

Advances in Machine Learning and Natural Language Processing

Research in Computing Science

Series Editorial Board

Editors-in-Chief:

Grigori Sidorov, CIC-IPN, Mexico
Gerhard X. Ritter, University of Florida, USA
Jean Serra, Ecole des Mines de Paris, France
Ulises Cortés, UPC, Barcelona, Spain

Associate Editors:

Jesús Angulo, Ecole des Mines de Paris, France
Jihad El-Sana, Ben-Gurion Univ. of the Negev, Israel
Alexander Gelbukh, CIC-IPN, Mexico
Ioannis Kakadiaris, University of Houston, USA
Petros Maragos, Nat. Tech. Univ. of Athens, Greece
Julian Padget, University of Bath, UK
Mateo Valero, UPC, Barcelona, Spain
Rafael Guzmán, Univ. of Guanajuato, Mexico

Editorial Coordination:

Alejandra Ramos Porras
Carlos Vizcaino Sahagún

Research in Computing Science es una publicación trimestral, de circulación internacional, editada por el Centro de Investigación en Computación del IPN, para dar a conocer los avances de investigación científica y desarrollo tecnológico de la comunidad científica internacional. **Volumen 148, No. 7**, julio de 2019. *Certificado de Reserva de Derechos al Uso Exclusivo del Título* No.: 04-2005-121611550100-102, expedido por el Instituto Nacional de Derecho de Autor. *Certificado de Licitud de Título* No. 12897, *Certificado de licitud de Contenido* No. 10470, expedidos por la Comisión Calificadora de Publicaciones y Revistas Ilustradas. El contenido de los artículos es responsabilidad exclusiva de sus respectivos autores. Queda prohibida la reproducción total o parcial, por cualquier medio, sin el permiso expreso del editor, excepto para uso personal o de estudio haciendo cita explícita en la primera página de cada documento. Distribuida por el Centro de Investigación en Computación, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, México, D.F. Tel. 57 29 60 00, ext. 56571.

Editor responsable: *Grigori Sidorov, RFC SIGR651028L69*

Research in Computing Science is published by the Center for Computing Research of IPN. **Volume 148, No. 7**, July 2019. The authors are responsible for the contents of their articles. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without prior permission of Centre for Computing Research.

Advances in Machine Learning and Natural Language Processing

Lourdes Martínez Villaseñor (ed.)



Instituto Politécnico Nacional
"La Técnica al Servicio de la Patria"



Instituto Politécnico Nacional, Centro de Investigación en Computación
México 2019

ISSN: 1870-4069

Copyright © Instituto Politécnico Nacional 2019

Instituto Politécnico Nacional (IPN)
Centro de Investigación en Computación (CIC)
Av. Juan de Dios Bátiz s/n esq. M. Othón de Mendizábal
Unidad Profesional “Adolfo López Mateos”, Zacatenco
07738, México D.F., México

<http://www.rcs.cic.ipn.mx>

<http://www.ipn.mx>

<http://www.cic.ipn.mx>

The editors and the publisher of this journal have made their best effort in preparing this special issue, but make no warranty of any kind, expressed or implied, with regard to the information contained in this volume.

All rights reserved. No part of this publication may be reproduced, stored on a retrieval system or transmitted, in any form or by any means, including electronic, mechanical, photocopying, recording, or otherwise, without prior permission of the Instituto Politécnico Nacional, except for personal or classroom use provided that copies bear the full citation notice provided on the first page of each paper.

Indexed in LATINDEX, DBLP and Periodica

Electronic edition

Editorial

Este número de la revista “Research in Computing Science” contiene 38 artículos relacionados con temas sobre aprendizaje máquina y procesamiento de lenguaje natural. Los trabajos aquí publicados fueron cuidadosamente seleccionados por el comité editorial y revisados por revisores externos considerando su calidad técnica y originalidad científica. En estos artículos se abordan temas y aplicaciones entorno a cuatro tópicos relacionados: aprendizaje máquina, minería de textos, clasificación de imágenes y procesamiento de lenguaje natural.

En este volumen, 20 artículos presentan aplicaciones con uso de aprendizaje máquina en diversos dominios como: medios sociales, detección de fallas tanto en sistemas eléctricos y procesos industriales, pronósticos de velocidad del viento y predicción de partículas contaminantes, detección de enfermedades agrícolas y análisis del polen, aplicaciones en ambientes de aprendizaje y videojuegos educativos, creaciones artísticas y clasificación de objetos astronómicos. Dentro de estos trabajos también hay propuestas para mejorar el balanceo de datos, la optimización de redes distribuidas, selección de atributos y reducción de dimensiones.

Dentro de los temas de clasificación de imágenes, tenemos 9 trabajos que permiten o mejoran la clasificación de imágenes de plantas medicinales, maderas, hojas de pimiento morrón y árbol del dzidzilché. Entre ellos se encuentran otros que detectan emociones o mejoran las condiciones para la clasificación de imágenes.

En minería de textos, se incluyen 4 artículos de análisis de twitter y clasificación de eventos. Relacionados con el procesamiento de lenguaje natural, se reportan trabajos de creación y manejo de mapas conceptuales distribuidos, extracción de relaciones no taxonómicas en corpus de dominio, extracción semiautomática de metadatos en documentos no estructurados y evaluación de resúmenes automáticos en inglés.

Cabe mencionar que el proceso de revisión y selección de artículos se llevó a cabo usando el sistema libremente disponible EasyChair (www.easychair.org).

Lourdes Martínez Villaseñor
Universidad Panamericana, México
Editor Invitado

Junio 2019

Table of Contents

	Page
Minería de opiniones sobre textos en Twitter <i>Belém Priego Sánchez, Luis Poot Terán</i>	13
Proceso de desarrollo de aplicaciones para el dominio social media usando técnicas de Deep Learning <i>Aarón Colmenares Morales, Laura Nely Sánchez Morales, Giner Alor Hernández, José Luis Sánchez Cervantes, Ulises Juárez Martínez</i>	23
Selección óptima de variables mediante metaheurísticas binarias para la detección de fallas en un proceso industrial multivariado usando aprendizaje de máquina <i>Jesús Alejandro Navarro Acosta, Edgar O. Reséndiz Flores, Irma D. García Calvillo</i>	37
Propuesta de red neuronal convolutiva para la predicción de partículas contaminantes PM10 <i>Ricardo Domínguez-Guevara, María del Carmen Cabrera-Hernández, Marco Antonio Aceves-Fernández, Jesús Carlos Pedraza-Ortega</i>	51
Clasificador de plantas medicinales por medio de Deep Learning <i>Juan Monroy-de-Jesús, Adriana Reyes-Nava, Fernando Olmos</i>	65
Clasificación de maderas por corte usando su propio contenido de textura usando la metodología CBIR <i>Gabriel de Jesús Celis-Escudero, Juan Villegas-Cortez, Carlos Avilés-Cruz, Arturo Zúñiga-López, Andrés-Ferreyra Ramírez, Roman A. Mora-Gutiérrez, Cesar Benavides-Alvarez</i>	79
Configuración de red neuronal para el análisis de polen <i>Leyra Martínez, Pedro Arguijo, Antonio Hiram Vázquez López, Roberto Ángel Meléndez Armenta</i>	91
Detección y diagnóstico de fallas en sistemas eléctricos de potencia combinando una red neuronal autoasociativa y una red neuronal probabilística <i>Juan Pablo Nieto González</i>	99

Balanceo de datos del Síndrome de Guillain-Barré utilizando SMOTE para la clasificación de subtipos	113
<i>Manuel Torres-Vásquez, José Hernández-Torruco, Betania Hernández-Ocaña, Oscar Chávez-Bosquez</i>	
Videojuego educativo para ayudar a comprender los principios básicos de la programación y desarrollar la habilidad lógica en niños de educación básica	127
<i>Ana Caren Ticante Hernández, Carlos Miguel Herrera Orduña, Pedro Arguijo, Roberto Ángel Meléndez Armenta, Antonio Hiram Vázquez López</i>	
Mapas conceptuales colaborativos: un modelo basado en grafos distribuidos	141
<i>Jorge A. Hernández-Perales, Manuel Hernández-Gutiérrez</i>	
Reconocimiento multimodal de emociones orientadas al aprendizaje.....	153
<i>Ramón Zatarain Cabada, María Lucía Barrón Estrada, Héctor Manuel Cárdenas López</i>	
Reconocimiento de gestos de la mano aplicado a una interfaz para ambientes de aprendizaje.....	167
<i>Ramón Zatarain-Cabada, María Lucía Barrón-Estrada, Brandon Antonio Cárdenas Sainz</i>	
Pronóstico de la velocidad del viento mediante técnicas de Inteligencia Artificial	181
<i>Juan Ramón V. Barraza, Héctor Rodríguez R., Víctor González Huitrón, Carlos A. Lara Alvarez, Juan J. Flores</i>	
Diseño de una red neuronal distribuida entre dispositivos Raspberry Pi conectados a Internet por medio de XMPP.....	197
<i>Alberto Martínez Contreras, David Tinoco Varela, Fernando Gudiño Peñalosa</i>	
Optimización Distribuida de Redes Convolucionales para la Clasificación de Imágenes	213
<i>Emmanuel F. Ramírez Hernández, Héctor Rodríguez Rangel, Víctor González Huitrón, Juan J. Flores, Vicenc Puig</i>	
Emotion Mining Mechanism over Texts in Social Media	227
<i>Luis Casillas, Alejandro Ramirez</i>	
Evaluación de resúmenes automáticos con y sin resúmenes de referencia para el idioma inglés.....	241
<i>Selene Arai Vilchis Sepúlveda, Yulia Ledeneva</i>	

Modelos neuronales pulsantes adaptados para el mejoramiento de luminosidad de imágenes cerebrales de gran resolución	253
<i>Manuel Mejía-Lavalle, Kevin Aguilar, Humberto Sossa, Dante Mujica, Andrea Magadan</i>	
Modelo neurodifuso para el control de un exoesqueleto para rehabilitación de brazo en pacientes con EVC	267
<i>Santos M. Orozco-Soto, Alberto I. Pérez-Sanpablo, Elisa Romero-Ávila, Catherine Disselhorst-Klug, Juan M. Ibarra-Zannatha</i>	
Algoritmo basado en reglas de asociación para la extracción de relaciones no taxonómicas en corpus de dominio	277
<i>Irvin Yair Cabrera Moreno, Mireya Tovar Vidal, José de Jesús Lavalle Martínez, Meliza Contreras González</i>	
Estudio comparativo de arquitecturas de CNNs en hojas de Pimiento Morrón infectadas con virus PHYVV o PEPGMV	289
<i>Eduardo Díaz-Gaxiola, Zurriel E. Morales-Casas, Oscar Castro-López, Gerardo Beltrán-Gutiérrez, Ines F. Vega-López, Arturo Yee-Rendón</i>	
Entrenamiento dinámico de redes convolucionales profundas para clasificación de imágenes	305
<i>Oscar Frausto-Pérez, Alfonso Rojas-Domínguez, Manuel Ornelas-Rodríguez, Héctor Puga, Martín Carpio</i>	
Comunicación política en Twitter y su análisis automático mediante el uso de datos ordenados y visualización de información.....	319
<i>Rocío Abascal Mena</i>	
Extracción semiautomática de metadatos en documentos no estructurados utilizando procesamiento de lenguaje natural y propiedades tipográficas	331
<i>Alberto Iturbe Herrera, Azucena Montes Rendón, Juan-Manuel Torres-Moreno, Gerardo Sierra Martínez, Noé Alejandro Castro Sánchez, Juan Gabriel González Serna</i>	
Detección de comandos de voz con modelos compactos de aprendizaje profundo	347
<i>Edoardo Bucheli-Susarrey, Miguel González-Mendoza, Oscar Herrera-Alcántara</i>	
Detección del árbol de dzidzilché usando los descriptores de Halarick y redes neuronales artificiales.....	363
<i>Juan Jose Negron-Granados</i>	

Métodos computacionales para la creación artística.....	375
<i>Sandra Rodríguez-Mondragón, Luis Jorge Soto-Walls, Manuel Martín Clavé-Almeida, Oscar Herrera-Alcántara</i>	
Red neuronal convolucional con extracción de características multi-columna para clasificación de imágenes	391
<i>Fidel López Saca, Andrés Ferreyra Ramírez, Carlos Avilés Cruz, Juan Villegas Cortéz</i>	
Clasificación de eventos académicos a partir de su descripción textual	405
<i>Ariadna Gutiérrez-Rosales, José A. Reyes-Ortiz, Maricela Bravo</i>	
Detección de enfermedades en el sector agrícola utilizando Inteligencia Artificial	419
<i>Balzhoyt Roldán Ortega, Rajesh Roshan Biswal, Eddy Sánchez DelaCruz</i>	
Estado del arte de clasificación de nódulos pulmonares utilizando Aprendizaje Automático	429
<i>Cecilia Irene Loeza Mejía, Rajesh Roshan Biswal, Eddy Sánchez-DelaCruz</i>	
Utilización de un sistema en tiempo real para la predicción de contaminación del aire	441
<i>Carlos Lino-Ramírez, Rogelio Bautista-Sánchez, Sandra P. Bombela-Jiménez</i>	
Algoritmos de aprendizaje automático de vanguardia para el diagnóstico de enfermedades	455
<i>José Carlos de Jesús Montero Rodríguez, Rajesh Roshan Biswal, Eddy Sánchez de la Cruz</i>	
Acceso automático por video detección de gestos manuales utilizando técnicas eficientes de segmentación dinámica de color de piel	469
<i>Julio Zamora Esquivel, José Rodrigo Camacho Pérez, Cruz Vargas Jesus Adan, Paulo López Meyer, Héctor Cordourier Maruri</i>	
Redes neuronales y árboles de decisión para la clasificación de objetos astronómicos	477
<i>Rogelio Hernández Montes, Cynthia Alejandra Martínez Pinto, Silvana Guadalupe Navarro Jiménez</i>	

MoCa: Una aplicación móvil para registrar síntomas de pacientes con cáncer.....	491
<i>Beatriz A. González-Beltrán, Lizbeth Gallardo-López, Roberto García-Madrid, Marco V. Ferruzca-Navarro, Alejandro Reyes-Ortíz</i>	
Selección de atributos mediante separación de centroides	505
<i>Héctor Jiménez-Salazar, Alfredo Mateos-Papis, Christian Sánchez-Sánchez</i>	
Reducción de la dimensión de registros de evaluaciones académicas aplicando el algoritmo K-means	515
<i>Victoria-Elizabeth Padilla-Morales, Saturnino Job Morales Escobar, Maricela Quintana López, José Martín Flores Albino, Oscar Herrera Alcántara</i>	

Minería de opiniones sobre textos en Twitter

Belém Priego Sánchez, Luis Poot Terán

Universidad Autónoma Metropolitana unidad Azcapotzalco,
Departamento de Sistemas,
Ciudad de México, México
abps@azc.uam.mx, lept28@hotmail.com

Resumen. Actualmente, la información se encuentra en constante expansión gracias a la posibilidad que ha proporcionado el Internet y las herramientas de comunicación que éste tiene. Con lo cual, es posible obtener desde textos científicos hasta opiniones que una persona tiene sobre cierto producto. Sin embargo, conocer e interpretar toda esta información, principalmente los comentarios, no es tarea sencilla por lo que se hace necesaria la construcción de herramientas que permitan beneficiar y ayudar con dicha tarea. En este artículo se presenta una aplicación de Internet enriquecida que permite procesar comentarios de Twitter; el objetivo es analizar y clasificar los textos, escritos, emitidos por los usuarios de un producto. Dicha herramienta trabaja en tiempo real y permite clasificar opiniones a partir de una frase de búsqueda y mediante la obtención de la acción de ésta, es decir, mediante la identificación del verbo.

Palabras clave: minería de opiniones, clasificación de opiniones, identificación de información.

Text Opinion Mining in Twitter

Abstract. Nowadays, information is continuously growing in internet due to the great communication tools that already exist in this worldwide platform. We are actually able to obtain any kind of documents from Internet which ranges from scientific documents to single opinions about a certain product. The semantic interpretation of such documents is, however, a very difficult task which requires to foster the construction of natural language processing tools. In this paper, we present a enriched application able to process Tweets with the aim to analyze and classify text excerpts such as customer opinions of some product. The tool developed is working real-time and allows to classify customer opinions from a query constructed on the basis of a part-of-speech analysis of the input tweet identifying two components: verbs and nouns.

Keywords: opinion mining, classification, natural language processing.

1. Introducción

Día a día la cantidad de información generada por los usuarios de Internet está en constante crecimiento, ésta va desde textos científicos hasta opiniones proporcionadas por una persona sobre un producto. Por lo que su análisis puede dar resultados útiles para afrontar problemas actuales. Sin embargo, obtener estos resultados no es tarea sencilla; en consecuencia, se han desarrollado nuevas tecnologías y herramientas.

El análisis de texto surge del Procesamiento del Lenguaje Natural. El cual “pretendía que una computadora interpretara cualquier texto con el fin de encontrar la semántica asociada al conjunto de palabras que lo conforman” [2]. A lo largo de los años se ha utilizado el análisis de texto para extraer patrones basados en palabras y temas significativos, obteniendo datos cuantitativos. Logrando por ejemplo predicciones sobre el comportamiento de un individuo, así también influyendo en su toma de decisiones de forma imperceptible.

Las redes sociales pueden brindar gran cantidad de información (comentarios que se generan a través del uso de las redes sociales) sobre gustos, opiniones, noticias, eventos, entre otros. Las personas comparten gran parte de sus actividades diarias en las redes sociales, la información, generada por este tipo de tecnologías, puede ser usada por dependencias, como empresas del sector privado para saber si un producto es del agrado de sus consumidores o un partido político para conocer si sus propuestas son aceptas por la mayoría de la población.

En el presente artículo se presenta un sistema web capaz de analizar opiniones en la red social Twitter. Para llevar a cabo dicho análisis se deben tener en cuenta dos aspectos fundamentales: el tema a buscar y su valoración. En el primer aspecto se descargarán tweets con el objeto de búsqueda. El segundo aspecto examinará la información de cada tweet transformándolo por medio de lexicones (listas de palabras obtenidas del mismo tweet). Se analizarán los lexicones obtenidos para llegar a una respuesta porcentual sobre el objeto de búsqueda y su valoración. La herramienta presentada, en este artículo, es una aplicación de Internet enriquecida que permite procesar comentarios de Twitter con el fin de analizar y clasificar los textos escritos emitidos por los usuarios de un producto. Dicha herramienta trabaja en tiempo real y permite clasificar opiniones a partir de una frase de búsqueda y mediante la obtención de la acción de ésta, es decir, mediante la identificación del verbo.

2. Minería de opiniones

El análisis de sentimientos es una tarea de clasificación de textos dentro del área del procesamiento del lenguaje natural, cuyo objetivo consiste en detectar la polaridad (positiva, negativa o neutra), de una opinión dada por un cierto usuario [5]. El conocer la opinión que una persona tiene hacia un producto o servicio es de gran ayuda para toma de decisiones, ya que permite, entre otras cosas, que posibles consumidores verifiquen calidad del producto o servicio antes de utilizarlo.

El análisis de la polaridad, en cualquier tipo de comentario, es una tarea que está teniendo un gran auge, debido a que actualmente existe un fuerte interés en determinar automáticamente si las opiniones publicadas en medios públicos tienen un carácter positivo o negativo. La minería de opiniones se enfoca en determinar la polaridad de las publicaciones para, generalmente, dar seguimiento a la reputación de una entidad. En este artículo se aborda esta problemática mediante el desarrollo de una aplicación que permite procesar comentarios de Twitter emitidos de un producto; categorizando éstos en opiniones positivas y negativas.

2.1. Trabajos relacionados

En esta sección se presentan los trabajos reportados en la literatura que permiten, de cierto modo, realizar una comparativa entre lo que se presenta y lo existente. Las opiniones, fabricadas mediante una conversación o comentario, es posible consultarlas a través de foros, blogs o redes sociales, éstas últimas siendo la novedad y teniendo el mayor auge actual. Un trabajo basado en una colección de entradas de blogs, es el presentado en [1] que realiza el análisis de sentimientos y minería de opiniones en dichas entradas, mostrando la relevancia de los sistemas de aprendizaje automático como recurso para la detección de información de opinión. Siguiendo con el auge actual, las redes sociales, se tiene el artículo [6] que proporciona información sobre la importancia y la usanza de la información publicada en redes sociales, así como una forma de hacerlo; esta importancia con el objetivo de realizar un análisis más detallado de la minería de opiniones y el análisis de sentimientos. Un trabajo asociado a la minería de opiniones y su alto grado de investigación en los últimos años es el presentado en [3] el cual plantea un sistema que extrae, procesa e identifica sentimientos para clasificar las opiniones sobre un dominio específico (hoteles).

En [4] se aborda el tema de minería de opiniones y análisis de sentimientos, lo cual permite identificar las opiniones de los usuarios expresando comentarios positivos, negativos o neutros y citas subyacentes al texto; se realiza la búsqueda de comentarios con opiniones propias de los usuarios con el fin de llegar a un resultado favorable o desfavorable del tema a buscar y, además, realizando una clasificación de sentimientos de los usuarios.

Hacer uso de los datos generados por personas, comentarios, es una gran oportunidad para ganar tiempo en las decisiones tomadas, debido a que proporcionan información que puede ser utilizada en diferentes ámbitos. A partir de los datos adquiridos se puede realizar un análisis automático y generar estadísticas sobre la opinión colectiva (positiva o negativa) de un producto, servicio o persona [5]. Dicho análisis es de gran utilidad para los analistas de medios, desde la disminución de tiempos hasta la disminución de costos. Por ello, en este artículo se ayuda a la automatización de ciertos procesos de análisis para empresas, consumidores, por citar algunos ejemplos. En las siguientes secciones se presenta el desarrollo del sistema que permite la minería de opciones de tweets.

3. Propuesta para la minería de tweets

La popularidad y el uso de las redes sociales, por personas de diversas edades y clases sociales, permiten que el desarrollo de este trabajo se lleve a cabo de una forma adecuada; debido a que actualmente las personas plasman su juicio sobre un producto o servicio de una manera sencilla y recurrente. Una de las formas en las que se transmite una opinión es mediante la escritura de un tweet y en ocasiones resulta ser más fácil escribir éste, de lo que se opina, que mandar un email o hacer una llamada. Por lo que recabar y analizar esta información es más sencillo para las empresas.

Esta plataforma puede tener alcances incluso de estudiar y predecir el comportamiento humano basado en lo que escribe, por ejemplo, conocer el estado anímico de un individuo para evitar atentados en escuelas como lo es en Estados Unidos de Norte América, evitar el suicidio de un adolescente, por citar algunos ejemplos.

Esta sección presenta la propuesta llevada a cabo para la minería de opiniones en tweets. La metodología está compuesta de cuatro módulos: extracción, preprocesado, clasificación y análisis y resultado, descritos a continuación.

3.1. Extracción de tweets

Este módulo es el encargado de recolectar los tweets, en tiempo real, que se analizarán; básicamente está compuesto de dos etapas principales: conexión y extracción.

La etapa de conexión consiste en realizar la conexión con los servidores de Twitter; para llevar a cabo dicho proceso, inicialmente se crea una cuenta de desarrollador de Twitter y posteriormente se hace el registro del proyecto. Cuando el proyecto es aceptado, Twitter proporciona cuatro llaves (*Consumer Key*, *Consumer secret*, *Acces token* y *Acces token secret*) únicas y son totalmente confidenciales; éstas permiten el acceso del proyecto a los servidores de Twitter. El desarrollo del proyecto fue realizado en Java, en la figura 1 se muestra el código de la clase que permite la conexión a los servidores de Twitter.

La segunda etapa es la extracción de los tweets, la cual prosigue a la etapa de conexión, que consiste básicamente de extraer los tweets que coinciden con el tema de búsqueda proporcionado por el usuario. A cada tweet se le da un formato específico y es almacenado en un archivo de texto plano; una de las condiciones de descarga, gratuita, es que únicamente se pueden descargar cien tweets por día. Por tal motivo, este conteo es reiniciado a las 24 horas de cada día. En la figura 2 se muestra la clase usada para extraer los tweets y darles formato.

3.2. Preprocesado del corpus

Este módulo está compuesto de tres etapas: la primera que se encarga de identificar los metadatos de los tweets, la segunda que le da un formato a los

```

public void Tweet() throws TwitterException{
    Twitter twitter;
    ConfigurationBuilder cb = new ConfigurationBuilder();
    cb.setDebugEnabled(true)
        .setOAuthConsumerKey("Consumer Key")
        .setOAuthConsumerSecret("Consumer Secret")
        .setOAuthAccessToken("Access Token")
        .setOAuthAccessTokenSecret("Access Token Secret");
    twitter = new TwitterFactory(cb.build()).getInstance();
}

```

Fig. 1. Clase, en Java, que permite la conexión a Twitter.

tweets para distinguirlos unos de otros y la tercera que se encarga de procesar los tweets.

Al extraer los tweets, se observa que el corpus está compuesto de diferentes metadatos como son apodo (*nickname*) que el usuario elige y puede cambiar, seguido por un nombre de usuario (*username*) el cual es único e inicia con el carácter @; posteriormente, se encuentra la fecha del día que se escribió el tweet y se presenta en el formato *ddmesaaaa* y finalmente, el tweet escrito por el usuario.

Una vez identificados los metadatos de los tweets, el siguiente paso fue el etiquetamiento; éste consiste en que cada tweet fue etiquetado con las etiquetas `< INICIOTWEET >` y `< /FINOTWEET >` que permiten delimitar el inicio y el fin del tweet. Posteriormente, se procede a identificar las palabras separadas por un espacio en blanco de cada oración en los tweets; este proceso se denomina Tokenización. El resultado de la tokenización es un nuevo corpus, el cual facilita la lematización de los tweets. El proceso de lematización se realizó con TreeTagger¹ con el archivo de parámetros o modelo del idioma español.

3.3. Clasificación del tweet basado en la valoración

Este módulo está compuesto de dos etapas, la primera es la identificación del verbo principal de la búsqueda y la segunda la clasificación de los tweets a partir de este verbo.

Para el caso de la etapa de identificación del verbo, se analizó la oración de búsqueda, que es ingresada por el usuario al sistema. El proceso es similar al de tokenización y lematización de los tweets, sin embargo, ahora el objetivo es identificar el verbo de la búsqueda ingresada. Esta identificación se realiza a partir de las etiquetas que el lematizador proporcionó y se selecciona el lema, que es lo que permite identificar la acción de búsqueda.

¹ Disponible en: <http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/>

```
public class TweetEx {
    private Twitter twitter;
    private SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy");

    public TweetEx() {
        twitter = TwitterFactory.getSingleton();
    }

    public List<Status> query(String query) throws TwitterException {
        QueryResult search = twitter.search(new Query(query));
        List<Status> tweets = search.getTweets();
        return tweets;
    }

    public void escribirTweet(Status status) throws IOException {
        File A = new File("/Resultado/Busqueda.txt");
        FileWriter w = new FileWriter(A);
        BufferedWriter bw = new BufferedWriter(w);
        PrintWriter wr = new PrintWriter(bw);
        wr.write(String.format(status.getUser().getScreenName()+" "+sdf.format(status.getCreatedAt())+"\n"));
        wr.write(status.getText()+"\n\n");
    }

    public void escribirTweet(List<Status> status) throws IOException {
        for (Status tweet : status) {
            escribirTweet(tweet);
        }
    }
}
```

Fig. 2. Clase, en Java, que permite la extracción de tweets.

La siguiente etapa consiste en clasificar los tweets de acuerdo al verbo principal de búsqueda, resultado obtenido en una etapa preliminar, mediante la transformación de este en su antónimo. Para la clasificación de los tweets se realizó una búsqueda sobre el corpus de tweets lematizados, en la que se deseaba encontrar el verbo original o su antónimo, obteniendo dos corpora para cada una de las búsquedas efectuadas en Twitter. De esta manera, se crean los corpora de tweets donde en el primer grupo se encuentran los tweets que contienen la valoración original (verbo principal) denominado *Corpus_Positivo* y en el segundo los tweets que contienen su antónimo denominado *Corpus_Negativo*.

3.4. Análisis y visualización del resultado

En este módulo se realizó un conteo de los tweets que se encontraban en cada corpora (*Corpus_Positivo* y *Corpus_Negativo*). Posteriormente, se hizo la sustracción del total de la suma de los dos grupos con el total de tweets descargados, para poder verificar el porcentaje de error; es decir, cuántos tweets no se clasificaron. Esto se puede expresar con la ecuación error de clasificación (1).

$$Erc = Td - (Tp + Tn). \quad (1)$$

Donde *Erc* corresponde al resultado de los tweets erróneamente clasificados, *Td* el total de tweets descargados, *Tp* el número de tweets del *Corpus_Positivo* y *Tn* el número de tweets del *Corpus_Negativo*.

Finalmente, se visualizan los resultados en una interfaz gráfica, en la figura 3 se muestra la pestaña principal con un ejemplo de búsqueda que permite

visualizar de todo el proceso del sistema; desde la entrada de texto, donde el usuario escribe la oración de búsqueda, hasta la visualización de los resultados en una gráfica de pastel.

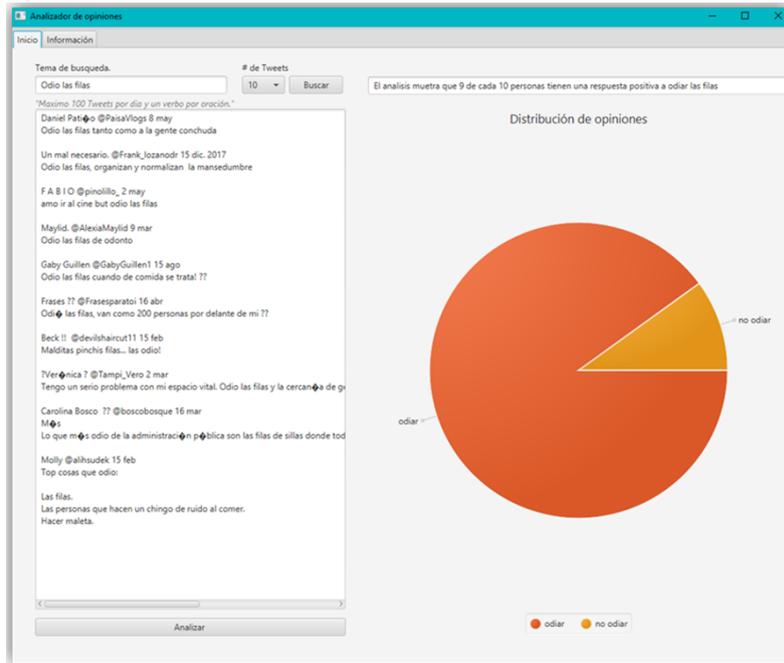


Fig. 3. Pestaña principal del sistema de minería de opiniones.

El sistema está construido de tal manera que si alguno de los datos solicitados no se incluye; por ejemplo, al momento de que un usuario no seleccione el número de tweets a buscar (10, 20, 40, 50, 80 o 100 tweets). Éste mostrará un mensaje al usuario indicando que algunos de los campos están vacíos. Además, se agregó una función en el módulo de clasificación, para evitar errores cuando se está buscando el verbo y no se encuentra. Es decir, cuando un usuario haya escrito una oración sin verbo.

4. Resultados obtenidos

En esta sección se muestran una serie de pruebas que permiten mostrar los resultados del sistema; gracias a que el sistema permite seleccionar el número de tweets (N) a descargar, es posible ejecutar la misma consulta con diferentes tamaños de corpora. En este sentido, los valores que se han corroborado son para $N = 10, 50, y 100$ tweets, teniendo un total de 400 tweets diferentes de análisis y

descargados en diferentes tiempos, resaltando que el sistema funciona en tiempo real.

La primera prueba se consistió en ejecutar diez veces el sistema, en diferentes tiempos, con el tema de búsqueda “odiar las filas” para un tamaño de $N=10$, obteniendo un total de 100 tweets para dicho tema. La siguiente prueba consistió en buscar el tema de búsqueda “me gusta la navidad” para $N=50$ en dos ejecuciones diferentes, recolectando en total 100 tweets. El tercer experimento consistió en utilizar como oración de búsqueda “Quiero que México cambie” para $N=100$ en dos ejecuciones de sistema, analizando un total de 200 tweets en diferentes tiempos.

Como primeras pruebas del funcionamiento del sistema, se utilizaron alrededor de 1,000 tweets, sin embargo, para el análisis de dichas pruebas únicamente se utilizaron 400 tweets. De éstos se obtuvieron alrededor de 8,000 tokens. La Tabla 1, muestra los resultados de las diferentes pruebas realizadas.

Tabla 1. Resultados de las diferentes pruebas realizadas.

Tamaño del corpus	No. de pruebas realizadas	No. de tokens lematizados	Porcentaje de tweets erróneamente clasificados
10	10	2,000	5 %
50	2	2,000	15 %
100	2	4,000	27 %

Como puede observarse en la Tabla 1 el porcentaje de tweets erróneamente clasificados tiene una relación directa con el tamaño del corpus, es decir, si el tamaño del corpus aumenta o disminuye, el porcentaje de tweets erróneamente clasificados se ve afectado de la misma manera. Este comportamiento puede deberse a varios factores como lo son:

- El módulo de extracción de tweets los almacena en codificación ANSI, provocando que los caracteres no identificados se conviertan en signos de interrogación. Cuando se lematiza, TreeTagger no logra reconocerlos y puede provocar un tweet sin clasificar.
- Existen problemas de ortografía, en algunos casos, los usuarios no separan las palabras adecuadamente, provocando que en el proceso de lematización, el lema no sea encontrado y TreeTagger devuelva la etiqueta *< unknow >*.
- TreeTagger no reconoce la escritura de los tweets que no está hecha en español. Por lo que asigna la etiqueta *< unknow >* (caso similar al anterior).

Finalmente, todas las pruebas se realizaron satisfactoriamente ya que siempre se cumplió con la función de analizar y clasificar los comentarios.

5. Conclusiones y perspectivas

El procesamiento de lenguaje natural es una área con un gran campo de acción, el presente trabajo es una muestra de lo que se puede realizar. Como se

pudo observar, se logró desarrollar, satisfactoriamente, el sistema para el análisis de opiniones en la red social Twitter, llegando a extraer tweets, lematizar tokens, clasificar y analizar corporas y finalmente, visualizar los resultados.

El sistema realiza el análisis adecuadamente, sin embargo, conforme aumenta el tamaño del corpus de tweets, los tweets erróneamente clasificados también lo hacen. El menor porcentaje de error en la clasificación de los tweets, se encontró en las pruebas realizadas con el corpus de tamaño $N=10$ tweets. Este porcentaje fue del 5%, no obstante, el resultado no se puede considerar representativo por la muestra tan pequeña de opiniones. Se puede considerar guardar el resultado para búsquedas posteriores pero el ligar los resultados puede llevar a una conclusión errónea, ya que se pueden presentar variaciones de opiniones a través del tiempo.

La información del funcionamiento de TreeTagger en Java es muy escasa y poco descriptiva, lo que dificulta utilizar esta herramienta. Sin embargo, TreeTagger está completo con todos sus módulos, solo es necesario el ensayo y error para encontrar el modo adecuado de la aplicación deseada. Para mayor éxito en la lematización, se recomienda entrenar a TreeTagger o crear un archivo de parámetros con emoticones y abreviaturas y posiblemente realizar el almacenamiento en formato UNICODE.

El presente trabajo puede ser ampliado al análisis de emociones sobre textos en Twitter, polarizando los tokens en el proceso de lematización y reincorporádoslos a su oración original. Esto crearía oraciones polarizadas y con su correcto análisis se podría conocer el estado de ánimo del usuario.

Referencias

1. Fernández, J., Boldrini, Gómez, E.J.M., Martínez-Barco, P.: Análisis de sentimientos y minería de opiniones: el corpus EmotiBlog. *Procesamiento del Lenguaje Natural*, Revista no. 47, pp. 179–187 (2011)
2. García Menier, E.: Análisis De Textos Por Computadora. *Boletín de Lingüística* 18(25), 121–134 (2006)
3. Henríquez, C.: Minería de Opiniones basado en la adaptación al español de ANEW sobre opiniones acerca de hoteles. *Procesamiento de Lenguaje Natural*, Revista no. 56, pp. 25–32 (2016)
4. Khairnar, J., Kinikar, M.: Machine Learning Algorithms for Opinion Mining and Sentiment Classification. *IJSRP* 3(6), 1–6 (2013)
5. Priego, B., Pinto, D., Castro, M., León, M.: Análisis de la polaridad en comentarios de estudiantes universitarios sobre el desempeño de sus profesores. *Pistas Educativas*, no. 130, pp. 946–961 (2018)
6. Sneka, G.: Algorithms for Opinion Mining and Sentiment Analysis: An Overview. *International Journal of Advanced Research in Computer Science and Software Engineering* 6(2), 1–5 (2016)

Proceso de desarrollo de aplicaciones para el dominio social media usando técnicas de Deep Learning

Aarón Colmenares Morales¹, Laura Nely Sánchez Morales¹, Giner Alor Hernández¹,
José Luis Sánchez Cervantes², Ulises Juárez Martínez¹

¹Tecnológico Nacional de México/ I.T. Orizaba,
Veracruz, México

²CONACYT-Tecnológico Nacional de México/ I.T. Orizaba, México
ingacm3000@gmail.com, lauransanchezmorales@gmail.com,
galor@ito-depi.edu.mx, jlsanchez@conacyt.mx, ujuarez@ito-depi.edu.mx

Resumen. Los avances del Deep Learning, son cada vez más notorios en el desarrollo de software. Sin embargo, a pesar de tener un gran avance aún no se explotan las técnicas de Deep Learning para los generadores automáticos de software, con atención especial en el desarrollo de aplicaciones *Social Media*. Por lo tanto, se propone un proceso de generación automática de aplicaciones basado en técnicas de Deep Learning, a partir de una imagen digital que representa una interfaz de usuario. Para lo cual se deben identificar UIDPs (*User Interface Design Patterns*) válidos para interfaces de aplicaciones de *Social Media* a través de técnicas de aprendizaje profundo. Las ventajas de este trabajo es el enriquecer las aplicaciones, reducir costos y mejorar el desempeño. Por último, se presenta un caso de estudio que muestra la implementación de las técnicas de Deep Learning en la identificación de UIDPs.

Palabras clave: generación automática de software, procesamiento de imágenes, proceso de desarrollo, aprendizaje profundo, social media, UIDPs.

Application Development Process for the Social Media Domain using Deep Learning Techniques

Abstract. Advances on Deep Learning techniques are becoming more notorious in software development. However, Deep learning techniques are not yet used for automatic software development into Social Media domain. This paper proposes an automatic development process of social media applications based on Deep Learning techniques. The proposed process generates applications from a digital image that represents a User Interface Design Pattern (UIDP). The advantages of this work are to enrich applications, to reduce costs and to improve performance. Finally, a case study that shows the implementation of Deep Learning techniques for identification of UIDPs is presented.

Keywords: automatic software generation, image processing, development process, deep learning, social media, UIDPs.

1. Introducción

Día a día se generan nuevos y variados métodos en el desarrollo de software más rápidos, certeros y confiables. Desde el punto de vista de la ingeniería del software y de forma específica dentro de la etapa de diseño de software, los UIDPs son una solución recurrente que resuelve problemas comunes de diseño, como son: las resoluciones de diferentes dispositivos, el diseño responsivo o problemas de navegación; todas estas soluciones se plantean a través de un lenguaje común para los diseñadores. Por su parte, el uso de aplicaciones de dominio *Social media*, cuenta con una gran demanda para los usuarios, debido a que se han expandido los nichos de mercado cada vez más al entorno social.

Por otra parte, el Deep Learning es un subconjunto de técnicas de inteligencia artificial que permite modelos computacionales compuestos de múltiples capas para aprender un conjunto de datos etiquetados, con el contenido de varias capas permiten el reconocimiento del lenguaje, la detección de algún objeto e inclusive la traducción. Dentro del Deep Learning encontramos las CNN (*Convolutional Neural Networks*), las cuales son diseños para procesar datos que llegan en forma de múltiples arreglos, generando una rápida detección de bordes, líneas, entre otras características simples.

De acuerdo a la literatura revisada en este documento encontramos que a pesar de que existen generadores de software similares y se encuentra orientados a otros dominios, existen también procesos poco orientados para las aplicaciones *Social Media* que manipulen técnicas de Deep Learning y ayuden en el reconocimiento de UIDPs en imágenes. Desde esta perspectiva, la identificación de cada UIDP es importante para clasificar y determinar el tipo de dispositivo adecuado para su funcionamiento. Tal situación, motiva a plantear un proceso de generación automática de aplicaciones basado en técnicas de Deep Learning.

El primer pilar de tres en esta investigación, es la generación de software aplicando el uso de una imagen o un boceto generado a mano alzada. El segundo pilar, es el desarrollo de software basado en el aprendizaje profundo, donde una de las características principales que se aprovechó en el desarrollo de software es el aprendizaje a través del entrenamiento, el cual mejora el tiempo de desarrollo. El tercer pilar son los UIDPs, ya que permiten identificar y restaurar ciertos errores de una aplicación, entre los cuales permiten mantener una resolución estándar y rectificar errores generados de diseño por el desarrollador, cumpliendo con una aplicación fácil de usar para el usuario.

Como contribuciones para la ingeniería de software, se tienen cuatro aspectos importantes: (1) generar beneficios para que el proceso de desarrollo de software sea mucho más intuitivo y que sea tan bueno como lo hace un ser humano, (2) diseñar un modelo de *machine learning* y un conjunto de reglas que permitan la generación automática de código, (3) tener un mejor desempeño para aumentar la precisión en la clasificación de imágenes que mejore los ciclos de desarrollo del software y (4) enriquecer las aplicaciones que se sumen a las actividades diarias y aumente la motivación para el desarrollo de software.

Este documento está estructurado de la siguiente manera, en la sección 2 se presenta el estado del arte referente a los diversos trabajos relacionados para la

generación automática de software, dividido en dos partes: herramientas para la generación de aplicaciones o desarrollo de software y técnicas Deep Learning en el desarrollo de software. En la sección 3 se hace una descripción de los UIDPs identificados, analizados y aplicados para la generación automática de aplicaciones del dominio *Social Media*. En la sección 4 se detalla el prototipo para la clasificación de interfaces de usuario. Finalmente, en la sección 5 se presentan las conclusiones y trabajo a futuro.

2. Estado del arte

A continuación, se presenta la revisión del estado del arte sobre los trabajos relevantes para la generación de aplicaciones o desarrollo de software y técnicas de Deep Learning en el desarrollo de software.

2.1. Herramientas para la generación de aplicaciones o desarrollo de software

Para las herramientas de generación de aplicaciones se reportan trabajos como Rosales-Morales et al. [1] donde proponen una metodología para el desarrollo de software, y a través de una evaluación cualitativa y cuantitativa, se obtienen resultados que favorecen a Visual Paradigm y Dreamweaver como herramientas para el desarrollo de software. En Cortes-Camarillo et al. [2] analizaron diversos patrones de diseño, para diferentes dispositivos y plataformas para el desarrollo aplicaciones educativas. Cortes-Camarillo et al. [3] presentaron Edugene, una herramienta enfocada en crear contenido educativo para tres tipos de aplicaciones como MOOC (*Massive Online Open Courses*), blog o wiki. Desde una perspectiva diferente, en Cortes-Camarillo et al. [4] presentaron Atila, un generador de aplicaciones educativas basado en UIDPs. La característica principal de Atila es generar aplicaciones rápidas; además, establece un lenguaje nativo para mejorar el desempeño de las aplicaciones, la compatibilidad y generar código a partir de un archivo XML. En Sánchez-Morales et al. [5] se propuso un proceso que consta de tres fases: análisis de imagen, configuración y generación de código fuente; el resultado dio una contribución para la ingeniería de software. Da-Cost, Neto y de-Oliveira [6] presentaron un estereotipo para MDD (*Model-Driven Development*). El estereotipo UI (*User Interface*) basado para portales Web, captura los procesos y las especificaciones de la UI. Este modelo reduce el desarrollo *time-to-market*, esfuerzo y costo, además contribuye a la calidad, productividad y conservando el funcionamiento en las aplicaciones Web. Por otra parte, Sánchez-Morales et al. [7] propusieron el desarrollo de un componente de software que utiliza técnicas de procesamiento de imágenes y reconocimiento de patrones para la generación automática de aplicaciones multidispositivo. Como resultados se obtiene el reconocimiento de diferentes elementos que forman parte de una imagen representativa de una interfaz de usuario.

2.2. Técnicas Deep Learning en el desarrollo de software

Dentro del desarrollo del software el uso de técnicas de Deep Learning son importantes y tienen diversas aplicaciones. Desde esta perspectiva, en Halbe y Joshi [8] presentaron un enfoque que recibe una imagen escaneada previamente del diseño de la GUI (*Graphical User Interface*), el sistema segmenta los controles HTML. El enfoque logró reconocer los siguientes controles: botón de radio, casilla de verificación, cuadro de texto y botón de comando. Por otra parte, en Baveye et al. [9] platearon MemoNet, los resultados sugieren que el modelo depende de las emociones inducidas por las imágenes y la información emocional incrementa el desempeño del modelo para imágenes neutrales y positivas.

En Bianco et al. [10] destacaron que, para el reconocimiento correcto de una imagen de un logotipo, la imagen debe de estar compuesta y ejecutada por una CNN. Los resultados confirmaron la factibilidad de los métodos propuestos superando los del estado del arte. Liu et al. [11] proponen que los métodos de imagen fusionada basados en Deep Learning se aplicaron para fotografía digital, multi modularidad de imagen e imágenes de teledetección. Los resultados sostienen que existen dificultades en los trabajos convencionales y son resumidas en métodos de fusión y evaluaciones. Pang et al. [12] crearon un gran conjunto de datos ad-hoc con ejemplos positivos y negativos desde la base de datos.

Los resultados muestran, que el método Deep Learning de seguimiento de objetivos visuales es más competitivo con respecto a los algoritmos del estado del arte. Krizhevsky, Sutskever y Hinton [13] emplearon un nuevo método de regularización llamado Dropout. Los resultados demostraron que las pruebas de la red visual tienen un alto índice de clasificación y asignación al nombrar correctamente las imágenes. Huang et al. [14] planteó que el MRBDL (*Multi-concept Retrieval using Bimodal Deep Learning*) capturó correlaciones semánticas entre una imagen visual con sus etiquetas libres contextuales.

Estas pruebas muestran la mejora en los datos semánticos, visuales y de texto. Nedzved et al. [15] presentó un esquema en la combinación de características de bibliotecas y de un intérprete con un conjunto de funciones para el procesamiento de imagen. Como resultado, se dividió en dos partes: (1) a desarrolladores de software profesional, son cambios sin necesidad de compilar; y (2) a usuarios, la cual cambia la interfaz gráfica para mejorar su entorno.

Los dominios abordados por los trabajos revisados se enfocan principalmente en: UIDPs, ingeniería de software, A.I. (*Artificial Intelligence*). Cada trabajo está ligado a uno, dos y hasta tres aspectos relacionados con: la generación de código fuente, el soporte de UIDPs, el soporte multidispositivo, y el uso de alguna técnica de Deep Learning. Sin embargo, se identificó que los trabajos no cubren todas las características analizadas al mismo tiempo.

Todos los trabajos soportan el desarrollo multidispositivo, pero no necesariamente permiten la generación de código fuente, o la implementación de UIDPs en el desarrollo de sus interfaces, y además el uso de técnicas de Deep Learning como parte de su proceso de desarrollo. Por tal motivo, se observa la oportunidad de abordar estas técnicas para plantear un proceso de desarrollo de software basado en técnicas de

Deep Learning, que parta del uso de imágenes generadas a mano alzada que representen interfaces de usuario.

3. UIDPs para social media

En esta sección se identifican y analizan los UIDPs adecuados y válidos para el desarrollo de aplicaciones *Social Media*. El análisis consistió un conjunto de aplicaciones de *Social Media* existentes, con el objetivo de identificar los UIDPs utilizados en cada aplicación.

Para las aplicaciones se identificaron dos tipos de UIDPs y se dividen en UIDPs por reputación e interacción. Los UIDPs por reputación permiten construir prestigio a los usuarios mediante el contenido, información compartida, la relación con más personas y seguir sus actividades. Los UDIPs por interacción por el contrario permiten compartir información, conectarse con más personas y seguir sus actividades. En la Tabla 1 se describen los UIDPs identificados por reputación. Y en la Tabla 2 se describen los UIDPs identificados por interacción.

Tabla 1. UIDPs por reputación.

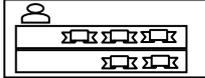
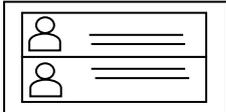
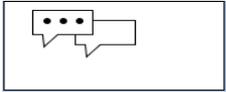
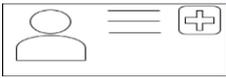
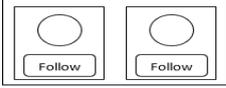
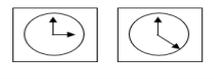
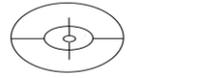
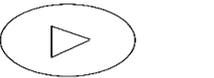
Reputación	Descripción	Imagen
Leaderboards	Se usan para mostrar usuarios altamente competitivos de las actividades que desarrollan.	
Achievements	Se aplica para desplegar las metas o el logro alcanzado en alguna actividad y compartirlo con tu círculo de amigos.	

Tabla 2. UIDPs por interacción.

Interacciones Sociales	Descripción	Imagen
Friend list	Ofrece una manera de explorar a los usuarios de tu círculo cercano, da oportunidad de encontrar contactos frecuentes o la forma de compartir la información con los demás.	
Chat	Permite a los usuarios contactar con otros usuarios de su mismo círculo o externos y provee una forma de comunicación privada de persona a persona.	
Friend	Crea un círculo de personas que desean interactuar y compartir contenido.	
Follow	Permite hacer un seguimiento de la información publicada por un usuario, temas o personas específicas de interés por parte del usuario.	

Interacciones Sociales	Descripción	Imagen
Reaction	Califica el contenido y da información de preferencias y gustos.	
Invite friends	Interactúa con amigos e invita a más amigos.	
Activity Stream	Actualiza las actividades de los contactos y que respondan a las acciones.	
Menú horizontal	Muestra en forma de lista las opciones de la aplicación en forma horizontal.	
Menú Vertical	Muestra en forma de lista las opciones de la aplicación en forma vertical.	
Galería	Da la forma de ver imágenes de modo secuencial, de principio a fin o cuando se desea ver una imagen lo más posible reservada de resolución.	
Panel	Permite ver las características o propiedades del aspecto.	
Ranking	Valora o califica por parte de un usuario un aspecto que contenga la aplicación.	
Picker	Permite seleccionar los ajustes de un calendario, paleta de color, entre otros.	
Music Player	Reproduce música y sonidos seleccionados.	

De los patrones de diseño identificados, se seleccionaron Chat, Galería, Menú horizontal, Menú vertical, Music player, Panel, Picker y Ranking, que son soportados hasta el momento por el prototipo generado.

4. Prototipo

En esta sección se describe un prototipo como prueba de concepto de implementación de las técnicas de Deep Learning para la identificación de UIDPs en imágenes generadas a mano alzada de tipo *Social Media*.

Es necesario mencionar que esta investigación está basada en el proceso de generación de software descrito por Cortes-Camarillo et al. [3], Cortes-Camarillo et al. [4], Sánchez-Morales et al. [5] y Sánchez-Morales et al. [7], el cual consta de tres etapas principales: análisis de la imagen, configuración y generación de código. De

manera particular, en esta investigación se realizaron cambios en la etapa de análisis de la imagen. En este caso, se usa el formato PNG para las imágenes soportadas con una resolución de 1050 por 650 píxeles. Es importante mencionar que el algoritmo no está limitado a esta única resolución de imágenes, es posible adaptar el algoritmo para diferentes resoluciones de forma automática. Sin embargo, para fines de pruebas se aplicó una única resolución. Por otra parte, en este trabajo se aplican algoritmos de Deep Learning a través de una red neuronal convolucional usando los modelos de AlexNet, LeNet o de propia creación; el proceso que realiza la red neuronal convolucional consiste en tomar una parte de una imagen y extraer un subconjunto de píxeles para la siguiente capa, así consecutivamente hasta la clasificación de cada UIDP encontrado en la imagen. Para la etapa de configuración, el tipo de aplicaciones soportadas son *Social Media* de tipo Microbloggin, Streaming, y Redes sociales; y se tiene soporte para aplicaciones móviles para las plataformas Android™ y iOS™ y para Web con HTML5

Finalmente, en la generación de código para la plataforma Web el código generado es HTML5 combinado con el *framework* de Bootstrap, este *framework* es utilizado ampliamente para navegar de forma responsiva en cualquier dispositivo tanto de escritorio como móvil; para la plataforma Android se tienen proyectos en Java; por último, para iOS™ se tiene Objective-C. A continuación, en la Fig. 1 se observa un diagrama de flujo con cada una de las etapas que retroalimenta el proceso de generación de software. De manera específica, en este prototipo la aportación principal se centra en la identificación de UIDPs a través de la CNN, donde los algoritmos utilizados dadas sus características permiten obtener mejores resultados de identificación de UIDPs en imágenes generadas a mano alzada con respecto a un trabajo previo reportado en Sánchez-Morales et al. [7].

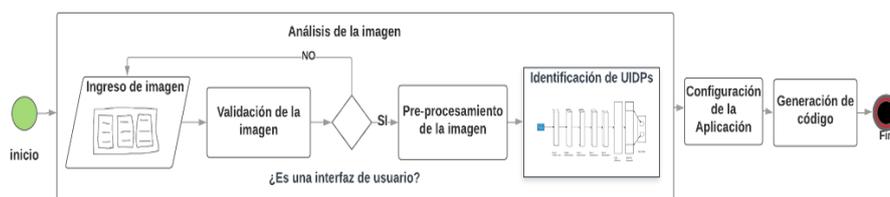


Fig. 1. Diagrama de Flujo del proceso de generación de software a partir de técnicas Deep Learning

El prototipo desarrollado se programó utilizando Python 3.6.0. y en el entorno de desarrollo Spyder con la versión 3.3.2. Python es un intérprete, orientado a objetos y un lenguaje de programación de alto nivel con semántica dinámica. Comúnmente usa un script o conexión de diálogo para conectar componentes ya existentes además es de código abierto [16].

Para desarrollar el prototipo se realizaron un conjunto de pruebas de entrenamiento para obtener el modelo final. En cada prueba se utilizaron un conjunto de imágenes clasificadas en 4 tipos: imágenes ideales, no ideales, reales en blanco y negro y reales a color. Las imágenes ideales son imágenes generadas mediante cualquier herramienta de dibujo y no presentan deformaciones (ver Fig. 2-a).

Las imágenes no ideales son aquellas generadas a mano alzada, desde cualquier aplicación o herramienta de dibujo o bien a través de una tableta digitalizadora (ver Fig. 2-b). Las imágenes reales en blanco y negro son imágenes generadas a mano alzada en hojas de papel y digitalizadas a través de una cámara o un scanner (ver Fig. 2-c). Finalmente, las imágenes reales a color son imágenes generadas a mano alzada, en hojas de papel de diferentes colores utilizando bolígrafos también de diferentes colores (ver Fig. 2-d).

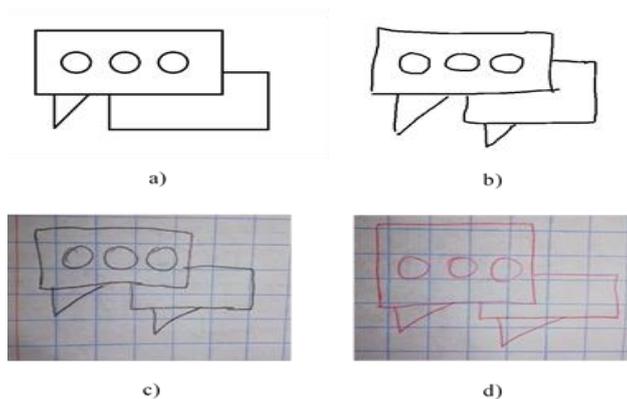


Fig. 2. Tipos de imágenes utilizadas para el entrenamiento de la red neuronal convolucional: a) imagen ideal, b) imagen no ideal, c) imagen real en blanco y negro, d) imagen real a color.

Los aspectos de cada prueba se encuentran organizados en Tablas de la siguiente forma. En la primera fila, se tiene el número de la prueba con el respectivo nombre de la API y el tipo de modelo aplicado, seguido por el total de imágenes usadas para el entrenamiento. En la segunda fila, se observan los encabezados para el nombre de los UIDPs que se emplearon en las pruebas, la precisión y el tiempo de respuesta del modelo. Al final de cada tabla, se tiene los batch y épocas realizados por cada prueba de entrenamiento. Para la prueba 1 ver la Tabla 3, para la prueba 2 ver la Tabla 4, para la prueba 3 ver la Tabla 5 y para la prueba 4 ver la Tabla 6.

Tabla 3. Resultados de la prueba 1.

Prueba 1. DeepLearning-Lenet		250 imágenes
UIDPs	Precisión	Tiempo (seg.)
Acordeon	.45	21.32
Barra de etiqueta	.45	21.32
Login	.45	21.32
Master Detalle	.45	21.32
Tarjeta	.45	21.32
256 batch y 3 épocas		

Tabla 4. Resultados de la prueba 2.

Prueba 2. Tensorflow-Libre		600 imágenes
UIDPs	Precisión	Tiempo (seg.)
Galería	1.0000000	30.00
Reproductor de música	3.1575022e-13	45.0

1699 batch y 100 épocas

Tabla 5. Resultados de la prueba 3.

Prueba 3. TensorFlow-Libre		800 imágenes
UIDPs	Precisión	Tiempo (seg.)
Galería	0.4339219	23.407269
Menú horizontal	0.422323233	23.407269
Menú vertical	0.123212333232	23.407269
Panel	0.0221412	23.407269

1401 batch y 10 épocas

Para la última prueba se realizó un pre-procesamiento, que consiste en la conversión de una imagen natural a una imagen binaria aplicando el procesamiento de imágenes. Para ejemplificar este proceso se usó una imagen generada a mano alzada que representa un Menú horizontal (ver Fig. 3-a). El resultado de convertir la imagen en escala de grises se observa en la Fig. 3-b. Posteriormente, el segundo paso es el uso del algoritmo de núcleo gaussiano que permite eliminar el ruido de los tonos grises [17] y ocupar solamente los bordes obtenidos (ver Fig. 3-c). Cabe mencionar que se pueden presentar restricciones ante el cambio de brillo de una imagen ocasionando que los algoritmos usados en el pre-procesamiento omitan algunos bordes del patrón. Lo anterior propicia que no se reconozcan los UIDPs correctamente dentro de la imagen, para solucionar este problema es necesario combinar otros algoritmos que mejoren el contraste y brillo de la imagen sin perder los bordes.

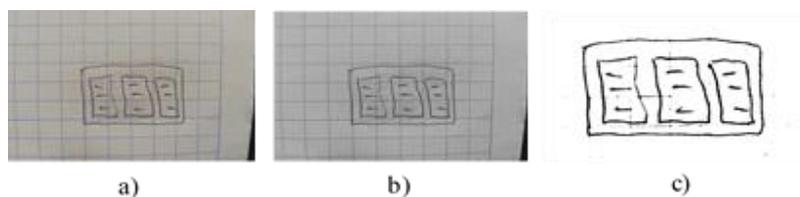


Fig. 3. Pasos del pre-procesamiento: a) imagen original de tipo real en blanco y negro, b) resultado del método de escala de grises y c) resultado del método de núcleo gaussiano.

Una vez realizado el pre-procesamiento de la imagen, se realiza el entrenamiento de la red neuronal convolucional y los resultados obtenidos se observan en la Tabla 6.

Tabla 6. Resultados de la prueba 4.

Prueba 4. TensorFlow-AlexNet		1400 imágenes
UIDPs	Precisión	Tiempo (seg.)
Galería	.99983644	8.24
Menú horizontal	.56006614	8.32
Menú vertical	.99995962	8.40
Music player	.9624471	8.18
Panel	.9980167	7.98
Picker	.8734556	7.91
Ranking	.98580235	7.85

699 batch y 10 épocas

Como se observa, los resultados obtenidos en la prueba 4 fueron mucho más exactos, obteniendo un 99% de eficiencia en la clasificación de los UIDPs y con un tiempo de respuesta de 8 segundos en promedio. Por lo tanto, este es el modelo final seleccionado para la clasificación de imágenes generadas a mano alzada que representen interfaces de usuario de aplicaciones *Social Media*.

5. Caso de estudio

En esta sección se aborda un caso de estudio para ejemplificar el funcionamiento del prototipo desarrollado.

En este caso de estudio, se seleccionaron 6 patrones de diseño adecuados para generar un prototipo de aplicación de Social Media. El prototipo es una aplicación para la reproducción de música y utiliza los UIDPs descritos a continuación:

- Galería: para desplegar al usuario las imágenes de álbumes disponibles para reproducir.
- Menú horizontal: despliega las opciones disponibles de la aplicación, como inicio, álbum y configuración.
- Music player: permite reproducir las diferentes pistas de música.
- Panel: permite ver las características del álbum o pista a reproducir.
- Picker: permite elegir el horario del temporizador para finalizar una canción.
- Ranking: Para calificar una canción o álbum.

Las imágenes utilizadas para representar la aplicación de este caso de estudio tienen dimensiones de 1050 por 650 píxeles en formato PNG. Los UIDPs generados se pueden observar en la Fig. 4.

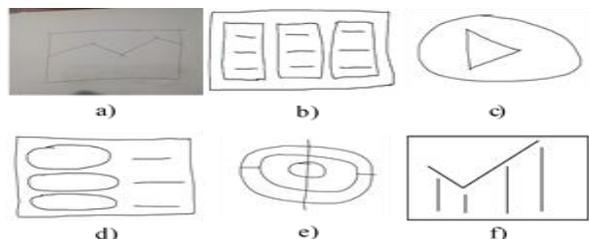


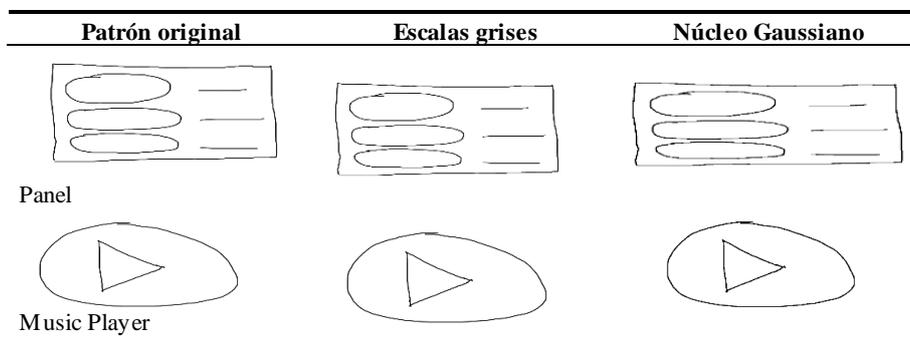
Fig. 4. UIDPs utilizados para la generación de una aplicación social media.: a) Galería, b) Menú horizontal, c) Music player, d) Panel, e) Picker y f) Ranking.

Para utilizar el prototipo de la red neuronal convolucional previamente entrenado, el primer paso es indicar la ruta de la imagen que se va utilizar, en este caso la primera imagen ingresada es una imagen real en blanco y negro que representa una Galería (ver Fig. 4-a), la segunda es una imagen no ideal que representa un Menú horizontal (ver Fig. 4-b), la tercera es una imagen no ideal que representa al Music player (ver Fig. 4-c), la cuarta es una imagen no ideal que representa un Panel (ver Fig. 4-d), la quinta es una imagen no ideal que representa un Picker (ver Fig. 4-e) y la sexta es una imagen ideal que representa un Ranking (ver Fig. 4-f).

El segundo paso, consiste en aplicar un pre-procesamiento a través de un conjunto de algoritmos de procesamiento de imágenes. En la Tabla 7 se observan los resultados obtenidos del pre-procesamiento aplicado a cada patrón.

Tabla 7. Imágenes de prueba: Ranking (ideal), Menú horizontal (no ideal) y Galería (real blanco y negro) con su respectivo preprocesamiento.

	Patrón original	Escalas grises	Núcleo Gaussiano
Ranking			
Menú horizontal			
Galería			
Picker			



Una vez aplicado el pre-procesamiento, el paso 3 consiste utilizar la red neuronal convolucional entrenada y los resultados obtenidos para cada patrón se observan en la Tabla 8.

Tal como se puede observar, los resultados de eficiencia de clasificación mostrados en la Tabla 8 obtienen porcentajes superiores al 70% en los patrones Rankin, Galería, Panel, Picker y Music player. Mientras que el Menú horizontal tiene resultados inferiores al 70%, sin embargo, su clasificación es correcta.

Tabla 8. Resultados obtenidos para la identificación de patrones: a = Ranking, b = Menú horizontal, c = Galería, d = Panel, e = Picker y f = Music player.

	a	b	c	d	e	f
a	.98580235	.0023476905	.0066808336	.0008515252	.0008665596	.0018575224
b	.008125944	.560063	.002517612	.0037316438	.00053650513	.0008372874
c	.0007908905	.0008986282	.9975389	2.825912e-05	.00013480928	.0006040252
d	.0005736141	.08623368	.0011079037	.7965732	.0006951616	.0010907252
e	.000964381	.007865683	.009596186	.010660858	.8734549	.09668942
f	.0001126410	.1.89209e-05	.0007694098	4.704886e-05	3.334518e-05	.99901175

Desde el entorno de Spyder se muestran las pantallas de salida con la clasificación de cada patrón (ver Fig. 5).

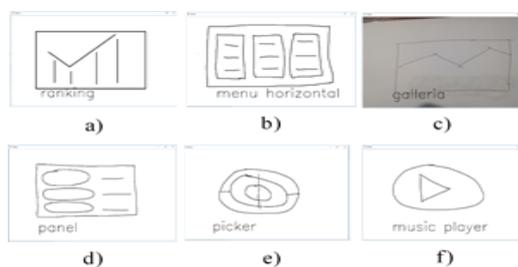


Fig. 5. Resultados de la clasificación de imágenes: a) Ranking, b) Menú horizontal, c) Galería, d) Panel, e) Picker y f) Music player.

Los resultados obtenidos determinan el correcto funcionamiento de la red neuronal convolucional y garantizan una adecuada clasificación, con lo que se tiene un margen de referencia que permitirá ampliar la investigación y abrir la oportunidad para crear un nuevo estándar o enfoque para el desarrollo de software. Lo anterior considerando que el tiempo promedio de respuesta es menor a 60 segundos. Con tal resultado se ve claramente el impacto que genera una red neuronal convolucional para la clasificación de imágenes.

6. Conclusiones y trabajo a futuro

Los resultados obtenidos gracias al apoyo de las técnicas de Deep Learning aportan un gran avance para estimular futuros desarrollos para diferentes ámbitos como *e-Commerce* y aplicaciones médicas. Este aporte en la ingeniería de software, da un panorama amplio para los desarrolladores que buscan procedimientos exactos, confiables y con costo menor de lo promediado. Cabe aclarar que la integración del Deep Learning, conlleva a una nueva línea de investigación para desarrollar e implementar aplicaciones en diferentes dominios, creando nuevas técnicas, métodos, paradigmas y metodologías que ampliarán una expectativa diferente a las técnicas convencionales que aún se trabajan.

El caso de estudio presentado demostró un claro ejemplo de que las diferentes imágenes de prueba (imágenes ideales, no ideales, reales en blanco y negro, y reales a color), pueden ser clasificadas sin ningún problema. Los resultados obtenidos no solamente obtuvieron un impacto importante para la clasificación, sino también el tiempo de espera para reconocer un patrón puede concluirse que es adecuado.

Este trabajo es una extensión de investigaciones previas que se presentan en Cortes-Camarillo et al. [3], Cortes-Camarillo et al. [4], Sánchez-Morales et al. [5] y Sánchez-Morales et al. [7] cuyas pruebas y resultados son un margen de referencia para mejorar los algoritmos de identificación de UIDPs en imágenes generadas a mano alzada.

Para el trabajo a futuro se pretende ampliar el abanico de patrones de diseño para este dominio y considerar nuevos dominios como *e-Commerce* y aplicaciones médicas. También se pretende generar nuevos modelos considerando otras APIs de Deep Learning para poner a prueba la eficiencia de la clasificación con nuevos UIDPs y diseños de aplicaciones más robustas que consideren la composición de patrones de diseño de interfaces de usuario.

Agradecimientos. Los autores agradecen al Instituto Tecnológico de México por su apoyo en este trabajo. Este trabajo de investigación fue patrocinado por el Consejo Nacional de Ciencia y Tecnología (CONACYT), así como por la Secretaría de Educación Pública (SEP) a través del PRODEP.

Referencias

1. Rosales-Morales, V.Y., Alor-Hernández, G., García-Alcaráz, J.L., Zatarain-Cabada, R., Barrón-Estrada, M.L.: An analysis of tools for automatic software development and automatic code generation. Facultad de Ingeniería, No. 77, pp. 75–87 (2015)

2. Cortes-Camarillo, C.A., Alor-Hernández, G., Olivares-Zepahua, B.A., Rodríguez-Mazahua, L., Peláez-Camarena, S.G.: Análisis comparativo de patrones de diseño de interfaz de usuario para el desarrollo. *Research in Computing Science*, vol. 126, pp. 31–41 (2016)
3. Cortes-Camarillo, C.A., Alor-Hernández, G., Sánchez-Morales, L.N., Rosales-Morales, V.Y., Rodríguez-Mazahua, L., Sánchez-Cervantes, J.L.: EduGene: A UIDP-Based Educational App Generator for Multiple Devices and Platforms. *International Journal of Human-Computer Interaction*, pp. 1–23 (2018)
4. Cortes-Camarillo, C.A., Rosales-Morales, V.Y., Sánchez-Morales, L.N., Alor-Hernández, G., Rodríguez-Mazahua, L.: Atila: A UIDPs-based educational application generator for mobile devices. In: *International Conference on Electronics, Communications and Computers (CONIELECOMP)*, pp. 1–7 (2017)
5. Sánchez-Morales, L.N., Alor-Hernández, G., Miranda-Luna, R., Rosales-Morales, V.Y., Cortes-Camarillo, C.A.: Generation of User Interfaces for Mobile Applications Using Neuronal Networks. *New Perspectives on Applied Industrial Tools and Techniques*, pp. 211–231 (2017)
6. da-Cost, S.L., Neto, V.V.G., de-Oliveira, J.L.: A User Interface Stereotype to build Web Portals. In: *9th Latin American Web Congress*, pp. 10–18 (2014)
7. Sánchez-Morales, L.N., Rosales-Morales, V.Y., Alor-Hernández, G., Posada-Gómez, R., Muñoz-Contreras, H., Juárez-Martínez, U.: Módulo de generación de aplicaciones multidispositivo a partir del procesamiento de imágenes. *Research in Computing Science on Computer science and computer engineering*, Issue 9, pp. 81–94 (2015)
8. Halbe, A., Joshi, A.R.: A Novel Approach to HTML Page Creation Using Neural Network. *Procedia Computer Science*, vol 45, pp. 197–204 (2015)
9. Baveye, Y., Cohendet, R., Da-Silva, M.P., Le-Callet, P.: Deep Learning for Image Memorability Prediction: The Emotional Bias. In: *Proceedings of the 2016 ACM on Multimedia Conference*, pp. 491–495 (2015)
10. Bianco, S., Buzzelli, M., Mazzini, D., Schettini, R.: Deep learning for logo recognition. *Neurocomputing*, vol. 245, pp. 23–30 (2017)
11. Liu, Y., Chen, X., Wang, Z., Wang, Z. J., Ward, R. K., Wang, X.: Deep learning for pixel-level image fusion: Recent Advances and future prospects. *Information Fusion*, vol. 42, pp. 158–173 (2018)
12. Pang, S., del-Coz, J.J., Yu, Z., Luaces, O., Díez, J.: Deep learning to frame objects for visual target tracking. *Engineering Applications of Artificial Intelligence*, vol. 65, pp. 406–420 (2017)
13. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: *NIPS'12 Proceedings of the 25th International Conference on Neural Information Processing Systems*, vol. 1, pp. 1–9 (2012)
14. Huang, C., Xu, H., Xie, L., Zhu, J., Xu, C.: Large-scale Semantic Web Image Retrieval Using Bimodal Deep Learning Techniques. *Information Sciences*, pp. 1–39 (2017)
15. Nedzved, A., Gurevich, I., Trusova, Y., Ablameyko, S.: Software Development Technology with Automatic Configuration to Classes of Image Processing Problems. *Software and Hardware for pattern recognition and image analysis*, vol. 23, no. 2, pp. 269–277 (2013)
16. What is Python? Executive Summary. <https://www.python.org/doc/essays/blurb/>. Último acceso: 2018/09/26
17. Gonzalez, R., Woods, R.: *Digital image processing*. 4th ed. Pearson, New Jersey (2017)

Selección óptima de variables mediante metaheurísticas binarias para la detección de fallas en un proceso industrial multivariado usando aprendizaje de máquina

Jesús Alejandro Navarro Acosta¹, Edgar O. Reséndiz Flores²,
Irma D. García Calvillo¹

¹ Universidad Autónoma de Coahuila, Centro de Investigación en Matemáticas
Aplicadas, Saltillo, Coahuila

² Instituto Tecnológico de Saltillo, División de Estudios de Posgrado e Investigación.
Saltillo, Coahuila

alejandro.navarro@uadec.edu.mx, eresendiz@itsaltillo.edu.mx

Resumen. Este trabajo presenta la comparación entre la optimización por enjambre de partículas y el algoritmo de búsqueda gravitacional en sus versiones binarias en combinación con la distancia Mahalanobis para la selección óptima de variables en un proceso de soldadura de discos de soporte para el mecanismo de los asientos en el automóvil. Los resultados muestran que la metodología implementada es capaz de seleccionar un subconjunto de variables capaz de mantener e incluso aumentar la exactitud en la detección de fallas en el proceso, así como brindar a los encargados del proceso las variables que intervienen en la variabilidad del mismo. Lo anterior se valida mediante la implementación de 16 algoritmos y enfoques de aprendizaje automático supervisado y otro no supervisado ampliamente utilizados para tareas de reconocimiento de patrones y detección de fallas.

Palabras clave: metaheurísticas binarias, distancia Mahalanobis, aprendizaje automático.

Optimal Feature Selection through Binary Metaheuristics for Fault Detection in a Multivariate Industrial Process Using Machine Learning

Abstract. This work presents the comparison between binary particle swarm optimization and binary gravitational search algorithm in combination with the Mahalanobis distance for the optimal feature selection in a welding process of support discs for the seat mechanism in the car. The results show that the implemented methodology is able to select a subset of variables capable of maintaining and even increasing the accuracy for fault detection in the process. This is validated through

the implementation of several supervised and one unsupervised machine learning algorithms, which are widely used for pattern recognition tasks.

Keywords: binary metaheuristics, Mahalanobis distance, machine learning.

1. Introducción

En la actualidad el sector industrial se enfrenta día a día con el reto de lograr altos niveles de calidad en sus productos. Por tal motivo, los diferentes departamentos involucrados en la producción de dichos productos implementan diversos procedimientos y técnicas con el fin de minimizar los defectos de calidad. Sin embargo debido al gran avance de la tecnología, los procesos son cada vez más complejos demandando técnicas y metodologías cada vez más sofisticadas y robustas para su análisis. En la industria actual los procesos pueden presentar características no lineales, presencia de ruido y una alta cantidad de variables que pueden presentar correlación entre si. Dichas características han hecho complicado y en muchas veces ineficiente el análisis de estos procesos mediante técnicas clásicas. Actualmente la inteligencia artificial ha demostrado resultados satisfactorios al analizar procesos y conjuntos de datos con las características antes mencionadas mediante el desarrollo de sistemas para la detección temprana de fallas, y así evitar defectos de calidad en componentes o en el producto final. De acuerdo con Chiang [4] una falla se define como una desviación no autorizada de al menos una característica o variable del sistema. En este sentido un algoritmo que sea capaz de reconocer patrones puede ser implementado con el fin de detectar fallas en un sistema o proceso. En [19], Vapnik establece que junto con regresión y estimación de la densidad, la clasificación es una de las tres tareas principales del aprendizaje automático, es decir clasificación se refiere a construir un algoritmo capaz de clasificar una muestra en un conjunto de clases dadas sin implicación humana. Siguiendo esta lógica un algoritmo de aprendizaje automático funcional será capaz de distinguir a partir de un conjunto de datos una falla presente en el proceso con un determinado porcentaje de error. Debido a que obviamente se busca disminuir el error, además del desarrollo de algoritmos más potentes se llevan a cabo varios enfoques para disminuir el error en algoritmos ya existentes. Uno de estos enfoques es el preprocesamiento de datos el cual tiene como objetivo adecuar la información para su posterior análisis, dicho preprocesamiento puede incluir la extracción de características, la limpieza de ruido en señales y la selección de variables entre otros. Debido al avance de la tecnología y la informática la capacidad de adquirir información de los procesos industriales ha aumentado en gran manera. Por lo tanto, los conjuntos de datos a analizar son cada vez más grandes no solo en el número de muestras u observaciones que se pueden obtener sino también en el número de variables que interactúan entre si. Por tal motivo el enfoque de selección de variables se ha vuelto uno de los preprocesamientos mas estudiados. El objetivo de este enfoque es encontrar un subconjunto de variables

el cual permita describir los procesos de igual manera o mejor que el conjunto completo de datos. Esta idea se sustenta en el hecho de que en conjuntos de alta dimensionalidad puede existir información redundante o poco útil la cual aumenta la complejidad de los modelos para la detección de fallas. Además estos se vuelven computacionalmente costosos [3], es decir produce modelos con baja exactitud y que consumen gran tiempo al llevar a cabo la fase de entrenamiento. Como se mencionó anteriormente el analizar un proceso industrial con el fin de detectar fallas es una tarea desafiante tomando en cuenta las características de los mismos en la actualidad. Por tal motivo este trabajo de investigación se enfoca en la implementación de dos metaheurísticas binarias para la construcción de un modelo el cual sea capaz de seleccionar las variables útiles y así mejorar la detección de fallas en un proceso industrial multivariado.

2. Técnicas

2.1. Metaheurísticas

La optimización estocástica es la clase general de algoritmos y técnicas que emplean cierto grado de aleatoriedad para encontrar soluciones óptimas (o tan óptimas como sea posible) a problemas difíciles. Las metaheurísticas son los más generales de este tipo de algoritmos, y se aplican a una gama muy amplia de problemas [9]. En la literatura no existe una única definición de metaheurística, una de ellas es: “Una metaheurística es un conjunto de conceptos que pueden ser utilizados para definir métodos heurísticos que se pueden aplicar a un amplio conjunto de problemas diferentes. En otras palabras, una metaheurística puede ser vista como un marco algorítmico general que puede aplicarse a diferentes problemas de optimización con relativamente pocas modificaciones para hacerlas adaptables a un problema específico” [13]. Los algoritmos metaheurísticos son capaces de resolver problemas complejos de optimización donde otros métodos de optimización no son eficaces o eficientes.

Optimización por enjambre de partículas (PSO). El algoritmo de optimización con enjambre de partículas (PSO) fue desarrollado por J. Kennedy y R. C. Eberhart, el cual es un método de optimización metaheurístico basado en la simulación del comportamiento social de las aves, abejas, bancos de peces, entre otros. A diferencia de otros algoritmos evolutivos como algoritmos genéticos (GA), PSO es más simple ya que no contiene operaciones de mutación o cruce, lo que reduce la complejidad del modelo [7]. En PSO cada individuo en el enjambre es representado por un vector en un espacio de búsqueda multidimensional. Este vector tiene un vector de velocidad asignado que determina el siguiente movimiento de la partícula. Cada partícula actualiza su velocidad basándose en la velocidad actual y la mejor posición que ha explorado hasta ahora. Y también basándose en la mejor posición explorada por todo el enjambre [17]. PSO fue originalmente desarrollado para espacios de valores continuos, sin embargo muchos problemas son definidos para espacios de valor discreto donde el dominio de las variables es finito.

PSO binario. En 1997, Kennedy and Eberhart [8] introducen una versión binaria de PSO para problemas de optimización discretos. En PSO binario (BPSO) cada partícula representa su posición en valores binarios, donde cada valor de la partícula puede ser cambiado de 1 a 0 o viceversa.

Considere un espacio de búsqueda d -dimensional y a la i -ésima partícula del enjambre denotada como $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{id})^t$ cuya velocidad es $\mathbf{v}_i = (v_{i1}, v_{i2}, \dots, v_{id})^t$. Por lo tanto las ecuaciones que guían la dinámica de la partícula en PSO continuo son:

$$\mathbf{v}_i(t+1) = w\mathbf{v}_i(t) + c_1\phi_1(\mathbf{p}_{ibest} - \mathbf{y}_i(t)) + c_2\phi_2(\mathbf{p}_{gbest} - \mathbf{y}_i(t)), \quad (1)$$

$$\mathbf{x}_i(t+1) = \mathbf{y}_i(t) + \mathbf{v}_i(t+1), \quad (2)$$

donde $\mathbf{v}_i(t+1)$ y $\mathbf{x}_i(t+1)$ son la velocidad y posición siguientes de cada partícula. Mientras c_1 y c_2 son constantes positivas, y ϕ_1 y ϕ_2 son variables aleatorias con distribución uniforme entre 0 y 1. Las mejores posiciones locales y globales son \mathbf{p}_{ibest} y \mathbf{p}_{gbest} respectivamente. El peso de inercia sobre un nuevo vector se denota por w [5].

La principal diferencia entre PSO y BPSO es la forma en que las velocidades cambian para actualizar las posiciones de las partículas. La versión binaria necesita una función para generar valores en el conjunto binario $\{0, 1\}$ que son los únicos estados posibles de la partícula, por lo tanto la función sigmoide es utilizada para decidir la probabilidad de generar tales números binarios. Esta función se aplica bit a bit en una partícula dada y se define como sigue:

$$sigm(v_{ik}) = \frac{1}{1 + e^{-v_{ik}}}. \quad (3)$$

La posición de la partícula correspondiente se actualiza en base a la siguiente regla

$$y_{ik}(t+1) = \begin{cases} 1 & \text{si } r < sigm(v_{ik}) \\ 0 & \text{De lo contrario,} \end{cases} \quad (4)$$

donde r es un número pseudo aleatorio seleccionado de una distribución uniforme entre $[0, 1]$.

Algoritmo de búsqueda gravitacional (GSA). GSA es un algoritmo de optimización basado en la ley de la gravedad y la interacción de las masas. En la ley gravitacional de Newton las partículas se atraen unas a otras con una fuerza gravitacional. Dicha fuerza entre dos cuerpos es directamente proporcional al producto de sus masas e inversamente proporcional al cuadrado de sus distancias. En este algoritmo cada masa cuenta con posición, masa inercial, masa gravitacional activa y otra pasiva. La posición de la masa corresponde a una posible solución, y sus masas gravitacionales e inerciales son determinadas

usando una función de ajuste. Los autores en [15] establecen que GSA obedece a las siguientes leyes:

- Ley de gravedad: Cada partícula atrae a todas las demás partículas y la fuerza gravitacional entre dos partículas es directamente proporcional al producto de sus masas e inversamente proporcional a la distancia entre ellas.
- Ley de movimiento: La velocidad actual de cualquier masa es igual a la suma de la fracción de su velocidad anterior y la variación en la misma. La variación en la velocidad o aceleración de cualquier masa es igual a la fuerza en el sistema dividida por la inercia.

La velocidad y la posición de cada masa está dada por:

$$\mathbf{v}_i^d(t+1) = rand_i \times \mathbf{v}_i^d(t) + a_i^d(t), \quad (5)$$

$$X_i^d(t+1) = x_i^d(t) + \mathbf{v}_i^d(t+1), \quad (6)$$

donde $a_i^d(t)$ es la aceleración del agente i en el tiempo t y en dirección a d , y se calcula de la siguiente manera:

$$a_i^d(t) = \frac{F_i^d(t)}{M_{ii}(t)}. \quad (7)$$

En un espacio d -dimensional la fuerza total que actúa sobre un agente i es

$$F_i^d(t) = \sum_{i \neq j} rand_j F_{ij}^d. \quad (8)$$

Las masas gravitacionales e inerciales están dadas por:

$$M_i(t) = \frac{q_i(t)}{\sum_{j=1} q_j(t)}, \quad (9)$$

donde,

$$q_i(t) = \frac{fit_i - worst(t)}{best(t) - worst(t)}, \quad (10)$$

para problemas de minimización $best(t)$ y $worst(t)$ se calculan como:

$$best(t) = \min fit_j(t), \quad (11)$$

$$worst(t) = \max fit_j(t), \quad (12)$$

$$F_{ij}^d = G(t) \frac{M_{pi}(t) \times M_{aj}(t)}{R_{ij}(t) + \varepsilon} (x_j^d(t) - x_i^d(t)). \quad (13)$$

M_{aj} es la masa gravitacional activa aplicada al agente j , M_{pi} es la masa gravitacional pasiva relacionada al agente i . $G(t)$ es la constante gravitacional en el tiempo t , ε es una constante, y $R_{ij}(t)$ es la distancia Euclídea entre los agentes i y j . Para detalles de las ecuaciones revise [15] y [11].

Al igual que PSO, GSA se modificó para resolver problemas binarios. En [16] se presenta una versión binaria de GSA (BGSA) donde la actualización del valor de la posición de la masa alterna entre 0 y 1. Esto se hace de acorde a su velocidad usando la función \tanh de la siguiente manera:

$$X_i^d(t+1) = \begin{cases} x_i^d(t) & \text{if } rand_i \leq |\tanh(v_i^d(t+1))|, \\ \overline{x_i^d(t)} & \text{De lo contrario.} \end{cases} \quad (14)$$

Para detalles del algoritmo revisar [12] y [16].

3. Distancia Mahalanobis (MD)

La MD es una distancia generalizada que es útil para determinar las similitudes entre conjuntos de datos desconocidos y conocidos. Mide las distancias en espacios multidimensionales; teniendo en cuenta las correlaciones entre cualquier variable o característica que pueda existir [18]. La MD es una medida de distancia que se ha utilizado en aplicaciones como la detección de anomalías, el reconocimiento de patrones y el control de procesos. Además, no es sensible a las diferentes escalas de los parámetros monitoreados, ya que los valores de MD se calculan utilizando parámetros normalizados. La distancia Mahalanobis de un conjunto de datos se calcula de la siguiente manera:

$$MD = \frac{1}{k} Z_{ij} C^{-1} Z_{ij}^T, \quad (15)$$

donde Z_i es el vector normalizado obtenido al normalizar los valores de X_i donde $i = 1, 2, \dots, k$:

$$Z_{ij} = \frac{X_{ij} - \overline{X}_i}{S_i}, \quad i = 1, 2, \dots, k; \quad j = 1, 2, \dots, n \quad (16)$$

$$\overline{X}_i = \frac{1}{n} \sum_{j=1}^n X_{ij}, \quad S_i = \sqrt{\frac{\sum_{j=1}^n (X_{ij} - \overline{X}_i)^2}{(n-1)}} \quad (17)$$

donde X_{ij} es el valor correspondiente a la i -ésima variable en la j -ésima observación y C es la matriz de correlación [14]:

$$C = \frac{1}{(n-1)} \sum_{j=1}^n Z_j Z_j^T. \quad (18)$$

4. Metodología para la detección de variables

Para llevar a cabo la selección de variables se implementa una metaheurística binaria con el fin de minimizar la siguiente función de aptitud:

$$\min f(\mathbf{x}) = \beta f_1(\mathbf{x}) + (1 - \beta) \frac{p_{\text{seleccionadas}}}{p}, \quad (19)$$

sujeto a:

$$\sum_{i=1}^p x_i \leq p, \quad \sum_{i=1}^p x_i = p_{\text{seleccionadas}}, \quad f_1(\mathbf{x}) \leq f_1^{\max},$$

donde $f_1(\mathbf{x})$ es una función que mide el error en la clasificación binaria de los datos, mientras $p_{\text{seleccionadas}}$ es el subconjunto de variables seleccionado. Siguiendo el enfoque de la metodología llamada sistema Mahalanobis-Taguchi (MTS) [6]. La distancia Mahalanobis es utilizada con el fin de clasificar los datos en dos clases, conocidas como datos saludables y no saludables. En este sentido $f_1(\mathbf{x})$ se calcula de la siguiente manera:

$$f_1(\mathbf{x}) = w_1 \frac{n_1^e}{n_1} + w_2 \frac{n_2^e}{n_2}, \quad (20)$$

donde n_1^e es el número de observaciones saludables clasificadas como no saludables, n_2^e es el número de las no saludables clasificadas como saludables, n_1 y n_2 es el total de observaciones saludables y no saludables respectivamente. Mientras que w_1 y w_2 son la contribución a cada tipo de clasificación errónea. Para obtener n_1^e y n_2^e se proceden a calcular las distancias mahalanobis de los datos saludables y no saludables y se calcula lo siguiente:

$$n_1^e = \sum_{j=1}^{n_1} MD_j^1 \quad \text{s.t.} \quad MD_1^2 \leq MD_j^1, \quad (21)$$

$$n_2^e = \sum_{j=1}^{n_2} MD_j^2 \quad \text{s.t.} \quad MD_{n_1}^1 \geq MD_j^2. \quad (22)$$

Por último, para llevar a cabo la selección óptima de variables se usa la metaheurística binaria de la siguiente manera:

Sea $\mathbf{x} = (x_1, x_2, \dots, x_p)^t$ un vector de dimensión p donde

$$x_i = \begin{cases} 0 & \text{si la variable } i \text{ no es seleccionada,} \\ 1 & \text{si la variable } i \text{ es seleccionada.} \end{cases} \quad (23)$$

De esta manera la metaheurística binaria retiene solo el subconjunto de variables ($p_{\text{seleccionadas}}$) que minimicen la función de aptitud 19. En este enfoque se comparará el desempeño de las metaheurísticas binarias BPSO y GSA.

5. Aplicación industrial: Soldadura en mecanismo para asiento de coche

A continuación se presenta la metodología antes descrita usando BPSO y BGSA con el fin de comparar el desempeño de cada metaheurística. Dicha

metodología se aplica en un proceso de soldadura de discos de soporte para el mecanismo de los asientos en el automóvil. La figura 1 muestra algunos de los componentes a los cuales el disco es unido mediante soldadura láser. El proceso de soldadura implica 18 variables dimensionales relevantes. Butwell y Lapwell son dos modos de soldadura en el proceso y las características se miden en diferentes cortes realizados en una unión de material, las variables dimensionales correspondientes a diferentes cortes en las uniones soldadas se muestran en la Figura 2 y se enlistan en la Tabla 1. Para fines experimentales, se recolectaron un total de 94 observaciones, de las cuales 63 corresponden a medidas correspondientes a piezas que satisfacen los criterios de calidad (operación normal) y las 31 restantes no las cumplieron (operación defectuosa).



Fig. 1. Soporte de disco y asiento de automóvil.

5.1. Resultados experimentales

Para el cálculo de la función de clasificación errónea (Ec. 20) se consideran tres configuraciones, caso 1: $w_1 = w_2 = 0,5$, caso 2: $w_1 = 0,95$, $w_2 = 0,05$ y caso 3: $w_1 = 0,05$, $w_2 = 0,95$. β se fija en 0.9 en concordancia con Vignolo [20], quien recomienda fijar este valor entre 0.7 y 0.9. Para las metaheurísticas BPSO y GSA, el número de individuos que conforman la población es de 100 y el número máximo de iteraciones es de 200. En la Tabla 2 se muestra que el valor alcanzado para la función de aptitud (Ec. 19) por BPSO y BGSA es el mismo, de igual manera en este caso de estudio las variables detectadas por ambos enfoques coinciden. En las Figuras 3-5 se muestra que BGSA converge mas rápido que BPSO en los tres casos de experimentación planteados. Con el fin de probar el efecto de las variables seleccionadas, se entrenan otros algoritmos de aprendizaje automático supervisado y uno no supervisado ampliamente usados en la detección de fallas. En este experimento se entrenan dichos enfoques con las 63 observaciones en operación normal y las 31 observaciones en operación defectuosa con el fin de obtener la exactitud en la clasificación de este conjunto de datos por parte de los algoritmos, dicho porcentaje de exactitud es obtenido

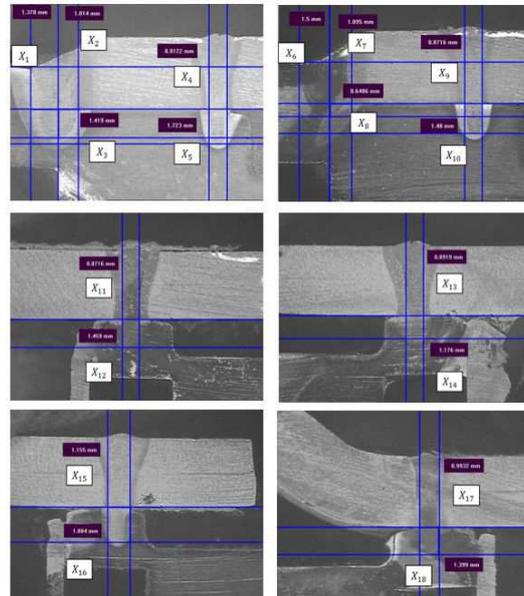


Fig. 2. Variables dimensionales en cortes del disco.

usando validación cruzada de 10 subconjuntos. Primero estos se entrenan considerando la totalidad de las variables (18 variables), y posteriormente se entrenan usando solo el subconjunto de variables seleccionadas. Podemos ver en la Tabla 3 que de 16 enfoques de aprendizaje automático supervisado cuatro de ellos obtuvieron el mismo porcentaje de exactitud para ambos experimentos. Otros cuatro obtuvieron un porcentaje mayor usando la totalidad de las variables, mientras que 8 de los algoritmos incluso aumentaron su exactitud al ser entrenados solo con el subconjunto de variables detectados por la metodología.

Finalmente se prueba el efecto de la selección de variables usando el algoritmo de aprendizaje no supervisado k-medias el cual es uno de los métodos más usados para agrupamiento o clustering. Sus posibles aplicaciones son agrupamiento por similitud, predicción no lineal, aproximaciones de distribuciones multivariadas y pruebas no paramétricas de independencia entre varias variables, además de ser relativamente sencillo de implementar y computacionalmente económico [10]. Para detalles referirse a [1]. Para este experimento se usa la base de datos con las 63 observaciones en operación normal y las 31 observaciones en operación defectuosa formando una base de datos de 94 observaciones con 18 variables para el primer caso y otra usando solo las variables seleccionadas, es decir de 94 observaciones y 7 variables. Posteriormente se elige $k=2$, es decir k-medias formará dos conglomerados de manera no supervisada. Posteriormente se evaluará el contenido de los mismos, es decir como agrupó las observaciones normales y las defectuosas ya que de antemano se conoce cuáles de estas observaciones pertenecen a una u otra clase. Para evaluar este algoritmo se usaron diversas

Tabla 1. Descripción de las variables

	Descripción
x_1	Ancho 1 de soldadura Butwell corte 1, en unión (disco A)
x_2	Ancho 2 de soldadura Butwell corte 1, en unión (disco A)
x_3	Penetración de soldadura, Butwell corte 1, en unión (disco A)
x_4	Ancho 1 de soldadura Butwell corte 2, en unión (disco A)
x_5	Ancho 2 de soldadura Butwell corte 2, en unión (disco A)
x_6	Penetración de soldadura, Butwell corte 2, en unión (disco A)
x_7	Ancho de soldadura Lapwell corte 1, en unión (disco A)
x_8	Penetración de soldadura, Lapwell corte 1, en unión (disco A)
x_9	Ancho de soldadura Lapwell corte 2, en unión (disco A)
x_{10}	Penetración de soldadura, Lapwell corte 2, en unión (disco A)
x_{11}	Ancho de soldadura Lapwell corte 1, en unión (disco B)
x_{12}	Penetración de soldadura, Lapwell corte 1, en unión (disco B)
x_{13}	Ancho de soldadura Lapwell corte 2, en unión (disco B)
x_{14}	Penetración de soldadura, Lapwell corte 2, en unión (disco B)
x_{15}	Ancho de soldadura Lapwell corte 3, en unión (disco B)
x_{16}	Penetración de soldadura, Lapwell corte 3, en unión (disco B)
x_{17}	Ancho de soldadura Lapwell corte 4, en unión (disco B)
x_{18}	Penetración de soldadura, Lapwell corte 4, en unión (disco B)

Tabla 2. Valores óptimos y variables seleccionadas para las diferentes configuraciones.

	Metaheurística	Valores óptimos	Variables seleccionadas
Caso 1	PSO	0.0389	$x_2, x_3, x_6, x_8, x_{10}, x_{11}, x_{13}$
	GSA	0.0389	$x_2, x_3, x_6, x_8, x_{10}, x_{11}, x_{13}$
Caso 2	PSO	0.0389	$x_2, x_3, x_6, x_8, x_{10}, x_{11}, x_{13}$
	GSA	0.0389	$x_2, x_3, x_6, x_8, x_{10}, x_{11}, x_{13}$
Caso 3	PSO	0.0389	$x_2, x_3, x_6, x_8, x_{10}, x_{11}, x_{13}$
	GSA	0.0389	$x_2, x_3, x_6, x_8, x_{10}, x_{11}, x_{13}$

métricas como la exactitud (Ac), la precisión (Pr), exhaustividad (Re) y la conocida como F-score [2], las cuales se calculan de la siguiente manera.

$$Ac = \frac{TP+TN}{TP+FN+FP+TN}, \quad (24)$$

$$Pr = \frac{TP}{TP+FP}, \quad (25)$$

$$Re = \frac{TP}{TP+FN}, \quad (26)$$

$$F\text{-score} = \frac{2 \times Re \times Pr}{Re+Pr}, \quad (27)$$

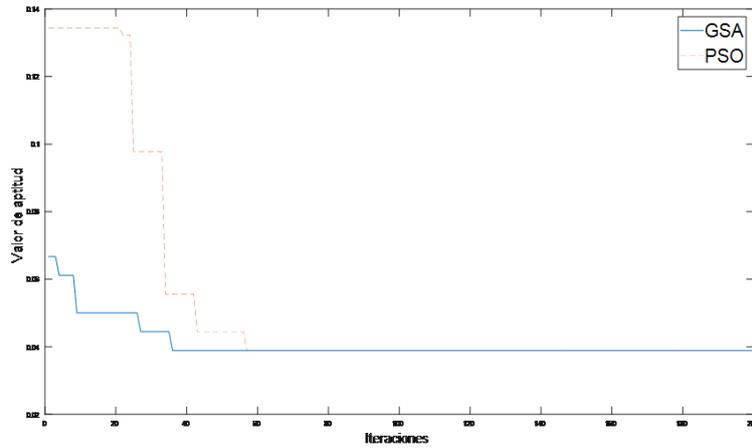


Fig. 3. Convergencia en el caso 1: $w_1 = w_2 = 0,5$

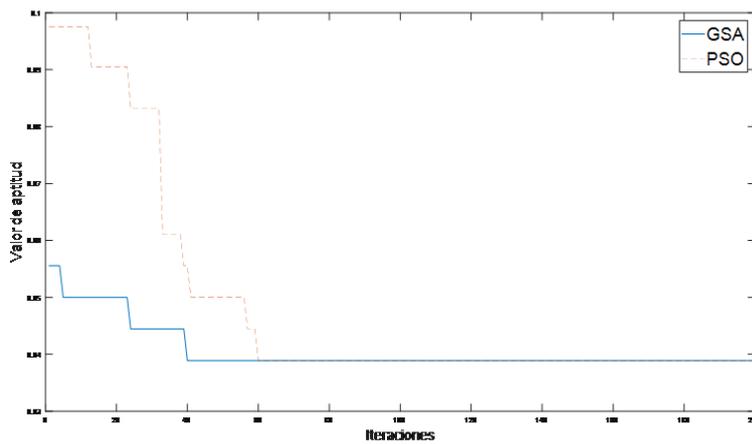


Fig. 4. Convergencia en el caso 2: $w_1 = 0,95, w_2 = 0,05$

donde TP y TN se refieren a los verdaderos positivos y negativos respectivamente, mientras FN y FP son los falsos negativos y falsos positivos. Los resultados de ambos casos se presentan en la Tabla 4.

Como se observa, los valores de las métricas alcanzados por k-medias son bajos para implementar dicho algoritmo para la detección de fallas. Sin embargo cabe señalar que se usó la versión básica del algoritmo, y es importante que el lector esté consciente de que existen diversos enfoques y estrategias para mejorar el rendimiento de k-medias. En este estudio solo se probó el efecto de implementar el algoritmo con el total de las variables y con las variables seleccionadas por el enfoque presentado. Siendo que este último logro valores ligeramente superiores.

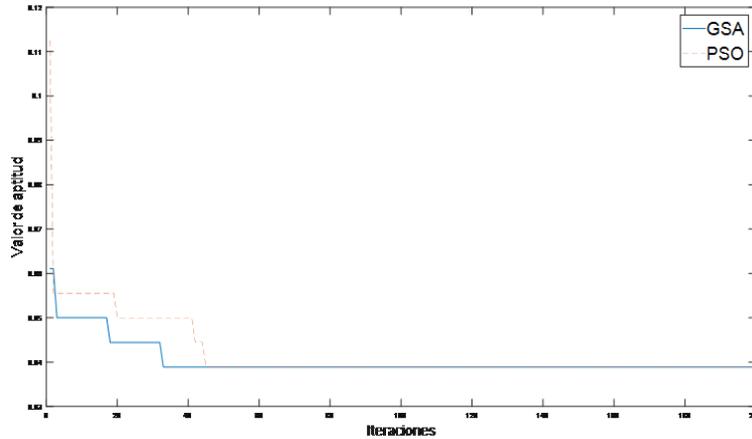


Fig. 5. Convergencia en el caso 3: $w_1 = 0,05, w_2 = 0,95$

Tabla 3. Exactitud % de otros algoritmos de aprendizaje de máquina supervisado

Algoritmo	Todas las variables	Variables seleccionadas
Complex tree	80.9	85.1
Medium tree	80.9	85.1
Simple tree	72.3	81.9
Logistic regression	64.9	71.3
Linear SVM	68.1	74.5
Quadratic SVM	84.0	86.2
Cubic SVM	79.8	80.9
Fine Gaussian SVM	69.1	69.1
Medium Gaussian SVM	94.7	90.4
Coarse Gaussian SVM	68.1	68.1
Fine KNN	80.9	79.8
Medium KNN	67.0	68.1
Coarse KNN	67.0	67.0
Cosine KNN	71.3	68.1
Cubic KNN	68.1	67.0
Weighted KNN	69.1	69.1

6. Conclusiones y trabajo futuro

En este artículo se presentó una herramienta basada en metaheurísticas binarias y la distancia Mahalanobis para la selección óptima de características aplicada a un proceso de soldadura de discos de soporte para el mecanismo de los asientos en el automóvil. La metodología combina las ventajas discriminantes de la distancia Mahalanobis y las capacidades de PSO y GSA para encontrar la solución óptima de un problema de optimización con variables binarias. Esto

Tabla 4. Evaluación del algoritmo k-medias

Métrica	Todas las variables	Variables seleccionadas
Exactitud	0.5638	0.5745
Precisión	0.7037	0.7018
Exhaustividad	0.6032	0.6349
F-score	0.6496	0.6667

con el fin de seleccionar las variables que impactan en los defectos del producto. Como se demostró en la experimentación, ambas metaheurísticas alcanzaron el mismo valor en la función de la aptitud. De igual manera ambas detectaron el mismo subconjunto de variables, lo que permite a las personas encargadas del proceso tomar acciones para controlar estas variables. Por otro lado GSA convergió mas rápido que PSO en los tres casos de experimentación. Además, el efecto de este subconjunto de características seleccionadas se ha implementado en diversos algoritmos de aprendizaje automático supervisado y no supervisado. Donde en ocho de los supervisados y el no supervisado obtuvieron valores mayores de exactitud al ser entrenados solo con el subconjunto de variables seleccionadas por la metodología. Como trabajo futuro se esta interesado en la incorporación de otras metaheurísticas híbridas para resolver el problema de minimización presentado en este trabajo. Así como implementar enfoques para mejorar la capacidad de clasificación de la distancia Mahalanobis.

Referencias

1. Abualigah, L.M., Khader, A.T., Hanandeh, E.S.: Hybrid clustering analysis using improved krill herd algorithm. *Applied Intelligence* 48(11), 4047–4071 (Nov 2018)
2. Abualigah, L.M., Khader, A.T., Hanandeh, E.S.: A new feature selection method to improve the document clustering using particle swarm optimization algorithm. *Journal of Computational Science* 25, 456–466 (2018)
3. Chandrashekar, G., Sahin, F.: A survey on feature selection methods. *Computers and Electrical Engineering* 40, 16–28 (2014)
4. Chiang, L.H., Russell, E.L., Braatz, R.D.: *Fault Detection and Diagnosis in Industrial Systems*. Springer-Verlag London Ltd. (2001)
5. Das, S., Abraham, A., Konar, A.: *Particle swarm optimization and differential evolution algorithms. technical analysis, applications and hybridization perspectives*. *Studies in Computational Intelligence* (2008)
6. Ghasemi, E., Aaghaie, A., Cudney, E.A.: Mahalanobis taguchi system: a review. *International Journal of Quality & Reliability Management* Vol. 32 (2015)
7. Haixiang, G., Yijing, L., Yanan, L., Xiao, L., Jinling, L.: BPSO-adaboost-KNN ensemble learning algorithm for multi-class imbalanced data classification. *Engineering Applications of Artificial Intelligence* (2016)
8. Kennedy, J., Eberhart, R.: *Particle swarm optimization*. *International conference on neural networks*, IEEE (1995)
9. Luke, S.: *Essentials of Metaheuristics*. Lulu, second edn. (2013), available for free at <http://cs.gmu.edu/~sean/book/metaheuristics/>

10. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics. pp. 281–297. University of California Press, Berkeley, Calif. (1967)
11. Mirjalili, S., Hashim, S.Z.M.: A new hybrid PSO-GSA algorithm for function optimization. In: 2010 International Conference on Computer and Information Application. pp. 374–377 (Dec 2010)
12. Mirjalili, S., Wang, G.G., Coelho, L.d.S.: Binary optimization using hybrid particle swarm optimization and gravitational search algorithm. *Neural Computing and Applications* 25(6), 1423–1435 (Nov 2014), <https://doi.org/10.1007/s00521-014-1629-6>
13. Osman, I., Laporte, G.: Metaheuristics: A bibliography. *Ann Oper Res* (1996)
14. Patil, N., Das, D., Pecht, M.: Anomaly detection for igbts using mahalanobis distance. *Microelectronics Reliability* 55(7), 1054–1059 (2015), <http://www.sciencedirect.com/science/article/pii/S0026271415000888>
15. Rashedi, E., Nezamabadi-pour, H., Saryazdi, S.: GSA: A Gravitational Search Algorithm. *Information Sciences* 179(13), 2232–2248 (2009), special Section on High Order Fuzzy Sets
16. Rashedi, E., Nezamabadi-pour, H., Saryazdi, S.: BGSA: binary gravitational search algorithm. *Natural Computing* 9(3), 727–745 (Sep 2010)
17. Reséndiz-Flores, E.O., Rull-Flores, C.A.: Mahalanobis-Taguchi system applied to variable selection in automotive pedals components using gompertz binary particle swarm optimization. *Expert Systems with Applications* (2013)
18. Rizal, M., Ghani, J., Nuawi, M., Haron, C.: Cutting tool wear classification and detection using multi-sensor signals and Mahalanobis-Taguchi system. *Wear* 376–377, 1759–1765 (2017), 21st International Conference on Wear of Materials
19. Vapnik, V.: *The Nature of Statistical Learning Theory*. Springer (2000)
20. Vignolo, L.D., Milone, D.H., Scharcanski, J.: Feature selection for face recognition based on multi-objective evolutionary wrappers. *Expert Systems with Applications* 40(13), 5077–5084 (2013)

Propuesta de red neuronal convolutiva para la predicción de partículas contaminantes PM10

Ricardo Domínguez-Guevara, María del Carmen Cabrera-Hernández,
Marco Antonio Aceves-Fernández, Jesús Carlos Pedraza-Ortega

Universidad Autónoma de Querétaro, Facultad de Ingeniería, Querétaro, México
ricardo.dominguezguevara@gmail.com, k.armenkabrera@hotmail.com,
marco.aceves@gmail.com, caryoko@yahoo.com

Resumen. La materia particulada o partículas suspendidas (PM) son partículas contaminantes que forman parte del aire que respiramos; un subgrupo de PM son las partículas con diámetro menor a 10 micrómetros, las cuales son nombradas PM 10. Este último subgrupo de partículas ha sido reconocido dentro del conjunto de contaminantes que mayor impacto tiene en la salud de las personas; son un factor importante en enfermedades cardiovasculares, que a su vez desarrollan otro tipo de padecimientos. Por esta razón, el poder contar con modelos que permitan determinar el comportamiento de PM 10 es fundamental para la prevención de algunas enfermedades. Por otra parte, en años recientes se ha explorado el uso de redes neuronales convolutivas (Convolutional Neural Networks o CNN) aplicadas a la predicción de series de tiempo, demostrando un gran potencial en dicho propósito. En el presente trabajo se implementa una arquitectura de CNN aplicada a la predicción de PM 10, utilizando distintas variables meteorológicas para el modelado de PM 10.

Palabras clave: red neuronal convolutiva, redes profundas, partículas contaminantes, PM 10, predicción, CNN.

Convolutional Neural Network Proposal for Particulate Matter PM10 Prediction

Abstract. Particulate matter or suspended particles (PM) are pollutant particles that form the air we breathe; a subgroup of PM are the particles with a diameter lower than 10 micrometers, those particles are called PM 10. This subgroup has been recognized in the group of pollutants that has higher impact on people health; they are an important factor on cardiovascular diseases that also develop another health problem. For this reason, counting with models that let us determine the behavior of PM 10 is crucial for the prevention of some diseases. In the other hand, in recent years it has been explored the application of convolutional neural networks (CNN) on time series predictions, showing a huge potential for that purpose. In this work a CNN architecture is implemented and applied to the prediction of PM 10, using meteorological variables to model PM 10.

Keywords: convolutional neural networks, deep learning, pollutant particles, PM 10, prediction, CNN.

1. Introducción

La contaminación ambiental ha sido en la actualidad un tema de gran interés para un amplio sector de la población, debido a los estudios que se han realizado acerca de su incremento y los potenciales daños que pudiera ocasionar, tanto a la naturaleza como a la salud humana.

La materia particulada o partículas suspendidas (PM) son partículas contaminantes compuestas por distintos tipos de materia, tanto natural como provocada por el hombre [1]. Un subgrupo de las PM son las PM₁₀, teniendo la característica de que son partículas con un diámetro menor a los 10 micrómetros. Su composición es muy variada, ya que una gran cantidad de materia está dentro de esas dimensiones. La composición de PM₁₀ ha sido estudiada en [1], donde se determina el porcentaje de cada elemento que compone las PM₁₀ siendo el carbono uno de los más grandes.

Por otra parte, el impacto de las altas concentraciones de PM₁₀ en la salud de las personas es elevado. Incluso concentraciones por debajo de las normas ambientales promedio tienen un impacto en la función respiratoria de los individuos [2]. Por dicha razón, el modelado de PM₁₀ es importante ya que es uno de los factores de mayor impacto a la salud de la población. En la figura 1 se observa un diagrama que permite hacer un comparativo entre los puntos donde se alojan comúnmente los distintos tipos de PM en el sistema respiratorio de las personas. Se pueden observar, aparte de PM₁₀, las PM_{2.5} que son partículas suspendidas con un diámetro menor a los 2.5 micrómetros y las menores a 0.1 micrómetros (PM_{0.1}).

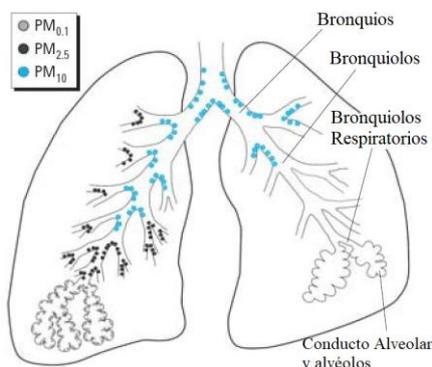


Fig. 1. Diagrama de puntos principales donde se concentran los subgrupos de PM en el sistema respiratorio [3].

Para el presente trabajo se hace uso de una base de datos pública de la cual se utilizan los datos de 7 variables meteorológicas que son medidas por 2 redes de monitoreo del Sistema de Monitoreo Atmosférico (SIMAT) de la ciudad de México, perteneciente a la Secretaría del Medio Ambiente (SEDEMA) del Gobierno Federal. Las redes de monitoreo consultadas tienen el nombre de Red Automática de Monitoreo Atmosférico (RAMA) y Red de Meteorología y Radiación Solar (REDMET). Estas redes de monitoreo son administradas por el Centro de Información de la Calidad del Aire (CICA), el cual recibe datos por minuto de todas las redes de monitoreo, posteriormente realiza

un promedio por hora para generar el Índice Metropolitano de la Calidad del Aire (IMECA) que se hace público cada hora en la página electrónica de la SEDEMA [4].

Las bases de datos utilizadas fueron consultadas en la página oficial de la SEDEMA, mediante la REDMET y RAMA. El SIMAT cuenta con 45 estaciones de monitoreo distribuidas en distintas zonas de la Ciudad de México y la zona Metropolitana (Fig. 2).



Fig. 2. Mapa de la localización de las redes de monitoreo del SIMAT [5].

Una vez que se obtuvieron los datos, se procedió a la implementación del algoritmo que permitiría el modelado de la concentración de PM10 y su variación, medido en unidades de microgramo por metro cúbico ($\mu\text{g}/\text{m}^3$). En los últimos años las CNN han sido utilizadas para predicción de series de tiempo, logrando un buen desempeño; demostrando que tienen un buen potencial en este campo. Las CNN tienen la cualidad de encontrar patrones en grandes volúmenes de datos, al aplicar transformaciones al conjunto de datos.

2. Antecedentes

Se han implementado muchas técnicas para el modelado y predicción de PM10, algunas de ellas mediante la clasificación de PM10 en grupos dependiendo de su concentración como en [6] y [7], pero este acercamiento no da un valor exacto de predicción, sino un intervalo. Posteriormente en acercamientos más recientes se ha realizado una predicción de forma continua, lo cual entrega un resultado con una mayor certeza.

En [8] se implementa una red híbrida de CNN con Memoria a Corto y Largo Plazo (LSTM) para la predicción de ozono, logrando obtener resultados de precisión superiores a los previamente obtenidos mediante otras técnicas. Lo cual permite reconocer el potencial de las CNN para la predicción de series de tiempo.

Acercamientos previos que se han tenido al modelado y predicción de PM10, mediante algoritmos de optimización por enjambre, se pueden consultar en [9], donde se utiliza Optimización por Colonia de Hormigas para el modelado de PM10 y en [10] donde se utiliza Optimización por Enjambre de Partículas (PSO) en un Sistema de Inferencia Neuro-Difuso Adaptativo (ANFIS) para la predicción de dicho contaminante, logrando grandes mejoras por medio de PSO. Implementaciones de Máquina de Vectores de Soporte (SVM) para la predicción de Ozono y PM10 pueden consultarse en [11], donde se prueban distintos kernels para la predicción.

En [12] se hace uso de técnicas de clusterización para la selección de las mejores características para una arquitectura de red neuronal artificial (ANN).

Por último, en [13] se hace uso del Perceptrón Multicapa (MLP) para la predicción de concentraciones de PM10.

En el estado del arte, hasta el momento, no se ha reportado el uso de redes convolutivas para la predicción de ningún tipo de PM. El uso de una técnica distinta, no reportada previamente, será la principal aportación del trabajo de investigación.

3. Materiales y métodos

3.1. Conjunto de datos

El conjunto de datos utilizado consta de la concentración de PM10 y 6 variables meteorológicas obtenidas de las redes de monitoreo. Algunas variables disponibles en la base de datos no fueron tomadas en cuenta debido a que tenían una gran cantidad de datos faltantes o solo se tenía registro reciente de ellas. Las variables elegidas se tomaron en cuenta debido a que son las que comúnmente se han utilizado, en trabajos previos relacionados, para la predicción de PM10 mediante otras técnicas y de igual manera porque son las que cuentan con una mayor cantidad de datos válidos dentro de la base de datos utilizada. En la Tabla 1 se pueden observar las variables seleccionadas junto con su métrica.

Tabla 1. Variables utilizadas para el conjunto de datos.

Variable	Métrica	Red de Monitoreo
PM10	Microgramo/metro cúbico ($\mu\text{g}/\text{m}^3$)	RAMA
Temperatura (TMP)	Grados Celsius ($^{\circ}\text{C}$)	REDMET
Dirección del Viento (WDR)	Grados Azimut	REDMET
Velocidad del Viento (WSP)	Metros/segundo (m/s)	REDMET
Humedad Relativa (RH)	Porcentaje (%)	REDMET
Radiación Solar Ultravioleta tipo A (UVA)	Mili Watt/centímetro cuadrado (mW/cm^2)	RAMA
Radiación Solar Ultravioleta tipo B (UVB)	Dosis mínima de eritema sobre hora (MED/h)	RAMA

Se utilizaron los datos por hora desde el año 2010 hasta 2018, por lo que el conjunto de observaciones es superior a los 166,000 por variable utilizada, dando un total aproximado de 1.1 millones de datos. Se creó una función en la que podía realizarse el procesamiento por mes y haciendo una comparación se determinó que al utilizar el mes de enero se podía crear un mejor modelo de predicción. Se experimentó con todos los meses y el conjunto de datos completo, pero finalmente se utilizaron los datos del mes de enero por lo que la cantidad de datos entre las 7 variables es aproximadamente de 100,000. De todas las estaciones de monitoreo disponibles se optó por utilizar la de Merced (MER), debido a que contaba con menor cantidad de datos faltantes en la mayor parte de los años utilizados. De igual manera en la estación MER se contaba con la

medición de las 7 variables utilizadas en los años seleccionados, cosa que en otras estaciones no se tenía disponibilidad.

De esta manera al contar con aproximadamente 100,000 datos en el conjunto al tomar el mes de enero, por variable contamos con poco más de 14,000 datos. Esto nos dice que tendremos aproximadamente 14,000 lotes para procesar en nuestras 3 arquitecturas. Al realizar el proceso de entrenamiento, validación y pruebas de 3 arquitecturas en repetidas iteraciones para la validación de los resultados deriva en una gran cantidad de tiempo de cómputo con esa cantidad de datos. Por lo que el uso del conjunto de datos completo tomaría 12 veces más en realizarse. Por esas razones se hace uso de una muestra mediante el mes de enero, resaltando que mediante experimentación fue el mes que generó un mejor resultado y por ende una mejor representación del conjunto total de datos.

3.2. Normalización de los datos

El conjunto de datos fue normalizado ya que durante la experimentación se determinó que presentaba mejores resultados de predicción al implementar la normalización.

Se presenta el estudio de diversos métodos de normalización en [14], en ese trabajo se concluye que la normalización por vector es la más eficiente para su trabajo de predicción. Sin embargo, por medio de la experimentación que determinó que para este trabajo existían otros métodos de normalización que generaban un mejor rendimiento.

Se utilizó la normalización z-score, dicha normalización consiste en restar la media del conjunto de datos a cada elemento y posteriormente dividir entre la desviación estándar como se muestra en la siguiente ecuación (ec. 1):

$$N_i = \frac{T_i - \mu_T}{\sigma_T}. \quad (1)$$

Una de las características de esta normalización es que el conjunto de datos resultante tiene una media de 0 y una desviación estándar de 1.

3.3. Creación de lotes de entrenamiento, validación y prueba

El conjunto de datos fue separado de la forma 80% de los datos para entrenamiento, 10% para validación y 10% prueba. Dado que se contó con un total de datos de 100,000 aproximadamente, se tuvieron 80,000 datos para entrenamiento, 10,000 para validación y 10,000 para pruebas.

El conjunto de datos fue organizado de tal manera que tuviera la forma necesaria de entrada para las arquitecturas de redes implementadas como se explica más adelante, esto debido a que el grupo de datos debía estar organizado en una matriz con el orden de los días previos de entrada, las 7 variables utilizadas y la cantidad de lotes de entrenamiento; por lo que se segmentó en una cantidad de entradas n (24 datos previos) para crear los lotes de entrenamiento, validación y prueba.

Debido a que se implementaron 3 arquitecturas, los datos fueron estructurados de la manera descrita a continuación. Las MLP requieren un vector bidimensional de la forma [Lotes, Entradas], las CNN de 1 dimensión (CNN1D) requieren una matriz 3D de la forma [Lotes, Entradas, Características] y las CNN de 2 dimensiones (CNN2D) requieren una matriz 4D de la forma [Lotes, Filas, Columnas, Canales].

3.4. Creación de las arquitecturas

Para tener un marco de comparación se utilizó un MLP (Fig. 3), similar al de [13], con una cantidad de neuronas de entradas dependiente de la cantidad n de datos de entrada, el cual fue determinado como 24, seguida de una capa oculta de 10 neuronas y una capa de salida con 1 neurona para la predicción de la hora siguiente. Se realizaron pruebas para determinar el mejor valor para la cantidad de neuronas en la capa oculta, resultando el valor de 10 como el más adecuado, a diferencia de los resultados que se obtuvieron en [13], donde se usaron dos modelos de MLP con 8 y 19 neuronas en la capa oculta respectivamente. El MLP se usa como comparación ya que es de las técnicas más recientes y con mayor uso en esta aplicación específica.

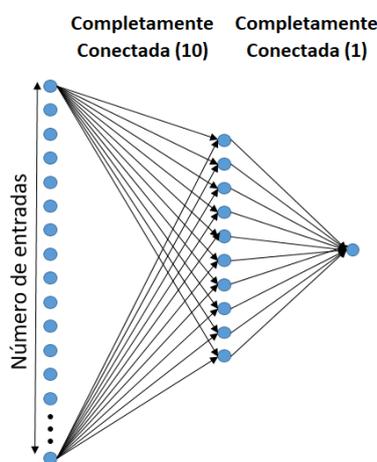


Fig. 3. Diagrama de MLP.

Se creó una CNN1D (Fig. 4) con una entrada $[n, 7]$, por las n entradas y 7 debido a que son 7 características para la predicción; esta capa fue seguida de una capa de aplanamiento que linealiza la salida de la CNN1D para que pueda ser procesada por la siguiente capa oculta de 10 neuronas, posteriormente se añadió una capa de deshabilitación de neuronas (Dropout) con una probabilidad de 15%, esto para evitar el sobreentrenamiento, finalmente una capa de salida de 1 neurona.

Finalmente se implementó una CNN2D (Fig. 5) con una entrada de $[n, 7, 1]$, las dos primeras dimensiones similares a la capa CNN1D y la última dimensión porque solo se utilizó un canal; posteriormente se insertó la capa de aplanamiento, una capa oculta de 10 neuronas, la capa de Dropout con 15% y una capa de salida de 1 neurona.

La principal diferencia entre las arquitecturas CNN1D y CNN2D es el kernel utilizado, ya que en la CNN1D el kernel es un vector de una fila por una cantidad n de columnas, por lo cual se realiza la convolución a cada fila de la matriz de entrada. Por otra parte, la CNN2D tiene un kernel de n filas por m columnas por lo que se realiza la convolución de manera bidireccional a la matriz de entrada; este último toma más datos para la convolución por lo que podría decirse que encuentra relaciones entre los datos en un mayor espacio de búsqueda.

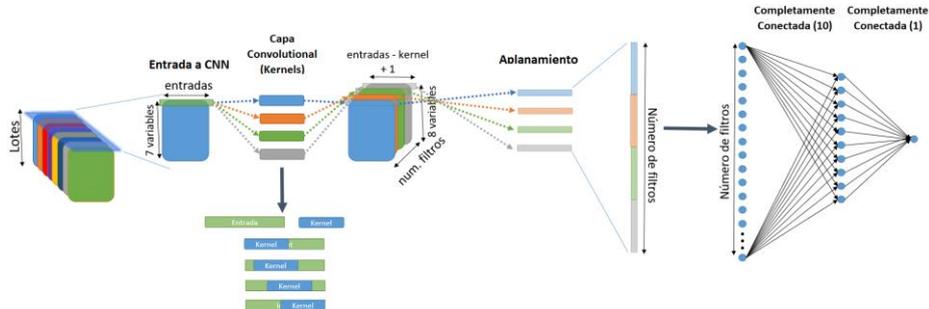


Fig. 4. Diagrama de CNN1D.

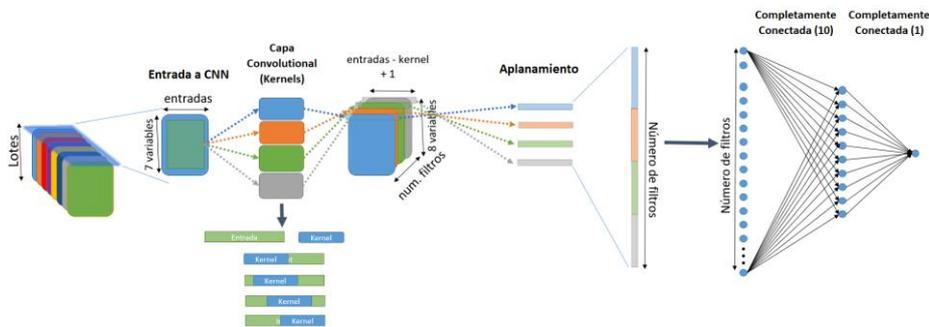


Fig. 5. Diagrama de la CNN2D.

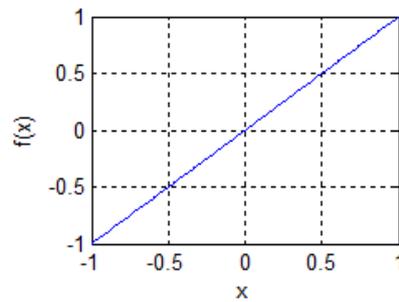


Fig. 6. Función de activación lineal.

Todas las capas de todas las arquitecturas tienen una función de activación lineal, como se muestra en la ecuación 2 y (Fig. 6), dicha función de activación se elige ya que los datos son normalizados y se requiere que las salidas puedan tener valores negativos. Otras funciones de activación tienden a llevar los valores de salida a 0 o limitan la salida a números positivos. Por otra parte, todas las arquitecturas usan el optimizador Adam; dicho optimizador ha demostrado tener un buen rendimiento en datos donde el gradiente es altamente variable [15]. Debido a que el conjunto de datos es altamente no lineal, mediante el uso de otro tipo de optimizadores como lo es el Gradiente de Descenso Estocástico (SGD), la búsqueda de un espacio de solución es más lento y no logra adecuarse a las grandes variaciones de los datos. Por su parte, Adam tiene un buen

desempeño en grandes conjuntos de datos y es apropiado para datos no estacionarios y con grandes variaciones en su gradiente [15]:

$$f(x) = x. \quad (2)$$

4. Resultados

Se probó cada arquitectura con los mismos parámetros, éstos pueden observarse en la Tabla 2, por 10 iteraciones para obtener un promedio de los valores resultantes.

Tabla 2. Parámetros para las arquitecturas implementadas.

Parámetro	Método / Variable
Número de Entradas	50
Tamaño de Lote	100
Función de Activación	Lineal
Función de Costo	Media del Error Absoluto (MAE)
Optimizador	Adam

Adicionalmente, las capas de convolución se definieron con un tamaño de kernel igual al número de características, en el caso de CNN1D, y número de características por número de entradas para el caso de CNN2D. Ambas redes aplicaron 100 filtros a cada lote de entrada.

La función de costo, MAE, fue seleccionada debido a que fue la que presentó un mejor desempeño en las 3 arquitecturas. Ésta es calculada por medio de la ecuación (3), donde y es el valor real y \hat{y} es el valor predicho:

$$MAE = \frac{1}{n} \sum_{j=1}^n |y_j - \hat{y}_j|. \quad (3)$$

Otra de las métricas utilizadas es la de la Raíz de la Media de los Errores al Cuadrado (RMSE), denotado por la ecuación (4), con la notación previamente definida:

$$RMSE = \sqrt{\frac{1}{n} \sum_{j=1}^n (y_j - \hat{y}_j)^2}. \quad (4)$$

La métrica de Media del Error Absoluto Porcentual da un valor que refleja la lejanía que tiene un punto predicho con respecto al valor real, una de las desventajas es que mientras mayores sean los valores del conjunto de datos, se puede presentar un mayor error. La ecuación (5) es utilizada para su cálculo:

$$MAPE = \frac{1}{n} \sum_{j=1}^n \left| \frac{y_j - \hat{y}_j}{y_j} \right|. \quad (5)$$

Adicionalmente se utiliza el Índice de Concordancia (IOA), obtenido mediante la ecuación (6). Se toma en cuenta la notación anterior y μ_T corresponde a la media del conjunto de datos:

$$IOA = 1 - \frac{\sum_{j=1}^n (y_j - \hat{y}_j)^2}{\sum_{j=1}^n (|\hat{y}_j - \mu_T| + |y_j - \mu_T|)^2}. \quad (6)$$

En la figura 7 se puede observar una gráfica con la comparación entre el RMSE de cada arquitectura, se puede notar que el desempeño de ambas arquitecturas con capa de convolución otorgan un resultado ligeramente superior al MLP. Cabe mencionar que a menor valor de RMSE, menor error; por lo tanto, el modelo tiene una mayor precisión. Esto es notorio para todas las iteraciones con capa de convolución (tanto CNN1D como CNN2D), con excepción de la prueba número 7 (iteración) en donde CNN1D da un resultado de error ligeramente superior que MLP. Sin embargo, en general los resultados son mejores utilizando esta métrica de error con las capas de convolución en comparación con MLP.

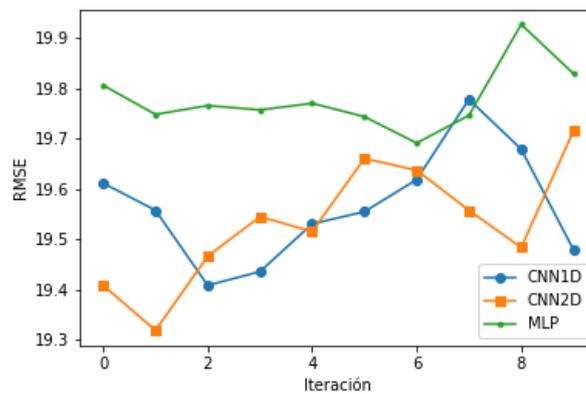


Fig. 7. Comparación de RMSE en 10 iteraciones con cada arquitectura.

Como otro método de visualización en la figura 8 se muestra un diagrama de cajas con las mismas 10 iteraciones para el RMSE.

En la figura 8, es notorio que a pesar que la variabilidad es mayor para los resultados con arquitecturas de convolución, los resultados son en su gran mayoría inferiores, en cuestión de error, comparado con la misma métrica en MLP, lo que muestra una repetitividad en resultados con mayor precisión (por lo tanto, con un menor RMSE).

Por otra parte, el MAPE, resulta ser ligeramente menor para las mismas iteraciones, lo cual refleja la variabilidad de dicha métrica. En la figura 9 se muestra la comparación de los resultados de la experimentación. En suma, MAPE para esta aplicación, no es recomendable para identificar métricas de error por su alta variabilidad en las 3 arquitecturas.

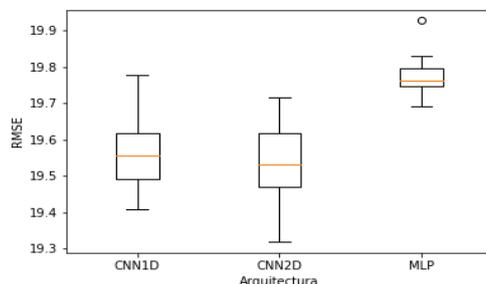


Fig. 8. Comparación de RMSE mediante diagrama de cajas.

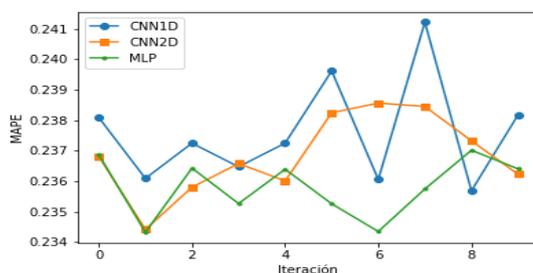


Fig. 9 Comparación de MAPE para las 3 arquitecturas.

El diagrama de cajas puede verse en la figura 10, en él puede notarse que en la CNN2D se alcanzan valores bajos similares a los de MLP, pero la media está por encima.

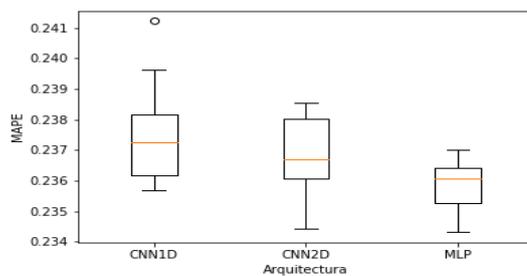


Fig. 10. Diagrama de cajas de MAPE.

Por último, se muestran los resultados del IOA, en la figura 11 mediante una gráfica con las 10 iteraciones y en la figura 12 mediante un diagrama de cajas. En el diagrama de cajas es más notorio que los resultados de MLP tienen mayor repetitividad, sin embargo, son más bajos.

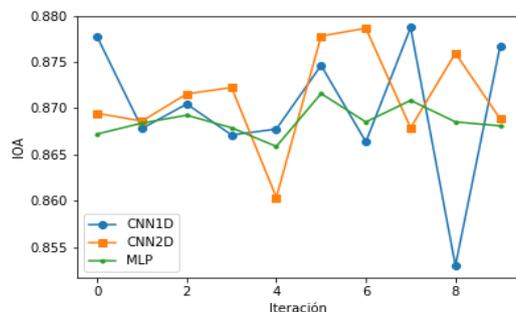


Fig. 11. Gráfica de comparación de IOA.

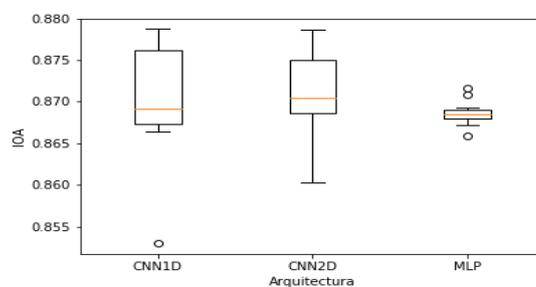


Fig. 12. Diagrama de cajas de IOA.

En la Tabla 3 se muestran los resultados en promedio del experimento realizado. Se puede observar que los resultados de las redes de convolución son ligeramente superiores a los de MLP en el caso de RMSE y de IOA, y menores en la métrica de MAPE, lo cual es un indicativo de que las redes convolutivas tanto de 1D como 2D, tienen mejores resultados en prácticamente todos los casos con respecto a una red ampliamente utilizada en modelado de PM10 como lo es MLP.

Tabla 3. Promedio de Resultados para las 3 Arquitecturas.

Arquitectura	Métrica	Resultado
CNN1D	RMSE	19.5657
	MAPE	0.2376
	IOA	0.87
CNN2D	RMSE	19.5311
	MAPE	0.2368
	IOA	0.8711
MLP	RMSE	19.779
	MAPE	0.2358
	IOA	0.8686

Haciendo un análisis de los resultados obtenidos puede observarse que el MLP, que en trabajos previos ha demostrado ser un mejor acercamiento a la predicción de PM10 que otros métodos, tiene un rendimiento ligeramente más bajo que ambas arquitecturas propuestas. La CNN1D obtuvo un mejor desempeño, con un RMSE ligeramente menor. Por su parte la CNN2D mejoró un poco más el RMSE. El RMSE es una de las métricas que mejor representa el desempeño junto con el IOA, ya que se basan en la diferencia del error predicho y el real por medio de las unidades que conciernen al modelo. Por su parte el MAPE es una unidad porcentual que tiene una gran variabilidad dependiendo de la magnitud de las mediciones, por lo que no es altamente confiable si se utiliza como única métrica de desempeño. Por medio de los resultados se puede decir que la CNN2D es la que otorga un modelo más confiable, aunque la diferencia es muy ligera comparativamente con CNN1D.

5. Conclusiones

Este trabajo presenta el uso de redes convolutivas para la predicción de PM10, dicha arquitectura no había sido utilizada hasta ahora con ese fin. De esta manera el trabajo de investigación tiene un impacto, ya que en el estado del arte no se reporta ningún trabajo que aborde el tema desarrollado. Las redes convolutivas tienen un buen desempeño para la predicción de series de tiempo, en el trabajo realizado han demostrado igualar y superar ligeramente una de las redes que en trabajos recientes habían demostrado el mejor desempeño para la predicción de PM10. Se hizo uso de un gran volumen de datos y de una gran cantidad de iteraciones para determinar que los resultados no son por la naturaleza estocástica de las redes neuronales; al hacer esto se asegura que los modelos propuestos son certeros y son robustos. Trabajos previos pueden dirigirse en el ajuste de los parámetros de las redes convolutivas para incrementar el desempeño, ya que existen varios parámetros que afectan el resultado; por mencionar algunos se tiene el tamaño de kernel, número de filtros, tamaño de lote, inicialización de bias y pesos, etc. Mediante el ajuste de los parámetros puede lograrse incrementar aún más la capacidad predictiva de las arquitecturas propuestas.

Referencias

1. Chow, J.C., Watson, J.G., Edgerton, S.A., Vega, E.: Chemical composition of PM2.5 and PM10 in Mexico City during winter 1997. *Science of the Total Environment* 287(3), 177–201 (2002)
2. Pope Iii, C.A., Dockery, D.W., Spengler, J.D., Raizenne, M.E.: Respiratory Health and PM10 Pollution A Daily Time Series Analysis. *American Review of Respiratory Disease* 144, 668–674 (1991)
3. Cormier, S.A., Lomnicki, S., Backes, W., Dellinger, B.: Origin and health impacts of emissions of toxic by-products and fine particles from combustion and thermal treatment of hazardous wastes and materials. *Environmental Health Perspectives* 114(6), 810–817 (2006)
4. Calidad del Aire, <http://www.aire.cdmx.gob.mx/default.php?opc=%27aKBhnmM=%27>. Último acceso: 2019/03/29
5. Calidad del Aire Página Principal, <http://www.aire.cdmx.gob.mx/default.php>. Último acceso: 2019/03/29

6. Sun, W., Zhang, H., Palazoglu, A., Singh, A., Zhang, W., Liu, S.: Prediction of 24-hour-average PM 2.5 concentrations using a hidden Markov model with different emission distributions in Northern California. *Science of the Total Environment* 443, 93–103 (2013)
7. Dong, M., Yang, D., Kuang, Y., He, D., Erdal, S., Kenski, D.: PM 2.5 concentration prediction using hidden semi-Markov model-based times series data mining. *Expert Systems with Applications*, 36(5), 9046–9055 (2010)
8. Pak, U., Kim, C., Ryu, U., Sok, K., Pak, S.: A hybrid model based on convolutional neural networks and long short-term memory for ozone concentration prediction. *Air Quality, Atmosphere & Health* 11(8), 883–895 (2018)
9. Aceves-Fernández, M.A., Estrada, A.L., Pedraza-Ortega, J.C., Gorrostieta-Hurtado, E., Tovar-Arriaga, S.: Design and Implementation of Ant Colony Algorithms to Enhance Airborne Pollution Models. *International Journal of Environmental Science and Toxicology Research* 3(2), 22–28 (2015)
10. De León, B.O., Aceves-Fernandez, M.A., Fernández-Fraga, S.M., Ramos-Arreguín, J.M., Gorrostieta-Hurtado, E.: An improved particle swarm optimization (PSO): method to enhance modeling of airborne particulate matter (PM10). *Evolving Systems* 2019(0), 1–10 (2019)
11. Sotomayor-Olmedo, A., Aceves-Fernández, M.A., Gorrostieta-Hurtado, E., Pedraza-Ortega, C., Ramos-Arreguín, J.M., Vargas-Soto, J.E.: Forecast Urban Air Pollution in Mexico City by Using Support Vector Machines: A Kernel Performance Approach. *International Journal of Intelligence Science* 2013(3), 126–135 (2013)
12. Cortina-Januchs, M.G., Quintanilla-Dominguez, J., Vega-Corona, A., Andina, D.: Development of a model for forecasting of PM10 concentrations in Salamanca, Mexico. *Atmospheric Pollution Research* 6(4), 626–634 (2015)
13. Paschalidou, A.K., Karakitsios, S.: Forecasting hourly PM 10 concentration in Cyprus through artificial neural networks and multiple regression models: implications to local environmental management. *Environ Sci Pollut Res* 18, 316–327 (2011)
14. Panigrahi, S., Karali, Y., Behera, H.S.: Normalize Time Series and Forecast using Evolutionary Neural Network. *International Journal of Engineering Research & Technology* 2(9), 2518–2522 (2013)
15. Kingma, D.P., Ba, J.L.: ADAM: A Method for Stochastic Optimization. *ICLR* 1–15 (2015)

Clasificador de plantas medicinales por medio de Deep Learning

Juan Monroy-de-Jesús, Adriana Reyes-Nava, Fernando Olmos

Tecnológico de Estudios Superiores de Jocotitlán, Jocotitlán, México
{juan.monroy, adriana.reyes}@tesjo.edu.mx, fol9602@gmail.com

Resumen. El presente trabajo plantea la implementación de un algoritmo basado en Deep Learning para la identificación de plantas medicinales. El propósito es que el usuario le pueda indicar al sistema que planta buscar e inmediatamente se active la búsqueda de esta o bien que cuando se le muestre una planta devuelva una respuesta donde se haga una descripción de la planta con sus características. Esto se logró mediante el desarrollo de una red neuronal de aprendizaje profundo, a través del entrenamiento con una colección de corpus (base de datos) propia. Se busca que el sistema de reconocimiento sea implementado a futuro en un robot recolector de basura, donde sus dos funciones principales sean recolectar desechos e identificar plantas medicinales.

Palabras clave: red neuronal convolucional, CNN, aprendizaje profundo, inteligencia artificial.

Plant Sorter Using Deep Learning

Abstract. The present work propose the implementation of an algorithm based on Deep Learning for identification of medicinal plants. The purpose is that user can tell the system which plant look for and immediately it active the search for it or that when plant is shown it return a response where will show a description about the plant with its characteristics. This achieved through the development of a neural network of deep learning, through a training with a collection of corpus (database) own it. The aim is that the recognition system to be implemented in a future in a garbage collection robot, where its two functions are wastecollection and identify medicinal plants.

Keywords: convolutional neural network, CNN, deep learning, artificial intelligence.

1. Introducción

Actualmente, con el uso de técnicas de aprendizaje profundo, se busca automatizar tareas en diversas áreas de estudio; campos como la medicina, industria automotriz o alimenticia se han visto beneficiados [1]; la botánica no es la excepción, en los últimos años se han implementado técnicas de clasificación con enfoques de estudio [2], de producción agrícola [3] e identificación de enfermedades que puedan padecer los cultivos [4, 5].

Partiendo de lo anterior, las Redes Neuronales Convolucionales, son un modelo donde las neuronas corresponden a campos receptivos que buscan simular la composición de las neuronas de la corteza visual primaria de un cerebro biológico.

La red se compone de múltiples capas; en el principio se encuentra la fase de extracción de características compuesta de neuronas convolucionales y de reducción; conforme avanzan las iteraciones del algoritmo se disminuyen sus dimensiones activando características cada vez más complejas; para llegados al final se utilicen neuronas sencillas las cuales realizan la clasificación.

En este tipo de arquitecturas se han logrado avances importantes en el campo de la Inteligencia Artificial, al proveer diferentes estructuras y mejorar los algoritmos de aprendizaje permitiendo mayor flexibilidad en los modelos de clasificación [6].

Sin embargo, este no es el único modelo que se emplea actualmente para que las máquinas sean capaces de aprender. Otros algoritmos son los árboles de decisión, las reglas de asociación, los algoritmos genéticos, las redes bayesianas o el aprendizaje por refuerzo [7].

Por otra parte, actualmente el uso de fármacos se ha utilizado para tratar cualquier enfermedad en las personas, sin embargo, estos fármacos se ha demostrado que pueden generar efectos secundarios en las personas, por ejemplo, problemas en los riñones, páncreas, acides, entre otras. Sin embargo, la comunidad en general ha olvidado el tratamiento tradicional de antaño, estos tratamientos se basan en remedios con plantas medicinales, las cuales se ha olvidado sobre sus propiedades curativas y aun mejor que no casusa problemas secundarios en las personas.

El presente trabajo muestra la implementación de un algoritmo convolucional clasificador de plantas medicinales con ayuda de Deep Learning.

2. Contexto general para la clasificación de imágenes

La clasificación de imágenes consiste en asignar a una imagen una etiqueta de un conjunto definido de categorías en función de sus características.

Factores como la escala, condiciones de iluminación, deformaciones u ocultamiento parcial de objetos hacen de la clasificación de imágenes una tarea compleja, a la que se ha dedicado un gran esfuerzo para desarrollar sofisticadas técnicas de reconocimiento de patrones, que no siempre producen los resultados esperados.

Desde el punto de vista del aprendizaje automático, la clasificación de imágenes es un problema de aprendizaje supervisado, en el que los algoritmos clasificadores generan un modelo a partir de un Dataset o conjunto de imágenes previamente categorizadas. El modelo obtenido se utiliza posteriormente para clasificar nuevas imágenes.

Para estos algoritmos las imágenes son matrices tridimensionales cuyas dimensiones son el ancho, alto y la profundidad de color, siendo el contenido de cada posición de la matriz un valor numérico que representa la intensidad de color de cada píxel de la imagen digital [8].

2.1. Redes neuronales

Las redes neuronales artificiales son redes interconectadas masivamente en paralelo de elementos simples, inspirados en el modelo biológico del ser humano [9],

actualmente existen diferentes modelos de redes neuronales, por ejemplo, las redes bayesianas, el perceptrón multicapa, redes recurrentes, convolucionales, entre otras. siendo esta última, tema de interés en este trabajo

2.2. Redes neuronales convolucionales

Son un tipo particular de red neuronal inspirada en el funcionamiento de la corteza visual del cerebro. Estas redes están diseñadas para resolver problemas de visión artificial como el reconocimiento de patrones, aunque pueden tener otros usos como la clasificación de textos o el procesamiento de lenguaje natural.

Al igual que las redes neuronales “convencionales”, reciben una entrada que transforman a través de una serie de capas de neuronas, pero en este caso la entrada es una imagen representada en forma de matriz tridimensional dada por ancho, alto y profundidad de color, siendo esta última la que contiene los valores numéricos de los píxeles respecto al tono que maneja. Las capas de neuronas también se organizan de manera tridimensional, y el resultado de las distintas transformaciones llevadas a cabo en la red es la clase o categoría a la que pertenece dicha imagen.

Una CNN (Red Neuronal Convolutional por sus siglas en inglés *Convolutional Neural Network*) es un conjunto de capas de procesamiento, de modo que puede verse como un diagrama secuencial de bloques [10], se obtienen apilando múltiples capas de características. Una capa está formada por K filtros lineales seguida de una función de respuesta no lineal.

Se caracterizan por el procesamiento de imágenes, sin embargo, a menudo es ventajoso explotar la estructura de la imagen. Por ejemplo, los píxeles que están muy juntos en la imagen (píxeles adyacentes) tienden a estar fuertemente correlacionados mientras que píxeles que están muy separados en la imagen tienden a ser débilmente correlacionados o no correlacionados [9].

Existen cuatro operaciones principales que conforman las redes convolucionales, estas se describen a continuación.

2.3. Capa convolucional

Las capas convolucionales se entienden como un conjunto de filtros comúnmente llamados campos receptivos [10]. Su principal propósito es extraer características de una imagen. Dicho conjunto de filtros son entrenables y realizan producto punto con los valores de la capa precedente. En la práctica, los valores de los filtros son aprendidos para su activación al encontrar ciertas características. Cuando son colocados en cascada se obtienen diferentes niveles de abstracción [11].

Permiten reducir el número de elementos que conforman la red y detectar características que poseen las imágenes. Considerando lo anterior, se sabe que las capas convolucionales son de gran utilidad en el análisis de imágenes permitiendo reducir la complejidad del sistema y al mismo tiempo, extraer rasgos útiles de las mismas [12].

En la convolución se realizan operaciones de productos y sumas entre la capa de partida y los filtros con lo que se genera un mapa de características. Las características extraídas corresponden a cada posible ubicación del filtro en la imagen original. Este proceso tiene la ventaja de que el mismo filtro puede extraer la misma característica en cualquier parte de la entrada, reduciendo el número de conexiones y de parámetros a entrenar en comparación con una red multicapa de conexión total [13].

Cada capa convolucional consta de 4 parámetros principales:

- Número de filtros N.
- Dimensión M X M: Campos receptivos que recorren toda la imagen de entrada a la capa.
- Stride: Representa el salto de cada filtro, si su valor es 1 saltara de pixel en pixel hasta recorrer toda la imagen.
- Pad: Representa el espacio a partir del cual comienza a recorrer cada campo, este determina el volumen de salida de cada capa [10].

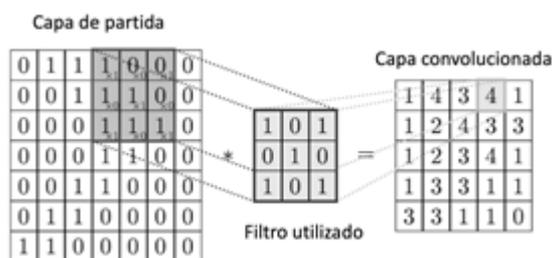


Fig. 1. Muestra el proceso de convolución en una imagen.

De manera general, una capa Convolutiva recorre una matriz (pixeles contenidos en la imagen) aplicando un filtro que es una matriz más pequeña, obteniendo las características de la imagen aplicando multiplicaciones, generando una nueva matriz más pequeña que conforme las iteraciones avanzan, se profundiza en las características de la imagen como se muestra en la figura 1.

2.4. Rectificador lineal de unidad

Son utilizados después de cada convolución actuando como de función de activación en términos de redes neuronales tradicionales. Son una operación que reemplaza los valores negativos por cero y su propósito es agregar no linealidad al modelo, eliminando la relación proporcional entre la entrada y salida [11].

Su función matemática se expresa de la siguiente manera:

$$h = \max(0, x), \tag{1}$$

donde todos los identificadores inferiores a 0 serán convertidos en 0 mientras los positivos se conservan [10].

El comportamiento por defecto y más habitual es que mientras la entrada tenga un valor por debajo de cero, la salida será cero, pero cuando la entrada se eleva por encima, la salida es una relación lineal con la variable de entrada de la forma $f(x) = x$. La función de activación ReLU ha demostrado funcionar en muchas situaciones diferentes, y actualmente es muy usada, gráficamente se puede observar en la figura 2.

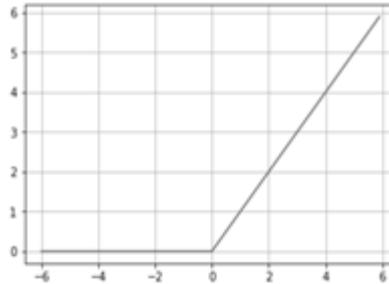


Fig. 2. Muestra el comportamiento de la función de activación gráficamente.

2.5. Pooling

Es la capa utilizada para reducir las dimensiones, con el objetivo de disminuir los tiempos de procesamiento reteniendo la información más importante [11], a su vez consigue reducir el sobreajuste (*overfitting*) en la red y algo de invarianza a la traslación. Como se muestra en la siguiente imagen la capa *pooling* se comporta conservando el valor mayor tras aplicar un filtro con una matriz, con lo que se crea una nueva matriz.

En la figura 3 se muestra una matriz de 4 X 4 que ejemplifica lo que se conoce como capa de partida misma que es un estado posterior a pasar por la capa convolucional, el filtro con el que se evaluó es de un tamaño 2 X 2, es decir la imagen se reducirá en una de 2 X 2 píxeles al ya tener la capa de reducción siendo evaluadas en total 4 secciones en la capa previa.



Fig. 3. Muestra el funcionamiento de la capa de reducción (*Pooling*).

2.6. Capa totalmente conectada (*Dense Layer*)

Capa totalmente conectada realiza la clasificación basada en las características extraídas por las capas de convolución y las reducidas por pooling. En esta capa todos los nodos están conectados con la capa precedente [11].

En la figura 4 se muestra el funcionamiento de esta capa, la primera capa de entrada es el resultante de la última reducción por parte de la capa de *pooling*, la capa oculta funciona como clasificador, a partir de la cual se encuentran las incidencias que determinen cual es el resultado de la clasificación, por último, se encuentran la capa de salida.

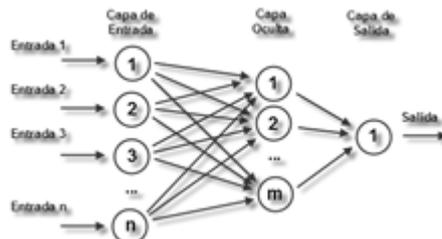


Fig. 4. Representación gráfica de una capa clasificatoria.

2.7. Funcionamiento de la red convolucional

Las capas de convolución y las de *pooling* se encargan de extraer características mientras que la capa totalmente conectada actúa como clasificador. Para el funcionamiento de este modelo debemos proceder al entrenamiento.

Esto implica:

- 1) Inicializar todos los parámetros o pesos con valores aleatorios.
- 2) Utilizar una imagen de entrenamiento y utilizarla en el modelo.
- 3) Calcular el error total de las probabilidades resultantes del modelo.
- 4) Propagar hacia atrás para calcular el error de gradiente de todos los pesos en la red y utilizar gradiente descendiente para actualizar estos valores y minimizar el error de salida [11].

2.8. Evaluación del nivel de clasificación en una red neuronal

Métrica Accuracy. Esta métrica trabaja en relación con la función que se utiliza para juzgar el rendimiento del modelo, matemáticamente se define de la siguiente manera:

$$\frac{TP + TN}{TP + TN + FP + FN} (\hat{p}), \quad (2)$$

donde TP son los Positivos Verdaderos (Clasificaciones esperadas durante el entrenamiento), FP son Falsos Positivos (Cuando se clasifica como perteneciente a una categoría a la que no pertenece), TN Negativos Verdaderos (Cuando un elemento perteneciente no es clasificado donde se esperaba) y FN Falsos Negativos (Cuando un elemento no perteneciente no se clasifica en alguna clase a la que no pertenece) y \hat{p} proporción de aciertos definida de la siguiente manera: s/n , donde s es el número de aciertos obtenidos y n se refiere al número de pruebas realizadas (de iteración del algoritmo) [14].

Función de pérdida Categorical Crossentropy. Una función de pérdida (función objetivo, o función de puntuación de optimización) es uno de los dos parámetros necesarios para compilar un modelo; cuando usa la función `categorical_crossentropy`, sus objetivos deben estar en formato categórico (por ejemplo, si tiene 10 clases, el objetivo para cada muestra debe ser un vector de 10 dimensiones que sea todo ceros, excepto un 1 en el índice correspondiente a la clase de la muestra). Para convertir *objetivos enteros en objetivos categóricos*.

Matemáticamente, la función `categorical_crossentropy` está definida entre una distribución aproximada y una distribución verdadera; esto quiere decir que mide el número promedio de bits necesarios para identificar un evento de un conjunto de posibilidades, si se usa un esquema de codificación basado en una distribución de probabilidad dada por q , en lugar de la distribución "verdadera" p expresada en la siguiente ecuación:

$$H(p, q) = - \sum_x p(x) \log(q(x)), \quad (3)$$

donde p está definido por una condición OR que contiene un vector dado de 1 a N donde en cada elemento representa la posición dada al valor "1", q representa las filas para la distribución y x es la entrada recibida [15].

3. Marco metodológico

Previo a la implementación del algoritmo de red se pensó en cómo debería estar estructurada la CNN a implementar por lo que se optó por el modelo que se muestra en la Fig. 5.

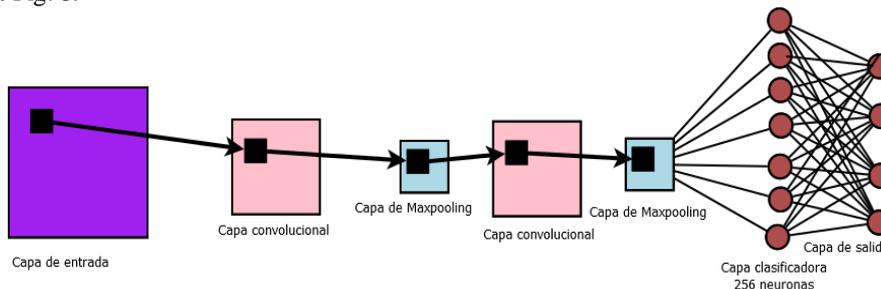


Fig. 5. Muestra la arquitectura de la red neuronal a utilizar.

Basándonos en la imagen anterior, se describirá a detalle el algoritmo a seguir para conseguir una correcta clasificación.

Capa de entrada. Esta capa mostrada en la figura anterior identificada con color morado es equivalente a las imágenes que se contendrán en el Corpus, además se describen algunos pasos de procesamiento de estas a fin de obtener una mejor clasificación, esto con ayuda de Keras.

Procesamiento de imagen con Keras

Tamaño estandarizado. A fin de reducir la carga de trabajo al procesarse redujo el tamaño de todas las imágenes que entraran al algoritmo a un tamaño de 150 X 150 píxeles.

Función rescale. Considerando que los valores posibles en un formato de color RGB son de 0 a 255, se implementó esta función con la finalidad de reducir el amplio rango que se puede obtener de tonalidad entre un píxel y otro, dándoles valores que van de 0 a 1 según el tono que adquiera cada píxel. Esto se muestra en la figura 6.



Fig. 6. Ejemplo de cambio de tono con la función Rescale.

Función shear_range. Esta ayuda a que aleatoriamente se seleccionen imágenes para inclinar su posición, esto para tener un mayor rango de clasificación y a su vez tener en cuenta que es posible que exista variación dada por el medio donde se tome o por la posición de la cámara. Esto se muestra en la figura 7.



Fig. 7. Ejemplos de la función shear_range.

Función zoom_range. selecciona aleatoriamente algunas entradas a fin de hacer un acercamiento que permita tener más muestras distintas a partir de una imagen. Como se muestra en la figura 8.



Fig. 8. Ejemplo en el que se muestra la aplicación de la función de zoom_range en la buganvilla.

Función horizontal_flip y vertical_flip. Estas ayudan a invertir la posición de la imagen para que cuando al algoritmo se le muestre incluso una imagen de cabeza pueda identificar la especie de planta a la que pertenece como se muestra en la figura 9.



Fig. 9. Muestra el reposicionamiento que sufren las imágenes previo a ser introducidas a la red neuronal para su entrenamiento.

Función flatten. Sirve para aplanar la imagen, es decir darle a la imagen el formato necesario para trabajar como una capa totalmente conectada que se pueda trabajar con

la capa Convolutiva que la recibirá, es decir, se le indica al algoritmo que se está trabajando con imágenes 2D.

Capas convolucionales. En la figura 5. estas capas se identifican con color rosa, la primera de ellas cuenta con un tamaño de 3 X 3 píxeles mismos que irán recorriendo la matriz de la capa de entrada; este proceso se realizará en 32 iteraciones por entrada recibida.

La segunda capa consta de un tamaño de 2 X 2 píxeles, en este caso se convolucionará la matriz obtenida por la capa de reducción; esta realizará su función durante 64 iteraciones por entrada recibida.

Como producto de cada capa convolutiva, se generará una nueva matriz por cada una, donde en base a la matemática expuesta en el marco teórico, se irán resaltando las características de cada imagen.

Capas de pooling (Maxpooling). Se propuso utilizar dos capas de este tipo, cada una después de una capa convolutiva, se busca con ello evitar el sobre ajuste de la red y a su vez reducir la carga computacional que aumenta a medida que se avanza en el algoritmo.

Para esto se hizo uso de la función MaxPooling con la que en base a una matriz de tamaño 2 X 2 píxeles, se toma el valor más relevante que contenga ese espacio de píxeles y se envía a una nueva matriz.

La primera de estas capas se ubica después de la primera convolución y la otra después de la segunda convolución; esta última se toma como entrada para la capa clasificadora que se describirá a continuación.

Capa de clasificación. Una vez que ya se han obtenido las características más sobresalientes en las capas anteriores, se procede a ir separando categóricamente cada una de las imágenes con las que se ha de entrenar el algoritmo. Para esto se implementó esta capa que consta de 256 neuronas.

A fin de evitar el sobre entrenamiento de la red, se desactivan aleatoriamente el 50% de las neuronas utilizadas; con ello se busca que la red no se acostumbre a seguir un solo camino para la clasificación obligándola a alternar rutas para llegar al mismo resultado.

Además de esto con Keras se le agrega la función Softmax, misma que le dice al algoritmo que tiene que buscar la probabilidad más alta de que la imagen analizada pertenezca a una categoría en específico.

Capa de salida. Esta última capa está definida como el número de clases con la que se está realizando la clasificación, es decir, se tendrán tantas capas de salida como especies de plantas que se deseen clasificar.

Entrenamiento de la red neuronal convolutiva

La arquitectura definida anteriormente se implementó en el lenguaje de programación Python. A fin de obtener una clasificación óptima se incluyeron los siguientes parámetros:

- Número de épocas: En total el algoritmo recorrerá 20 veces el corpus de imágenes.

- Número de Iteraciones: Se definió un estándar de 1000 iteraciones por épocas y por cada iteración se analizan 32 imágenes.
- Validación del algoritmo: Al final de cada época se realizan 300 iteraciones donde se obtienen del directorio de validación las imágenes que contiene a fin de determinar qué tan eficiente ha sido la clasificación.

Ahora bien, se utiliza la función *categorical_crossentropy* a fin de minimizar la función de pérdida y se registra el porcentaje de aciertos en la clasificación aplicando la métrica *accuracy* para saber cómo está aprendiendo la CNN. Para la actualización de pesos se utiliza el optimizador Adam con una tasa de aprendizaje de 0.0005.

Solo resta utilizar la función *save* para guardar el modelo y la función *save_weights* para guardar los pesos, estos se guardaron con extensión .h5 para poder consumirlos posteriormente en una aplicación.

Con el modelo de la Red Neuronal queda ejecutar el script en Python y esperar el modelo entrenado como se muestra en la figura 10. donde se ve la época en que está entrenando y el número de iteraciones transcurridas, además de mostrar el valor de la función de pérdida (variable *loss*) y el porcentaje de aciertos (variable *acc*) dado por la métrica *accuracy*, como podemos notar en la primera iteración el valor es bastante elevado para la función de pérdida y el porcentaje de aciertos bastante bajo-mismos que mejoraran sus números con el avanzar de las épocas.

```
Epoch 1/20
2018-11-13 10:05:06.636077: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supp
ensorFlow binary was not compiled to use: SSE4.1 SSE4.2 AVX
2018-11-13 10:05:06.659872: I tensorflow/core/common_runtime/process_util.cc:69] Creating new
r op setting: 2. Tune using inter_op_parallelism_threads for best performance.
14/1000 [.....] - ETA: 17:26 - loss: 3.6768 - acc: 0.4911
```

Fig. 10. Entrenamiento de la red.

4. Pruebas y resultados

Para probar el modelo se desarrolló una aplicación de consola con el lenguaje de programación Python y con ayuda de Keras, en la que se consumieron los archivos de modelo.h5 y pesos.h5.

- Las predicciones se guardan en un arreglo cuya dimensión es equivalente al número de especies a clasificar (En este caso 11).
- La clasificación obtenida se expresa con un “1” y los demás espacios del vector se rellenan con “0”, donde la posición del vector es equivalente al orden establecido por los directorios que contienen las imágenes.
- Se evaluaron el número de clasificaciones correctas obtenidas y se modificó el algoritmo hasta obtener la mejor clasificación posible.

A fin de determinar que la clasificación realizada por el algoritmo era aceptable, se realizó la siguiente prueba cargándole imágenes diferentes a las utilizadas durante el entrenamiento por lo que se obtuvieron los siguientes resultados (ver Tabla 1).

Se desarrolló una interfaz web a partir de la cual se pueden visualizar los resultados obtenidos por el clasificador de manera clara como se muestra en la Fig. 11. donde se aprecia una sencilla interfaz con la que iniciaremos la búsqueda al pulsar el botón de búsqueda intensiva ubicado en la parte superior.

Tabla 1. Resultados de las pruebas a las que se sometió el algoritmo.

Planta	Imagen de prueba	Resultado obtenido
Buganvilla		pred: Buganbilia [1. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
Árnica		pred: Arnica [0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]
Lengua de suegra		pred: Lengua [0. 0. 1. 0. 0. 0. 0. 0. 0. 0.]
Manzanilla		pred: Manzanilla [0. 0. 0. 1. 0. 0. 0. 0. 0. 0.]
Mastuerzo		pred: Mastuerzo [0. 0. 0. 0. 1. 0. 0. 0. 0. 0.]
Ruda		pred: Ruda [0. 0. 0. 0. 0. 1. 0. 0. 0. 0.]
Yerba de sapo		pred: Y. de sapo [0. 0. 0. 0. 0. 0. 1. 0. 0. 0.]
Sauco		pred: Sauco [0. 0. 0. 0. 0. 0. 0. 1. 0. 0.]
Savile		pred: Savila [0. 0. 0. 0. 0. 0. 0. 0. 1. 0.]
Siempreviva		pred: Siempreviva [0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0.]
Yerba de burro		pred: Y. de burro [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1.]



Fig. 11. Muestra la pantalla de bienvenida al clasificador.

Posteriormente se empiezan a tomar fotos y se van almacenando las incidencias hasta que finalmente se nos envía un resultado de la búsqueda (imagen izquierda) contrastada con una imagen precargada (derecha) donde además se describe de manera sencilla la planta identificada como se muestra en la figura 12.



Fig. 12. Se muestra la interfaz con resultado posterior a haber determinado que se ha encontrado una sábila.

En la parte del backend el programa ejecuta dos acciones, la primera es tomar fotos del entorno haciendo uso de la librería OpenCV y la segunda es identificar la planta ejecutando el algoritmo de la red neuronal explicado en la sección del procedimiento; lo anterior se hace mediante la ejecución de dos hilos que trabajan simultáneamente, mientras con uno se toman muestras del entorno el otro las toma para su análisis (ver Fig. 13).

```
pred: sábila
Foto tomada
pred: sábila
pred: sábila
Foto tomada
pred: sábila
Foto tomada
Foto tomada
```

Fig. 13. Acciones realizadas por el software.

En cuanto al grado de acierto del clasificador se han monitorizado 2 valores durante el entrenamiento de la red neuronal que son la función de pérdida y la métrica accuracy donde nuestra función de pérdida obtuvo un valor del 0.4196 siendo el mejor valor obtenido debido a que en los primeros experimentos superaba las 3 unidades según la métrica implementada, nuestro clasificador identifica con un 84% de eficiencia como se muestra en la figura 14.

loss: 0.4196 - acc: 0.8407

Fig. 14. Resultados obtenidos de la función de pérdida y la métrica implementada.

5. Conclusión y trabajos futuros

A continuación, se muestra la Tabla 2. En ella se realiza la comparativa de los resultados obtenidos con trabajos analizados en el estado del arte.

El primero es el resultado de esta investigación con 84.07% de clasificación, con 11 plantas analizadas de formas completa, el segundo tiene una clasificación de 79.60 % con 27 especies analizando únicamente las hojas.

El tercer proyecto muestra un rango de efectividad de 93.08%, sin embargo, este proyecto se enfoca únicamente en la identificación mediante hojas.

El último proyecto muestra un porcentaje de clasificación del 92.64 % trabajando únicamente con la planta de café y su madurez.

Tabla 2. Comparación del trabajo realizado y similares.

Clasificador	Porcentaje de acierto
Clasificador de plantas medicinales en la región de Jocotitlán	84.07%
Leaves recognition system using a neural network	79.60%
Neural Network Application on Foliage Plant Identification	93.08%

En conclusión, el trabajo propuesto evidencia una mejor clasificación, no obstante, como trabajo futuro se pretende mejorar haciendo uso de más plantas medicinales y posteriormente implementarlo en un robot recolector de basura.

Referencias

1. Calvo, D.: <http://www.diegocalvo.es/red-neuronal-convolucional-cnn/>, último acceso: 2018/10/18
2. Luna González, J.A., Paris, S., Nakano Miyatake M., Robles Camarillo D.: Comparación de Arquitecturas de Redes Neuronales Convolucionales para la Clasificación de Imágenes de Ojos. En: Simposio Iberoamericano Multidisciplinario de Ciencias e Ingeniería., vol. 1, no. 1, pp. 94 –101 (2016)
3. Antea Herrera, A.: Detección de texto utilizando Redes Neuronales Convolucionales. Cataluña: Universidad Politécnica de Cataluña (2015)

4. Quintero, C., Merchán, F., Cornejo, A.: Uso de Redes Neuronales Convolucionales para el Reconocimiento Automático de Imágenes de Macro invertebrados para el Biomonitorio Participativo. Conference Paper, vol. 2018, n° 1, pp. 585–596 (2017)
5. Durán Suárez, J.: Reconocimiento de caracteres escritos a mano. Sevilla: Universidad de Sevilla (2017)
6. Puy, P.: Facultad de Matemática, Astronomía y Física, Universidad Nacional de Cordoba, http://www.famaf.proed.unc.edu.ar/pluginfile.php/19400/mod_resource/content/1/02.evaluacion.pdf, último acceso: 2018/12/10
7. Hernandez Avila, R.: Deep Learning. Una revisión. Innovación y Gestión del Conocimiento, vol. 1, no. 1, pp. 2–7 (2018)
8. Núñez Sánchez, A.: Diseño de un sistema de reconocimiento automático de matrículas de vehículos mediante una Red Neuronal Convolutiva. Cataluña: Universidad Oberta de Cataluña (2016)
9. Matich, D.J.: Redes Neuronales: Conceptos Básicos y Aplicaciones. Universidad Tecnológica Nacional, vol. 1, n° 1, pp. 10–32 (2001)
10. Swain, M., Dash, S. K., Dash, S., Mohapatra, A.: An approach for iris plant classification using neural network. International Journal on Soft Computing, vol. 3, no. 1, pp. 79–89 (2012)
11. Pitarque, A., Ruiz, J.C., Roy, J.F.: Las redes neuronales como herramientas estadísticas no paramétricas de clasificación. Psicothema, vol. 12, no. 2, pp. 459–463 (2000)
12. Gavarini, S., Trípode, M., Tosini, M., Ceccatto, A.: Clasificación de semillas de malezas utilizando Redes Neuronales Artificiales. Instituto Computación aplicada (INCA), vol. 1, no. 1, pp. 1–5 (2010)
13. Sekeroglu, B., Inan, Y.: Leaves recognition system using a neural network. In: 12th International Conference on Application of Fuzzy Systems and Soft Computing, ICAFS, Vols. 1877-0509, no. 102, pp. 579–592 (2016)
14. Gang Wu, S., Sheng Bao, F., You Xu, E., Wang, Y. X., Chang, Y. F., Xiang, Q. L.: A Leaf Recognition Algorithm for Plant Classification Using Probabilistic Neural Network. Dept. of Computer Science, Texas Tech University, vol. 1, no. 0707-4289, pp. 1–5 (2007)
15. Grajales-Múnera, J.E., Restrepo-Martínez A.: Clasificación de Mariposas por Modelos de Color HSI y RGB Usando Redes Neuronales. Tecnológicas, vol. 1, no. 0123-7799, pp. 669–679 (2013)

Clasificación de maderas por corte usando su propio contenido de textura usando la metodología CBIR

Gabriel de Jesús Celis-Escudero¹, Juan Villegas-Cortez¹, Carlos Avilés-Cruz¹, Arturo Zúñiga-López¹, Andrés-Ferreyra Ramírez¹, Roman A. Mora-Gutiérrez², Cesar Benavides-Alvarez³

¹ Universidad Autónoma Metropolitana, Unidad Azcapotzalco, Departamento de Electrónica. San Pablo Xalpa No. 180, Col. Reynosa Tamaulipas, CP 02200, Ciudad de México, México.

² Universidad Autónoma Metropolitana, Unidad Azcapotzalco, Departamento de Sistemas.

³ Universidad Autónoma Metropolitana, Unidad Iztapalapa, Departamento de Ing. Eléctrica, San Rafael Atlixco 186, Vicentina. CP 09340, Ciudad de México, México. gjcelis90@gmail.com, {juanvc, caviles, azl, mgra, fra}@azc.uam.mx, cesarba@xanum.uam.mx

Resumen. En la actualidad la identificación adecuada de la madera a través de sus elementos anatómicos macroscópicos es una actividad importante en la industria y su comercialización. Sin embargo, la identificación por elementos macroscópicos requiere un adiestramiento previo que varios usuarios no poseen. Por ende surge la necesidad de proveer a los usuarios de maderas una herramienta capaz de identificar la especie o género arbóreo a la que pertenece una muestra de madera (tabla, tablón, polín etc.), sin ser expertos agrónomos o botánicos. La textura en las imágenes digitales macroscópicas de madera poseen el potencial para caracterizar, estudiar y clasificar una muestra. En el presente trabajo proponemos clasificar imágenes de muestras de maderas mexicanas con base en su imagen y sus propias características, utilizando la técnica CBIR (Content Based Image Retrieval), siendo la primera aproximación sencilla y efectiva para la madera de especies mexicanas. Se ha entrenado un clasificador de distancia con una base de datos de 39 especies, y se logró una clasificación del 100% de éxito por medio de la técnica de resustitución, con una implementación basada en código abierto en lenguaje Python y librería OpenCV.

Palabras clave: Texturas en las imágenes, Clasificación de maderas mexicanas, CBIR, OpenCV, Reconocimiento de patrones, Tecnología de maderas

Classification of Woods by Cutting Texture by Means of CBIR

Abstract. According to current literature, proper wood identification through its atomic macroscopic elements is an important procedure for the

industry and wood commercialization. Nevertheless, the identification by its macroscopic elements requires previous training which several users do not possess. Therefore, users of wood have the need to identify the taxonomy of woods in order to determine the origin of a wood's sample (such as tablet, plank and pollin) without having expertise on the field. The texture of woods in digital images has the potential to analyze and classify the sample with accuracy. The following paper states a method for classifying Mexican woods samples with the application of Content Based Image Retrieval technique (CBIR). Previous data show a hundred percent success rate in sample classification of 39 species with the CBIR method in conjunction with computer languages such as Python and library OpenCV.

Keywords: Texture of Woods in Digital Images, Classifying Mexican Woods, CBIR, OpenCV, Pattern Recognition

1. Introducción

La inteligencia artificial (IA) busca desarrollar métodos y algoritmos que le permitan comportarse a las computadoras de modo inteligente. La IA comprende un amplio conjunto de algoritmos, y su efectividad esta estrechamente relacionada con la calidad de la información de la cual se aprende con base en el objetivo a alcanzar para tomar decisiones [1].

México es uno de los países mega diversos del mundo; las características geográficas, climáticas y ambientales del mismo determinan que más del 70 % del territorio sea de aptitud preferentemente forestal. El gran potencial forestal de México ha promovido que en los últimos gobiernos se busque incrementar la producción maderable nacional. El potencial maderable del país incluye a una variedad amplia de coníferas tales como: pinos, cedros blancos, entre otros); además de varias latifoliadas como cedro rojo, caoba, palofierro, etc [2].

La gran cantidad de especies arbóreas con aptitud de explotación maderables, plantean la necesidad de un instrumento que permita la identificación y control de la madera para los industriales (aserraderos, transporte de maderas, muebles), en supervisores gubernamentales y usuarios finales. En otras palabras, en la tipificación de maderas es deseable poder conocer la especie maderable con la que se esta trabajando o que se va adquirir. En el presente trabajo la clasificación de maderas se realiza a partir de la imagen de su corte, o en términos técnicos a partir de su textura del corte.

En términos generales se define a la madera como la parte sólida y rígida que se encuentra bajo la piel de los tallos leñosos de los árboles, i.e., es el tejido leñoso que sirve de sostén al árbol [3].

De manera frecuente se clasifica a la madera para diferentes propósitos. La madera se forma por albura y duramen. La albura es madera joven, la cual al paso del tiempo y de procesos físico-químicos se transforman en duramen (también llamada madera vieja, o bien, albura ha pasado por el proceso de duraminización).

En este trabajo mostramos el desarrollo e implementación de un sistema CBIR, aplicado en una base de datos de un grupo de fotografías de maderas mexicanas, aunque se podrá utilizar en otras maderas con un aprendizaje o entrenamiento previo. Al parecer es el primer trabajo de este tipo aplicado a maderas mexicanas, y además en comparación con otros trabajos [4], presentamos un alcance basado en el estudio de textura macroscópica, más sencillo y con la aplicación de caracterizar el patrón de la textura de superficie a partir de rasgos estadísticos. Por ende, las propiedades macroscópicas a considerar en las imágenes de madera son: color, brillo (alto, medio o bajo), textura (efecto visual de la superficie de la madera resultado de la porosidad, anillos de crecimiento y demás estructuras celulares), hilo (forma de crecimiento de las células).

La manera usual de clasificar y por ende determinar las maderas es de manera visual o química, ya sea resinas naturales o artificiales halladas. Muchas veces depende de la habilidad técnica desarrollada por el especialista, o si en algún momento llevo un tratamiento la madera [2]. La metodología CBIR se ha utilizando para el reconocimiento de imágenes por su contenido, por medio de la textura, es así que es prudente su utilización para la clasificación y reconocimiento de maderas por medio de su textura visualizada en el corte.

En la sección 2 presentamos el Estado del arte de la metodología CBIR y sus aplicaciones previas, en la sección 3 la metodología a detalle se explica, en la sección 4 la experimentación y resultados son presentados con explicaciones específicas orientadas para que pueda ser replicada, y los resultados con su análisis se muestran en la sección 5, finalmente en la sección 6 ofrece las conclusiones y el trabajo futuro.

2. Estado del arte

En la última década con el aumento de las imágenes en la Internet, a la par de la masificación de las redes sociales, el aumento de la capacidad de almacenamiento en la red o en la nube junto al auge de las imágenes y el hecho de que no estén clasificadas todas ha propiciado el desarrollo de herramientas que traten de paliar este problema; hablando de cantidades reportadas en el año 2017, se estimó en un billón la cantidad de imágenes digitales acorde a una investigación reportada por el servicio mylio.com (véase URL: <http://tinyurl.com/y2jo5wal>), tal que el 85 % de ellas fueron tomadas por un teléfono móvil. Ante esta realidad se tiene la posibilidad de crear nuevos repositorios de bases de datos de imágenes, pero el reto es clasificarlas acorde a su contenido, de forma no-supervisada, y con una alta eficiencia de forma simple y rápida, todo ello orientado al trabajo fundamental de organizar, proteger y compartir [5].

La técnica CBIR (Content Based Image Retrieval) trabaja y funciona para atacar este problema y tiene resultados probados en identificación y clasificación de imágenes de interés, tales como los escenarios naturales [6] y la clasificación de rostros humanos [7]. Muchos sistemas y herramientas CBIR se han desarrollado para la consulta de imágenes, tal como Query By Image Content (QBIC) desarrollado en los años 90s por IBM, que permite al usuario consultas dentro

de una gran base de datos con base a atributos de la imagen [8].

CBIR es un sistema que tiene dos funciones principales: (i) Inserción de datos, que es un método no supervisado para la extracción de información, tal como color, forma o texturas, de la base de datos para ser guardados autoagrupados por medio de un algoritmo para su futura utilización; y (ii) Procesamiento de consultas, que mediante una imagen de muestra, obteniendo sus características, serán comparadas con métricas específicas hacia los racimos de datos autoagrupados en el paso previo, y así obtener el valor dentro de la clasificación previamente hecha de la base de datos.

Anteriormente la clasificación de las imágenes se llevaba a cabo de forma manual y por medio de metadatos insertados en las imágenes, pero, en una base de datos muy grande conlleva un costo enorme para su clasificación, y otro problema era cuando estos metadatos se realizaban en diferentes idiomas. Con el uso de CBIR se puede automatizar esta tarea de clasificación, ya que se realiza por medio de características de la misma imagen sin necesidad de metadatos, etiquetas o intervención externa.

Cabe mencionar que la investigación anatómica de maderas mexicanas se desarrolló fuertemente a mitad del siglo pasado en México de la mano Investigadores como Juana Huerta Crespo, Ramón Echenique Manrique, Gózaló Novelo Gózales entre otros. Como se mencionó con anterioridad, los estudios de identificación de maderas con base a las propiedades macroscópicas y organolépticas son de suma importancia en el sector forestal, sin embargo en la actualidad la clasificación correcta de la madera depende fuertemente habilidad técnica desarrollada por el especialista [2].

En otros trabajos relacionados, podemos citar a [4], ahí realizan un estudio de la textura de las maderas a nivel microscópica y macroscópica, además de imágenes en estereogramas, a diferencia de la simplicidad de nuestro trabajo donde únicamente usamos fotografías macroscópicas obtenidas de forma controlada en nuestro laboratorio, con muestras comunes de campo; adicionalmente ellos hacen un preprocesado de las imágenes más costoso basado en aplicación de máscaras, y finalmente su mejor aproximación es de un reconocimiento del 90 %, mientras con nuestra propuesta logramos el 100 % de clasificación. Es recomendable revisar ese trabajo para ver los estudios de los intentos previos a su propuesta, mismos que aquí sólo referenciamos.

Se definen como características macroscópicas aquellas cualidades de la madera que pueden ser determinadas a simple vista o haciendo uso de una lupa de 10 aumentos; en contraste, las propiedades organolépticas son aquellas que se aprecian por a través de los sentidos. Las principales cualidades macroscópicas y organolépticas de las maderas a considerar en las clasificaciones son: color, olor sabor, brillo, textura e hilo.

3. Metodología

Para el problema de la clasificación de maderas se propone el modelo mostrado en la figura 1 [7].

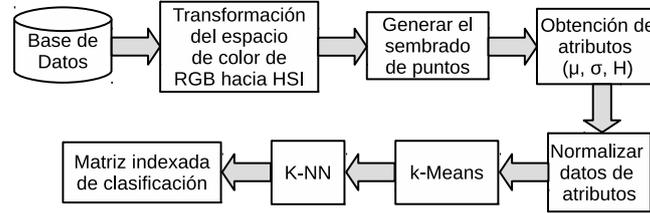


Fig. 1. Metodología aplicada para la clasificación de imágenes de cortes de maderas.

En ella se plantean los pasos siguientes:

- Base de datos de fotografías macroscópicas de maderas, de la cual serán extraídas las imágenes para ser analizadas en su contenido,
- Transformar el espacio de color de RGB hacia HSI [8], esto nos proporcionara la información necesaria, tal como es la textura, luminosidad, saturación de color, etc.
- Generar el sembrado de puntos fijos[8] en forma de malla de 10×10 , véase una muestra en la siguiente página en la figura 2
- Obtener los atributos estadísticos de textura de la imagen, siendo la media μ , desviación estándar σ , y homogeneidad h [9].
 - Para la homogeneidad es necesario calcular la probabilidad de cada una de las combinaciones posibles de los elementos, como se muestra en la ecuación 1, para a continuación realizar el calculo de homogeneidad [10] de la región establecida con la ecuación 2.

$$P_{i,j} = \frac{V_{i,j}}{\sum_{i,j=0}^{N-1} V_{i,j}} \quad (1)$$

$$\sum_{i,j=0}^{N-1} \frac{P_{i,j}}{1 + (i + j)^2} \quad (2)$$

- Normalizar los datos de la matriz obtenida, para ello es necesario obtener el valor máximo y mínimo de cada columna y aplicar la ecuación 3.

$$\frac{Valor - V_{Max}}{V_{Max} - V_{min}} \quad (3)$$

- Una vez normalizada, véase la tabla 1, se crea la matriz indexada, mostrada en la tabla 2 realizada con 20 centroides, donde se encuentran clasificados los datos, observando como están distribuidas las texturas de las imágenes en los k centroides, calculados por k -means,
- Finalmente los datos de la matriz se disponen en orden a partir de la aplicación del algoritmo K -NN basado en métrica euclidiana.

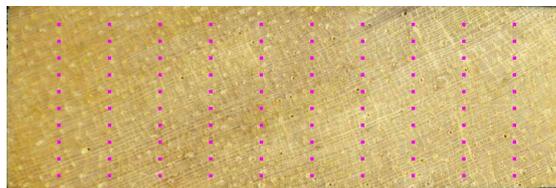


Fig. 2. Distribución regular del sembrado de puntos en la imagen de un corte de madera.

Tabla 1. Datos normalizados de los atributos.

	μ de H	σ de H	h de H	μ de S	σ de S	h de S	μ de I	σ de I	h de I
1	0.1258	0.01025	0.6816	0.2270	0.1254	0.7569	0.8904	0.1894	0.4989
2	0.1262	0.009393	0.8282	0.2956	0.292	0.7677	0.8458	0.209	0.6136
3	0.1287	0.01052	0.7696	0.209	0.1172	0.8099	0.9114	0.1994	0.5722
...	...								
19500	0.09129	0.01496	0.5713	0.6231	0.2982	0.1110	0.3557	0.2583	0.1746

Tabla 2. Matriz indexada, "Matriz de clasificación".

	Cen 1	Cen 2	Cen 3	Cen 4	Cen 5	Cen 6	Cen 7	Cen 8	Cen 9	Cen 10
1	0	0	0	100	0	0	0	0	0	0
2	0	0	0	100	0	0	0	0	0	0
3	0	4	0	96	0	0	0	0	0	0
...	...									
195	35	0	63	0	1	0	0	0	1	0

Una vez realizado el entrenamiento, el siguiente paso es la recuperación. En este proceso realizaremos tres tipos de pruebas de recuperación [11]: La primera por resustitución, estos es, se tomara una imagen dentro de la misma base de aprendizaje para ser consultada y obtener su clasificación; la segunda consta de dejar uno fuera LOO (Leave One Out), donde se toma una muestra de la base de datos que no sera introducida en el proceso de aprendizaje; y tercera por validación cruzada CV (Cross Validation), donde cierto porcentaje de la base de datos no se toma en cuenta en el aprendizaje y se utilizara para prueba como consulta de muestra desconocida, en este trabajo se tomara el 60% de la base de datos para el aprendizaje, y el 40% restante para prueba. Cada uno de los métodos de recuperación son básicamente el mismo, tal como lo muestra la figura 3, la diferencia es la cantidad de imágenes insertadas para el aprendizaje y la consulta. El algoritmo K -NN se ocupa para realizar la recuperación, ya que la principal tarea es realizar una clasificación de las imágenes con los centroides dados por k -Means. En este caso los centroides son la clasificación obtenida en el aprendizaje. Para determinar el porcentaje de precisión se comparara las

primeras 5 imágenes más cercanas a la de consulta.

La implementación se desarrolló en lenguaje Python, de forma secuencial, utilizando la librería OpenCV para el uso de k -Means. El Hardware empleado es una computadora tipo PC con especificaciones: Procesador AMD A8-9600, 4 cores y 4 threads a 3.9Ghz; memoria RAM de 6GB a 2133Mhz; Sistema operativo Linux Ubuntu 18.04 de 64 bits. Las imágenes fueron tomadas con una cámara NIKON D3200 en formato RAW y convertidas a JPG con los mismos valores de revelado.

En el presente trabajo, para obtener una base de datos de imágenes ideales, las fotografías fueron realizadas en un ambiente controlado en el laboratorio de Multimedia de la UAM AZC (URL: <http://zeus.azc.uam.mx/multimedia/>), donde los parámetros: velocidad de obturación, apertura de diafragma, valor de exposición, balance de blancos y cantidad de luz son los mismos. La base de datos consta de 195 elementos y la cual es parte de la contribución del presente trabajo.

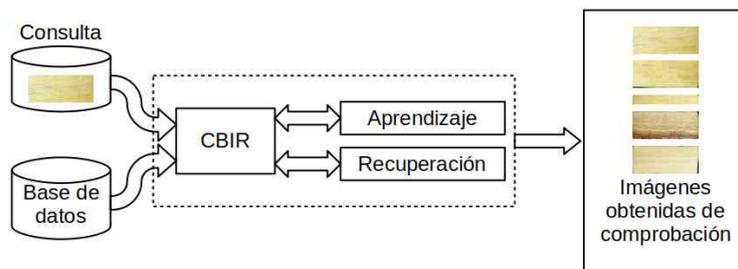


Fig. 3. Metodología de recuperación.

4. Experimentación y resultados

En esta sección mostraremos los resultados de los experimentos, obtenidos por la metodología propuesta.

Para la obtención de resultados, se realizaron dos variaciones en las pruebas; el número de centroides, $K = \{5, 10, 15, 20, 25, 30, 40, 50, 60\}$, y el tamaño de la base de datos introducida en el aprendizaje, tal que son el 100% (toda la base de datos dentro del aprendizaje), 99.49% (se deja una imagen fuera del aprendizaje), y 60% (117 imágenes de la base de datos son introducidas en el aprendizaje).

Las fotografías de la base de datos son de dimensiones variables. El número de iteraciones en k -Means fue de 100, siendo el mismo número en cada proceso, la variación del tiempo de aprendizaje es con base en el número de centroides que se establezcan y la cantidad de imágenes introducidas, como se muestra en la figura 4.

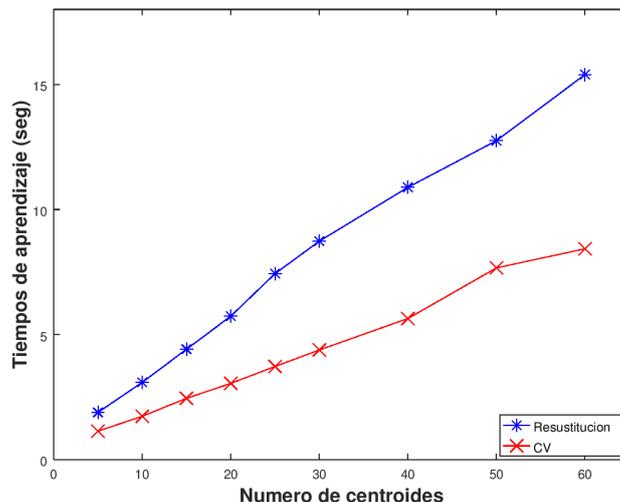


Fig. 4. Tiempos de aprendizaje para recuperación y CV.

La recuperación se basa en encontrar por medio de la clasificación a las imágenes mas cercanas usando la métrica euclidiana. Esto es:

- Carga de la *imagen de consulta*.
- *Transformación del espacio* de color RGB a HSI.
- *Sembrado de puntos*.
- Obtención de *atributos estadísticos*.
- Normalizar los valores de μ , σ y h . Para normalizar se utilizaron los valores de mínimos y máximos obtenidos en el aprendizaje.
- Aplicación de *k-Means* para obtener los centroides, y *K-NN* para clasificar la consulta con base al aprendizaje.
- Por ultimo, se debe introducir el resultado de la clasificación de la consulta a *K-NN* y utilizar como centroides a la clasificación obtenida en el aprendizaje.

Usando la matriz de confusión se realizó el cálculo de porcentaje de precisión de nuestra propuesta, utilizando la ecuación 4, siendo los valores de la diagonal de la matriz de confusión y el calculo de porcentaje de recuperación se lleva acabo con la ecuación 5. En la figura 5 se muestra un ejemplo de las clases obtenidas en el aprendizaje. La clase 1 son maderas claras de brillo alto de textura media e hilo recto; la clase 2 corresponde a maderas pardas de textura media brillo medio; clases 4 a 6 son duramen de maderas con color característico, textura fina de hilo recto; las clases 3, 7 a 9 corresponden a maderas de textura gruesa de hilo recto, la 3 y 7 presentan albura y duramen en la misma muestra, mientras que la muestra 9 muestra un defecto en la madera.

$$Precision = \frac{Numero\ de\ imagenes\ relevantes\ recuperadas}{Total\ de\ numero\ de\ imagenes\ relevantes} \times 100\% \quad (4)$$

$$Recuperacion = \frac{Numero\ de\ imagenes\ relevantes\ recuperadas}{Total\ de\ numero\ de\ imagenes\ de\ la\ base\ de\ datos} \times 100\% \quad (5)$$



Fig. 5. Imágenes de muestra de la base de datos.

Algunos ejemplos de los resultados obtenidos se muestran a continuación. En la figura 6(a) se muestra del método por resustitución las respuestas 1, 2 y 4 son del mismo tipo de madera que la consulta. En la figura 6(b) las imágenes de respuesta 1 y 3, por el método LOO, son del mismo tipo que la de consulta y; en la figura 6(c) por el método de CV la imagen de la respuesta 4 se encuentra en el mismo tipo de madera que la consultada. Estos resultados son los obtenidos con $K = 60$.

5. Resultados

En este trabajo se declara un acierto si una imagen o mas, de las primeras cinco obtenidas por la recuperación pertenecen a la clase de la consulta, ya que las maderas llegan a tener diferentes texturas dependiendo del corte realizado, como el ejemplo mostrado en la figura 7. Como se observa en la figura 6, tres de los elementos de recuperación pertenecen a la clase de la imagen de consulta, y las 2 imágenes faltantes no estarán en las 5 primeras imágenes de recuperación aun cuando son del mismo tipo de madera, ya que en estas se observan los poros de

Consulta	Res #1	Res #2	Res #3	Res #4	Res #5	
						(a)
						(b)
						(c)

Fig. 6. Resultados de la recuperación de 3 imágenes diferentes en los tres tipos de métodos de recuperación: (a) por resustitución, (b) por LOO y (c) por CV.

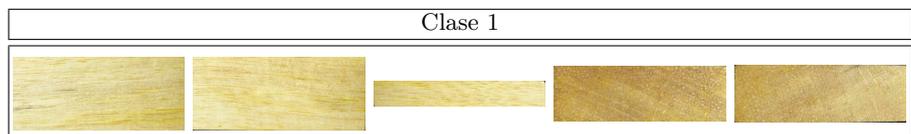


Fig. 7. Ejemplo de los cortes de las fotografías de muestras.

la madera que son típicos en cortes radiales, y en la imagen de consulta los poros son líneas o rayas como los observados en cortes longitudinales o tangenciales.

En la gráfica de la figura 8 se muestra la precisión obtenida de cada uno de los métodos. En las pruebas por resustitución los resultados que se obtienen aun con un número pequeño de centroides, $K = 5$, son de un 100 %; esto mismo sucede con LOO, donde el porcentaje de precisión es de 100 % en todos los casos. Y en CV la precisión varía significativamente, tal que con $K = 5$ la precisión es menor a al 50 % y con $K = 40$ se obtiene una precisión poco mayor al 80 %, siendo esta la mejor.

6. Conclusiones

Hemos presentado en este artículo la primera aproximación a clasificación e identificación de tipos de maderas mexicanas a partir de su imagen y textura de corte, haciendo uso de la técnica CBIR para poder generar un patrón descriptivo de su textura sin el uso de palabras o etiquetas asociadas a un lenguaje o idioma, teniendo como fuente directa de la información al contenido de la propia imagen de estudio, logrando una clasificación del 100 % con las condiciones de imágenes fijas y fotografiadas en ambiente de iluminación controlado.

También estamos compartiendo nuestra base de datos generada para este primer estudio, con la finalidad de que más estudiosos del tema, o personas interesadas en la preservación e identificación de patrones de adecuación de las especies arbóreas, deseen hacer en seguimiento de sus proyectos de investigación y estudios vinculados.

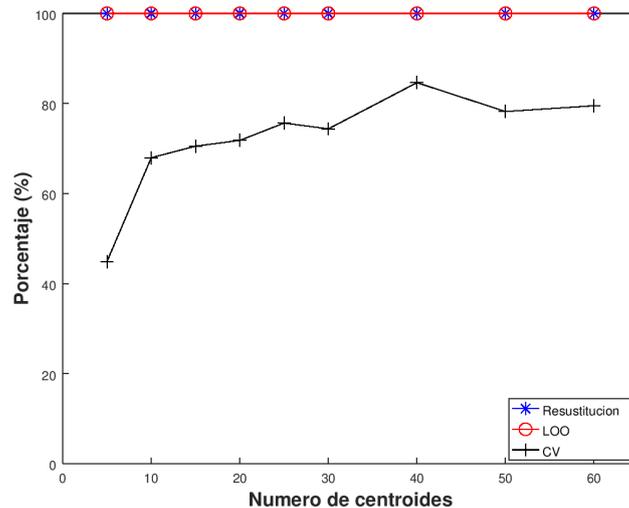


Fig. 8. Porcentaje de precisión en: Resustitución, LOO y CV.

En este primer trabajo de aplicación de visión por computadora apoyada en algoritmos de clasificación por auto agrupamiento y usando una métrica de distancia, consideramos que marca el siguiente paso hacia las vertientes de poder realizar la identificación de las maderas en ambientes no controlados, específicamente en imágenes con ruido, o bien tomadas en entornos laborales, tales como talleres, aserraderos, museos, etc.

Como parte del fortalecimiento de la implementación, también se considera poder aplicar otras metodologías más complejas de cómputo numérico, para comparar el rendimiento versus la simplicidad de la técnica CBIR reportada. En trabajos futuros se integrara más información que permitirá asociar la identificación de la especie y la usabilidad de la madera.

Agradecimientos. Este trabajo ha sido resultado del proyecto de investigación divisional CBI EL006-18 de la Universidad Autónoma Metropolitana unidad Azcapotzalco.

Referencias

1. Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach. Pearson Education, 2 ed. (2003)
2. Vignote-Peña, S., Morales-Peris, F.J., Morales-Mendez, E., Gómez-Pérez, C.: Tecnología de la madera en la construcción arquitectónica. Madrid: Ediciones Mundi-Prensa, 1 ed. (2000)
3. Echenique-Manrique, R.: Características de la madera y su uso en la construcción. México DF.: Lito Ediciones Olimpia, 1 ed. (1971)

4. Wang, H. et al: Wood Recognition Using Image Texture Features. PLoS One. 8, 10, 1–12 (2013)
5. Tyagi, V.: Content-Based Image Retrieval - Ideas, Influences, and Current Trends. Springer, (2017)
6. Serrano-Talamantes, J.F., Aviles-Cruz, C., Villegas-Cortez, J., Sossa-Azuola, J.H.: Self organizing natural scene image retrieval, Expert Systems with Applications, no. 0 (2012)
7. Benavides Alvarez, C., Villegas Cortez, J., Román Alonso, G., Aviles Cruz, C.: Face Classification by Local Texture Analysis through CBIR and SURF Points. Revista IEEE América Latina, Vol. 14, Issue: 5, pp. 2418–2424 (2016)
8. Villegas-Cortez, J., Pérez-Pimentel, Y., Osuma-Galán I.: Implementación de un sistema de reconocimiento de imágenes por contenido usando algoritmos genéticos. México DF. Departamento de electrónica, Universidad Autónoma Metropolitana (2014)
9. Gonzalez R.C., Woods, R.E.: Digital Image Processing. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (1992)
10. Presutti, M.: La matriz de co-ocurrencia en la clasificación multispectral: Tutorial para la enseñanza de medidas texturales en cursos de grado universitario. La Plata. Departamento de ambiente y recursos naturales, Universidad Nacional de La Plata, Facultad de Ciencias Agrarias y Forestales (2004)
11. Benavides-Alvares, C., Villegas-Cortez, J., Román-Alonso, G., Avilés-Cruz, C.: Reconocimientos de rostros a partir de la propia imagen usando técnica CBIR. México DF. Departamento de Ing. Eléctrica, Universidad Autónoma Metropolitana, Iztapalapa (2015)

Configuración de red neuronal para el análisis de polen

Leyra Martínez, Pedro Arguijo, Antonio Hiram Vázquez López,
Roberto Ángel Meléndez Armenta

Tecnológico Nacional de México/Instituto Tecnológico Superior de Misantla,
Misantla, Ver., México
leyramartinez.lm@gmail.com, pedro_arguijo@excite.com,
jahvazquez@itsm.edu.mx, ramelendeza@itsm.edu.mx

Resumen. En la palinología se han aplicado con éxito las redes neuronales para la clasificación de granos de polen. Con este fin se seleccionó RPROP como algoritmo de entrenamiento de la red neuronal para la clasificación de un dataset previamente reportado, se consideraron las variantes con y sin retroceso de peso. Se utilizó un algoritmo genético para optimizar la topología de la red neuronal (número de capas ocultas y número de neuronas por capa), razones de entrenamiento y prueba, pesos iniciales y número de iteraciones. Las topologías encontradas y reportadas en este documento exhiben resultados de clasificación superiores a los previamente reportados. Aunque no es el objetivo final, se pretende que este desarrollo sea útil en la identificación de los tipos de polen en la miel producida en la región de Misantla.

Palabras clave: palinología, dataset, red neuronal, RPROP.

Neural Network Configuration for Pollen Analysis

Abstract. In palynology, neural networks have been successfully applied for the classification of pollen grains. For this purpose, RPROP was selected as training algorithm of a neural network for the classification of a dataset previously reported. The RPROP variants with and without weight backtracking were considered. A genetic algorithm was used to optimize the ANN topology (number of hidden layers and number of neurons per layer), training and test ratios, initial weights and number of iterations. The topologies found and reported in this document shows better classification results than those previously reported. Although it is not the final goal, this development is intended to be useful in identifying the pollen types in honey produced in the Misantla region.

Keywords: palynology, dataset, neural network, RPROP.

1. Introducción

La palinología es una disciplina de la botánica dedicada al estudio del polen y las esporas [1], se centra fundamentalmente en el análisis de la morfología externa que presenta patrones estructurales diferentes al de las variaciones en la exina, que es la pared externa de los granos de polen. El estudio y análisis microscópico de su simetría,

apertura en las paredes, contorno, forma, tamaño, etc., tiene un valor taxonómico y permite distinguir diferentes taxones a distintos niveles: familia, géneros, especies.

El estudio de los granos de polen resulta ser una tarea difícil, en sus diferentes fases, a partir de pequeñas muestras microscópicas. El análisis de estos son una fuente importante de información para muchas aplicaciones científicas e industriales, lo que hace de la palinología una herramienta valiosa para diversas áreas del conocimiento [2].

Desde hace tiempo se han propuesto métodos de sistemas semiautomáticos o automáticos para el análisis de polen basados en técnicas de análisis de imágenes e inteligencia artificial, en particular redes neuronales y máquinas de soporte vectorial [3-6]. En términos generales, las metodologías propuestas segmentan la imagen y extraen características de textura tanto en imágenes de intensidad, como en algún espacio de color. Santiago Tello Mijares y Francisco Flores [6] plantean una metodología para segmentar granos de polen superpuestos. El método separa el polen en función del proceso de cambio medio y la erosión por mínimos basados en la serie de Fibonacci.

De la imagen segmentada extraen características de forma, y del espacio de color CIELab obtienen características de textura de primer y segundo orden. Para el reconocimiento de los granos de polen consideran tres clasificadores: bosque de árboles al azar, perceptrón multicapa y red de Bayes. Encontrando resultados de clasificación del 96.2% y 96.1% en sensibilidad y exactitud con un perceptrón multicapa.

En este documento, se analiza el desempeño de tres topologías de red neuronal entrenadas con el algoritmo RPROP con y sin retroceso de peso para la clasificación de un dataset de granos de polen, el cual consta de 12 clases palinológicas de polen con 618 instancias [6]. Las topologías mencionadas se obtuvieron utilizando un algoritmo genético, el cual determina la arquitectura adecuada que maximiza la exactitud. En la sección 2 se describe RPROP y las topologías utilizadas. El dataset se aborda en la sección 3 y se muestran los resultados obtenidos en la sección 4.

2. Marco teórico

2.1. Redes neuronales

Una red neuronal artificial es un sistema de procesamiento de información que funciona de manera similar a las redes neuronales biológicas [7]. La base de estos modelos son los siguientes supuestos:

- El procesamiento de la información se produce en un gran número de elementos simples llamadas neuronas.
- Las señales se transmiten entre las neuronas a lo largo de los enlaces de conexión.
- A cada enlace de conexión se le asigna un peso que multiplica la señal transmitida.
- Cada neurona aplica una función de activación en su entrada de red (que es la suma de las señales de entrada ponderadas) para obtener su señal de salida.

Existen dos estructuras en el desarrollo de las redes neuronales, redes de una sola capa y redes multicapa. En la Fig. 1 se muestra un ejemplo de una red neuronal, en la cual se indican las interconexiones de la red, así como los pesos entre estas.

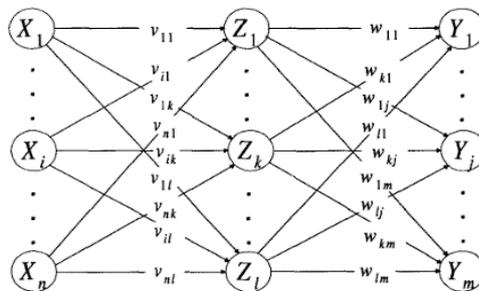


Fig. 1. Estructura de una red neuronal multicapa.

2.2. RPROP

Riedmiller y Braun [8] introdujeron RPROP como una extensión del algoritmo clásico de entrenamiento de propagación hacia atrás (*backpropagation*). RPROP reemplaza la tasa de aprendizaje constante ρ por tamaños de pesos adaptativo Δ_{ij} que se controlan durante el aprendizaje de los pesos w_{ij} . El paso del tamaño Δ_{ij} de los pesos se actualiza como:

$$\Delta_{ij}^{(t-1)} = \begin{cases} \Delta_{ij}^{(t-1)} \cdot \eta^+ & \text{if } \frac{\partial E^{(t)}}{\partial w_{ij}} \cdot \frac{\partial E^{(t-1)}}{\partial w_{ij}} > 0 \\ \Delta_{ij}^{(t-1)} \cdot \eta^- & \text{if } \frac{\partial E^{(t)}}{\partial w_{ij}} \cdot \frac{\partial E^{(t-1)}}{\partial w_{ij}} < 0 \\ \Delta_{ij}^{(t-1)} & \text{else} \end{cases} \quad (1)$$

con parámetros $0 < \eta^- < 1 < \eta^+$, que especifica la magnitud de adaptación del tamaño de paso, si el signo del gradiente $\frac{\partial E^{(t-1)}}{\partial w_{ij}}$ cambia. Donde E indica el error a la salida de la red. Los pesos se actualizan con

$$\Delta w_{ij}^{(t)} = -\text{sin} \left(\frac{\partial E^{(t)}}{\partial w_{ij}} \right) \cdot \Delta_{ij}^{(t)} \quad (2)$$

cada vez que la derivada parcial de la función de error con respecto al peso correspondiente cambia su signo, indica que el valor de la última actualización fue demasiado grande y el algoritmo ha saltado por encima de un mínimo local. El valor de actualización Δ_{ij} se reduce en un factor η^- . Si la derivada conserva su signo, el valor de actualización aumenta ligeramente con el factor η^+ para acelerar la convergencia en regiones poco profundas.

Una de las ventajas del algoritmo RPROP, en comparación con la propagación hacia atrás, es que la magnitud de la derivada parcial no afecta la actualización de peso y depende sólo de los signos de la derivada parcial. Por lo tanto, permite una convergencia más rápida de lo que puede hacer la propagación hacia atrás. RPROP

realiza una adaptación directa de la etapa de ponderación basada en la información del gradiente local. En la Tabla 1, se muestra una comparación entre el algoritmo de entrenamiento *Backpropagation* y su variante RPROP [9].

RPROP considera propagación hacia atrás con y sin retroceso de peso, definida por los algoritmos RPROP⁺ y RPROP⁻, respectivamente [8].

Tabla 1. Comparación de algoritmos.

<i>Backpropagation</i>	<i>RPROP</i>
<ul style="list-style-type: none"> • Emplea un ciclo propagación-adaptación de dos fases. • Es iterativo. • Su convergencia es tardada. • No requiere la normalización de los vectores de entrada. • Es capaz de generalizar, clasificando correctamente patrones ruidosos o incompletos. 	<ul style="list-style-type: none"> • Su escala de requisitos de tiempo y memoria es lineal con el número de parámetros a optimizar. • Proporciona un entrenamiento y una tasa de convergencia más rápido. • Es robusto. • Es considerado como uno de los mejores métodos de aprendizaje entre los algoritmos. • Es complicado de programar.

2.3. Topología

Un aspecto importante de las aplicaciones de redes neuronales es la determinación de una topología adecuada, es decir, el número de capas ocultas y el número de neuronas por capa oculta, razones de entrenamiento y prueba, pesos iniciales y número de iteraciones (épocas). Si bien existen algunas reglas generales, no se puede determinar una solución óptima a este problema en un tiempo razonable. Generalmente, una topología de red se selecciona por prueba y error, optando por la mejor.

Resulta complejo el hecho de descubrir cuántas capas ocultas deben utilizarse, cuántas neuronas debe tener cada capa oculta, qué tipo de función de activación utilizar, qué algoritmo y tasa de aprendizaje debe emplearse para el entrenamiento de la red neuronal. Los algoritmos genéticos (GA) junto con las redes neuronales se han utilizado en diversas aplicaciones [10-13]. Estos se han empleado para encontrar la topología de las capas ocultas, el número de conectividad y parámetros de entrenamiento óptimos para analizar datos biomédicos relacionados con padecimientos cardíacos [13]. Al emplear GA se reemplazan los métodos de prueba y error para determinar la topología óptima de la red neuronal y aumentar la velocidad, precisión y eficiencia de la fase de entrenamiento para así obtener un mayor grado de exactitud en la clasificación. Los GA se utilizan para la optimización de múltiples variables y tienen la capacidad de resolver una amplia gama de problemas de optimización, con el uso de tres operaciones simples: selección, cruce y mutación [13]. Consecuentemente, al combinar algoritmos genéticos y redes neuronales se obtienen arquitecturas óptimas de redes en menos tiempo. Con este enfoque, además de reducir el tiempo de ejecución al ejecutar el algoritmo logrando puntos óptimos. Para este trabajo, dentro del espacio de búsqueda del algoritmo genético se consideró maximizar la exactitud de las configuraciones de la red neuronal. En la Tabla 2 se muestran las 3 configuraciones las

cuales arrojan la mayor exactitud con distintas razones de entrenamiento y prueba. Cabe mencionar que cada una de estas configuraciones tienen 33 neuronas de entrada y 12 neuronas de salida.

Tabla 2. Configuraciones seleccionadas.

	Capa de entrada	Capas ocultas			Épocas	Razón de entrenamiento y prueba
		Primera	Segunda	Tercera		
Configuración 1	33	42	28	18	26	70-30
Configuración 2	33	11	21	45	31	90-10
Configuración 3	33	31	44	19	37	80-20

Como se puede observar en la Tabla 2, el número de neuronas de las capas ocultas para cada una de las configuraciones consideradas son demasiadas, por lo cual no mostramos figura alguna de las topologías analizadas ya que no se alcanzan a distinguir las capas ocultas con sus respectivas neuronas, sus interconexiones y pesos w_{ij} .

3. Dataset

El dataset empleado en este documento consta de 618 instancias divididas en 12 clases diferentes de polen [6]. En la Figura 2 se muestra una imagen representativa de cada una de las clases de polen que se analizan. Las clases que tiene el dataset, con su respectivo nombre palinológico y el número de observaciones por clase se muestra en la Tabla 3.



Fig. 2. Granos de polen por clases del 1- 12 y se representan así: 1-Huizache, 2-Alfalfa,3-Ortiga, 4-Pasto, 5-Hierba, 6-Morera, 7-Pacana, 8-Oliva, 9-Mesquite, 10-Sauce, 11-Pimienta, 12-Sorgo.

Las características reportadas en el dataset se extrajeron de imágenes segmentadas en el espacio de color CIELab. CIELab es un modelo de color perceptual uniforme en el que la componente L representa la luminosidad de un objeto desde el negro (0) al blanco (100), a representa la variación del verde (-) al rojo (-), y b representa la variación del azul (-) al amarillo (+). Una variación numérica de estos valores corresponde aproximadamente a la misma cantidad de cambio percibido visualmente. De las imágenes segmentadas se obtuvieron las características de forma, y del espacio de color CIELab las características de textura de primero y segundo orden, descritas

por matrices de coocurrencia de niveles de gris. Dichos datos se muestran en la Tabla 4. Como se mencionó son un total de 33 características, 6 de forma, 15 de textura de primer orden y 12 de textura de segundo orden.

Tabla 2. Clasificación palinológica del conjunto de datos de imágenes de especie de polen.

Clase	Nombre polen	Observaciones
1	Huisache	46
2	Alfalfa	46
3	Ortiga	56
4	Pasto	37
5	Hierba	42
6	Morera	56
7	Pacana	32
8	Oliva	52
9	Mesquite	80
10	Sauce	64
11	Pimienta	73
12	Sorgo de Alepo	34

Tabla 3. Características del dataset.

Características		
Forma	Textura (CIE-L*a*b*)	
	Primer orden	Segundo orden
Ancho Alto Perímetro Área Circularidad Compactancia	Promedio Media Varianza Desviación estándar Entropía	Contraste Correlación Energía Homogeneidad

4. Resultados

Para probar cada una de las topologías descritas en la Tabla 1 se utilizó el software estadístico R y su interfaz gráfica de usuario RStudio, en una laptop con procesador i3, memoria RAM de 8GB, un sistema operativo de 64 bits.

En la Tabla 5 se muestra un ejemplo de la matriz de confusión obtenida en la configuración 1 utilizando RPROP+, con la proporción de muestreo 70-30, o sea, 70% del dataset para entrenar la red neuronal y 30% para realizar pruebas. Dicha configuración obtuvo una exactitud del 99.35%. En la matriz de confusión el color rojo indica que existió un error en la clasificación, por ejemplo, la alfalfa se confundió tanto con el pasto y con el mesquite.

En la tabla 6 se muestran los resultados obtenidos de cada una de las configuraciones mencionadas previamente, cabe mencionar únicamente se muestran los resultados de las proporciones que obtuvieron la mejor exactitud. Como se mencionó los algoritmos de entrenamiento de las topologías fueron Rprop+ y Rprop-.

Tabla 4. Matriz de confusión 70-30 con el algoritmo Rprop+.

		Referencia												Total	
		70%	1	2	3	4	5	6	7	8	9	10	11		12
Clasificación	1	46	0	0	0	0	0	0	0	0	0	0	0	0	46
	2	0	46	0	1	0	0	0	0	1	0	0	0	0	48
	3	0	0	56	0	0	0	0	0	0	0	0	0	0	56
	4	0	0	0	36	0	0	0	0	0	0	1	0	0	37
	5	0	0	0	0	42	0	0	0	0	0	0	0	0	42
	6	0	0	0	0	0	55	0	0	0	0	0	0	0	55
	7	0	0	0	0	0	0	32	0	0	0	0	0	0	32
	8	0	0	0	0	0	0	0	52	0	0	0	0	0	52
	9	0	0	0	0	0	0	0	0	79	0	0	0	0	79
	10	0	0	0	0	0	1	0	0	0	64	0	0	0	65
	11	0	0	0	0	0	0	0	0	0	0	72	0	0	72
	12	0	0	0	0	0	0	0	0	0	0	0	34	0	34
Total		46	46	56	37	42	56	32	52	80	64	73	34	618	

En las configuraciones 1 y 2, el algoritmo Rprop con retroceso de peso (Rprop+) obtuvo el mejor resultado de exactitud alcanzando valores del 99.35% al 99.83%, mientras que en la configuración 3 el algoritmo Rprop sin retroceso de peso (Rprop-) alcanzó una exactitud del 99.35%, siendo este mayor que el resultado de exactitud de Rprop+ para esta configuración.

La validez de los resultados se comprobó a través de validación cruzada, obteniendo resultados de exactitud entre el 94 y 97%, tal como se muestra. La validación cruzada es un método estadístico utilizado para evaluar, comparar y evitar el sobreentrenamiento de algoritmos de aprendizaje, mediante el uso del subconjunto de datos etiquetados de validación.

En la validación cruzada la información se divide en dos subconjuntos: uno se utiliza para el entrenamiento del modelo, mientras que el otro se utiliza para validarlo, su forma más básica es conocida como k-folds donde los subconjuntos de entrenamiento y validación se intercambian a través de las iteraciones de tal manera que cada subconjunto de datos pueda ser validado.

Tabla 5. Resultados.

	Razón de entrenamiento y prueba	Rprop +	Rprop -	Validación Cruzada
Configuración 1	70-30	99.35%	98.70%	95.40
Configuración 2	90-10	99.83	99.19	94.03
Configuración 3	80-20	98.70	99.35	96.61

5. Conclusión

En este trabajo, se consideraron tres topologías de redes neuronales entrenadas con el algoritmo RPROP para realizar la clasificación polen. Los resultados obtenidos con éstas son superiores a los obtenidos previamente [6], pues en ambos casos se utiliza el mismo dataset. Nuestras topologías arrojan resultados de exactitud por encima del 98%, mientras que en los resultados reportados la exactitud es del 96%, utilizando también

redes neuronales artificiales, pero se desconoce la topología que ellos emplearon para hacer sus respectivos análisis.

Referencias

1. Polen: Propiedades, utilización y beneficios, <https://mejorconsalud.com/polen-propiedades-utilizacion-beneficios/>. Último acceso: 2019/20/03
2. Fundación UNAM Ecología, <http://www.fundacionunam.org.mx/ecologia/la-ciencia-del-polen/>. Último acceso: 2019/03/20
3. Sánchez Mesa, J.A., Galán, C., Martínez Heras, J.A., Hervás Martínez, C.: The use of a neural network to forecast daily grass pollen concentration in a Mediterranean region southern part of the Iberian Peninsula. *Clinical & Experimental Allergy*, vol. 32, no. 11, pp. 1606–1612 (2002)
4. Li, P., Treloar, W.J., Flenley, J.R., Empson L.: vf: vf Towards automation of palynology 2: the use of texture measures and neural network analysis for automated identification of optical images of pollen grains. *Journal of Quaternary Science: Published for the Quaternary Research Association*, vol. 19, no. 8, pp. 755–762 (2004)
5. Li, P., Flenley, J.R.: Pollen texture identification using neural networks. *Grana*, vol. 38, no. 1, pp. 59–64 (1999)
6. Tello-Mijares, S., Flores, F.: A Novel Method for the Separation of Overlapping Pollen Species for Automated Detection and Classification. *Computational and Mathematical Methods in Medicine*, Vol. 2016, Article ID 5689346, 12 pages, (2016)
7. Friedman, M., Kandel, A.: Introduction to Pattern Recognition Statistical, Structural, Neuronal and Fuzzy Logic Approaches. World Scientific Pub Co Inc., March 2, (1999)
8. Riedmiller, M., Braun, H.: Direct adaptive method for faster backpropagation learning: the RPROP algorithm. In: *Proceedings of the IEEE International Conference on Neural Networks*, pp. 586–591 (1993)
9. Reddy, C.S., KVSVN, R.: An optimal neural network model for software effort estimation. *Int. J. of Software Engineering, IJSE*, vol. 3, no. 1, pp. 63–78. (2010)
10. Venkatesan, D., Kannan K., Seravanan, R.: A genetic algorithm-based artificial neural network model for the optimization of machining processes. *Neural Computing and Applications*, vol. 18, no. 2, pp. 135–140 (2009)
11. Karimi, H., Yousefi, F.: Application of artificial neural network-genetic algorithm (ANN-GA) to correlation of density in nanofluids. *Fluid Phase Equilibria* 336, pp. 79–83 (2012)
12. Tonnizam Mohamad, E., Shirani Faradonbeh, R., Jahed Armaghani, D., Monjezi, M., Abd Majid, M.Z.: An optimized ANN model based on genetic algorithm for predicting ripping production. *Neural Computing and Applications* 28, Supplement 1, pp. 393–406 (2017)
13. Rubio Escamilla, U., Arguijo Hernández, S.P.: Configuración de arquitectura de una RNA mediante algoritmos genéticos. *Academia Journals*, vol.10, no. 3, pp. 3058–3064 (2018)

Detección y diagnóstico de fallas en sistemas eléctricos de potencia combinando una red neuronal autoasociativa y una red neuronal probabilística

Juan Pablo Nieto González

Universidad Autónoma de Coahuila, Facultad de Ingeniería
Coahuila, México
juan.nieto@uadec.edu.mx

Resumen. El monitoreo de los sistemas de potencia es particularmente retador debido a la presencia de cambios dinámicos de carga en modo de operación normal en los nodos de la red, presencia de variables continuas y discretas, información con presencia de ruido y falta o exceso de datos. Por ello, la necesidad de desarrollar metodologías más poderosas que combinen técnicas de inteligencia artificial ha sido reconocida. Este artículo propone un sistema de monitoreo basado en los datos históricos del sistema compuesto por dos fases. En la primera fase aprende el comportamiento de la operación normal del sistema utilizando una red neuronal autoasociativa (RNAA), la cual lleva a cabo el proceso de detección. En la segunda fase se da el diagnóstico final empleando una red neuronal probabilística (RNP), la cual clasifica el tipo de falla presente y proporciona su tiempo de ocurrencia.

Palabras clave: detección de fallas, diagnóstico de fallas, sistemas eléctricos de potencia, red neuronal autoasociativa, red neuronal probabilística.

Fault Detection and Diagnosis in Electrical Power Systems Combining an Autoassociative Neural Network and a Probabilistic Neural Network

Abstract. Power systems monitoring is particularly challenging due to the presence of dynamic load changes in normal operation mode of network nodes, the presence of both continuous and discrete variables, noisy information and lack or excess of data. Due to this, the need to develop more powerful approaches combining artificial intelligence techniques has been recognized. This paper proposes a monitoring system based on the system history data composed by two phases. In the first phase it learns the normal operation behavior of the system using an autoassociative neural network (AANN) which carries out the detection process. In the second phase the final diagnosis is given using a probabilistic neural

network (PNN), which classifies the type of fault present and gives its time of occurrence.

Keywords: fault detection, fault diagnosis, electrical power systems, autoassociative neural network, probabilistic neural network.

1. Introducción

El monitoreo de los sistemas de ingeniería está relacionado a la detección y el diagnóstico de fallas. Primeramente se detectan patrones específicos de comportamiento en los datos observados y posteriormente se genera un diagnóstico completo del sistema.

Un moderno sistema eléctrico de potencia (SEP) es un ejemplo de tal sistema de ingeniería complejo. En él existen un gran número de sensores, controladores y módulos de cómputo que recolectan una gran cantidad de señales. Considerando que los modernos SEP, ya sean de una sola planta industrial, de una ciudad, región o país completo, son sistemas de gran escala. Y puesto que de ellos se puede extraer una gran cantidad de datos, es que se propone una metodología de monitoreo que emplea solamente datos históricos del proceso. Para llevar a cabo la detección y diagnóstico de fallas, en el presente artículo, se combina una red neuronal autoasociativa (RNAA) y una red neuronal probabilística (RNP). La metodología se aplica a un SEP de 24 nodos que contiene cambios dinámicos de carga propuesto por la IEEE mostrando resultados prometedores. La organización del artículo es la siguiente. La sección 2 revisa el estado del arte.

La sección 3 da los preliminares matemáticos de la RNAA y de la RNP. La sección 4 proporciona la descripción general de la propuesta. La sección 5 muestra un caso de estudio. Finalmente la sección 6 da la conclusión del artículo.

2. Estado del arte

Desde el punto de vista de seguridad y confiabilidad de los sistemas eléctricos, es necesario tener un oportuno diagnóstico de fallas. Lo anterior para poder detectar, aislar y diagnosticar fallas, así como dar aviso a los operadores del sistema para tomar las correspondientes acciones correctivas. Durante un disturbio, hay un gran número de eventos relacionados a las fallas, haciendo que el diagnóstico y la decisión de tomar acciones correctivas se torne una tarea difícil.

En este dominio, la necesidad de desarrollar técnicas más poderosas ha sido reconocida, y las técnicas híbridas que combinan varios métodos de razonamiento se han empezado a emplear. [1] considera la configuración de elementos automáticos en los modernos sistemas de potencia eléctrica, tales como relevadores de protección y de recierre automáticos para mejorar un modelo analítico y de optimización para el diagnóstico de fallas de sistemas de potencia. De acuerdo al principio de protección de los relevadores, el diagnóstico de fallas es expresado como un problema de programación entera y resuelto por el algoritmo genético de búsqueda Tabú. [2] presenta una metodología que utiliza redes neuronales

integradas con varias técnicas estadísticas. Entre las herramientas numéricas y estadísticas utiliza análisis de Fourier, valores RMS, valores de sesgo y de curtosis así como componentes simétricas para la detección y la identificación de las fallas es llevada a cabo mediante una red neuronal perceptrón multicapa. [3] propone un método compuesto de dos fases: En la primera fase una red neuronal probabilística es entrenada con los eigenvalores de los datos de voltaje obtenidos en operación normal así como con fallas simétricas y asimétricas. La segunda fase emplea una comparación entre las muestras para detectar y localizar la presencia de una falla. [4] utiliza lecturas de la corriente de fase solo durante el primer cuarto de ciclo empleando un método que combina componentes simétricas con un análisis de componentes principales para identificar y clasificar una falla. [5] propone una red bayesiana y minería de datos para diagnosticar fallas en una red eléctrica. El estatus de la información de las protecciones es tomado como atributos condicionales y la región de falla como un atributo de decisión. [6] propone una metodología capaz de localizar nodos de un sistema eléctrico en modo de falla. La metodología está compuesta por dos fases: en la primer fase una red neuronal probabilística se entrena con los eigenvalores de voltaje para dar una clasificación del tipo de falla presente. En la segunda fase se utiliza un ANFIS para dar el diagnóstico final. [7] se enfoca en el diseño de algoritmos de diagnóstico de fallas empleados como requisito para un control tolerante a fallas de un sistema eléctrico de distribución de energía. Emplea métodos basados en modelo y en particular utiliza relaciones analíticas redundantes para detectar cuando es que la red eléctrica ha tenido cambios dinámicos de carga. Este artículo emplea las ecuaciones básicas de la teoría de circuitos eléctricos y la topología de redes para llevar a cabo la detección de fallas y su correspondiente diagnóstico. [8] propone una metodología para la detección y diagnóstico de fallas de SEP utilizando un conjunto de rasgos característicos tiempo-frecuencia para la identificación, supervisión y detección de fallas en un controlador de flujo de potencia. La metodología extrae algunos rasgos característicos de una ventana de datos conteniendo un ciclo de la onda e incluye 3/16 de ciclo post falla empleando la transformada S rápida discreta ortonormal. Luego emplea una máquina de soporte vectorial para llevar a cabo la clasificación de las fallas. [9] muestra que la integración de sensores inteligentes para la detección y aislamiento de la falla en los SEP incrementa la resiliencia y robustez del control. Se monitorea el estado de los sensores sin emplear redundancia en el hardware. Propone un sensor que trabaja en tiempo real para la detección y aislamiento de fallas mediante el empleo de un algoritmo inteligente. La propuesta es probada en el sistema de 14 nodos propuesto por la IEEE. [10] propone una metodología de detección y clasificación de fallas en SEP utilizando el ángulo del factor de potencia. Los ángulos del voltaje y la corriente de los dos extremos de las líneas son comparados para realizar la detección y clasificación de las fallas. La técnica es aplicada al SEP de 15 nodos propuesto por la IEEE. [11] propone un sistema de detección y clasificación de fallas en SEP basado en el método de transformada wavelet empírica y la energía local. La transformada wavelet empírica es empleada para el proceso de detección y para la obtención del tiempo

de ocurrencia. Luego emplea una máquina de soporte vectorial para llevar a cabo la clasificación del tipo de falla basada en rasgos característicos de la energía local del sistema. [12] muestra una técnica descentralizada para la detección de fallas para SEP con puesta a tierra resonante. El propósito del artículo es detectar fallas de fase a tierra e identificar en que nodo sucedió dentro de los tres ciclos posteriores a la ocurrencia de la falla. Las fallas se detectan basandose en el desplazamiento del voltaje del neutro. Los voltajes de prefalla y post falla son utilizados para detectar la fase que se encuentra en modo de falla. Para identificar el nodo que se encuentra en modo de falla, se emplea una herramienta de procesamiento de señales llamada morfología matemática. La metodología se prueba en un SEP propuesto por la IEEE.

3. Preliminares

3.1. Red neuronal autoasociativa (RNAA)

Nieto [6] propuso una RNAA utilizada como un método de Análisis de Componentes Principales No-Lineales (ACP NL) para identificar y remover correlaciones entre las variables de un problema como una ayuda para reducir la dimensionalidad, visualización de datos, y análisis exploratorio de datos.

ACP NL opera entrenando una red neuronal tipo feed-forward para llevar a cabo un mapeo idéntico de las entradas de la red y reproducirlas en la capa de salida. La red contiene una capa interna que actúa como un cuello de botella (la capa contiene una menor cantidad de neuronas que las capas de entrada o de salida), lo cual obliga a la red a desarrollar una representación compacta de los datos de entrada. La RNAA está compuesta por cinco capas. La Figura 1 muestra la arquitectura de la RNAA.

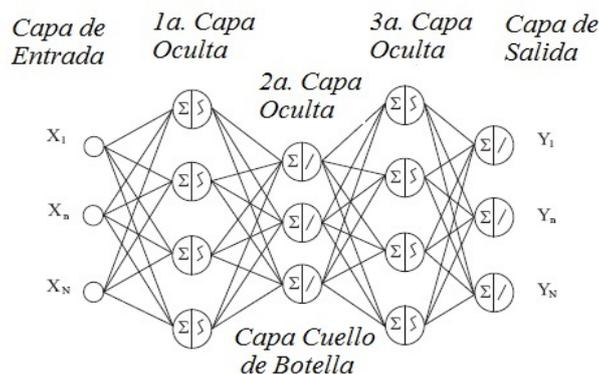


Fig. 1. Arquitectura de una RNAA.

Esta RNAA tiene una capa de entrada y una capa de salida, cada una con N neuronas y tres capas ocultas con H_1 , H_2 , y H_3 neuronas respectivamente.

Cuando una observación x es presentada en la entrada de la red, la ecuación de la neurona de salida de la l^{th} capa es función de las neuronas en la $(l-1)^{th}$ capa, dada por la ecuación 1 y mostrada en la Figura 2:

$$y_j = f\left(\sum_{i=1}^{n(l-1)} w_{i,j}^{(l)} y_i^{(l-1)}\right), \quad l = 1, \dots, 4, \quad (1)$$

donde $y_j^{(0)} = x_n$, $n = 1, \dots, N$ representa las componentes del vector de observaciones a la entrada de la red; $y_n^{(4)}$, $n = 1, \dots, N$, representa las componentes de la estimación de x dada a la salida de la red; $n(l-1)$ da el número de neuronas en la $(l-1)^{th}$ capa. La función $f(\cdot)$ es la función de activación de la neurona, la cual es sigmoideal. Las observaciones deben ser estandarizadas para que caigan dentro de un hipercono unitario.

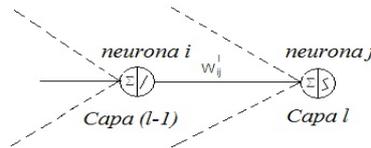


Fig. 2. Conexión de la i^{th} neurona de la $(l-1)$ capa con la j^{th} neurona de la l capa.

Si la red tiene que aprender una tarea específica, es necesario ajustar los pesos de las conexiones entre las neuronas para minimizar la diferencia entre la salida esperada y la salida dada por la RNAA. Esta minimización es llevada a cabo cuando se calcula el error diferencial. El método más comunmente usado es el de retropropagación.

3.2. Red neuronal probabilística (RNP)

Las redes neuronales probabilísticas (RNP) son conceptualmente similares a los modelos de los K vecinos más cercanos (KNN) por sus siglas en inglés. La idea básica se muestra en la Figura 3, la cual es que una variable es clasificada dentro de la clase a la que pertenezcan los K vecinos más cercanos de las variables predictoras. De la Figura 3 se asume que cada caso en el conjunto de entrenamiento tiene dos variables predictoras x y y . Los casos son graficados utilizando sus coordenadas x, y como se muestra en la figura.

También se asume que la variable objetivo tiene dos categorías, positiva la cual es denotada por un cuadrado y negativa, la cual es denotada por un guión. Puede notarse que el triángulo es posicionado casi exactamente sobre el guión, representando un valor negativo. Pero el guión se encuentra en una posición inusual comparado con los otros guiones, los cuales están agrupados por debajo de los cuadrados y a la izquierda del centro. Por lo anterior, parecería que el nuevo guión es un caso aislado. La clasificación de acuerdo al vecino más cercano

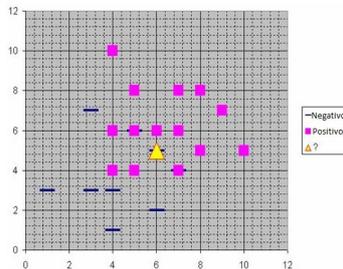


Fig. 3. Las RNP son conceptualmente similares a los K vecinos más cercanos.

dependerá de que tantos puntos en la vecindad sean considerados. Si se considera solamente un punto en la vecindad, se observa que el nuevo punto debería ser clasificado como negativo ya que se encuentra encima de un punto negativo conocido. Por otro lado, si se considera una clasificación empleando los 9 vecinos más cercanos, los 9 puntos más cercanos son considerados y entonces el efecto de los 8 puntos que rodean al punto nuevo lo clasificarán como un punto positivo.

Una red neuronal probabilística (RNP) se construye bajo este fundamento y se generaliza para considerar a todos los puntos presentes para la clasificación de una nueva observación. La distancia es calculada desde el punto siendo evaluado hacia cada uno de los otros puntos, y una función de base radial (FBR), también llamada función kernel es aplicada a la distancia para calcular el peso o influencia de cada uno de los puntos. La función de base radial es llamada así porque la distancia radial es el argumento de la función. $Peso:FBR(distancia)$ por lo que mientras mas lejano se encuentre algún punto con respecto a la nueva observación, menos influencia tiene sobre ella. Se pueden emplear diferentes tipos de funciones de base radial, pero la mas común es la función *Gaussiana*. La arquitectura de la RNP se muestra en la figura 4. El modelo tiene dos capas: la capa de la base radial y la capa competitiva.

Hay Q pares de vectores de entrada/objetivo. Cada uno de los vectores objetivo tiene K elementos. Uno de estos elementos es 1 y el resto es 0. Así, cada vector de entrada esta asociado con una de las K clases. Cuando se presenta una entrada, el bloque $||dist||$ produce un vector cuyos elementos indican que tan cerca se encuentra la entrada de los vectores del conjunto de entrenamiento. Un vector de entrada cercano a un vector de entrenamiento se representa por un numero cercano a 1 en la salida de a^1 . Si una entrada esta cerca de varios vectores de entrada de una sola clase, ésta es representada por varios elementos de a^1 cercanos a 1. Cada vector tiene un 1 solamente en el renglón asociado con la clase particular de la entrada y 0 en los demás elementos. La multiplicación Ta^1 suma los elementos de a^1 debidos a cada una de las K clases de entrada. Finalmente, la segunda capa, produce un 1 correspondiente al mayor elemento de n^2 y 0 en cualquier otro elemento. Así, la red ha clasificado el vector de entrada dentro de una de las K clases debido a que esa clase tuvo la máxima probabilidad de ser la correcta.

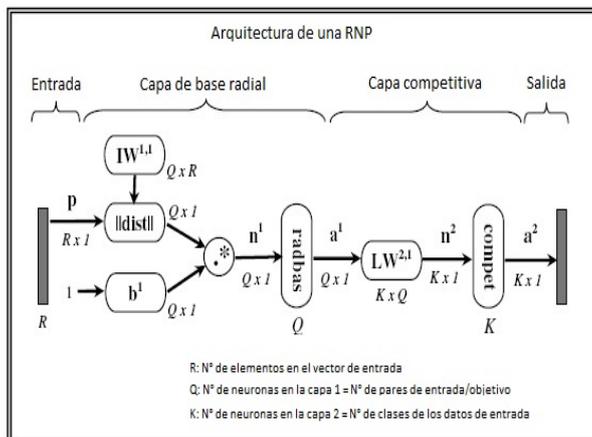


Fig. 4. Arquitectura de una RNP.

4. Descripción de la propuesta

La presente propuesta es una metodología basada en el historial del proceso. Es necesario obtener bases de datos en modo de operación normal del SEP para entrenar una RNAA y así llevar a cabo el proceso de detección. Adicionalmente, se requieren bases de datos, que contengan información de las diferentes fallas que se pudieran presentar en el sistema que se monitorea. Las bases de datos de las posibles fallas se emplean para llevar a cabo el aprendizaje de la RNP. La cual se encarga de dar el diagnóstico final cuando el sistema se encuentra en modo de falla. El proceso general de entrenamiento para la fase de detección y para la de diagnóstico del sistema de monitoreo se muestra en la figura 5

El algoritmo para el proceso de entrenamiento se resume de la siguiente manera:

1. Localizar las mediciones provenientes de los sensores de las variables que se desean monitorear.
2. De manera aleatoria tomar un subconjunto ($\approx 80\%$) de la cantidad total de datos para entrenar las herramientas de inteligencia artificial empleadas por la propuesta en la primera y segunda fase.

Para la Primera Fase

3. Tomar bases de datos de la operación normal del sistema.
4. Estandarizar el subconjunto de datos. Entrenar la RNAA y aprender el modelo.
5. Obtener los residuos de las condiciones de operación normal y sus correspondientes límites (umbrales).

Para la Segunda Fase

6. Tomar bases de datos de las diferentes fallas que pudieran presentarse en el sistema.

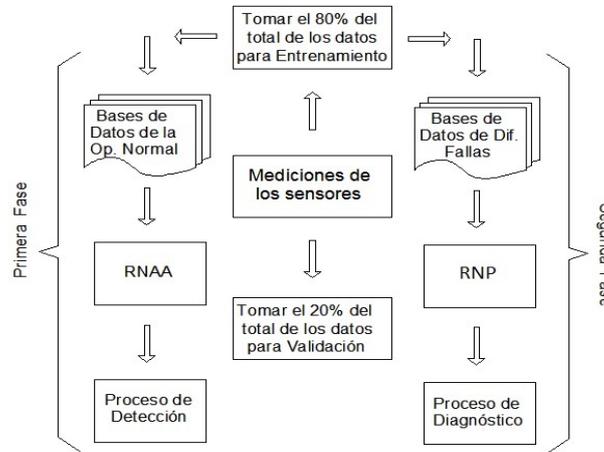


Fig. 5. Proceso general de entrenamiento para la fase de detección y la de diagnóstico del sistema de monitoreo.

7. Estandarizar el subconjunto de datos. Entrenar la RNP y aprender el modelo.
8. La RNP arroja el diagnóstico final.

Durante la primera fase del diagnóstico, el proceso de detección se lleva a cabo al obtener los residuos generados por la RNAA. Primeramente, se debe de realizar un proceso de estandarización para poder manipular las diferentes variables que se monitorean sobre una misma escala. Posteriormente se genera aleatoriamente un subconjunto de muestras formado por el 80 % del total de la base de datos. Dicho subconjunto de muestras es aprendido por la RNAA. Una RNAA de cinco capas, es una red cuyas salidas son entrenadas para emular las entradas sobre un adecuado rango dinámico. Esta característica de la red es muy importante para monitorear variables de sistemas complejos que presentan un cierto grado de correlación entre ellas, ya que cada salida recibe información de cada una de las entradas. Durante el entrenamiento, para hacer que cada salida iguale a su correspondiente entrada, las interrelaciones entre todas las variables de entrada y cada salida individual se refleja en los pesos de conexión de la red. Como resultado de lo anterior se tiene que cada salida específica e incluso la correspondiente salida, refleja solo una pequeña fracción del cambio de la entrada sobre un rango de valores razonablemente amplio. Ésto permite a la RNAA detectar la presencia de una falla, al comparar simplemente cada una de las entradas con su correspondiente salida, obteniendo de esta manera los residuos. Posteriormente se calculan los límites de dichos residuos para condiciones de operación normal del sistema.

Una vez que el proceso de detección es llevado a cabo, la segunda fase de la propuesta empieza a trabajar. Ésta utiliza la RNP para clasificar la falla presente previamente detectada por la primera fase. La salida de la RNP indica cual variable se encuentra en modo de falla, y el tiempo en el cual dicha falla ha

ocurrido. De esta manera, se proporciona un diagnóstico completo del sistema que se monitorea. La Figura 6 muestra el esquema general de operación en línea, del sistema de diagnóstico propuesto.

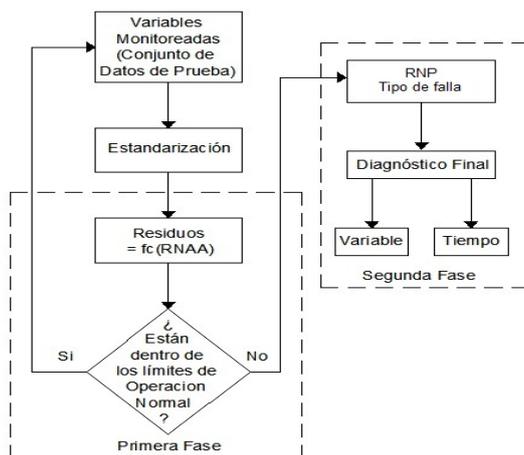


Fig. 6. Esquema de operación en línea del sistema de diagnóstico propuesto.

5. Caso de estudio

En esta sección se muestra el desempeño de la metodología propuesta. Para ello se lleva a cabo la simulación del SEP de 24 nodos propuesto por la *IEEE* y que presenta cambios dinámicos de carga. Tal SEP se muestra en la figura 7.

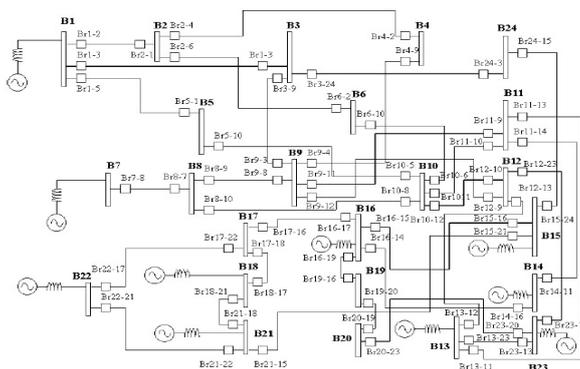


Fig. 7. Diagrama unifilar del SEP de 24 nodos propuesto por la IEEE.

El sistema de monitoreo propuesto se adecuó al caso de estudio como se muestra en el esquema de la figura 8 y se describe como sigue:

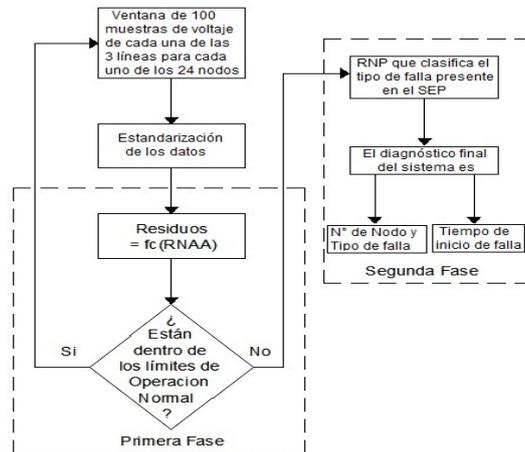


Fig. 8. Esquema de operación en línea del sistema de diagnóstico propuesto.

Se llevaron a cabo 24 simulaciones de falla para determinar el desempeño de la propuesta. Se incluyeron fallas simétricas y fallas asimétricas de forma aleatoria en los nodos 3, 9 10 y 13. Se tomaron en consideración diferentes escenarios y la presencia de múltiples fallas. Se combinaron fallas como: una línea a tierra (A GND), dos líneas a tierra (A-B GND), tres líneas a tierra (A-B-C GND) o fallas entre líneas (A-B o B-C) y el modo de no falla o modo de operación normal (NO FAULT).

La metodología propuesta se aplica como sigue:

1. Obtener ventanas de 100 muestras de los voltaje de cada una de las 3 líneas de los 24 nodos que conforman el SEP que se monitorea.
2. Estandarizar los valores.

Primera Fase

3. La salida de la RNAA muestra los residuos entre las entradas y salidas. Si los residuos están dentro de los límites de operación normal, regresar al paso 1. En caso contrario continuar con el paso 4.

Segunda Fase

4. La RNP clasifica sus entradas y en su salida arroja como diagnóstico final el número de nodo en el que está presente la falla, el tipo de falla presente y el tiempo en el cual da inicio el modo de falla.

En las siguientes tablas se muestra el desempeño de la propuesta tomando en cuenta 3 posibles casos:

Caso 1. El SEP trabaja adecuadamente durante las primeras 25 muestras del total de las 100 observaciones, es decir, 25 muestras están en modo de operación

normal y 75 en modo de falla.

Caso 2. Se toman 50 muestras en modo de operación normal y 50 en modo de falla.

Caso 3. Toma 75 muestras en modo de operación normal y 25 en modo de falla.

Las tablas 1 y 2 muestran un ejemplo de como fueron obtenidos los porcentajes. Las tablas 3 y 4 presentan un resumen de los porcentajes obtenidos para cada uno de los casos considerados.

Tabla 1. Detalle de desempeño en la detección y diagnóstico según el estado del nodo con 25 muestras en modo de operación normal y 75 con falla presente (caso 1).

Estado del Nodo	Diagnóstico Correcto	Falsa Alarma	Precisión
A-B-C GND	14	0	100 %
A-B GND	10	0	100 %
A GND	14	0	100 %
A-B	16	2	88.88 %
B-C	14	2	87.5 %
SIN FALLA	17	7	70.8 %

Tabla 2. Detalle de desempeño en la detección y diagnóstico por número de nodo con 25 muestras en modo de operación normal y 75 con falla presente (caso 1).

Nodo Número	Diagnóstico Correcto	Falsa Alarma	Precisión
3	22	2	91.66 %
9	20	4	83.33 %
10	22	2	91.66 %
13	20	4	83.33 %

Tabla 3. Precisión en la detección y diagnóstico del estado del nodo para los diferentes casos

Estado del Nodo	Caso 1	Caso 2	Caso 3
A-B-C GND	100 %	100 %	100 %
A-B GND	100 %	100 %	100 %
A GND	100 %	85.71 %	100 %
A-B	88.88 %	83.33 %	77.77 %
B-C	87.5 %	87.5 %	75 %
SIN FALLA	83.33 %	83.33 %	79.16 %

Cabe mencionar que se llevaron a cabo varias pruebas con el total de las muestras en la ventana de datos en modo de operación normal y la metodología propuesta detectó el 100 % de las veces como nodos *SIN FALLA*.

Tabla 4. Precisión en la detección y diagnóstico por número de nodo para los diferentes casos.

Estado del Nodo	Caso 1	Caso 2	Caso 3
3	91.66 %	87.5 %	83.33 %
9	83.33 %	87.5 %	79.16 %
10	91.66 %	87.5 %	83.33 %
13	83.33 %	87.5 %	79.16 %

5.1. Comparación contra dos sistemas de diagnóstico

Para observar el desempeño general, se llevo a cabo una comparación contra dos sistemas de diagnóstico similares. El primero de ellos es un sistema basado en lógica probabilística tomado de [13] y el otro es la propuesta de [3]. Las tablas 5 y 6 muestran el desempeño del sistema de diagnóstico basado en lógica probabilística. Las tablas 7 y 8 muestran el desempeño del sistema propuesto en [3].

Tabla 5. Desempeño en la detección y diagnóstico por nodo del sistema basado en lógica probabilística.

Estado del Nodo	Correcto	Falsa Alarma	Precisión
A-B-C GND	14	0	100 %
A-B GND	10	0	100 %
A GND	12	2	85.7 %
A-B	15	3	83.3 %
B-C	16	0	100 %
SIN FALLA	17	7	70.8 %

Tabla 6. Desempeño en la detección y diagnóstico por número de nodo del sistema basado en lógica probabilística

Nodo Número	Diagnóstico Correcto	Falsa Alarma	Precisión
3	19	5	79.1 %
9	21	3	87.5 %
10	21	3	87.5 %
13	23	1	95.8 %

Comparando los resultados de las 3 propuestas, puede notarse que en general, las 3 tienen un desempeño muy similar. Si se compara el caso 1 de la propuesta del presente artículo contra el caso 1 de los otros dos artículos, se observa que la presente metodología tiene un mejor desempeño. Otro punto importante es que la nueva propuesta es relativamente más fácil de implementar y de actualizar cuando el SEP crece.

Tabla 7. Desempeño en la detección y diagnóstico por nodo del sistema utilizando la propuesta de [3].

Estado del Nodo	Correcto	Falsa Alarma	Precisión
A-B-C GND	14	0	100 %
A-B GND	10	0	100 %
A GND	14	0	100 %
A-B	18	0	100 %
B-C	16	0	100 %
SIN FALLA	13	11	54.16 %

Tabla 8. Desempeño en la detección y diagnóstico por número de nodo del sistema utilizando la propuesta de [3].

Nodo Número	Diagnóstico Correcto	Falsa Alarma	Precisión
3	20	4	83.33 %
9	19	5	79.16 %
10	22	2	91.66 %
13	24	0	100 %

6. Conclusión

El artículo presenta una metodología de un sistema de monitoreo para la detección y diagnóstico de fallas en SEP con cambios de carga dinámica. La propuesta, se basa en los datos históricos del sistema por monitorear. Lo anterior es debido a que en la práctica tales datos son relativamente fácil de obtener ya que en la actualidad los sistemas complejos contienen al menos una computadora para su control, la cual almacena el comportamiento diario de éstos.

Una ventaja sobre los métodos basados en modelo, es que esta metodología necesita los datos históricos del sistema o proceso en modo de operación normal, y de las posibles fallas que se puedan presentar.

La metodología esta compuesta por 2 fases. En la primera fase se lleva a cabo el proceso de detección de falla presente en el sistema empleando una RNAA. Se emplea esta red, porque tiene la peculiaridad de que sus salidas siguen a sus entradas. De esta manera se pueden obtener los límites de operación normal, así como los residuos correspondientes entre sus salidas y entradas. Logrando así llevar a cabo el proceso de detección.

Una vez detectada una falla se pasa a una segunda fase compuesta por una RNP. Es en esta segunda fase dónde la RNP clasifica el tipo de falla presente en el sistema. Arrojando como diagnóstico final el número del nodo que se encuentra en modo de falla, el tipo de falla presente en éste nodo y el tiempo en cual inicia. Se decidió el empleo de una RNAA por la facilidad de obtener los residuos y lograr así la detección de la falla y la RNP para el diagnóstico debido a que este tipo de red neuronal presenta un buen desempeño al trabajar con problemas de clasificación. Adicionalmente, el empleo de estos 2 tipos de redes ofrece la ventaja de que necesitan poco tiempo para su entrenamiento.

Referencias

1. Liao, Z., Wen, F., Guo, W., He, X., Jiang, W., Dong, T., Liang, J., Xu, B.: An Analytical Model and Optimization Technique Based Methods For Fault Diagnosis in Power Systems. In: IEEE Third International Conference on Electric Utility Deregulation and Restructuring and Power Technologies, Nanjing, China, pp. 1388–1393 (2008)
2. Flauzino, R.A., Ziolkowski, V., da Silva, I.N., de Souza, D.M.B.: Hybrid Intelligent Architecture for Fault Identification in Power Distribution Systems. In: PES'09, IEEE Power & Energy Society General Meeting, pp. 1–6 (2009)
3. Nieto, J.P., Garza, L.E., Morales, R.: Multiple Fault Diagnosis in Electrical Power Systems with Dynamic Load Changes Using Probabilistic Neural Networks. *Computación y Sistemas*, vol. 14, no. 1, pp. 17–30 (2010)
4. Alsafasfeh, Q., Abdel-Qader, I., Harb, A.: Symmetrical Pattern and PCA Based Framework for Fault Detection and Classification in Power Systems. In: Conference on Electro Information Technology, pp. 1–5 (2010)
5. Qianwen, N., Youyuan, W.: An Augmented Naive Bayesian Power Network Fault Diagnosis Method based on Data Mining. In: Asia-Pacific Power and Energy Engineering Conference (APPEEC), pp. 1–4 (2011)
6. Nieto, J.P.: Multiple Fault Diagnosis in Electrical Power Systems with Dynamic Load Changes Using Soft Computing. In: 11th Mexican International Conference on Artificial Intelligence, MICAI 2012, San Luis Potosi, Mexico, vol. 2, pp. 319–330 (2012)
7. Knuppel, T., Blanke, M.: Fault diagnosis for electrical distribution systems using structural analysis. *International Journal of Robust and Nonlinear Control*, vol. 24, issue 8-9, pp 1446–1465, Special Issue: Fault Tolerant Control Power of Grids (2014)
8. Moravej, Z., Pazoki, M.: New Pattern-Recognition Method for Fault Analysis in Transmission Line With UPFC. *IEEE Transactions on Power Delivery*, vol. 30, issue 3, pp. 1231–1242 (2015)
9. Yang, H., Touria, E., Edrington, C.: Real-time sensor fault detection and isolation in power system with hardware implementation. In: North American Power Symposium (NAPS), Denver, CO, USA (2016)
10. Heydari, O., Khodabakhshi-Javinani, N., Askarian-Abyaneh, H.: A novel technique in detection and classification of faults using power factor angle. In: Conference on Electrical Power Distribution Networks Conference (EPDC), Semnan, Iran (2017)
11. Huang, N., Qi, J., Li, F., Yang, D., Cai, G., Huang, G., Zheng, J., Li, Z.: Short-Circuit Fault Detection and Classification Using Empirical Wavelet Transform and Local Energy for Electric Transmission Line. *Journal of Sensors*, vol. 17, issue 9, pp. 2133–2145 (2017)
12. Barik, Md. A., Gargoom, A., Mahmud, Md. A., Haque, Md. E., Al-Khalidi, H., Maung, A.: A Decentralized Fault Detection Technique for Detecting Single Phase to Ground Faults in Power Distribution Systems With Resonant Grounding. *IEEE Transactions on Power Delivery*, vol. 33, issue 5, pp. 2462–2473 (2018)
13. Garza, L.: Hybrid Systems Fault Diagnosis with a Probabilistic Logic Reasoning Framework. PhD Thesis (2001)

Balanced data of the Guillain-Barré Syndrome using SMOTE for the classification of subtypes

Manuel Torres-Vásquez, José Hernández-Torruco, Betania Hernández-Ocaña,
Oscar Chávez-Bosquez

Universidad Juárez Autónoma de Tabasco, División Académica de Informática y
Sistemas, Cunduacán, Tabasco, México

181H18001@alumno.ujat.mx, {jose.hernandezt,betania.hernandez,
oscar.chavez}@ujat.mx

Resumen. El trastorno neurológico llamado Síndrome de Guillain-Barré (SGB) afecta principalmente al sistema nervioso periférico y se caracteriza por un rápido desarrollo, en algunos casos es mortal. Los subtipos del SGB son principalmente Polineuropatía Desmielinizante Inflamatoria Aguda (AIDP), Neuropatía Axonal Motora Aguda (AMAN), Neuropatía Axonal Sensorial Aguda (AMSAN) y Síndrome de Miller-Fisher (MF). Cada subtipo debe tratarse de manera diferente, el tratamiento y los costos varían según el subtipo desarrollado. Por lo tanto, es esencial diagnosticar temprano el subtipo de SGB que sufre el paciente. El conjunto de datos incluye 129 registros de pacientes mexicanos. Este conjunto de datos se ha utilizado para construir una serie de modelos predictivos utilizando clasificadores simples y combinados Debido a que los datos originales del SGB están desbalanceados, balanceamos los datos a través del sobremuestreo sintético y utilizamos clasificadores simples para mejorar los modelos predictivos previos. En este estudio, usamos la Técnica de Sobremuestreo de Minorías Sintéticas (SMOTE) para balancear los datos, y aplicamos tres clasificadores simples con diferentes enfoques: C4.5, SVM, JRip. Utilizamos la técnica OvO (*One for One*) para obtener 6 subproblemas binarios de subtipos SGB. Usamos la curva ROC (*Receiver Operating Characteristic*) y la prueba de Wilcoxon para evaluar el rendimiento de los modelos obtenidos con datos desbalanceados en comparación con los modelos obtenidos con los datos balanceados para cada caso. Los resultados mostraron que la técnica SMOTE combinada con el clasificador JRip obtuvo los mejores resultados. Sin embargo, en algunos casos, los resultados con datos desbalanceados mejoraron los resultados utilizando datos balanceados.

Palabras clave: desbalanceo de datos, clasificación multiclase, SVM, JRip, C4.5, curva ROC, prueba Wilcoxon.

Balancing Guillain-Barré Data Using SMOTE for Subtype Classification

Abstract. The neurological disorder called Guillain-Barré Syndrome (GBS) mainly affects the peripheral nervous system and it is charac-

terized by a fast development, in some cases is mortal. The GBS subtypes are mainly Acute Inflammatory Demyelinating Polyneuropathy (AIDP), Acute Motor Axonal Neuropathy (AMAN), Acute Motor Sensory Axonal Neuropathy (AMSAN), and Miller-Fisher Syndrome (MF). Each subtype must be treated differently, treatment and costs differ according to the subtype developed by the patient. So, it is essential to early diagnose the subtype of GBS that the patient suffers. The dataset includes 129 records of Mexican patients. This dataset has been used to build a number of predictive models using single classifiers and ensemble methods. As the original GBS data is imbalanced, we balanced the data through synthetic oversampling and we used single classifiers to possible improve previous predictive models. In this study, we use SMOTE (Synthetic Minority Over-sampling Technique) for balancing data and we applied three single classifiers with different approaches: C4.5, SVM, JRip. We used OvO (One versus One) technique to obtain 6 binary subproblems of SGB subtypes. The ROC (Receiver Operating Characteristic) curve and Wilcoxon test both evaluated performance of the imbalanced models versus better balanced data for each case. Results showed that the SMOTE technique combined with the JRip classifier obtained the best results. However, some cases, results with imbalanced data improved results using balanced data.

Keywords: imbalance data, multiclass classification, SVM, JRip, C4.5, ROC curve, Wilcoxon test.

1. Introducción

El Síndrome de Guillain-Barré (SGB) es un trastorno autoinmune en el que el sistema inmunológico del cuerpo ataca los nervios, causando debilidad muscular y, a veces, parálisis. Mientras que la mayoría de las personas eventualmente se recuperan completamente del SGB, otras se quedan con daño nervioso permanente. En raras ocasiones el SGB puede llevar a la muerte. Algunas de sus principales características son [21]:

- La causa es desconocida, pero está asociada con infección respiratoria o gastrointestinal.
- Causa serios daños a la habilidad motora.
- Afecta de 1.3 a 2 por cada 100,000 personas anualmente.
- Tiene una tasa de mortalidad del cinco al quince por ciento.

Los cuatro subtipos principales del SGB son: Polineuropatía Desmielinizante Inflamatoria Aguda (AIDP, *Acute Inflammatory Demyelinating Polyneuropathy*), Neuropatía Axonal Motora Aguda (AMAN, *Acute Motor Axonal Neuropathy*), Neuropatía Axonal Sensorial Aguda (AMSAN, *Acute Motor Sensory Axonal Neuropathy*) y Síndrome de Miller-Fisher (MF). En la Tabla 1 se describen las características de cada uno de los subtipos de SGB.

Es esencial diagnosticar el subtipo de SGB que el paciente sufre de manera temprana. Cada subtipo debe tratarse de manera diferente, el tratamiento y

Tabla 1. Características de los subtipos del SGB.

Polineuropatía Desmielinizante Inflamatoria Aguda		
Características patológicas	Características clínicas	Enfermedades previas
Desmielinización periférica multifocal. Remielinización lenta. Probablemente mecanismos inmunitarios tanto humorales como celulares.	El subtipo más común (hasta el 90 por ciento de los casos de SGB en los Estados Unidos). Debilidad progresiva, simétrica; hiporreflexia o arreflexia.	Virus del herpes. Campylobacter jejuni. Micoplasma. Neumonía.
Neuropatía Axonal Motora Aguda		
Características patológicas	Características clínicas	Enfermedades previas
Anticuerpos contra los gangliósidos GM1, GD1a, GalNAc-GD1a y GD1b en los axones del nervio motor periférico; no desmielinización.	La neuropatía axonal aguda del motor o del sensor sensorial representa del 5 al 10 por ciento de los casos de SGB. Fuertemente asociado con Campylobacter jejuniinfection; Más común en el verano, en pacientes más jóvenes y en China o Japón. Sólo síntomas motores. Los reflejos profundos del tendón pueden ser preservados.	Campylobacter jejuni.
Neuropatía Axonal Sensorial Aguda		
Características patológicas	Características clínicas	Enfermedades previas
Mecanismo similar al de la neuropatía axonal motora aguda, pero con degeneración axonal sensorial.	Similar a los de la neuropatía axonal motora aguda, pero con afectación predominantemente sensorial.	Campylobacter jejuni.
Síndrome Miller Fisher		
Características patológicas	Características clínicas	Enfermedades previas
Desmielinización. Anticuerpos de inmunoglobulina G contra los gangliósidos GQ1b, GD3 y GT1a.	Raro (3 por ciento de los casos de SGB en los Estados Unidos) Oftalmoplejia bilateral. Ataxia. Areflexia. Facial, debilidad bulbar ocurre en el 50 por ciento de los casos. Tronco, la debilidad de la extremidad se produce en el 50 por ciento de los casos.	Ocasionalmente, Campylobacter jejuni, la mayoría de los casos con historia desconocida o Mycoplasma pneumoniae.

los costos varían según el subtipo desarrollado por el paciente. El método de diagnóstico tradicional de los subtipos SGB es a través de estudios fisiológicos y de laboratorio, que son tardados y costosos.

Hoy en día, el aprendizaje automático se utiliza ampliamente para diagnosticar enfermedades, como el cáncer y el Alzheimer. Sin embargo, un número limitado de trabajos aborda el SGB. En [16] se aplicó la regresión logística multinomial para identificar los predictores clínicos y electrodiagnósticos de la parálisis respiratoria neuromuscular en el SGB. Los predictores encontrados fueron la progresión temprana hacia discapacidad pico, la disfunción bulbar y la inestabilidad autonómica. En otro trabajo [13], aplicaron regresión logística para identificar las variables clínicas asociadas con la necesidad de ventilación mecánica en los pacientes con SGB. Las variables identificadas fueron el inicio simultáneo de la debilidad motora de las extremidades superiores e inferiores como síntoma inicial, el grado de potencia de la extremidad superior $<3/5$ en el nadir y la debilidad bulbar. En [19] se aplicó análisis discriminante lineal disperso para optimizar la precisión electrodiagnóstica de los subtipos del SGB.

En este estudio, nuestro objetivo fue investigar si a través de técnicas de balanceo de datos es posible mejorar los modelos predictivos para el SGB creados anteriormente, donde el conjunto de datos estaba desbalanceado. Estos modelos se crearon utilizando clasificadores simples y combinados, utilizando los datos desbalanceados originales [1,2]. Dado que este es un estudio exploratorio, para balancear los datos utilizamos SMOTE (*Synthetic Minority Over-sampling TEchnique*) [3], una técnica ampliamente utilizada con resultados probados.

Aplicamos tres clasificadores simples con diferentes enfoques: C4.5 (árbol de decisión, basada en instancias), SVM (*Support Vector Machine*) basado en kernel, y JRip (basado en reglas). Comparamos su rendimiento en experimentos que consisten en balancear los datos al 100 %, 200 %, 300 %, 400 %, 500 %, 1000 % para 6 subproblemas binarios de subtipos SGB utilizando el enfoque OvO (*One versus One*). El rendimiento de los modelos se evaluó mediante la curva de características operativas del receptor (ROC del inglés *Receiver Operating Characteristic*), medida típica para conjuntos de datos desbalanceados. Utilizamos la prueba no paramétrica de Wilcoxon [5] para investigar una diferencia estadística entre los modelos desbalanceados y balanceados.

Este artículo está organizado de la siguiente forma: En la sección 2 presentamos una descripción del conjunto de datos, los algoritmos de aprendizaje automático y la medida de rendimiento utilizada en el estudio. La sección 3 describe el procedimiento experimental. En la sección 4 mostramos y discutimos los resultados experimentales. Finalmente, en la sección 5 resumimos los resultados, damos conclusiones del estudio y también sugerimos algunos trabajos futuros.

2. Materiales y métodos

2.1. Conjunto de datos

Los datos se recopilaron en el Instituto Nacional de Neurología y Neurocirugía (México) de 1993 a 2002. Se tienen registros de 129 pacientes, cada uno

clasificado con uno de los subtipos del SGB: 20 AIDP, 37 AMAN, 59 AMSAN, 13 Miller-Fisher.

De acuerdo con un estudio anterior [9], seleccionamos los atributos relevantes que se muestran en la Tabla 2. Las primeras cuatro variables son valores enteros, las siguientes son reales.

Tabla 2. Variables utilizadas en este trabajo.

Etiqueta	Nombre de las variables
v22	Simetría (en debilidad)
v29	Afectación de los músculos extraoculares
v30	Ptosis
v31	Implicación cerebelosa
v63	Amplitud del nervio motor mediano izquierdo
v106	Área bajo la curva del nervio motor cubital izquierdo
v120	Área bajo la curva del nervio motor cubital derecho
v130	Amplitud del nervio motor tibial izquierdo
v141	Amplitud del nervio motor tibial derecho
v161	Área bajo la curva del nervio motor peroneo derecho
v172	Amplitud del nervio sensorial mediano izquierdo
v177	Amplitud del nervio sensorial mediano derecho
v178	Área bajo la curva del nervio sensorial mediano derecho
v186	Latencia del nervio sensorial cubital derecho
v187	Amplitud del nervio sensorial cubital derecho
v198	Área bajo la curva del nervio sensorial sural derecho

2.2. Algoritmos de aprendizaje automático

SMOTE. Utilizamos la técnica de preprocesamiento SMOTE para sobremuestrear la clase minoritaria hasta igualar la clase mayoritaria de nuestro conjunto de datos desbalanceado.

En [3] se introdujo uno de los métodos de muestreo más utilizados y con mayor éxito en casos de problemas de desbalanceo de clases binario, la Técnica de Sobremuestreo de Minorías Sintéticas: SMOTE (*Synthetic Minority Over-sampling TEchnique*). Esta técnica sobremuestra la clase minoritaria al crear datos sintéticos o artificiales basado en las similitudes del espacio de características entre los ejemplos de minoría existentes, a diferencia de otras técnicas que solo reemplazan los datos en forma aleatoria. SMOTE toma la clase minoritaria y la sobremuestra tomando cada muestra de clase minoritaria e introduciendo ejemplos sintéticos a lo largo de los segmentos de línea que unen a cualquiera de los vecinos más cercanos a la clase k minoritaria. Dependiendo de la cantidad de sobremuestreo requerido, los vecinos de los vecinos más cercanos se eligen al azar. Estos nuevos datos creados en forma sintética mejoran las técnicas

anteriores que reemplazan el sobremuestreo en forma simple. Los datos sintéticos aumentan el conjunto de datos original ayudando al clasificador a crear regiones de decisión mas grande y menos específicas logrando una mejora significativa en el clasificador [6].

El algoritmo SMOTE realiza los siguientes pasos:

- Recibe como parámetro el porcentaje de ejemplos a sobremuestrear.
- Calcula el número de ejemplos que tiene que generar.
- Calcula los k vecinos más cercanos de los ejemplos de la clase minoritaria.
- Genera los ejemplos siguiendo este proceso:
 - Para cada ejemplo de la clase minoritaria, elige aleatoriamente el vecino a utilizar para crear el nuevo ejemplo.
 - Para cada atributo del ejemplo a sobremuestrear, calcula la diferencia entre el vector de atributos muestra y el vecino elegido.
 - Multiplica esta diferencia por un número aleatorio entre 0 y 1.
 - Suma este último valor al valor original de la muestra.
 - Devuelve el conjunto de ejemplos sintéticos.

En la Figura 1 podemos observar el funcionamiento de SMOTE. Este crea objetos sintéticos en la clase minoritaria mediante la interpolación de un objeto y sus k vecinos más cercanos. En la Figura 1a podemos observar el conjunto de datos formado por dos clases una clase mayoritaria y otra minoritaria. En la Figura 1b muestra los vecinos más cercanos seleccionados para aplicar SMOTE. También se observan las instancias creadas en forma sintética de la clase minoritaria. En la Figura 1c se muestra el conjunto de datos balanceados a través del sobremuestreo.

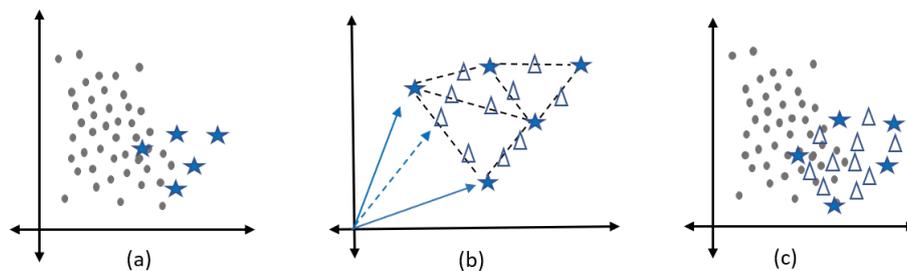


Fig. 1. Generación de datos sintéticos con SMOTE.

Clasificadores simples C4.5 (árbol de decisión, basada en instancias), Máquina de Soporte Vectorial (SVM del inglés *Support Vector Machine*, basado en kernel), JRip (basado en reglas). Utilizamos estos clasificadores para discriminar las instancias de cada clase y comparar los resultados.

2.3. Medidas de rendimiento

Utilizamos la curva ROC, como medida de rendimiento para el conjunto de datos desbalanceado. Es un gráfico bidimensional que proporciona un buen resumen del rendimiento de un modelo de clasificación en presencia de conjuntos de datos desbalanceados con costos de error desiguales [8].

El área de la gráfica tiene un valor entre 0.5 y 1, donde un valor de 1 representa un diagnóstico perfecto y 0.5 representa una prueba sin diagnóstico de capacidad discriminatoria.

2.4. Técnica de binarización

Dividimos el conjunto de datos en subproblemas binarios utilizando el enfoque OvO. Esta técnica entrena un clasificador para cada par de clases posibles $(n-1)/2$ (aprendizaje por pares), por lo tanto, obtuvimos seis pares de conjuntos de datos desbalanceados con el mismo número de combinaciones.

Utilizamos la técnica de binarización OvO, ampliamente utilizada en problemas de clasificación. OvO tiene un mejor desempeño al aplicar la prueba de Wilcoxon en comparación con la técnica OvA [7]. Desde una perspectiva médica, OvO puede ayudar a los médicos a distinguir un subtipo de otro.

2.5. Validación

Utilizamos la evaluación *train-test* para cada clasificador individual, empleando dos tercios de los datos para el entrenamiento y un tercio para las pruebas.

3. Procedimiento experimental

Comenzamos creando 6 conjuntos de datos binarios a partir del conjunto de datos original utilizando OvO:

- SGB1 (57 instancias): AIDP (20 instancias) vs AMAN (37 instancias).
- SGB2 (79 instancias): AIDP (20 instancias) vs AMSAN (59 instancias).
- SGB3 (33 instancias): AIDP (20 instancias) vs MF (13 instancias).
- SGB4 (96 instancias): AMAN (37 instancias) vs AMSAN (59 instancias).
- SGB5 (50 instancias): AMAN (37 instancias) vs MF (13 instancias).
- SGB6 (72 instancias): AMSAN (59 instancias) vs MF (13 instancias).

Luego, dividimos cada uno de los conjuntos de datos en dos conjuntos, 66 % para los datos de entrenamiento y 34 % para los datos de prueba. Balanceamos los datos de entrenamiento utilizando SMOTE. La clase minoritaria de cada conjunto de entrenamiento se sobremuestrea al 100 %, 200 %, 300 %, 400 %, 500 %, 1000 %, según la literatura [3].

La Tabla 3 muestra el número de instancias de la clase minoritaria para cada conjunto de datos binarios. La primera columna muestra los seis casos de SGB n formados por la técnica OvO. La segunda columna muestra las instancias

originales correspondientes a la clase minoritaria del conjunto de datos original para cada caso. En la tercera columna, se puede ver el número de instancias de entrenamiento para cada caso. En las columnas 4 a 9, el resultado del sobremuestreo de las instancias de entrenamiento se muestra en 100 %, 200 %, 300 %, 400 %, 500 % y 1000 % consecutivamente.

Tabla 3. Sobremuestreo de la clase minoritaria utilizando SMOTE.

SGB	Original	Entrenamiento	100 %	200 %	300 %	400 %	500 %	1000 %
SGB1	20	11	22	33	44	55	66	121
SGB2	20	14	28	42	56	70	84	154
SGB3	13	9	18	27	36	45	54	99
SGB4	37	24	48	72	96	120	144	264
SGB5	13	9	18	27	36	45	54	99
SGB6	13	10	20	30	40	50	60	110

Para cada conjunto de datos SGB_n , calculamos el promedio del resultado de la curva ROC de 60 ejecuciones independientes con diferentes semillas utilizando los datos desbalanceados y balanceados al 100 %, 200 %, 300 %, 400 %, 500 % y 1000 % para los tres clasificadores. Comparamos en cada caso el promedio del modelo desbalanceado contra el mejor promedio del modelo balanceado. Realizamos la prueba no paramétrica de Wilcoxon [5] para investigar una diferencia estadística entre los modelos desbalanceados y balanceados. Utilizamos un valor de significación de 0.05. Utilizamos pruebas no paramétricas porque las condiciones iniciales que garantizan la credibilidad de las pruebas paramétricas pueden no cumplirse, lo que hace que el análisis estadístico sea menos confiable con este tipo de pruebas [7].

Los experimentos de R se realizaron en RStudio 1.1.423, con SMOTE del paquete DMwR 0.4.1 [18]. Utilizamos el paquete RWeka 0.4-39 [22], incluidos los clasificadores C4.5 y JRip. Clasificadores SVM del paquete e1071 1.7-0 [11]. Otros paquetes utilizados fueron rJava [20], caret [10], rpart [17], imbalance [4], Proc [14], lattice [15], y rpart.plot [12].

SVM se optimizó a través de la función *tune*, asignando los valores 0.001, 0.01, .1, 1, 10, 50, 80, 100 para el parámetro C.

4. Resultados y discusión

La Tabla 4 muestra la comparación de los resultados de la clasificación utilizando un conjunto de datos desbalanceados con los resultados de la clasificación utilizando el conjunto de datos balanceados al 100 %, 200 %, 300 %, 400 %, 500 % y 1000 % para cada clasificador, ROC fue la métrica utilizada. Cada valor es el promedio de los resultados obtenidos en las 60 ejecuciones. El mejor resultado de cada clasificador se resalta en negrita. La primera columna muestra los casos de SGB y su número de instancias. La última columna muestra el resultado de la

Tabla 4. Resultados de clasificación promedio de datos desbalanceados contra datos balanceados.

Subtipo (instancias)	Clasificador	Datos des-		Balanceo (%)						Resultados de prueba Wilcoxon
		balanceados		100	200	300	400	500	1000	
		ROC	ROC	ROC	ROC	ROC	ROC	ROC	ROC	
SGB1	C4.5	0.9563	0.9479	0.9438	0.9493	0.9528	0.9556	0.9576	NS	
AIDP-AMAN (20-37)	SVM	0.9618	0.9618	0.9632	0.9639	0.9625	0.9632	0.9632	NC	
	JRip	0.9507	0.9458	0.9424	0.9403	0.9382	0.9319	0.9389	NA	
SGB2	C4.5	0.8656	0.8430	0.8485	0.8375	0.8502	0.8607	0.8551	NA	
AIDP-AMSAN (20-59)	SVM	0.9618	0.9618	0.9632	0.9639	0.9625	0.9632	0.9632	NS	
	JRip	0.8472	0.8455	0.8285	0.8480	0.8561	0.8480	0.8549	NS	
SGB3	C4.5	0.8132	0.8313	0.7986	0.7889	0.7729	0.7958	0.7965	NS	
AIDP-MF (20-13)	SVM	0.7097	0.6306	0.6486	0.6472	0.6465	0.6563	0.6535	NA	
	JRip	0.8458	0.8250	0.7778	0.7750	0.7646	0.7292	0.7382	NA	
SGB4	C4.5	0.9132	0.9079	0.9096	0.9172	0.9062	0.9207	0.9093	NS	
AMAN-AMSAN (37-59)	SVM	0.8863	0.8836	0.8844	0.8843	0.8840	0.8821	0.8827	NA	
	JRip	0.8809	0.8955	0.9042	0.9019	0.9043	0.9071	0.9065	S	
SGB5	C4.5	0.8736	0.8868	0.8792	0.8833	0.8701	0.8861	0.8868	NS	
AMAN-MF (37-13)	SVM	0.8910	0.8611	0.8715	0.8792	0.8840	0.8847	0.8847	NA	
	JRip	0.8799	0.9014	0.8799	0.8875	0.8903	0.8861	0.8889	NS	
SGB6	C4.5	0.8007	0.8697	0.8185	0.8287	0.8084	0.8041	0.7839	S	
AMSAN-MF (59-13)	SVM	0.7388	0.7882	0.7646	0.7651	0.7522	0.7469	0.7534	S	
	JRip	0.8430	0.8720	0.8393	0.8306	0.8213	0.8111	0.8061	S	

prueba de Wilcoxon, donde *NS* no representa una diferencia significativa entre los resultados que utilizan datos desbalanceados y el mejor resultado después de balancear los datos; *NC* representa que la prueba no se pudo realizar debido a la similitud entre los datos comparados; *NA* significa que no se realizó ninguna prueba porque el mejor valor se obtuvo utilizando datos desbalanceados; **S** representa que los resultados son estadísticamente significativos.

Para SGB1, dos clasificadores mejoraron el rendimiento utilizando datos balanceados: C4.5 con un valor ROC de 0.9576 y balanceo de 1000 %, y SVM con valor ROC de 0.9639 y un balanceo de 300 %. En el primer caso, no hubo diferencia significativa con respecto a los datos desbalanceados, según la prueba de Wilcoxon. En el segundo caso, no se pudo realizar la prueba de Wilcoxon debido que los resultados obtenidos con el conjunto de datos desbalanceado y el balanceado al 300 % fueron similares. En el tercer caso, JRip no obtuvo un mejor rendimiento con los datos balanceados en ningún caso.

Para SGB2, dos clasificadores mejoraron el rendimiento utilizando datos balanceados: SVM con un valor ROC de 0.9639 y un balanceo del 300 %, y JRip con un valor ROC de 0.8561 y un balanceo de 400 %. En el primer caso, el clasificador C4.5 obtuvo un mejor rendimiento con datos desbalanceados con respecto a los datos balanceados. En el segundo y tercer caso, la prueba de Wilcoxon no encontró diferencia significativa con respecto a los datos desbalanceados.

En SGB3, C4.5 mejoró el rendimiento utilizando datos balanceados con un valor de ROC 0.8313 y un balanceo del 100 %. En el primer caso, la prueba de

Wilcoxon no encontró diferencia significativa con respecto a los datos desbalanceados. En el segundo y tercer caso, no hubo mejor rendimiento para SVM y JRip utilizando datos balanceados.

Para SGB4, dos clasificadores mejoraron el rendimiento utilizando datos balanceados: C4.5 con un valor ROC de 0.9207 y un balanceo del 500 %, y JRip con un valor ROC de 0.9071 y un balanceo del 500 %. En el primer caso, la prueba de Wilcoxon no encontró diferencia significativa con respecto a los datos desbalanceados. En el segundo caso, SVM no obtuvo un mejor rendimiento utilizando datos balanceados. En el tercer caso, la prueba de Wilcoxon encontró una diferencia significativa entre los datos desbalanceados y balanceados.

En SGB5, dos clasificadores mejoraron el rendimiento utilizando datos balanceados: C4.5 con un valor ROC de 0.8868 y un balanceo del 100 %, y JRip con un valor ROC de 0.9014 y un balanceo del 100 %. En el primer y tercer caso, la prueba de Wilcoxon no encontró diferencia significativa con respecto a los datos desbalanceados. En el segundo caso, no hubo mejor rendimiento utilizando datos balanceados.

Para SGB6, los tres clasificadores mejoraron el rendimiento utilizando datos balanceados: C4.5 con un valor ROC de 0.8697 y un balanceo del 100 %, SVM con un valor ROC de 0.7882 y un balanceo del 100 % y JRip con un valor ROC de 0.8720 y un balanceo del 100 %. La prueba de Wilcoxon encontró una diferencia significativa entre los datos desbalanceados y los datos balanceados en los tres casos.

Los tres clasificadores mejoraron el rendimiento en 6 casos. C4.5 mejoró un caso para datos desbalanceados y cinco casos utilizando datos balanceados. Sin embargo, solo un caso obtuvo diferencia significativa. SVM mejoró tres casos para datos desbalanceados y tres utilizando datos balanceados. Sin embargo, solo un caso obtuvo diferencia significativa. JRip mejoró dos casos para datos desbalanceados y cuatro casos con datos balanceados, dos casos con diferencias significativas. Todos los casos mejorados de los tres clasificadores se obtuvieron con datos balanceados.

Se obtuvo un mejor rendimiento en 12 casos utilizando datos balanceados y 6 casos obtuvieron un mejor rendimiento con el conjunto de datos desbalanceados. El balanceo al 100 % obtuvo el mayor rendimiento (seis casos, tres de ellos con una diferencia significativa). El balanceo a 200 % no obtuvo un mejor rendimiento. El balanceo a 300 % mejoró el rendimiento en dos casos (sin una diferencia significativa). El balanceo a 400 % mejoró el rendimiento en un caso (sin una diferencia significativa). El balanceo en 500 % mejoró el rendimiento en dos casos (una diferencia significativa). El balanceo a 1000 % mejoró el rendimiento en un caso (sin una diferencia significativa).

Estos resultados podrían ser un indicio de que el número de instancias influye en la clasificación una vez que los datos están balanceados. Se requieren más experimentos para confirmarlo.

5. Conclusiones

En este trabajo nos propusimos investigar si es posible mejorar los modelos predictivos para los subtipos de SGB creados en un estudio previo, mediante el uso de técnicas de balanceo de datos. Para balancear los datos utilizamos SMOTE, ya que es una técnica ampliamente utilizada con resultados probados. Este trabajo constituye el primer esfuerzo en el uso de técnicas de balanceo de datos para mejorar los modelos predictivos de SGB.

Para este primer estudio, seleccionamos tres clasificadores simples con diferentes enfoques: C4.5 (árbol de decisión, basada en instancias), Máquina de Soporte Vectorial (SVM del inglés *Support Vector Machine*, basado en kernel), JRip (basado en reglas). Comparamos su rendimiento en experimentos que consistían en balancear los datos al 100 %, 200 %, 300 %, 400 %, 500 %, 1000 % para 6 subproblemas binarios de subtipos de SGB (OvO).

El mejor resultado se obtuvo en el conjunto de datos SGB6 ya que los tres clasificadores crearon modelos predictivos utilizando datos balanceados que mejoraron significativamente los modelos con datos desbalanceados. Este resultado se logró al balancear los datos al 100 %.

En cuanto al rendimiento de los clasificadores, JRip fue el mejor, ya que se obtuvo modelos predictivos utilizando datos balanceados que mejoraron significativamente los modelos con datos desbalanceados. Esto mejoró los modelos en dos casos (SGB4 y SGB6). En el nivel de balanceo de datos, llegamos a la conclusión de que los mejores resultados se obtuvieron con un balance del 100 %, ya que en seis casos los modelos con datos balanceados superaron a los modelos con datos desbalanceados, tres de ellos con una diferencia significativa. Observamos que los casos con el mayor número de instancias obtuvieron el mejor rendimiento: SGB4 con 37 instancias para la clase minoritaria y 59 para la clase mayoritaria, y SGB6 con 13 instancias para la clase minoritaria y 59 para la clase mayoritaria.

Los resultados logrados en este estudio muestran que es factible construir un modelo predictivo para apoyar a los especialistas en la predicción automática que permita la distinción entre las diferentes variantes de esta enfermedad. La gravedad y el tratamiento del SGB varía entre los diferentes subtipos, por lo que la diferenciación entre ellos ayuda a la correcta identificación del subtipo que padece el paciente y de esta forma los especialistas pueden tomar las medidas necesarias para atenderlo adecuadamente.

Desde el punto de vista del aprendizaje automático, encontramos que SMOTE nos permitió crear algunos modelos con datos balanceados que superaron a los modelos con datos desbalanceados. Lo anterior representa una contribución en la exploración del rendimiento de las técnicas de balanceo de datos usando conjuntos de datos reales. Como trabajo futuro, experimentaremos con más variantes de la técnica SMOTE, y también con diferentes clasificadores simples y combinados para construir modelos predictivos más precisos.

Agradecimientos. Esta publicación fue financiada con recursos del Programa de Fortalecimiento de la Calidad Educativa (PFCE) 2019, bajo el proyecto

número P/PFCE-2019-27MSU0018V-11. Queremos agradecer al CONACYT por el apoyo al Doctorado en Ciencias de la Computación de la Universidad Juárez Autónoma de Tabasco.

Referencias

1. Canul-Reich, J., Frausto-Solís, J., Hernández-Torruco, J.: A predictive model for Guillain-Barré syndrome based on single learning algorithms. *Computational and Mathematical Methods in Medicine* 2017, 1–9 (2017)
2. Canul-Reich, J., Hernández-Torruco, J., Chávez-Bosquez, O., Hernández-Ocaña, B.: A predictive model for Guillain-Barré syndrome based on ensemble methods. *Computational Intelligence and Neuroscience* 2018, 1–10 (2018)
3. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research* 16, 321–357 (2002)
4. Córdón, I., García, S., Fernández, A., Herrera, F.: *imbalanced: Preprocessing Algorithms for Imbalanced Datasets* (2018), <https://CRAN.R-project.org/package=imbalanced>, r package version 1.0.0
5. Cuzick, J.: A wilcoxon-type test for trend. *Statistics in Medicine* 4(4), 543–547 (1985)
6. Fernandez, A., Garcia, S., Herrera, F., Chawla, N.V.: SMOTE for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary. *Journal of Artificial Intelligence Research* 61, 863–905 (2018)
7. Fernández, A., López, V., Galar, M., del Jesus, M.J., Herrera, F.: Analysing the classification of imbalanced data-sets with multiple classes: Binarization techniques and ad-hoc approaches. *Knowledge-Based Systems* 42, 97–110 (2013)
8. Gu, Q., Zhu, L., Cai, Z.: Evaluation measures of the classification performance of imbalanced data sets. In: *Communications in Computer and Information Science*, pp. 461–471. Springer Berlin Heidelberg (2009)
9. Hernández-Torruco, J., Canul-Reich, J., Frausto-Solís, J., Méndez-Castillo, J.J.: Feature selection for better identification of subtypes of Guillain-Barré Syndrome. *Computational and Mathematical Methods in Medicine* 2014, 1–9 (2014)
10. from Jed Wing, M.K.C., Weston, S., Williams, A., Keefer, C., Engelhardt, A., Cooper, T., Mayer, Z., Kenkel, B., the R Core Team, Benesty, M., Lescarbeau, R., Ziem, A., Scrucca, L., Tang, Y., Candan, C., Hunt, T.: *caret: Classification and Regression Training* (2018), <https://CRAN.R-project.org/package=caret>, R package version 6.0-81
11. Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., Leisch, F.: *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien* (2018), <https://CRAN.R-project.org/package=e1071>, R package version 1.7-0
12. Milborrow, S.: *rpart.plot: Plot 'rpart' Models: An Enhanced Version of 'plot.rpart'* (2018), <https://CRAN.R-project.org/package=rpart.plot>, r package version 3.0.5
13. Paul, B., Prasad, K., Singh, M., Tripathi, M., Bhatia, R., Padma, M.: Clinical predictors of mechanical ventilation in Guillain-Barré syndrome. *Neurology India* 60(2), 150 (2012)
14. Robin, X., Turck, N., Hainard, A., Tiberti, N., Lisacek, F., Sanchez, J.C., Muller, M.: *proc: an open-source package for r and s+ to analyze and compare roc curves*. *BMC Bioinformatics* 12, 77 (2011)

15. Sarkar, D.: *Lattice: Multivariate Data Visualization with R*. Springer, New York (2008), <http://lmdvr.r-forge.r-project.org>, ISBN 978-0-387-75968-5
16. Sundar, U., Abraham, E., Gharat, A., Yeolekar, M., Trivedi, T., Dwivedi, N.: Neuromuscular respiratory failure in Guillain-Barré syndrome: Evaluation of clinical and electrodiagnostic predictors. *Journal of Association of Physicians of India* 53(1), 764–768 (2005)
17. Therneau, T., Atkinson, B.: *rpart: Recursive Partitioning and Regression Trees* (2018), <https://CRAN.R-project.org/package=rpart>, r package version 4.1-13
18. Torgo, L.: *Datamining with R learning with case studies*. Chapman & Hall/CRC (2011)
19. Uncini, A., Ippoliti, L., Shahrizaila, N., Sekiguchi, Y., Kuwabara, S.: Optimizing the electrodiagnostic accuracy in Guillain-Barré syndrome subtypes: Criteria sets and sparse linear discriminant analysis. *Clinical Neurophysiology* 128(7), 1176–1183 (2017)
20. Urbanek, S.: *rJava: Low-Level R to Java Interface* (2018), <https://CRAN.R-project.org/package=rJava>, r package version 0.9-10
21. Wijdicks, E.F., Klein, C.J.: Guillain-Barré syndrome. *Mayo Clinic Proceedings* 92(3), 467–479 (2017)
22. Witten, I.H., Frank, E., Hall, M.A., Pañ, C.: *Data Mining, Practical Machine Learning Tools and Techniques*. Morgan Kaufmann (2017)

Videjuego educativo para ayudar a comprender los principios básicos de la programación y desarrollar la habilidad lógica en niños de educación básica

Ana Caren Ticante Hernández, Carlos Miguel Herrera Orduña, Pedro Arguijo, Roberto Ángel Meléndez Armenta, Antonio Hiram Vázquez López

Tecnológico Nacional de México / Instituto Tecnológico Superior de Misantla,
Departamento de Maestría en Sistemas Computacionales
Misantla, Veracruz, México

anacarenticante@gmail.com, herrera.orduna.carlosmiguel@gmail.com,
pedro_arguijo@excite.com, ramelendeza@itsm.edu.mx,
jahvazquezl@itsm.edu.mx

Resumen. Debido a la dificultad de aprender la lógica necesaria para la programación se diseña y desarrolla un videjuego educativo que tiene como objetivo ayudar a niños a introducirse en el mundo de la programación. El videjuego busca desarrollar la lógica de los infantes en etapa de primaria y la forma en que se realiza es dando al jugador el control de un personaje el cual se maneja mediante líneas de pseudocódigo que deben ser escritas en un campo de texto de manera correcta. El videjuego consta de 20 niveles y su dificultad es incremental. La metodología de desarrollo utilizada es Scrum, además, se utilizaron herramientas de software, como el motor gráfico Unity 3D, Asesprite, Tiled y Adobe Photoshop para el diseño y desarrollo del videjuego.

Palabras clave: videjuego, aprendizaje reforzado, Q-learning, educación, programación.

Educational Game to Help Understand Basic Topics of Programming and Develop Logical Skills in Children of Basic Education

Abstract. Due to the difficulty in learning the logic necessary for programming, an edu-game is designed and developed which the purpose to help children get started in the world of programming. The videogame seeks to develop the logic of kids in the basic education, the way it is done is giving to the player the control of a character which is handled by pseudocode lines that must be written in a text field so correct. The video game consists of 20 levels with an incremental difficulty. This was created based on the Agile Scrum methodology and software tools were used, such as the Unity 3D graphics engine, Asesprite, Tiled and Adobe Photoshop for the design and development of the videogame.

Keywords: videogame, reinforcement learning, Q-learning, education, programming

1. Introducción

Actualmente, la tecnología se ha convertido en la mano derecha de las personas, esta ayuda a eliminar obstáculos de comunicación, ha permitido avances importantes en medicina, contribuye en la efectividad de los procesos empresariales y permite implementar nuevas técnicas de enseñanza, entre muchas otras aplicaciones que mejoran la calidad de vida de la sociedad. Gran parte de estas tecnologías operan con base en código escrito en un lenguaje de programación que permite automatizar tareas tediosas y reducir el tiempo en el cual se ejecutan los procesos. La programación se encuentra en la vida cotidiana de la sociedad, por lo que actualmente, es común referirse a los jóvenes como “nativos digitales” que tienen la capacidad de diseñar, inventar y crear con la computadora [1]. Es por lo que la enseñanza de la programación es vital para el desarrollo tecnológico de la sociedad.

La psicología educativa establece que la edad apropiada para aprender a programar depende de la complejidad del código de programación, se han realizado varios estudios sobre cómo los estudiantes jóvenes utilizan la codificación de Entornos de Aprendizaje Inicial (ILE), y estos estudios proporcionan pistas sobre el tipo de programación que están realizando, hay evidencia de que algunos conceptos de programación pueden enseñarse a estudiantes de 5 a 7 años de edad mientras el lenguaje utilice conceptos accesibles para ese grupo de edad [2]. Además, debe convertirse en una actividad de aprendizaje, debe resultar divertido y motivacional, por lo que los videojuegos son una herramienta que puede aportar distintos beneficios a este proceso de aprendizaje.

Desde su aparición, los videojuegos son parte del desarrollo social y cultural de las personas y llegan a influir en su desarrollo físico, sensorial, mental y creativo. Dada su importancia, en la vida cotidiana, se han creado videojuegos educativos con el objetivo de generar estrategias y métodos de enseñanza alternativos a los tradicionales, animando a los infantes y jóvenes a obtener conocimiento con una herramienta atractiva y motivadora a través de retos y actividades que van presentando dichos videojuegos. En este artículo se describe el diseño y desarrollo de un videojuego capaz de ayudar a comprender los principios básicos de la programación para el desarrollo de la habilidad lógica en niños de educación básica.

2. Trabajos relacionados

Un videojuego es un juego digital interactivo, creado con fines de entretenimiento en general y se basa en la interacción con personas o personajes ficticios.

Los videojuegos pueden clasificarse de acuerdo a su género, existen juegos de acción donde su objetivo es básicamente eliminar enemigos, de tipo arcade, son juegos de plataformas, laberintos y aventuras, también existen los videojuegos deportivos que promueven la competitividad y valores propios de los deportes, los de estrategia que hacen uso de planificación de recursos o acciones para lograr un objetivo, los de simulación que se caracterizan por permitir al jugador experimentar e investigar el funcionamiento de máquinas, fenómenos, situaciones y asumir el mando. También existen los juegos serios que están diseñados con un propósito más formativo que de entretenimiento.

Los juegos serios tienen principalmente un propósito educativo, explícito y cuidadosamente planeado, este tipo de juegos ayudan a la comprensión y aprendizaje de nuevos conceptos y habilidades. La taxonomía de los juegos serios se rige por siete modalidades [3], juegos para la salud, publicitarios, para la formación, para la ciencia y la investigación, educativos y de producción.

Los juegos serios se caracterizan por hacer uso de simulaciones principalmente para el ramo empresarial, un ejemplo es Virtual Safety and Education Platform [4], que es un juego que tiene como finalidad capacitar operarios de servicios de emergencia, operarios de maquinaria, personal de hospitales, ejercito, etc., SimCityEdu, es otro videojuego basado en las simulaciones, en el que estudiante de primaria debe crear y gestionar una ciudad, trabajando conceptos de ciencia, tecnología, matemáticas y economía [5]. Los juegos serios pueden usar cualquier tecnología de juegos y estar desarrollados para cualquier plataforma, además, están dirigidos a todo público, principalmente a profesionales, consumidores y estudiantes.

En el ámbito educativo, los juegos son una herramienta útil para la obtención de conocimiento, donde a través de retos y actividades, el estudiante es el protagonista de su propio aprendizaje. Existen videojuegos educativos para aprender muchas cosas, los más básicos enseñan a leer, sumar, restar o hasta tocar instrumentos, aprender algún idioma, o sobre temas más específicos como la biología, tal es el caso de Proyecto Kokori, este videojuego pone a los jugadores al mando de una pequeña nave capaz de introducirse en una célula para estudiarla, además de identificar y solucionar los problemas que puedan ir encontrando [6].

Incluso existen videojuegos para el área médica, por ejemplo, Immune Attack, su objetivo es salvar la vida de un paciente que sufre una grave infección bacteriológica, para ello, los jugadores tratan de encontrar todos los focos de la infección y devolver la salud al paciente, además, los jugadores acceden y ponen en práctica conocimientos sobre los procesos biológicos que permiten detectar y combatir las infecciones [7], otra aplicación de los juegos educativos, es la enseñanza de la programación.

El aprendizaje de conceptos de programación no es como cualquier otro aprendizaje, no es un proceso definido y no se trata sólo de memorización, para este propósito los videojuegos educativos pueden ser una herramienta efectiva, donde el jugador puede obtener conocimientos de programación en un ambiente educativo-virtual e interactivo [8], donde pueden aumentar la motivación para el aprendizaje y mejorar la comprensión y adquisición de nuevos conocimientos.

Existen distintos juegos enfocados a enseñar lenguajes de programación específicos, por ejemplo, Code combat, este es una es una plataforma interactiva en línea, un juego con estilo de rol enfocado a niños de más de 9 años o adultos y está pensado para que los jugadores puedan aprender a programar lenguajes de programación específicos, por ejemplo, JavaScript acompañado del lenguaje de etiquetas HTML, en Code combat el jugador toma el papel del personaje que el elija, y mediante hechizos y acciones a través de comandos de programación el juego avanza con diferentes niveles, aumentando el nivel complejidad [9].

Otro videojuego es CodinGame, este juego funciona sobre el navegador, el cual le permite al jugador practicar sus habilidades de programación combinándolas con un videojuego de disparos y aventura. En este juego se derrotan enemigos, programando comandos en la pantalla que se asemeja a una consola [10].

3. Materiales y métodos

La investigación que se reporta es de carácter cualitativo y su principal objetivo es explorar el uso de ambientes virtuales de videojuegos para el desarrollo de la lógica y de las habilidades de programación.

3.1. Materiales

- Hardware: computadora con requerimientos mínimos, uso mínimo de 2Gb de memoria RAM y 4 núcleos de procesador 1.2 Giga Hertz.
- Escenarios: como escenario se definen los mapas del videojuego con los que interactúa el jugador.
- Participantes: Los participantes se eligieron con base en el público objetivo, por lo que se contactó a la escuela primaria urbana "Niños Héroes" ubicada en la colonia Espaldilla de Misantla, Veracruz. Los participantes fueron estudiantes de quinto y sexto de primaria, como se puede ver en la figura 1, participaron 4 niñas y 2 niños de 10 años, 7 niñas y 11 niños de 11 años, 3 niñas y 1 niño de 12 años, contabilizando un total de 28 participantes.

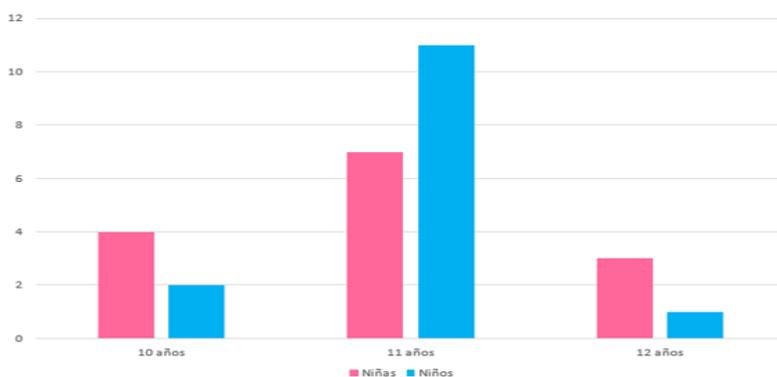


Fig. 1. Rango de edades de los participantes.

3.2. Herramientas de desarrollo

- Unity: Es un motor de videojuego multiplataforma creado por Unity Technologies. Está disponible como plataforma de desarrollo para Microsoft Windows, OS X, Linux. La plataforma tiene soporte de compilación con diferentes tipos de plataformas [11].
- TiledMap: Tiled es un editor de nivel de software libre. Admite la edición de mapas en varias proyecciones (ortogonal, isométrica, hexagonal) y también admite niveles de construcción con imágenes libremente posicionadas, giradas, escaladas o anotándolas con objetos de diversas formas [12].
- Adobe Photoshop: Adobe Photoshop es un editor de gráficos desarrollado por Adobe Systems Incorporated y utilizado principalmente para el retoque de fotografías y gráficos. Traducido al español significa “taller de fotos” y es el

líder mundial dentro del mercado de las aplicaciones de edición de imágenes en general [13].

3.3. Métodos

Metodología de desarrollo de software. Una metodología de desarrollo de software es una serie de pasos que se realizan de tal forma que permite a los desarrolladores elegir las técnicas para cada fase del proyecto y definir principalmente los participantes, sus actividades a desarrollar y una estructura de tiempo en el que deben realizarse. Seguir una metodología de desarrollo facilita la planificación, control, gestión de requisitos, operación, mantenimiento y validación de proyectos [14]. El conjunto de etapas y subetapas que sigue un proyecto, se denominan ciclo de vida del software, éste es fundamental para elegir la metodología de desarrollo a seguir ya que define las necesidades desde su concepción, los cambios o ajustes que puedan presentarse, el mantenimiento necesario, la realización de pruebas, hasta llegar a su finalización.



Fig. 2. Diagrama de la metodología Scrum para el desarrollo de videojuegos.

Se eligió utilizar la metodología de desarrollo ágil Scrum, porque demuestra un desarrollo que se adapta dinámicamente a las necesidades y requerimientos de la creación de un videojuego, además, se pueden realizar tareas simultáneamente, permite flexibilidad, mitigación de riesgos y adaptación a los cambios a los que se pueda someter en todo su ciclo de vida. Scrum permite trabajar en una serie de iteraciones en equipo.

El proceso de Scrum para este desarrollo se define en 3 fases secuenciales: pre-game, game y postgame, la fase de pre-game se realiza en una única iteración que comprende la definición del concepto del juego y la planificación de este, la fase game comprende a la implementación del videojuego y se trabaja de forma iterativa e incremental, la última fase tiene como objetivo evaluar y ajustar distintos aspectos del juego, eliminar errores detectados y, por último, entregar la versión final [15]. En la figura 3, se muestra un diagrama de la metodología aplicada en el videojuego.

Algoritmo de aprendizaje. El aprendizaje por refuerzo es un área del aprendizaje automático, el cual consiste en que un aprendiz recibe información de lo que es apropiado y con respecto a esto, realice una acción, en el caso de que la acción sea correcta, el aprendiz recibe algún tipo de valoración o premio, ante circunstancias erróneas, no se le notifica cual es el fallo, sino que, se informa que el comportamiento no fue apropiado y cuanto error se ha cometido. Este trabajo se centra en la técnica Q-learning, una forma de aprendizaje por refuerzo.

El algoritmo de Q-learning basado en el aprendizaje por refuerzo y utilizado en el aprendizaje automático, cuya ocupación es determinar qué acciones se deben escoger en un entorno dado, con el fin de maximizar alguna noción de "recompensa" o premio acumulado, esta recompensa representa la "calidad" de una acción tomada en un estado, en este caso, la recompensa representa la calidad de las soluciones introducidas por el jugador en el videojuego, esto con el objetivo de validar las soluciones. La introducción de esta técnica permite, evaluar la calidad de conocimiento obtenido y aportar una recompensa que pueda indicarle al usuario que ha llevado a cabo una acción correcta. En la figura 4, se muestra la representación del algoritmo de aprendizaje por refuerzo, implementado en el videojuego.

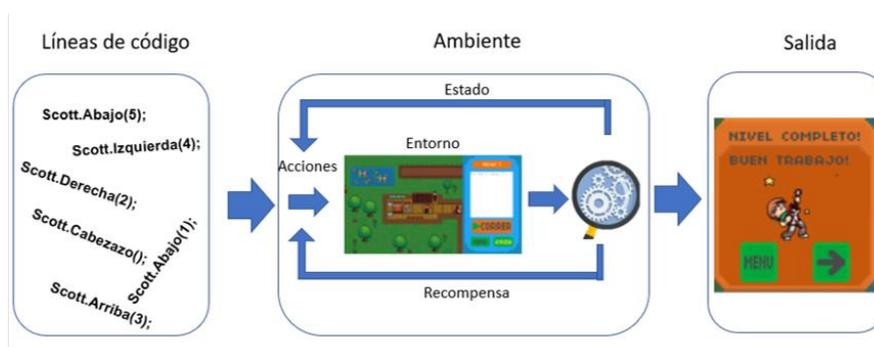


Fig. 3. Representación del algoritmo Q-learning en el videojuego.

4. Resultados

El resultado final consistió en el diseño y desarrollo de un videojuego, se dividió en 3 principales fases, pregame, game y posgame, de acuerdo con la metodología de desarrollo planteada.

4.1. Pre-game

La primera fase consistió en el planteamiento del concepto, la planificación y las especificaciones del videojuego.

Título. El título del videojuego es "Code Adventure".

Género. "Code Adventure" se considera un juego de estrategia ya que se caracteriza por manipular personajes para lograr objetivos haciendo uso de la inteligencia y la

planificación. También contiene, aunque a menor medida, características de los juegos de rol, ya que se interactúa con el personaje y se complementa con una historia. Por otro lado, el género puede considerarse como educativo, ya que está dirigido a aportar conocimientos y habilidades al jugador durante el transcurso del juego.

Arquitectura. El videojuego puede ser jugado desde cualquier sistema operativo sobre plataforma PC.

Mecánica. En esta fase también se definió la mecánica del videojuego, en la figura 5 se muestra una vista general de la pantalla de juego del nivel 1, misma que es similar para los demás niveles, únicamente cambiando el mapa, la posición y existencia de enemigos y obstáculos.



Fig. 4. Nivel 1 del videojuego.

Para completar el nivel el personaje principal debe pasar por cada uno de los obstáculos que se le presenten en el mapa, éstos pueden ser enemigos o rocas que deberá de destruir, también debe conseguir la guitarra de cada nivel, la función de la guitarra es orientar al jugador hacia el final del mapa y como un incentivo que motive al usuario a concluir todos los niveles del videojuego. En la figura 6 se muestran todas las guitarras diseñadas.

El jugador debe de introducir una serie de comandos que se traducirán en acciones que el personaje realizará. En la pantalla del juego existe un campo de texto donde el usuario debe escribir el código, que será verificado. Las líneas introducidas en el campo de texto permitidas son 5, hay cuatro acciones relacionadas con movimiento, al ejecutarse éstas permiten el desplazamiento del personaje hacia la izquierda, derecha, arriba o abajo, dependiendo de la acción que se trate, el jugador debe introducir el número de pasos que dará el personaje hacia la dirección indicada, el juego permite un número de pasos entre 1 y 20.

El personaje también puede realizar una acción de ataque, éste se realiza hacia donde se encuentre orientado el personaje, en la tabla 1, se muestra la sintaxis de las acciones posibles, donde “#” corresponde al número de pasos.

Tabla 1. Acciones posibles.

Acción	Sintaxis
Moverse a la derecha.	Scott.Derecha(#);
Moverse a la izquierda.	Scott.Izquierda(#);
Moverse hacia abajo.	Scott.Abajo(#);
Moverse hacia arriba.	Scott.Arriba(#);
Ataque	Scott.Cabezazo();



Fig. 5. Objetos recompensa.

Después de que el jugador escribe líneas de código en el panel de texto, debe de ejecutarlo, a continuación, el código se analiza para evaluar cada una de las líneas introducidas y encontrar posibles errores de sintaxis, si las líneas de código introducidas son correctas estas proceden a interpretarse, traduciéndose en acciones que realiza el personaje del juego.

Cámara. Para el caso de “Code Adventure” la plataforma de desarrollo es en 2D, lo recomendado para este tipo de juegos es una cámara fija, su posición, distancia focal y campo de visión no son cambiantes, por lo que, además, permite consumir menos recursos.

Periféricos. Ya que el videojuego es para PC, los periféricos fundamentales para jugar son teclado y ratón. El teclado permite capturar las instrucciones dadas por el jugador y el ratón permite desplazarse por las diferentes pantallas y niveles, así como poner en marcha las líneas de código escritas, haciendo clic en los botones que se encuentran en el videojuego.

Dificultad. Para determinar la dificultad de un videojuego se requiere considerar aspectos como el diseño de los escenarios, la historia, el modo y el ritmo de juego. Para el caso de “Code Adventure”, se determinó que la dificultad sea incremental, que los primeros niveles se resuelvan con pocas líneas de código y al avanzar se requieran de más líneas de código para resolver niveles más complejos. Por lo que la dificultad en este juego se compone por fácil, medio y difícil.

4.2. Game

En esta fase, se diseñaron los elementos visuales, así como las funcionalidades del videojuego.

Diseño. Se diseñaron los botones, íconos y demás elementos que componen las interfaces, como recursos que sirven para mostrar información al usuario para que logre identificar los diferentes niveles del videojuego, así como paneles contenedores de los botones, textos y demás elementos de la interfaz.



Fig. 6. Distribución de contenido.



Fig. 7. Mapa general de los niveles del videojuego.

Se diseñaron interfaces para la navegación dentro del videojuego, la primera corresponde al menú principal donde se pueden seleccionar las opciones de continuar, seleccionar un nivel en específico información sobre los desarrolladores y la opción de salir. Se diseñó una interfaz para que el usuario pueda seleccionar un nivel específico para jugar. También, se diseñó la interfaz que muestra la información de los desarrolladores y un botón para regresar al menú principal. Para la pantalla de juego principal se distribuyó el contenido como se puede observar en la figura 7, tomando como base una resolución de 1280x720 píxeles.

Desarrollo. En el proceso de desarrollo se añadió la funcionalidad al videojuego, desde poder navegar entre las interfaces del videojuego a través de los botones, se añadió un botón para mostrar información sobre la sintaxis del videojuego. Se añadió una función al botón de “Correr”, cuyo objetivo es, notificar que el usuario ha introducido líneas de

código, para que sean analizadas, también se desarrollaron todas las animaciones necesarias para el videojuego. Así, si el usuario introdujo las líneas de código correctas, se pueda verificar a través de movimientos realizados por el personaje al recorrer el mapa de acuerdo con las acciones introducidas.

Se diseñaron 20 niveles, los primeros doce se desarrollan dentro de la casa de Scott y los ocho restantes se ambientan en jardines y áreas verdes. Para el diseño de cada uno de estos niveles se usó la herramienta Tiled que permite construir mapas utilizando los tiles descargados previamente. Posteriormente, dichos mapas se exportaron a la plataforma de desarrollo del videojuego. En la figura 8 se muestra el mapa final que compone los 20 niveles.

4.3. Postgame

La última fase comprende la evaluación y verificación del software. Las pruebas de usabilidad son uno de los métodos más comunes para medir un videojuego. La usabilidad es un atributo de calidad que evalúa la calidad de experiencia del usuario e identifica la facilidad de uso de un sistema; en una prueba de usabilidad el videojuego se presenta a un grupo de usuarios objetivo, es decir, al público al que va dirigido principalmente el juego.

Lo que se busca es identificar posibles errores que los desarrolladores hayan pasado por alto, no es necesario presentar el producto final, basta con un prototipo o una versión avanzada del software, ya que se recomienda que dicha prueba se aplique al menos tres semanas antes de la entrega final [16].

La validación del videojuego se llevó a cabo mediante una prueba de usabilidad, la cual se desarrolló a partir de la implementación del prototipo del videojuego que incluye dos niveles de cada grado de dificultad dando un total de 6 niveles. Este prototipo fue aplicado a estudiantes de quinto y sexto de primaria, dichos participantes están descritos con más detalle en la sección 3.1. La manera en la que se llevó a cabo la prueba fue la siguiente, se dio una introducción a los estudiantes acerca de la importancia de su participación en esta actividad y de los objetivos del videojuego, además, se resolvieron todas las dudas existentes. Cabe mencionar que se contaba con escasas 3 PC y ya que el tiempo asignado para esta actividad fue limitado, debido a las diversas actividades programadas para ese día escolar, se optó por agrupar a los estudiantes en parejas de manera aleatoria y se dio la instrucción de que se alternarían para resolver los niveles por sí mismos, así, los estudiantes pudieron interactuar con cada grado de dificultad del videojuego en un lapso más corto.

Se diseñó una encuesta de 10 preguntas para medir la satisfacción de los estudiantes ante el videojuego, la cual fue aplicada después de que todos interactuaran con él. Para dicha encuesta se utilizaron preguntas con un lenguaje sencillo para que los estudiantes pudieran comprenderlas fácilmente, además, se incluyeron respuestas de opción múltiple por la misma razón y para que la información pudiera ser plasmada estadísticamente, las cuales pueden observarse en la tabla 2.

Para obtener datos de la encuesta, se utilizó una escala en la que a cada posible respuesta se le asigna un valor, por ejemplo: no=1, tal vez=2, si=3, dichos valores se encuentran asignados en la tabla 2. En algunas de las preguntas se pedía asignar una calificación del 1 al 5, porque lo que no fue necesario realizar una conversión.

Tabla 2. Prueba de usabilidad.

NUM	PREGUNTAS	POSIBLES RESPUESTAS				
1	¿Te gustó el juego? Califícalo del 1 al 5.	1	2	3	4	5
2	Del 1 al 5, ¿Qué tan fácil es de jugar?	1	2	3	4	5
3	¿Es divertido?	1. Poco	2. Regular	3. Mucho		
4	¿Te gustó el minijuego?	1 Poco	2.Regular	3.Mucho		
5	Del 1 al 5, ¿Qué tanto te agradó el personaje?	1	2	3	4	5
6	¿Volverías a jugarlo?	1 No	2.Tal vez	3. Si		
7	¿Lo recomendarías con tus amigos?	1. No	2. Tal vez	3. Si		
8	¿Aprendiste algo nuevo?	1. No	2. Tal vez	3. Si		
9	¿Encuentras interesante el mundo de la programación después de jugar este juego?	1. Poco	2. Regular	3. Mucho		
10	¿Calificación final del videojuego?	1	2	3	4	5

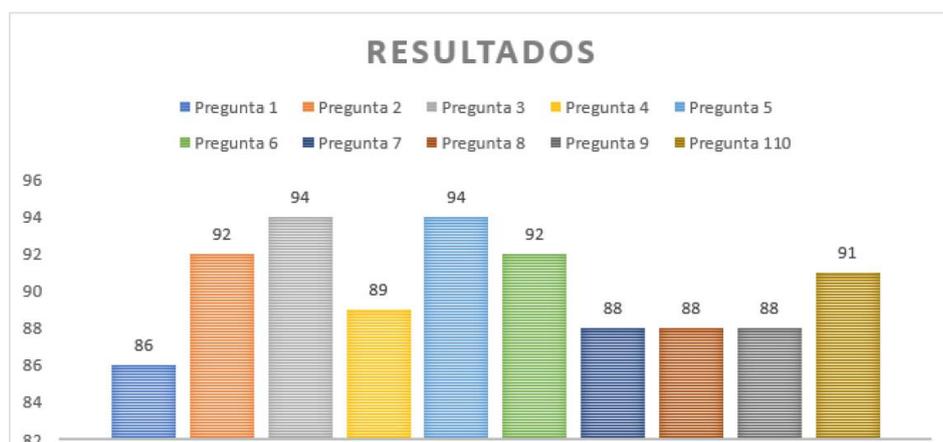


Fig. 8. Porcentajes obtenidos.

Una vez teniendo solamente valores ordinales, se obtuvieron los promedios de cada una de las preguntas, los cuales se muestran en la figura 9.

En conclusión, se desarrolló un videojuego 100% funcional, con un total de 20 niveles con distintos grados de dificultad, para la evaluación del software obtenido, se realizaron las pruebas pertinentes, se visitó a una escuela primaria de la ciudad de Misantla, Veracruz, donde 28 estudiantes de dicha institución educativa interactuaron con el software y posteriormente realizaron una encuesta de satisfacción.

Una vez realizadas las validaciones y cada una de las pruebas, se identifica que el juego satisface con las acciones verificadas anteriormente, brindando confiabilidad y funcionalidad en cada uno de los niveles del juego.

Se evidencia el correcto funcionamiento de los elementos del videojuego, dando un visto bueno en cada una de las pruebas realizadas y confirmando que el juego cumple con cada uno de los objetivos esperados.

5. Discusión

Para tener una perspectiva de “Code Adventure” en comparación a lo que ofrecen otros videojuegos educativos, se revisaron las diferentes propuestas de videojuegos educativos, principalmente, aquellos que tienen como objetivo ayudar a personas a aprender a programar, se analizaron factores como: plataforma, público objetivo, dificultad, tópicos de programación que tratan y sus costos. En la tabla 3 se muestra una comparación entre algunos de los principales videojuegos analizados.

Tabla 3. Comparación entre videojuegos para aprender a programar.

Videojuego	Plataforma	Lenguajes que enseña	Público objetivo	Costo
Code Monkey	PC WEB	CoffeeScript	de 10 a 12 años	29 dólares anuales
CodeCombat	PC WEB	Javascript Python.	de 10 a 17 años	39.99 dólares
Tynker	PC WEB Android	Python, HTML/CSS, JavaScript	de 7 a 9 años	\$20-\$180 dólares
CheckiO	PC WEB	Python JavaScript	de 17 a 18 años	24.99 dólares anuales
Code adventure	Pc Escritorio	Pseudocódigo	De 7 a 12 años	Gratis

Analizando la comparación presentada en la tabla 3, Code Adventure se diferencia en el tipo de plataforma, ya que, al ser una aplicación instalable offline no requiere de una conexión a internet, esto puede traer beneficios, ya que escuelas y hogares que no cuenten con este servicio pueden utilizar el videojuego.

En cuanto a los lenguajes que enseñan todos los videojuegos comparados se enfocan en enseñar lenguajes de programación específicos, Code Adventure tiene un enfoque

completamente distinto puesto que, se basa en enseñar la lógica computacional necesaria para comprender los principios básicos de la programación y así aplicar este método de razonamiento a cualquier lenguaje de programación. Por otro lado, los costos de los videojuegos analizados comprenden desde los 20 hasta los 180 dólares, costos que pueden no ser accesibles para algunos usuarios, por el contrario “Code Adventure” al ser una plataforma desarrollada en una institución educativa se realizó sin fines de lucro.

Referencias

1. Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., Kafai, Y.: Scratch: Programming for all. *Commun. Acm*, 52(11), 60–67 (2009)
2. Duncan, C., Bell, T., Tanimoto, S.: Should your 8-year-old learn coding? In: *Proceedings of the 9th Workshop in Primary and Secondary Computing Education*, pp. 60–69 (2014)
3. Gros Salvat, B.: Certezas e interrogantes acerca del uso de los videojuegos para el aprendizaje. *Revista Internacional de Comunicación Audiovisual, Publicidad y Literatura* 1(7), 251–264 (2009)
4. VStep, RescueSim-Virtual Emergency Response Training, www.rescuesim.com, Último acceso: 2018/12/03
5. Hao, J., Mislevy, R.J.: The Evidence Trace File: A Data Structure for Virtual Performance Assessments Informed by Data Analytics and Evidence-Centered Design (2018)
6. Mellado, M., Roa, J., Baez, M., Carpinelli, J., Garretón, V., Mercovich, E., Szwarcberg, M.: Kokori, set de herramientas TIC gratuito para la enseñanza y aprendizaje en biología celular (1), 1–16 (2018)
7. Stegman, M.: Immune Attack players perform better on a test of cellular immunology and self-confidence than their classmates who play a control video game. *Faraday discussions* 169, pp. 403–423 (2014)
8. Balerdi, F.E.: Videojuegos y educación. *Comunicar. Revista científica iberoamericana de comunicación y educación* 10, pp. 171–180 (1998)
9. CodeCombat, Acerca de, <https://codecombat.com/about>, Último acceso: 2019/03/08
10. CodinGame, Acerca de, <https://www.codingame.com/about/team>, Último acceso: 2019/03/02
11. Unity 3D, Documentation, <https://docs.unity3d.com/es/current/Manual/> Último acceso: 2019/03/08
12. MapEditor, About Tiled, <https://doc.mapeditor.org/en/stable/manual/introduction/>, Último acceso: 2019/03/02
13. Adobe, Introducción a PhotoShop <https://helpx.adobe.com/mx/photoshop/get-started.html>, Último acceso: 2019/03/02
14. Nader, J.R.: Metodología de Desarrollo de Software: MBM (Metodología Basada en Modelos). *Ingeniare* 16, pp.113–127 (2014)
15. Musil, J., Schweda, A., Winkler, D., Biffel, S.: Improving video game development: Facilitating heterogeneous team collaboration through flexible software (2010)
16. D. E. de Videojuegos. Libro Blanco del Desarrollo Español de Videojuegos 2016. Desarrollo Español de Videojuegos (2016)

Mapas conceptuales colaborativos: un modelo basado en grafos distribuidos

Jorge A. Hernández-Perales, Manuel Hernández-Gutiérrez

Universidad Tecnológica de la Mixteca, Huajuapán de León, Oaxaca
jahdezp@mixteco.utm.mx, manuelhg@mixteco.utm.mx

Resumen. Conforme el tiempo transcurre, el acervo de conocimiento humano se hace más y más grande, y con el advenimiento de medios alternativos de comunicación, este conocimiento tiende a fragmentarse y desorganizarse. Los mapas conceptuales basados en grafos permiten que una persona exprese de una forma efectiva, ágil y amena sus conceptos. La tecnología actual también permite que estos mapas puedan compartirse colaborativamente, así logrando crear mapas grupales en los que intervienen los conceptos de varias personas. En este trabajo presentamos un modelo epistemológico para la creación y el manejo de mapas conceptuales distribuidos. Este modelo tiene el propósito de enmarcar en una teoría de grafos con anotaciones temporales la forma en que los humanos adquieren, organizan y clasifican cierto tipo de conocimiento.

Palabras clave: aprendizaje a distancia, grafos, conocimiento, colaboración.

Collaborative Conceptual Maps Based on Distributed Graphs

Abstract. As time goes by, human knowledge is getting bigger and bigger, and with the advent of alternative means of communication, this knowledge tends to fragment and get disorganized. Conceptual maps based on graphs allow a person expresses in an effective, agile and entertaining way his/her concepts. Current technology also allows these maps to be shared collaboratively, thus achieving the creation of collaborative maps in which the concepts of several people intervene. In this paper we present an epistemological model for the creation and management of conceptual and distributed maps. This model has the purpose of framing in a theory of graphs with temporal annotations the way in which humans acquire, organize and classify a certain type of knowledge.

Keywords: learning, graphs, knowledge representation, collaborative.

1 Introducción

La adquisición y la organización del conocimiento humano es una de las actividades principales en el aprendizaje y la enseñanza; también es clave en la

información compartida y forma una de las bases principales del procesamiento de la información en la inteligencia artificial [9]. Tal manejo del conocimiento humano es útil en aplicaciones modernas tales como la *web semántica* [1], así como en clasificaciones generadas por usuarios que escriben anotaciones (por ejemplo, a imágenes) para después formar redes semánticas. Los mapas conceptuales son objetos gráficos que muestran componentes básicos de información relacionados entre sí. Tales componentes de información entrelazan y condensan *conceptos*. Una de las formulaciones gráficas de mapas conceptuales es por medio de visualizaciones inspiradas en la teoría de grafos [5], en donde cada nodo de un grafo representa un concepto (expresado mediante una palabra), y en donde cada lado enlaza un nodo con otro si tal par de nodos satisfacen una relación binaria. En este escrito mostraremos que la creación distribuida de mapas conceptuales tiene implicaciones positivas para el manejo estructurado de información, dentro de un modelo epistemológico básico; además, se analizarán y propondrán algunas herramientas para la creación computacional y la construcción distribuida de este tipo de mapas.

Este trabajo está organizado como sigue. En la sección 2 se plantea el concepto de *epistemología* y el de *ontología*, y una formulación de esta última como mapa conceptual. En la sección 3 plantearemos una relación que existe entre un mapa conceptual y un grafo. En la sección 4 se plantearán un conjunto posible de estrategias para la construcción de mapas conceptuales, intentando resolver el problema de *condición de carrera* (*race condition*) en la creación de grafos colaborativos. En la sección 5 se da una implementación de mapas conceptuales como grafos en Prolog, y en la sección 6, Conclusiones, se describen alcances y limitantes de la modelación; también se finaliza con algunas notas de resumen, se plantean algunas aplicaciones en la educación, y se dan notas de trabajo a futuro.

2 Ontologías y mapas conceptuales

Consideraremos en este escrito un cierto concepto de *epistemología* (definida tradicionalmente como un parte de la filosofía que estudia los principios, extensiones y métodos del conocimiento humano), en una vertiente pragmática parecida a como se menciona “ontología” en los medios computacionales.

Así pues, para nuestros propósitos, definimos un *modelo epistemológico relacional* como un conjunto I_t de información tomada de un universo U con otro conjunto de relaciones R_t (de un universo de relaciones R) definidas binariamente en I_t , y tal que t está dado como un parámetro de *tiempo*. La forma en que I_t cambia es mediante las operaciones: al eliminar de U uno de sus elementos (eli) o al agregar (ag) a U algún otro elemento, en tanto que R_t cambia por eliminar una relación (eliR), agregar una nueva (agR) o bien por modificar una relación existente (modR). Este modelo fundamenta adecuadamente un caso particular, a tratar a continuación, de *ontología*.

La ontología es una definición formal de tipos, propiedades y relaciones entre entidades que real o fundamentalmente existen para un dominio de discurso en

particular; en este artículo utilizamos este concepto de ontología para el manejo de conocimiento disperso. Comenzaremos investigando el impacto de la creación de mapas conceptuales en la síntesis de ontologías temáticas. Afirmamos por ahora que los mapas conceptuales son una aproximación básica al concepto de ontología: un mapa conceptual representará, en principio, los conceptos básicos que, a su vez, son enmarcados en otro concepto mayor (quizás metaconcepto), permitiendo con esto ya sea a nivel pedagógico o computacional explorar un *espacio de búsqueda conceptual*; un mapa conceptual a la vez sirve como una *guía* tanto a inteligencias artificiales como a la mente humana para decidir entre diversas opciones de exploración dentro del entramado del conocimiento humano; por la parte computacional, un mapa conceptual se puede ver como una forma constructiva de considerar ontologías [8]; por la parte humana, su interpretación da algunas pistas tanto del conocimiento humano como de su posible uso en tareas de razonamiento ([2], [10]). Es necesario enfatizar que los mapas conceptuales *distribuidos* tienen un alto valor informativo *social*, pues como se menciona en [6] (p.12-3, 2.1.1.1 *Jean Piaget's Perspective of Constructivism*), la interacción de cada individuo con su ambiente hace que sus modelos cognitivos sean únicos, y compartirlos hará que los demás participantes incrementen sus propias ideas así como, a su vez, el acervo cognitivo grupal. Nuestra suposición básica, en resumen, es que, dado un modelo epistemológico relacional como el previamente definido para un individuo A , las ontologías permiten un manejo y utilización de su acervo epistemológico de tal manera que este acervo se sustenta en consideraciones de conceptos atómicos (palabras) y relaciones binarias entre estos conceptos.

Por razones de eficiencia, los métodos automatizados tales como los de [8] son altamente deseables, aunque si bien su misma propuesta mecanicista indica que la información se auto-contiene y no es posible “incrementar” las posibilidades de mayores conceptos ontológicos, y si fuera posible, la forma desorganizada que adoptarían (por razón del crecimiento exponencial) las haría inútiles para consumo humano. Una mayor estructuración tal como la mencionada en [8], por ser jerárquica, es una buena forma de organizar el conocimiento [10], pero puede considerarse como un caso particular del modelo epistemológico y distribuido propuesto aquí. En suma, cuando la intervención humana es posible, algunos sistemas automatizados mejoran su funcionamiento (como un ejemplo, la adición de *etiquetas (tags)* por medios sociales a imágenes brinda un *contexto* para la apropiada clasificación de estas [4]), tal como acontece en redes sociales de orientación gráfica tales como *Instagram*® o *Pinterest*®.

3 Mapas conceptuales como grafos

En [10] hay una discusión amplia de los mapas conceptuales, sus bases teóricas y algunas de sus aplicaciones; de forma explícita, los grafos trabajados ahí tienen etiquetas en sus lados. Para simplificación de tratamiento, en este artículo omitiremos las etiquetas en los lados.

Modelados mediante grafos, los mapas conceptuales brindan una información considerable en formato bidimensional acerca de un tópico. Aunque tal formato

es bidimensional por restricciones físicas, es también posible contar con una *historia* de la creación y desarrollo de un mapa conceptual, así ampliando en una dimensión, la del tiempo, su formulación.

Primeramente, se notará que, por un lado, el número de nodos indicará una *densidad informativa* del mapa conceptual; por otro lado, el número de lados que unen estos nodos dará una medida de la *densidad relacional* de este mapa. En un grafo de 5 nodos con todos los nodos conectados entre sí, con el grafo completo llamado K_5 se tendrá una densidad conceptual moderada, pero una densidad relacional alta, pues por un teorema combinatorio, éste será el mayor número de relaciones establecidas para un grafo completo arbitrario no dirigido $K_n: n * (n - 1)/2$.

La elección de un grafo tiene también un acierto computacional: La extracción de la información subyacente es bidireccionalmente sencilla. En efecto, la recolección de información textual y las relaciones es una primitiva en XPCE¹, por ejemplo, para generar de manera inmediata un conjunto de *hechos* como programa lógico [11]. De hecho, por medio de *diccionarios* o *listas asociativas* (evitando autoreferencias [*loops*]) de nodos se tendrá listo un archivo de texto para su respectiva versión gráfica.

Diremos que un grafo es un *grafo conceptual* si tal grafo se enmarca en una historia de mapas conceptuales, cada uno representado por un grafo por el mapeo mencionado previamente. La característica novedosa que es digna de notar en los *grafos conceptuales* es que estos son objetos gráficos *dinámicos*, y al agregar la dimensión del tiempo afloran algunos resultados que veremos a continuación. Los grafos conceptuales requieren estar ordenados cronológicamente, y tener por cada grafo convencional anexo una etiqueta de fecha (que puede ser física o de causalidad), así enlazando estos grafos con el modelo epistemológico propuesto. Notemos que un grafo visto como mapa conceptual puede originar varios otros grafos; el grafo conceptual que trabajamos aquí solo toma una posible trayectoria de entre todas las posibles ramificaciones: esta situación resultó inevitable en las consideraciones de modelación basadas en temporalidad.

3.1 Mapas conceptuales como objetos dinámicos

Supongamos que cada grafo conceptual se representa con $G_n(t)$ para un usuario dado con identidad n en un tiempo t dado. En cada caso, cada $G_n(t)$ tendrá asociados sus dos conjuntos principales: Su conjunto de nodos $G_n(t)[nodos]$ y su conjunto de lados $G_n(t)[lados]$. La parte pragmática abordada aquí es que los nodos $G_n(t)$ representan cierta información (codificada mediante palabras) I_t , y similarmente, los lados $G_n(t)[lados]$ representan algunas relaciones R_t que se mantienen entre los nodos de I_t . Notemos que desde tanto un punto de vista local (por un usuario) como global (por un tablero común, en el caso de un sistema distribuido), el total de nodos se incrementará por cada nuevo usuario que se incorpore, y en posible mayor número, el total de conexiones que se puedan establecer entre estos nodos. De esta manera, algunos de los resultados de teoría

¹ Paquete gráfico incluido en Swi-Prolog, www.swi-prolog.org

de grafos solo son aplicables a grafos convencionales que se “congelan” en un tiempo t_o (grafos estáticos), pero no a grafos que cambian con el tiempo.

3.2 Operaciones directas sobre mapas conceptuales

En esta sección se enfatiza qué operaciones serían posibles sobre grafos que modelan mapas conceptuales, de tal forma que se aporte un estándar de presentación, así como algunas *reglas detransformación* que incrementen la confianza de un usuario en aportar nuevos conceptos. Describimos algunas operaciones que se pueden obtener mediante la combinación de grafos en el tiempo, teniendo en mente las posibles aplicaciones a mapas conceptuales (ver Fig. 1).

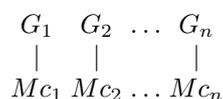


Fig. 1. Grafos (G) relacionados con mapas conceptuales (Mc), donde cada subíndice indica un tiempo i de construcción.

Unión de mapas conceptuales. Ahora, consideremos la unión de dos grafos en la formulación siguiente: La unión de dos grafos A y B , $A \cup B$ es un nuevo grafo C tal que C tiene como conjunto de nodos el conjunto de nodos de A en unión el conjunto de nodos de B , $C[nodos] = A[nodos] \cup B[nodos]$, y de tal forma que el conjunto de lados de C es igual a la unión del conjunto de lados de A en unión con el conjunto de lados de B : $C[lados] = A[lados] \cup B[lados]$. Esta definición tiene una propiedad epistemológica interesante desde el punto de vista de cada usuario, pues resulta que (bajo el impedimento de borrar nodos o lados ya establecidos) el conocimiento adquirido es siempre *monótono* (no en un sentido de uniformidad, si no en el sentido de que la información que puede veraz y positivamente concluirse ahora puede concluirse posteriormente, cuando la información se incrementa en el mapa; para casos no monótonos, información adicional puede invalidar conclusiones previas) y *consistente*. También, conforme t se incrementa y bajo el impedimento del borrado de nodos, es posible considerar que la complejidad del grafo (medida como el número de nodos por el número de lados, que establece una medida adecuada para nuestros propósitos) se incrementa a su vez.

La operación unión y su impacto en los usuarios. Cabe mencionar que al realizarse las operaciones como conjuntos en la unión de grafos, los nodos en común se fusionan automáticamente, así como los lados, y por que además para conjuntos arbitrarios $A \cap B \subseteq A \cup B$. Para etapas posteriores a que se haya diseminado distribuidamente el estado actual de los usuarios (bajo una política de rondas, por ejemplo), cada usuario se concentrará probablemente en una sección del

grafo total. Esto lleva a un interesante planteamiento de un *índice de atención* a una sección de un mapa conceptual: Hemos podido comprobar con algunos voluntarios (incluidos los autores) un usuario a lo más ve (con lectura dispersa) 10 nodos simultáneamente, y que toma una decisión conciente de enfocarse en algunos de los 10 elementos para así plantear la creación de nuevos nodos (unos 3 bajo una circunstancia usual de comienzo, ya que generalmente los conceptos también ser irán gastando y terminando, por lo que es necesario considerar que conforme pasa el tiempo habrá mayores retardos para colocar nuevos conceptos), aunque para dar un enfoque medible sería mejor utilizar técnicas de rastreo de la vista (*eye tracking*) [3], procediendo por medir *periodos de vista fija* (*fixations*) y *frecuencia de vistazos* (*saccades*), particularmente por que aquí la *contigüidad espacial* (también mencionada en [3]) entre texto e imagen es difusa, así como el respectivo efecto de desconcentración (*split attention*). Otros aspectos psicológicos son explorados en [10].

Así que el crecimiento del tiempo $f(n)$ de respuesta (cuando n (instante) crece) está asintóticamente en $\Omega(2^n)$ (haciendo que exista una constante c , $c > 0$, con $f(n) \geq c * 2^n$ a partir de un número natural dado) y empíricamente nos dice que después de unas cuantas rondas lo mejor que un moderador puede hacer es cerrar los permisos para agregar nuevos nodos, a menos que se abran algunos de estos nodos para continuar agregando elementos (*zooming*). De todas formas, por la misma base de promover la lluvia de ideas (*brainstorming*), sesiones cortas de unos minutos parecen la mejor decisión.

Plegado de nodos. Otra operación interesante es la de doblar o plegar (*folding*) algunos nodos para esperar que posteriormente se abran a solicitud de un usuario. Nodos que tienen esta característica pueden tener un indicador de apertura; la idea es que un gran diagrama conceptual probablemente no será requerido en ciertas etapas y puede obviarse su contenido.

Transitividad. Otra operación más es la cerradura transitiva (*transitive closure*). La operación conlleva la posibilidad de ocultar (no borrar) nodos si tenemos que si n_1 , n_2 y n_3 son nodos, con una nueva relación R sirviendo de conexión entre ellos, con $(n_i, n_j) \in R$ si tal conexión es estable, escrita como $n_i R n_j$, se tiene que $n_1 R n_2$ y $n_2 R n_3$ podríamos implicar $n_1 R n_3$, asumiendo que R es una relación transitiva (un resultado que es alguna vez sorprendente, pero el hecho es que es una conclusión correcta aunque con una pérdida sustancial de información; sería posible, no obstante, agregar un “etiquetado histórico” a cada nodo para evitar tal pérdida de información).

Temporalidad. La última operación que mencionamos es la de incorporar un *rastreo temporal*, un “estado de edad” de un nodo, donde se asocia secuencialmente una nueva edad a cada nodo (con un contador que se incrementa en cada ocasión). Esta información puede no ser relevante a nivel grupal (a menos que se utilicen relojes lógicos o vectoriales, que son necesarios para propagar en un sistema distribuido la información básica de causalidad [7, Capítulo 3]), pero sí sería necesario que fueran adjuntos a la identidad de un usuario, pues

la información entonces diseminada sería más profusa y de alguna utilidad (tal como saber qué usuarios participan más activamente, digamos).

3.3 Trayectorias, subgrafos y árboles generadores

Explotando un poco más la representación de grafos, es conocido que una trayectoria permite enlazar dos conceptos (nodos) distantes. En un contexto epistemológico la trayectoria es también una fuente importante de estructuración de conocimiento: nos indica una respuesta de cómo dos conceptos se enlazan. En sí, estas trayectorias son también interpretables como subgrafos. Los subgrafos, por otro lado, siendo de índole general, no siempre serán interpretables como conocimiento estructurado. Un ejemplo positivo de subgrafo es la obtención de un concepto y todos sus vecinos adyacentes con algún radio de adyacencia. En el caso de radio 1, estamos caracterizando nodos con forma de estrella: el nodo con sus vecinos inmediatos, que en términos de conocimiento humano parece algo particularmente valioso.

Supongamos, como otro ejemplo, que tenemos un grafo con ciclos como diagrama conceptual. Eso quiere decir que las conexiones son profundas, así que se puede pensar un algoritmo de árbol generador, para de esta manera obtener un nuevo diagrama que mantiene solo las relaciones “esenciales”. La pérdida de información subyacente, no obstante, puede no siempre valer la pena en tal posible “optimización”, pues la interpretación humana puede valorarse más que meramente la orientación de la información a la máquina. Además, los mismos ciclos pueden detectar casos de razonamiento de *petición de principio* o de *términos primitivos o indefinibles*. Los lados de todas formas pueden “ponderarse” (subjetivamente) por parte de los usuarios, en una escala apropiada (sugiriendo 1 a 5, de menor a mayor apreciada “fuerza” de interrelación); bajo este esquema, el árbol generamente maximal dice qué conceptos se asocian “fuertemente” entre sí.

3.4 Radios y subgrafos radiales

Como ya se ha mencionado, el concepto de *radio* en un grafo es también importante en términos de estructuración conceptual. Partiendo del radio 1, los elementos adyacentes están *directamente* relacionados con el nodo central. Cuando el radio se incrementa, se empieza a perder este tipo de interrelación. Una propuesta básica al respecto es pensar que los mismos radios son una medida objetiva de *cercanía conceptual*, por lo que empíricamente se puede concluir la siguiente ley de radios al infinito: Cuando el radio n desde un (nodo) punto central n_1 se incrementa, la relación conceptual entre el nodo n_1 y aquellos n_j ubicados a una distancia n se va perdiendo.

Los *subgrafos* tienen una interpretación particular en el caso de mapas conceptuales, ya que al no incorporar la información total inicial, pueden originar nuevas formulaciones a partir de su obtención, conduciendo con ello a “ramificaciones” conceptuales (sirviendo como un aliciente de creatividad). Es de notoria relevancia la construcción de subgrafos en su formulación de *árboles*, dada una visión jerárquica y de lo general-a-particular en la organización de ideas [10].

3.5 Desventajas de una representación de grafos

Algunas desventajas de los mapas conceptuales en la representación mostrada de grafos son heredadas de la simplicidad de los grafos mismos. En los grafos convencionales, la unión mediante un lado de un par de nodos no categoriza en absoluto ni a los nodos ni a los lados: todos los lados tienen la misma jerarquía, e igualmente en los nodos. En el caso de *lados ponderados*, los valores otorgados pueden ciertamente guiar algún tipo de jerarquía ya sea implícita o explícita, pero aún así los nodos siguen teniendo el problema de carecer de clasificación o de jerarquía.

Otra desventaja detectada es que los grafos rápidamente crecen en un tratamiento individualizado y todavía más si son creados por varios usuarios (en orden exponencial). El significado de esto es que en pocas rondas los usuarios son fácilmente abrumados por la acumulación de información, detractando así la idea de que los mapas conceptuales deberían ser utilizados para organizar información y hacerla humanamente asimilable.

4 Estrategias de formación de mapas conceptuales colaborativos

Tres son las estrategias colaborativas para diseñar mc's, en donde cada estrategia propuesta es: la estrategia casual, la estrategia de concurrencia planificada, y la estrategia asíncrona.

Estrategia casual. La *estrategia casual* procede con la confección independiente de un grafo para posterior fusión, así generando un nuevo grafo que se distribuye para nuevo análisis y diseño. No se permite el borrado de conceptos en cada iteración, siendo preferible comenzar otra sesión si alguno de los colaboradores está en franco desacuerdo con el logro parcial alcanzado. El proceso continúa con los usuarios hasta que un usuario simplemente decide quedarse con su parte tal como en una iteración fue obtenida.

Estrategia de concurrencia planificada. La siguiente estrategia es con concurrencia planificada: Al agregarse un concepto *A*, se distribuye entre los participantes. En caso de que se agregue otro concepto se verifica si el concepto es igual al de alguien más que lo haya planteado, o bien se agrega tal cual. Notemos que no es necesario borrar nodo alguno. Una heurística sencilla señalará qué tan vívido es un concepto: por el grado total ya sea de incidencia o de extroversión (exterioridad) o ambos. No habrá nodos aislados (solo al inicio hay uno), pero los nodos que tengan un grado 1 durante un buen lapso de tiempo serán conceptos que brindaron poca o nula aportación al trabajo grupal, presumiblemente.

La posible síntesis computacional está también basada en el grado de un nodo: Por medio de una simple exploración sobre cada nodo, incluyendo forzosamente el nodo inicial, se tendrá un conjunto de conceptos que abarcarán tanta importancia como el grado que se haya solicitado. Por ejemplo, si los grados asociados a un concepto son como siguen:

Concepto inicial: auto
{auto:1, llantas:2, volante:2: espejos:3, interior: 1}

se seguirá que las personas tienen un fuerte conocimiento consensado de integrar auto con espejos.

Estrategia sincronizada. La tercera estrategia se formula como un juego de varias personas por turnos, en donde cada persona en cada turno debe o agregar un concepto o enlazar ese concepto con algunos de los existentes. Señalado un número de rondas (distribuidas) el juego terminará con un grafo con riqueza conceptual alta (número de nodos) o riqueza relacional alta (número de lados) o bien con conocimiento profuso (suma de ambas riquezas). El problema con este enfoque, como ocurre frecuentemente en aplicaciones sincronizadas, es que el juego puede bloquearse si alguien decide abandonar el juego o existe una desconexión del grupo.

Una forma de aumentar pesos a los lados es por “palomear” una conexión, así que comenzando con peso de lado 1, cada vez que se confirmara la validez de una conexión el peso aumentaría en uno. Esta estrategia de creación de grafos con pesos brinda una considerable información de la “fuerza” de conexión entre conceptos.

El concepto de arbitraje no se requiere como tal en la creación y desarrollo de mapas conceptuales; sin embargo, sería deseable programar un “monitor” o “moderador” que permita anotar algunas informaciones relevantes con respecto a los usuarios, tal como el tiempo dedicado a desarrollar un mapa conceptual o bien sus contribuciones medidas directamente por los nodos agregados; también, el monitor puede decidir cuando dos nodos son iguales vía la discrepancia de una palabra con otra mediante la distancia de edición por ejemplo (o mediante corrección ortográfica directa) así como por posibles sinónimos con un *agente monitor* guiado semánticamente.

5 Implementación local en Prolog

Se ha realizado para experimentación local, y apoyándose en algunas herramientas gráficas provistas por XPCE de Swi-Prolog, un ambiente para el tratamiento epistemológico vía mapas conceptuales mediante grafos (ver Fig. 2). Se tienen planeadas extensiones computacionales tanto de índole experimental mediante Erlang o bien práctica, mediante la utilización de una plataforma educativa a distancia llamada Chamilo².

La parte siguiente es un extracto de la codificación realizada hasta ahora en Prolog:

```
graph_editor :-  
    new(GE, graph_editor),  
    send(GE, open).  
:- pce_begin_class(graph_editor, frame).
```

² <https://campus.chamilo.org/>

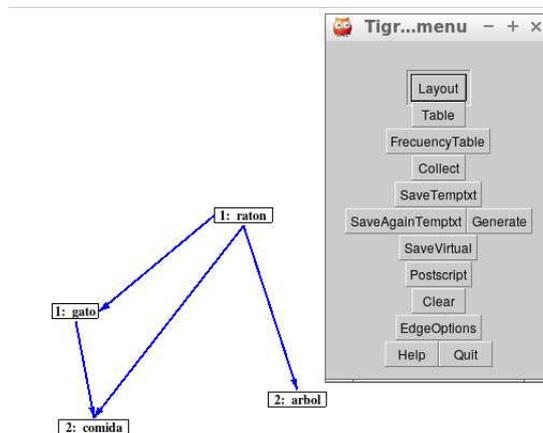


Fig. 2. Una implementación con anotaciones en nodos de un ambiente gráfico para grafos como mapas conceptuales.

```
initialise(GE) :->  
    "Create graph editor"::  
    send(GE,send_super,initialise,'The Conceptual Map Editor.'),  
    send(GE,append,  
    new(E, picture('Graph Editor',size(800,600))), {...})
```

6 Conclusiones

Como limitantes detectadas del esquema propuesto se tiene: No hay manejo de información negativa, pues poner información negativa rápidamente saturaría los grafos con lados. Solo información positiva es posible, del tipo “A se relaciona con B”, advirtiendo a los usuarios que afirmaciones del tipo “A no se relaciona con B” deberían ser excluidas. Otra limitante es aquella de permitir solo una relación genérica, pero sin duda diversos tipos de relaciones serían epistemológicamente más interesantes y no nos limitaría a solo estructuración simple de ideas. También se hace necesaria de cuando en cuando una “depuración” y “vaciado” de información, ya sea implícita o explícita (para guardar información, el programa propuesto sí puede salvar en una base de datos textual la información introducida); también hemos notado que debería haber algunas técnicas para aminorar la complejidad computacional en el crecimiento de la información y las relaciones.

Algunas aplicaciones posibles a la educación serían: Como moldes para exámenes educativos: Un mapa conceptual parcialmente llenado por completar, por ejemplo; o bien como bases de datos para rescatar conceptos importantes. En el mismo tenor, se puede solicitar la descripción de un concepto y que se de la información y después se relacione con el resto de los componentes. Otra

aplicación sería la formación de plantillas nodales sin enlaces, para ver qué tanto puede recuperar el usuario. Se puede utilizar la noción de subgrafo para reforzar la comprensión en un tema, solicitando al estudiante la escritura de ensayos mediante los conceptos interrelacionados dados. Se tiene planeado una versión distribuida en dos posibles vertientes: utilizando Erlang para experimentación distribuida, o utilizando el sistema Chamilo (plataforma educativa distribuida de fuente libre) para incrustarle alguna componente que incorpore mapas conceptuales como grafos.

Agradecimientos. Los autores desean expresar un agradecimiento especial a las autoridades administrativas y académicas de la Universidad Tecnológica de la Mixteca, por su siempre atenta y pronta respuesta a ayudarnos a realizar nuestra investigación.

Referencias

1. Davies, J., Fensel, D., van Harmelen, F. (eds.): *Towards the semantic web: Ontology-driven knowledge management*. John Wiley & Sons (2003)
2. Glasgow, J., Narayanan, N.H., Chandrasekaran, B. (eds.): *Diagrammatic Reasoning: Cognitive and Computational Perspectives*. MIT Press, Cambridge, MA, USA (1995)
3. van Gog, T., Jarodzka, H.: Eye tracking as a tool to study and enhance cognitive and metacognitive processes in computer-based learning environments. In: *International Handbook of Metacognition and Learning Technologies*. Springer (2013)
4. Johnson, J., Ballan, L., Li, F.: Love thy neighbors: Image annotation by exploiting image metadata. CoRR abs/1508.07647 (2015), <http://arxiv.org/abs/1508.07647>
5. Jungnickel, D.: *Graphs, networks and algorithms*. Springer (2008)
6. Konert, J.: *Interactive Multimedia Learning*. Springer Theses, Springer (2015)
7. Kshemkalyani, A.D., Singhal, M.: *Distributed Computing Principles, Algorithms, and Systems*. Cambridge University Press (2008)
8. Makagonov, P., Figueroa, A.R., Gelbukh, A.: Natural Language Processing and Information Systems: 11th International Conference on Applications of Natural Language to Information Systems, NLDB 2006, Klagenfurt, Austria, May 31 - June 2, 2006. Proceedings, chap. Studying Evolution of a Branch of Knowledge by Constructing and Analyzing Its Ontology, pp. 37–45. Springer Berlin Heidelberg, Berlin, Heidelberg (2006), http://dx.doi.org/10.1007/11765448_4
9. Makagonov, P., Figueroa, A.R., Sboychakov, K., Gelbukh, A.: Learning a domain ontology from hierarchically structured texts. In: *Proc. of Workshop Learning and Extending Lexical Ontologies by using Machine Learning Methods*. 22nd International Conference on Machine Learning, ICML 2005, August 7-11, 2005, Bonn, Germany (2005)
10. Novak, J., Cañas, A.: La teoría subyacente a los mapas conceptuales y a cómo construirlos (06 2015)
11. Sterling, L., Shapiro, E.: *The art of Prolog*. The MIT Press (1992)

Reconocimiento multimodal de emociones orientadas al aprendizaje

Ramón Zatarain Cabada, María Lucia Barrón Estrada,
Héctor Manuel Cárdenas López

Instituto Tecnológico de Culiacán,
Culiacán, Sinaloa, México
{rzatarain, lbarron, hector_cardenas}@itculiacan.edu.mx

Resumen. Las emociones centradas en el aprendizaje tienen un rol significativo en el proceso pedagógico del estudiante. Por esta razón, es relevante en los ambientes de aprendizaje virtual tomar en consideración no solo los aspectos cognitivos del estudiante si no también los aspectos afectivos. Algunos métodos como el auto reporte, observacional y las imágenes de actividad neuronal, han sido usados extensivamente para medir emociones. Sin embargo, hoy en día otros métodos de análisis de sentimientos y reconocimiento facial usando inteligencia artificial han demostrado ser una mejor alternativa para el reconocimiento automatizado de afecto. Los resultados son superiores cuando se combinan diferentes métodos para combinar el reconocimiento de expresiones faciales, expresiones textuales, aplicadas en ambientes de aprendizaje. Para la implementación de reconocedores, utilizamos nuestro propio corpus para la clasificación que emplea técnicas de aprendizaje profundo. Evaluamos la eficiencia de tres métodos de fusión comparándolas contra métodos unimodales de reconocimiento de emociones. La mejora de uno de los métodos (Representación basada en embebidos) de fusión fue de 4% (precisión de 86.20% y pérdida 1.35%). Los resultados son muy prometedores y pensamos que serán mejor con tres o más reconocedores participando en el sistema multimodal.

Palabras clave: reconocimiento de emociones, reconocimiento multimodal, método de fusión.

Multimodal Recognition for Learning Centered Emotions

Abstract. Learning-centered emotions have a significant role in the pedagogical process of a student. For this reason, it is relevant that virtual learning environments take into account not only the cognitive aspects of the student but also the affective ones. Some approaches such as self-report, observational, and neuroimaging have been used extensively to measure emotions. However, today other methods of artificial intelligence such as the recognition of facial expressions and sentimental analysis have proven to be a better alternative in the automatic recognition of affect, and the results are superior when we combine and integrate several modes of recognition. In this work, we present three different methods to combine the recognition of facial expressions and textual expressions, applied to virtual learning environments. For the implementation of recognizers, we use our own corpora with classifiers that employ deep learning techniques. We evaluate the efficiency of the three fusion methods by comparing them against unimodal methods of emotion recognition. The improvement of one

of the methods (Embedding-based representation) of fusion was 4% (precision of 86.20% and loss of 1.35%). The results are very encouraging, and we think they will be better with three or more recognizers participating in the multimodal system.

Keywords: emotion recognition, multimodal recognition, fusion method.

1. Introducción

El mundo que nos rodea contiene múltiples modalidades. Vemos objetos, escuchamos sonidos y olemos olores. De esta manera el ser humano relaciona cada una de estas modalidades con un método de obtención de datos, sea la vista, el oído y el olfato respectivamente. De la misma manera nuestro cerebro relaciona a algunos olores con percepción de sentidos diferentes como la vista, creando relaciones importantes entre el olor de una tarta de fresa y la figura de dicha tarta. De la misma manera la inteligencia artificial busca el uso de estas diferentes modalidades para entender el mundo alrededor de nosotros y crear relaciones entre imágenes, voz, texto, etc.

Por otro lado, las emociones tienen un rol fundamental en la mayoría de las actividades de los seres humanos, ya sea en el proceso cognitivo y en entornos laborales [1], así como también se ha mostrado que las emociones están fuertemente ligadas con el proceso de aprendizaje y los niveles de concentración de las personas [2,3]. De la misma manera las emociones están ligadas a los procesos socioeconómicos [4], en los cuales se ha mostrado que existe una correlación entre el movimiento de la bolsa de valores y los movimientos generales económicos con el sentimiento generalizado de la población.

En la última década ha habido un gran interés en el desarrollo de tecnologías y estudios que permitan realizar un reconocimiento de emociones de manera automatizada, utilizando técnicas de aprendizaje máquina (ML) y aprendizaje profundo (DL). El reconocimiento de emociones se logra a través de la extracción de características en diferentes modalidades, como la voz, expresión facial, expresiones corporales, análisis de sentimientos (SA) y minería de opiniones (OM). Dentro del área de aplicación de tecnologías para el aprendizaje y más específicamente en el uso de ambientes de aprendizaje (AA) el reconocimiento de emociones es una herramienta de gran valor que permite mejorar el proceso de enseñanza-aprendizaje ya que una retroalimentación de un estudiante a través de un sistema de aprendizaje ayuda a comprender de mejor manera el proceso de aprendizaje del alumno. Esta retroalimentación contiene opiniones abiertas que permiten al AA realizar cambios en la programación de los materiales, estrategias y otros aspectos del proceso de enseñanza.

Uno de los principales problemas del uso de sistemas de clasificación se encuentra en la robustez que presentan estos sistemas para ser utilizados en ambientes no controlados que comúnmente generan muestras con ruido. Dentro de las técnicas de ML y DL se ha descubierto que el uso de diferentes modalidades para la clasificación genera modelos que presentan una mayor precisión y robustez. Debido a esto nosotros proponemos la implementación de un sistema multimodal para el reconocimiento de emociones orientadas al aprendizaje para su aplicación en AA.

En este artículo, presentamos tres metodologías creadas para la fusión de datos de emociones orientadas al aprendizaje para el reconocimiento multimodal (rostro y texto) utilizando DL. En la primera metodología utilizamos una fusión de representación de datos basada en imágenes para el entrenamiento de una red neuronal convolucionada (CNN). En la segunda metodología aplicamos una fusión de representación de datos basada en embebidos N-dimensionales para el entrenamiento de una red neuronal convolucionada combinada con memorias largas de corto plazo (CNN-LSTM). Por último, se utilizó un sistema híbrido de capas convolucionadas (CC) y con memorias largas de corto plazo (CLSTM) para la extracción de características y una capa de fusión de vectores de características para entrenar una red densamente conectada (FC).

Para la implementación de la fusión de datos se utilizaron tres diferentes corpus; uno de ellos basado en imágenes de rostros de personas recopilados durante el proceso de aprendizaje de programación java y dos corpus recopilados de opiniones en texto relacionadas también con el proceso de aprendizaje.

Nuestra principal contribución es el uso de metodologías multimodales para el reconocimiento de emociones orientadas al aprendizaje para ambientes no controlados, realizando una comparativa entre el desempeño de diferentes metodologías unimodales y nuestras metodologías multimodales centradas en la fusión de datos.

2. Trabajos relacionados

En esta sección se describen algunos trabajos de investigación relacionados con la aplicación de técnicas de DL para el reconocimiento de emociones, el uso de sistemas multimodales para la creación de modelos de DL y algunos trabajos de reconocimiento multimodal de emociones. Estos trabajos contienen similitud con nuestro trabajo y algunas técnicas de ellos fueron consideradas y abordadas para el desarrollo de este proyecto.

La relación entre las emociones básicas y las expresiones faciales fue estudiada extensivamente por Paul Ekman [5]. Su trabajo ha servido como una guía para el desarrollo de sistemas computacionales que sean capaces de reconocer emociones a través de las diferentes modalidades de entrada, como el rostro, la voz, expresiones faciales, etc.

Hoy en día existen muchos trabajos de investigación enfocados al área de reconocimiento de emociones utilizando modalidades como la voz y el rostro para la extracción de características que han logrado crear modelos de alta precisión utilizando técnicas de ML [6] y DL [7] con el uso de redes de creencia profunda (BLN). Además, se ha observado un gran avance en la minería de opiniones y la clasificación de emociones en texto utilizando técnicas de DL [8].

Las técnicas de reconocimiento de emociones también han sido utilizadas en conjunto con ambientes de aprendizaje utilizando diferentes clasificadores de DL. En [9] se utilizó minería de opiniones para obtener datos relacionados con el proceso de aprendizaje de Twitter para entrenar una CNN-LSTM para la clasificación de emociones. De la misma manera se ha utilizado el reconocimiento de emociones en el rostro para la clasificación de emociones utilizando diferentes técnicas de extracción de características y reconocimiento de expresiones faciales [10].

Sin embargo, el reconocimiento de emociones es un desafío, especialmente si los datos utilizados para clasificar las emociones son ruidosos. Esta es una de las razones por las cuales muchos sistemas de predicción o reconocimiento de emociones no reconocen correctamente en ambientes no controlados (fuera de laboratorios). En busca de resolver este problema se han creado concursos con corpus en ambientes no controlados y DL ha mostrado ser una de las técnicas más efectivas para este problema [11].

Existen diferentes trabajos que utilizan sistemas multimodales que demuestran un mejor desempeño comparados con sus contrapartes unimodales para diferentes propósitos, como el reconocimiento y auto etiquetado de contenido multimedia para búsquedas en línea [12], la clasificación de géneros de música [13], reconocimiento de contexto y actividades [14] y reconocimiento de objetos más robustos utilizando sistemas multimodales con DL [15]. Se ha realizado mucho trabajo en el campo de reconocedores de emociones multimodales. Un ejemplo es el uso de Tumblr para el análisis de sentimientos y reconocimiento de emociones [16] donde se realizó una comparación de desempeños entre dos sistemas unimodales y un sistema multimodal para el etiquetado de imágenes según su relación con alguna emoción o sentimiento.

Otro trabajo importante ha sido el reconocimiento multimodal de emociones utilizando arquitecturas de DL con datos fisiológicos, de voz y rostro [17], en donde se utilizó una red convolucional de creencia profunda para clasificar emociones sutiles o de baja intensidad.

De la misma forma, se ha realizado investigación acerca de modelos de reconocimiento de emociones multimodales utilizando datos duros de sensores obtenidos de dispositivos móviles y accesorios [18] en donde se utilizó un modelo multimodal CNN-LSTM y unimodal CNN para reconocimiento de emociones en humanos, en el cual se prueba que los modelos multimodales con técnicas de DL tienen un mejor desempeño que los modelos unimodales.

3. Reconocimiento multimodal

El reconocimiento es un proceso para predecir una etiqueta o una clase dada una cierta cantidad de información. Los reconocedores multimodales son una aproximación de la inteligencia artificial (IA) para darle la capacidad a un modelo de procesar y relacionar información de múltiples modalidades para discriminar y clasificar información.

Realizar este tipo de reconocedor requiere de crear sistemas que permitan procesar la información de cada una de estas modalidades para obtener una representación generalizada de cada una de ellas, ya sea a través de la fusión de datos en su representación, a la creación de sistemas de reconocimiento independientes, fusionando las clasificaciones directamente o a la creación de sistemas que permitan extraer características directamente de los datos y fusionar estas características para entrenar modelos de clasificación.

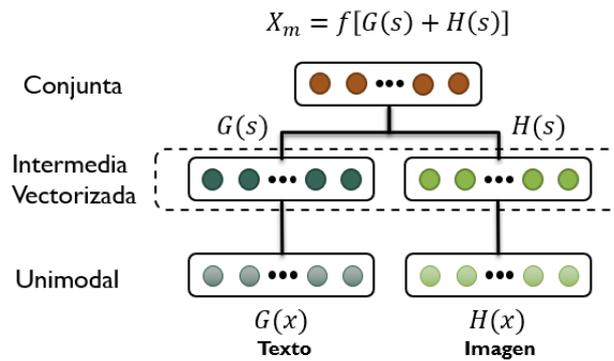
3.1. Representación multimodal de datos

La representación de datos se refiere a la creación de estructuras numéricas que representen de manera adecuada alguna entrada de valor de datos. Representar datos

duros en un formato que un modelo computacional pueda trabajar es uno de los más grandes retos del área de ML y DL multimodal. La representación de datos está fuertemente ligada al modelo de clasificación que se pretende implementar. Las redes neuronales convolucionadas por ejemplo utilizan imágenes como su medio de entrada que son pasadas por filtros para la extracción de características.

Para la representación multimodal de datos utilizamos una representación conjunta la cual consiste en la creación de representaciones intermedias basadas en modelos de interpretación de datos para ambas modalidades, de imagen y texto para posteriormente realizar una operación matemática creando así una nueva representación conjunta usada para entrenar modelos de DL. Un ejemplo de representación conjunta se puede observar en la figura 1.

Fig. 1. Representación conjunta de datos.



Para la representación de los datos de la modalidad de imagen se utilizaron dos modelos de color; el primero fue el modelo de color RGB con el cual se representaron los datos individuales de cada píxel a través de su valor hexadecimal creando una matriz representativa de cada uno de los colores contenidos en la imagen. El segundo fue el modelo de color a escala de grises, con el cual se determinaron números enteros representativos de cada uno de los valores de intensidad del color blanco dentro de la escala de grises, creando una representación matricial de la imagen a través de los valores individuales de cada píxel contenido en ella. Ambos modelos se observan en la figura 2.

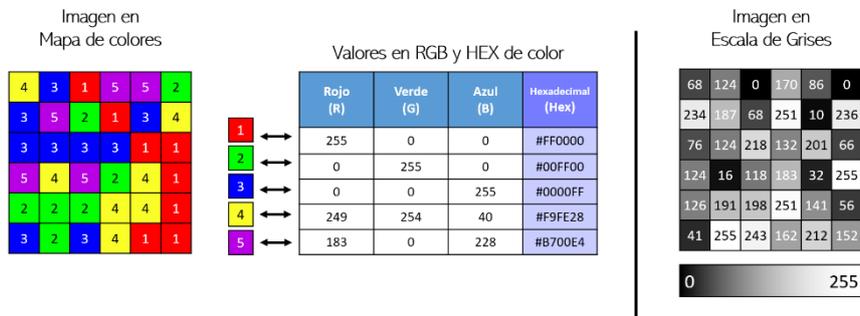


Fig. 2. Modelos de color para representación de imágenes.

Para la representación de los datos en la modalidad de texto se crearon diccionarios que utilizaban los índices para crear representaciones numéricas de cada una de las palabras contenidas dentro del corpus, creando un nuevo índice por cada palabra nueva encontrada dentro del corpus cuando este fue analizado. Un ejemplo de este diccionario se muestra en la figura 3.

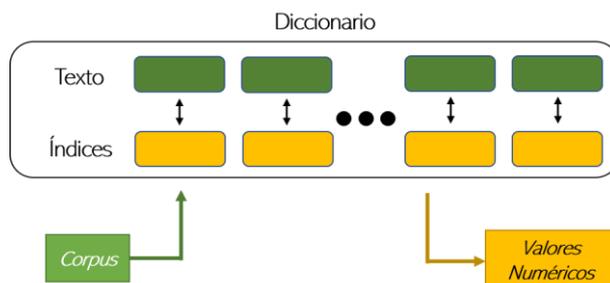


Fig. 3. Representación de diccionario para la traducción.

Una vez obtenidas las representaciones unimodales de datos se crearon corpus con representaciones conjuntas a través de sistemas de representación multimodal, tomando en cuenta los modelos a utilizar para la clasificación y el reconocimiento de emociones. En el caso de la CNN se utilizó una representación conjunta basada en imágenes debido a los métodos de extracción de características de este tipo de modelos y para el entrenamiento de una CNN-LSTM se utilizó una representación con embebidos de información. Para esto se realizó una representación intermedia en forma de vectores tanto de imágenes como de texto y se utilizó una capa de creación de embebidos previa al entrenamiento de este modelo.

3.2. Descripción de los corpus

El corpus es de suma importancia para la creación de cualquier modelo de clasificación o reconocimiento. Para este experimento utilizamos tres diferentes corpus unimodales, dos de ellos de texto y uno de imagen. Los corpus fueron creados en diferentes tiempos con técnicas diferentes. A continuación, se describen los tres corpus de manera breve, así como una pequeña explicación de su metodología de creación. Una descripción más detallada de la metodología de creación de estos corpus de imagen se puede observar en [9] y de texto en [10].

- **Corpus Sentitext (CST):** Consiste en un corpus recopilado de Twitter con 23,173 mensajes enfocados en el área del aprendizaje, etiquetados de manera manual buscando polaridad positiva o negativa en el texto.
- **Corpus Edusere (CES):** Consiste en un corpus recopilado de Twitter con 9,963 mensajes con enfoque en el área del aprendizaje, etiquetado de manera manual utilizando 5 de las emociones orientadas al aprendizaje (aburrido, enganchado, emocionado, concentrado, interesado)
- **Corpus Insight (CEI):** Este corpus de emociones en imágenes de rostro fue recopilado usando una diadema EMOTIV Insight con 5 canales de electroencefalograma a través del uso de un programa computacional para enseñar

JAVA utilizado en 38 estudiantes, 28 hombres y 10 mujeres con el cual se obtuvo un total de 5560 imágenes etiquetadas en 5 emociones orientadas al aprendizaje (aburrido, enganchado, emocionado, concentrado, interesado)

Estos corpus fueron utilizados para la creación de nuevos corpus multimodales utilizando las técnicas de representación conjunta de datos. Para esto se utilizó el CST en conjunto con el CEI para la creación de un corpus con una representación basada en imágenes (CSTEIBI). Además de esto se creó un segundo corpus con una representación conjunta utilizando los corpus CST y CEI creando así un nuevo corpus con representación basada en embebidos (CSTEIBE).

3.3. Sistemas de fusión de representación

Para llevar a cabo la fusión de representaciones unimodales en una representación multimodal, se debe convertir una modalidad en función de otra. Para ello se llevan a cabo tres pasos. Primero se hace un preprocesamiento de la información contenida en los corpus unimodales, donde se da a cada entidad del corpus una representación intermedia, ya sea a través de números, vectores o datos duros. Después se define el medio de representación final de los datos a la cual se convertirá esta representación intermedia a través de modelos de conversión o diccionarios. Por último, se realiza la creación de una metodología de conversión para los datos en su representación intermedia a su representación final. Este proceso se aprecia en la figura 4.

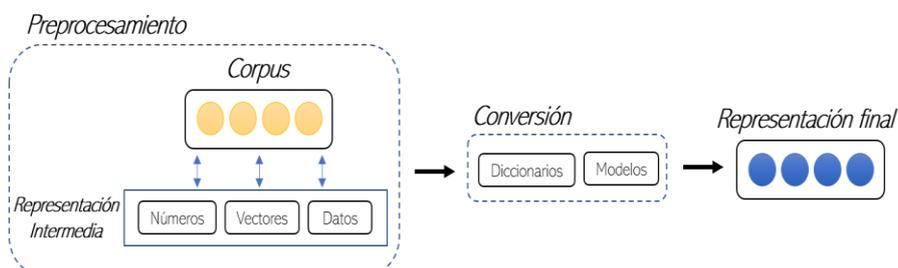


Fig. 4. Proceso de conversión para la fusión de modalidades.

Ambos sistemas de fusión con representación de datos siguen este patrón y sus implementaciones se muestran a continuación de manera detallada.

Fusión con representación basada en imágenes (FRBI). En la fusión con representación basada en imágenes se utilizaron dos corpus; el CEI para la modalidad de imagen con cinco etiquetas de emociones orientadas al aprendizaje y el CST para la modalidad de texto con dos etiquetas de polaridad en texto. Los corpus fueron fusionados para crear el nuevo CSTEIBI con diez etiquetas de polaridad en conjunto con emoción orientada al aprendizaje.

El preprocesamiento se realizó a través de un diccionario para la conversión del CST a una representación intermedia vectorizada. Primero creamos tokens de los documentos del corpus, después creamos un vector para contener y ordenar esos tokens.

Se indexaron los tokens no repetidos para asignarles índices en el diccionario, utilizando el índice como la representación de la palabra en número entero.

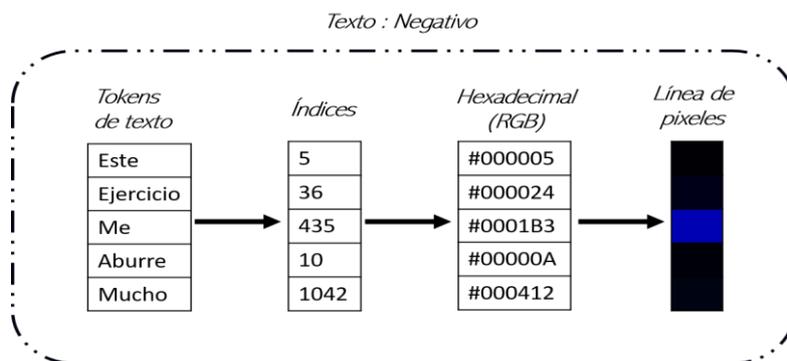


Fig. 5. Proceso de conversión de texto a imagen.

Dado que la representación final se encuentra en formato de imagen se utilizó el valor entero del índice de los tokens convertido a valor hexadecimal para crear un nuevo vector de 150 de tamaño. Utilizando el formato RGB se creó una representación en imagen del vector de texto relleno los espacios vacíos con ceros. Esto se convirtió a su representación de una imagen de 1 x 150 pixeles de tamaño. Este proceso se muestra en la figura 5.

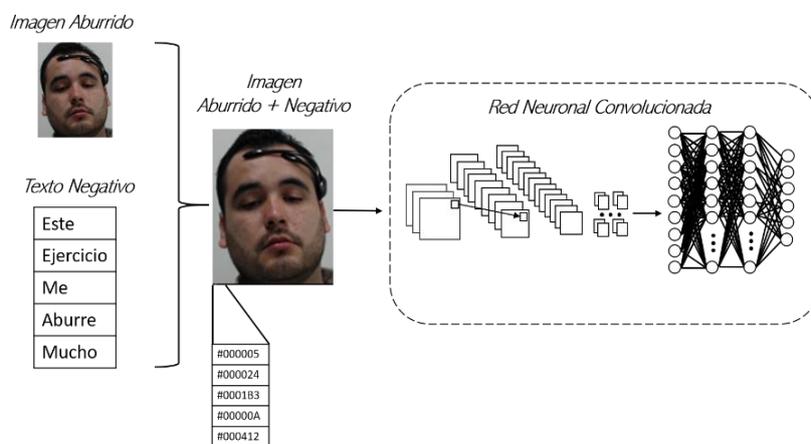


Fig. 6. Sistema de fusión de representación basada en imágenes.

Después se obtuvo la imagen del CEI y se preproceso para su transformación en una imagen de 150x150 pixeles de tamaño con 3 dimensiones en formato RGB. Una vez que ambas representaciones se encuentran en su formato de imagen se concatenan formando una nueva imagen de 150x151 pixeles de tamaño que representan la imagen del rostro y el texto. Se concatenaron las etiquetas del texto con la imagen para crear nuevas etiquetas que representaran ambas modalidades. Esta nueva imagen

representativa de ambas modalidades se anexo al nuevo CSTEIBI. Por cada imagen se obtuvieron polaridades positivas y negativas para crear representaciones conjuntas en forma de imagen.

Por último, se creó un modelo de CNN en el cual se utilizó el CSTEIBI para el entrenamiento y la validación. Esto se puede observar en la figura 6.

Fusión con representación basada en embebidos (FRBE). Para la fusión con representación basada en embebidos se utilizaron dos corpus; el CEI para la modalidad de imagen con cinco etiquetas de emociones orientadas al aprendizaje, y el CST para la modalidad de texto con dos etiquetas de polaridad, los cuales fueron fusionados para crear el nuevo CSTEIBE con diez etiquetas con pares de polaridad y emoción orientada al aprendizaje.

Para el preprocesamiento se utilizó un diccionario con los primeros 255 índices reservados para la conversión del CST y el CEI. Se obtuvieron las imágenes del CEI de las cuales se obtuvo un subconjunto de imagen centrada en el rostro. Se redimensionó la imagen a un tamaño de 150x150 píxeles y se convirtió a escala de grises, creando una matriz con la cual representa la intensidad del color blanco en la escala de grises con un máximo de 255 de valor. Por último, se procedió a convertir la matriz a vector, concatenando el último número de una fila con el primero de la siguiente. Este proceso se observa en la figura 7.

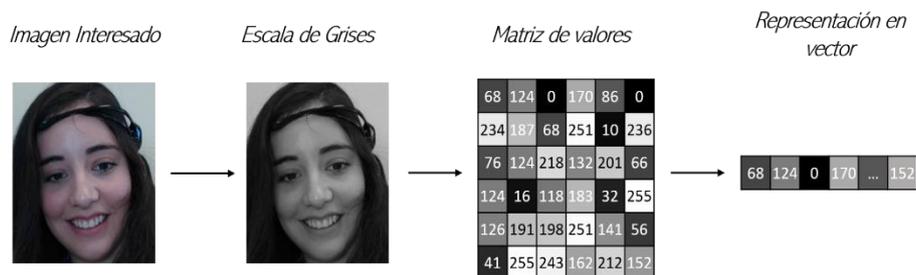


Fig. 7. Representación de imagen a vector.

De la misma manera preprocesamos el CST utilizando la misma técnica anteriormente mencionada a una representación intermedia vectorizada. Primero creamos tokens de los documentos del corpus, después creamos un vector para contener y ordenar esos tokens. Se indexaron los tokens no repetidos a partir del índice 256 para asignarles índice en el diccionario, utilizando el índice como su representación de la palabra en número entero. Se creó un vector para contener la información de cada uno de los documentos y se creó la representación vectorizada utilizando el diccionario.

Dado que la representación final para el entrenamiento de la red era un espacio de embebidos, se procedió a concatenar los vectores de imagen y texto para crear un nuevo vector conjunto, y de la misma forma se concatenaron las etiquetas para formar el nuevo CSTEIBE, el cual contenía la información en vector del texto y la imagen. Se utilizaron dos polaridades de texto por imagen para la creación de este nuevo corpus.

Finalmente, se creó un modelo de CNN-LSTM en el cual se colocó una capa de embebidos inicial, se utilizó el siguiente número superior al índice total del diccionario

para la creación de las dimensiones de entrada de la capa de embebidos, y se utilizó un valor arbitrario de 128 para las dimensiones del espacio vectorial para la creación de los embebidos. Se realizó el entrenamiento y se validó el desempeño del modelo.

3.4. Sistema de fusión de características (SFC)

Comúnmente se utilizan los términos característica y representación de manera intercambiable, ambos refiriéndose a una representación de tensor o vector de una entidad. Sin embargo, para la definición de este experimento, la palabra característica se define como el conjunto numérico vectorizado que representa una porción de dicha entidad, mientras que la representación es el medio en el cual se ve reflejada dicha entidad, pudiendo ser esto, una imagen, un texto, etc.

Para el sistema de fusión de características se realizó un proceso de preentrenamiento de un modelo CNN para la detección de cinco emociones orientadas al aprendizaje en imágenes utilizando el corpus CEI, y un modelo de CNN-LSTM con una capa de embebidos para la detección de cinco emociones en texto utilizando el corpus CES. Una vez que ambos modelos fueron entrenados y validados se guardó la configuración del grafo de cada modelo y sus pesos.

Se creó un nuevo modelo híbrido con las capas de extracción de características del modelo CNN y las capas de extracción de características del modelo CNN-LSTM con la capa de embebidos. Se anexó a este nuevo modelo una capa de concatenación de características con la cual se combinan las características extraídas de ambos modelos unimodales y se anexó al modelo una red neuronal densamente conectada para la clasificación. Esta arquitectura se muestra en la figura 8.

Finalmente, se inicializaron los pesos utilizando los valores obtenidos de las capas de extracción de características de los dos modelos unimodales de texto y de imagen, se entrenó y evaluó este nuevo modelo.

Para el entrenamiento se preprocesaron las imágenes del CEI obteniendo una imagen de la cara de 150 x 150 píxeles de tamaño con tres dimensiones en formato RGB. De la misma manera se preprocesó el texto utilizando un proceso de creación de tokens de texto, creación de un diccionario y creación de un vector representativo de los documentos del corpus.

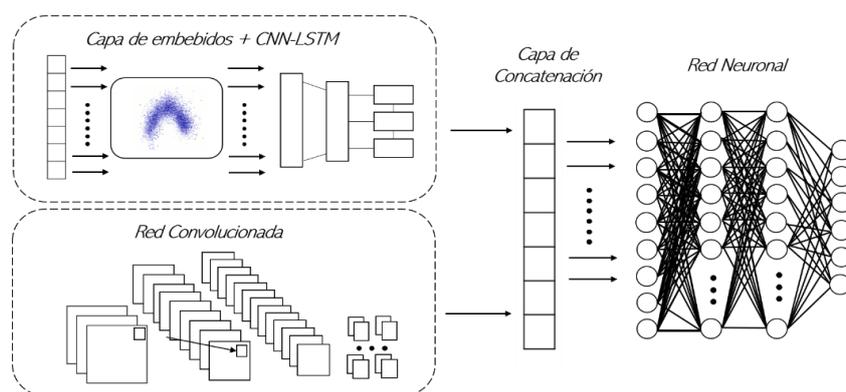


Fig. 8. Modelo de fusión de características.

Debido a que los corpus provenían de diferentes medios de información, se buscó que el entrenamiento se realizara con etiquetas idénticas en los datos utilizados.

4. Pruebas y discusión

Para los experimentos se utilizó el corpus CEI para el entrenamiento de un modelo de CNN y modelos de CNN-LSTM con una capa de embebidos para el CST y el CES. Los experimentos fueron realizados con validación cruzada de k iteraciones con $k = 5$ debido a que el entrenamiento de este tipo de modelos tomaba entre 1 y 2 días, especialmente con los modelos que utilizan una capa de embebidos. De la misma manera se tomó en consideración que la distribución de clases de los corpus dentro de las iteraciones fuera uniforme. En cada uno de los modelos entrenados la distribución de los corpus fue 80% para el entrenamiento y 20% para la validación del modelo.

Se utilizaron diferentes versiones de los dos corpus de texto (CST 2018, CST 2019a, CST 2019b, etc.). Las diferentes versiones que fueron creadas se diferencian en algunas expresiones que fueron agregadas y corregidas en cada uno de ellos.

Las métricas utilizadas para la evaluación de estos modelos y sistemas fueron basadas en los siguientes parámetros: la precisión en la clasificación, la pérdida de clasificación, el tamaño total de los corpus y el número de clases. La Tabla 1 muestra los resultados de las pruebas.

El modelo multimodal FRBI presentó los peores resultados (62.13 %) comparado con los otros modelos unimodales y multimodales. El modelo multimodal FRBE presentó resultados prometedores con 86.20 % de precisión en la etapa de validación. Los modelos unimodales que utilizaron el mismo corpus obtuvieron 91% de precisión en clasificación de polaridad (solamente dos clases) en texto usando el CST2019b y 74.16% en clasificación de emociones para expresiones faciales (cinco clases) utilizando el CEI. Como podemos ver en la tabla, el desempeño del FRBE fue mejor en la clasificación de diez clases comparado con los modelos unimodales.

Tabla 1. Comparación de los diferentes modelos y corpus del experimento.

Corpus	Modelo	Precisión	Perdida	Tamaño	Clases
CST 2018	CNN+LSTM	79.08%	1.72%	7492	2
CST 2019a	CNN+LSTM	87.28%	1.25%	164990	2
CST 2019b	CNN+LSTM	91.80%	1.10%	22554	2
CES 2018	CNN+LSTM	69.16%	2.76%	3017	5
CES 2019a	CNN+LSTM	69.47%	3.36%	4504	5
CES 2019b	CNN+LSTM	69.47%	3.45%	4504	5
CEI	CNN	74.16%	2.41%	5056	5
(CEI+CST2019b)	FRBI	62.13%	4.20%	50164	10
(CEI +CST2019b)	FRBE	86.20%	1.35%	50164	10
(CEI+CES2019b)	SFC	81.16%	1.57%	9560	5

El modelo SFC presentó buenos resultados con una precisión del 81.16 % comparado con los modelos unimodales para el reconocimiento de emociones en imágenes utilizando el CEI donde el modelo de CNN presentó 74.16 % de precisión. Por el otro lado los modelos unimodales de reconocimiento de texto usando diferentes versiones del corpus CES presentaron resultados menores al 70% de predicción durante su validación.

5. Conclusiones y trabajo futuro

En este artículo propusimos tres diferentes aproximaciones al reconocimiento de emociones multimodal, utilizando técnicas de representación conjunta con fusión de datos y un modelo de fusión de características. El objetivo de esta investigación fue el buscar una mejoría en el reconocimiento de emociones utilizando técnicas multimodales en vez de crecer los corpus de manera directa.

Usando técnicas multimodales logramos mejorar la precisión de modelos de reconocimiento utilizando corpus unimodales a través de representación conjunta de datos para el entrenamiento. Se ha mostrado también que esta mejoría solamente es posible cuando la representación de datos es apropiada. En la FRBI concluimos que la representación de los datos de texto (a través de agregar texto como una línea de píxeles) no era adecuada para el entrenamiento de un modelo de CNN por lo cual entregó resultados de baja precisión. Sin embargo, en la FRBE de ambos imagen y texto mostró resultados prometedores con el uso de un modelo de CNN-LSTM. De la misma manera utilizando un SFC ha demostrado ser valioso en cuanto al desempeño en precisión de modelos para detección de emociones.

También a través de los resultados mostrados en este estudio podemos concluir que se pueden utilizar diferentes corpus unimodales para el entrenamiento de sistemas multimodales mientras se mantenga una relación semántica similar como en el caso del SFC donde las clases utilizadas para la clasificación son iguales o se haga una representación conjunta de los corpus.

Existen varias limitaciones con el estudio existente que necesitan ser observadas en trabajos futuros. Una limitación es el tamaño del corpus CEI que resulta pequeño para una tarea de reconocimiento de emociones. Otra limitante es la cantidad de métodos de representación conjunta utilizados dentro de este experimento, la cual es baja, además de que otros métodos de representación conjunta pueden ser utilizados, como por ejemplo el uso de una representación basada en filtros del texto aplicado al corpus de imágenes. Otra limitante de este estudio es que las pruebas de desempeño se realizaron en ambientes de laboratorio y solamente con datos de los corpus de entrenamiento originales. Además de esto también se vio limitada la cantidad de K en las técnicas de validación cruzada, ya que se puede utilizar una K más grande u otras técnicas de validación para corroborar el desempeño de estos modelos.

Como trabajo futuro proponemos realizar evaluaciones de desempeño de los modelos de este artículo utilizando ambientes de aprendizaje en aplicaciones con ambientes no controlados. También proponemos ampliar el corpus CEI y realizar una mayor cantidad de balanceo de clases para obtener aún mejores resultados. También proponemos continuar con la investigación de diferentes aproximaciones para mejorar el desempeño de modelos de reconocimientos de emociones como la transferencia de aprendizaje, algoritmos genéticos para selección de hiperparámetros en modelos multimodales para obtener mejores resultados.

Referencias

1. Fisher, C.D., Ashkanasy, N.M.: The emerging role of emotions in work life: an introduction, *J. Organ. Behav.* 21(2), 123–129 (2000)

2. Bollen, J., Mao, H., Zeng, X.-J.: Twitter mood predicts the stock market. (2010)
3. Tyng, C.M., Amin, H.U., Saad, M.N.M., Malik, A.S.: The Influences of Emotion on Learning and Memory. *Front. Psychol.*, vol. 8, p. 1454 (2017)
4. Linnenbrink-Garcia, L., Pekrun, R.: Students' emotions and academic engagement: Introduction to the special issue, *Contemp. Educ. Psychol.* 36(1), 1–3 (2011)
5. Ekman, P.: An argument for basic emotions. *Cogn. Emot.* 6(3–4), 169–200 (1992)
6. Crangle, C.E., Wang, R., Perreau-Guimaraes, M., Nguyen, M.U., Nguyen, D.T., Suppes, P.: Machine learning for the recognition of emotion in the speech of couples in psychotherapy using the Stanford Suppes Brain Lab Psychotherapy Dataset (2019)
7. Hassan, M.M., Alam, M.G.R., Uddin, M.Z., Huda, S., Almogren, A., Fortino, G.: Human emotion recognition using deep belief network architecture. *Inf. Fusion*, vol. 51, pp. 10–18 (2019)
8. Chatterjee, A., Gupta, U., Chinnakotla, M.K., Srikanth, R., Galley, M., Agrawal, P.: Understanding Emotions in Text Using Deep Learning and Big Data. *Comput. Human Behav.*, vol. 93, pp. 309–317 (2019)
9. González-Hernández, F., Zatarain-Cabada, R., Barrón-Estrada, M.L., Rodríguez-Rangel, H.: Recognition of learning-centered emotions using a convolutional neural network. *J. Intell. Fuzzy Syst.* 34(5), 3325–3336 (2018)
10. Oramas Bustillos, R., Zatarain Cabada, R., Barrón Estrada, M.L., Hernández Pérez, Y.: Opinion mining and emotion recognition in an intelligent learning environment. *Comput. Appl. Eng. Educ.* (2018)
11. Levi, G., Hassner, T.: Emotion Recognition in the Wild via Convolutional Neural Networks and Mapped Binary Patterns. In: *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction - ICM1 '15*, pp. 503–510 (2015)
12. Yuhas, B.P., Goldstein, M.H., Sejnowski, T.J.: Integration of acoustic and visual speech signals using neural networks. *IEEE Commun. Mag.* 27(11), 65–71 (1989)
13. Bartolini, I., Ciaccia, P.: Scenique. In: *Proceedings of the working conference on Advanced visual interfaces - AVI '08*, p. 476 (2008)
14. Oramas, S., Barbieri, F., Nieto, O., Serra, X.: Multimodal Deep Learning for Music Genre Classification. *Trans. Int. Soc. Music Inf. Retr.* 1(1), 4–21 (2018)
15. Radu, V., Bhattacharya, S., Lane, N.D., Marina, M.K., Tong, C., Mascolo, C., Kawsar, F.: Multimodal Deep Learning for Activity and Context Recognition, vol. 1, no. 4, p. 27 (2017)
16. Jost, A.E., Springenberg, T., Spinello, L., Riedmiller, M., Burgard, W.: Multimodal Deep Learning for Robust RGB-D Object Recognition.
17. Hu, A., Flaxman, S.: Multimodal Sentiment Analysis to Explore the Structure of Emotions, (2018)
18. Ranganathan, H., Chakraborty, S., Panchanathan, S.: Multimodal emotion recognition using deep learning architectures. In: *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1–9 (2016)

Reconocimiento de gestos de la mano aplicado a una interfaz para ambientes de aprendizaje

Ramón Zatarain-Cabada, María Lucía Barrón-Estrada,
Brandon Antonio Cárdenas Sainz

Instituto Tecnológico de Culiacán, Culiacán, Sinaloa, México
{rzatarain, lbarron, brandon_cardenas}@itculiacan.edu.mx

Resumen. En este artículo presentamos un prototipo de una aplicación web interactiva 3D, la cual combina la interfaz de Google Blockly con una interfaz de interacción con implementación de reconocimiento de gestos de manos humanas. La aplicación consiste en que el usuario final tenga las posibilidades de programar sus propios gestos en 3D, asociarlos a eventos e implementarlos dentro de un entorno gráfico, mientras se utiliza programación basada en bloques para la creación de aplicaciones. Todo con el fin de desarrollar el pensamiento computacional del individuo y que este se familiarice con conceptos de interfaces humano-máquina naturales, programación algorítmica y pensamiento espacial. Aplicando evaluaciones de diseño centrado en el humano, se demuestra que las interfaces naturales y gráficas presentadas en este proyecto cumplen con los principios básicos de interacción natural y se tienen expectativas que indican que nuestro enfoque impacta de forma positiva el proceso de aprendizaje, aumentando la motivación y disfrute.

Palabras clave: interacción humano-máquina, interfaces naturales de usuario, pensamiento computacional, reconocimiento de gestos.

Recognition of Hand Gestures Applied to an Interface for Learning Environments

Abstract. This paper presents a prototype of an interactive 3D web application, which combines the Google Blockly interface with an interaction interface using an implementation of recognition of human-hand gestures. The application allows the end-user to be able to program their own 3D gestures, associate them to events and implement them within a graphical environment while using block-based programming for the creation of applications. The main goal is to develop the computational thinking of the individual and that becomes familiar with concepts of natural human machine interfaces, algorithmic programming, and spatial thinking. By applying human-centered design evaluations, it is demonstrated that the natural and graphic interfaces presented in this project comply with the basic principles of natural interaction and have expectations that indicate that our approach positively affects the learning process, increasing motivation and enjoyment.

Keywords: human-machine interaction, natural user interfaces, computational thinking, gesture recognition

1. Introducción

El sector educativo busca más que nunca en la actualidad, diversas formas para el desarrollo del pensamiento computacional. Esta es una serie de habilidades que integran la resolución de problemas, el diseño de sistemas y algoritmos, así como el

entendimiento del comportamiento humano, aplicando para ello conceptos estructurales de la computación. Este proceso se enfoca en el reconocimiento de aspectos de la informática en el mundo que nos rodea, y aplica herramientas y técnicas de la informática para comprender y razonar sobre los sistemas y procesos tanto naturales como artificiales [1].

Usualmente, cuando se habla de informática y de la computación, o de temas como la programación e interacción con una computadora, nos imaginamos a alguien sentado sobre una mesa, escribiendo sobre un teclado y moviéndose en pantalla por medio de un ratón o touchpad. Esta forma de interactuar con las computadoras no ha cambiado significativamente desde 1960 [2], época en la que estos periféricos fueron inventados. Actualmente existen diversos dispositivos en el mercado tales como tabletas y teléfonos inteligentes, en donde se presentan interfaces más innovadoras: pantallas con capacidad táctil y el uso de gestos que representan una forma más natural de interacción entre el humano y la máquina. También se han realizado diversas implementaciones de reconocimiento de gestos en dispositivos con sensores o cámaras con la finalidad de ofrecer ciertas simplicidades de uso [3].

El reconocimiento de gestos es un tema en las ciencias de la computación y de la tecnología del lenguaje con el objetivo de interpretar gestos humanos a través de algoritmos matemáticos. Los gestos pueden ser cualquier movimiento corporal o estado, pero comúnmente se originan a partir de un rostro o de las manos. El reconocimiento de gestos puede ser visto como la manera en que las computadoras interpretan el lenguaje corporal humano, considerándose como una interfaz natural humano-computadora.

Esto nos lleva a los conceptos de interacción humano-máquina (HMI) y las interfaces naturales de usuario (NUI), donde esta última define que los humanos se pueden comunicar con la máquina e interactuar naturalmente sin dispositivos mecánicos [2]. Utilizando el concepto de reconocimiento de gestos, es posible usar los dedos en un espacio libre para relacionar el movimiento del cursor con los movimientos del usuario.

Actualmente, la presencia de interfaces humano-máquina más naturales se ha vuelto una necesidad, ya que estas pueden proporcionar nuevas capacidades y facilidad de uso en diversos dispositivos computacionales, ya sean computadoras, teléfonos inteligentes, tabletas, máquinas del sector industrial, etc. Estas interfaces ofrecen al usuario una mejora en el tiempo de aprendizaje, la velocidad de desempeño, a la tasa de error de uso y la forma en que un usuario opera el software de un dispositivo y la satisfacción que se experimenta en el uso [4].

Por otra parte, el desarrollo del pensamiento computacional y de que un individuo sea capaz de programar sigue siendo hoy en día un problema abordado por diversas investigaciones. Desde la creación de herramientas digitales llamadas e-learning [5] para facilitar el aprendizaje, sistemas o mecanismos de asistencia de acuerdo al rendimiento del individuo (sistemas tutores inteligentes STI) [6], la comprensión de la conducta humana y de sus estados emocionales y como estos afectan su rendimiento a la hora de aprender (sistemas tutores afectivos STA) [7], son algunos de los enfoques que ofrecen diversas soluciones. Uno de estos enfoques es cómo el usuario llega a interactuar con un dispositivo de cómputo, herramientas en las cuales se realiza el ejercicio cognitivo computacional.

Como se mencionaba anteriormente, algunos estudios indican que la utilización de interfaces humano-máquina con un enfoque a la interacción natural ayuda a la visualización, la comprensión y por ende el aprendizaje del usuario [8] por lo que puede ser de gran ayuda para el sector educativo.

Por ello, el objetivo de esta investigación es la exploración de nuevas formas de interacción implementadas en entornos de aprendizaje, así como el impacto sobre el proceso de aprendizaje, aumentando la motivación, el disfrute del estudio y el desarrollo del pensamiento computacional.

Este artículo está organizado de la siguiente forma: la Sección 2 describe los trabajos relacionados. La Sección 3 presenta la estructura y funcionamiento de la herramienta. En la Sección 4 se muestra una sesión de trabajo. La Sección 5 contiene algunos de los experimentos realizados y los resultados obtenidos. Por último, en la Sección 6 se describen las conclusiones y trabajos futuros.

2. Trabajos relacionados

En esta sección se describen investigaciones y trabajos en áreas relacionadas con el campo del reconocimiento de gestos y el pensamiento computacional en la educación.

Una de las herramientas más importantes que se pueden mencionar es Scratch [9], una comunidad de aprendizaje creativo en línea, donde los usuarios pueden crear sus proyectos a partir de bloques gráficos, donde un bloque representa un elemento del lenguaje de la programación, como son: control, operadores, variables, funciones, etc. Otra herramienta importante es Alice [10], en donde principiantes pueden programar animaciones y juegos 3D utilizando modelos ya predefinidos a partir de una interfaz visual basada en la programación por bloques.

En el marco del uso de interfaces naturales, mediante el reconocimiento de gestos en el campo de la educación, los trabajos de investigación más importantes han estado dedicados principalmente a estudiar factores como la usabilidad y la motivación en prototipos educativos. S. Bowe, M. Antoniou, C. Garrett et al. [11] presentan el aprendizaje de niños y jóvenes utilizando dispositivos con pantalla táctil, donde se realiza una actividad con un juego de la Torre de Hanoi, dentro de una aplicación Android, para luego resolver uno real, concluyendo que el conocimiento y habilidades aprendidas en una tarea desde un dispositivo móvil pueden ser transferidas en el mundo real. De acuerdo con el trabajo de M. Ebner et al. [12] el uso de interfaces innovadoras como el controlador Leap Motion (LMC), mejora significativamente el proceso de aprendizaje dentro de aplicaciones que educan mediante el juego. Estas mejoran la motivación, el enganche y la diversión de los estudiantes mientras juegan aprendiendo. G. Zhu, S. Cai, y. Ma et al. [13] usan LMC para analizar la efectividad para la rehabilitación motriz de niños con autismo a partir de una serie de ejercicios en aplicaciones para el sector escolar. V. Tran, J. Lee, D. Kim et al. [14] en colaboración con The LEGO Group, LEGO System A/S Denmark usaron los dispositivos LMC y Kinect para presentar en VR una actividad interactiva que permitía al usuario diseñar y construir edificaciones con piezas LEGO.

Hasta el momento, no existe un sistema integral, con el enfoque del cómo el usuario interactúa con un dispositivo de cómputo y con la premisa de que el usuario aprenda y pueda dominar los conceptos que componen el pensamiento computacional, a partir de la programación e implementación de las mismas interfaces interactivas dentro de un entorno de aprendizaje.

La contribución principal de este proyecto es presentar un prototipo de un entorno de aprendizaje que englobe diferentes interfaces de interactividad natural, así como el uso de interfaces gráficas de programación mediante bloques y de entornos gráficos interactivos, con el fin de ofrecer al usuario final la capacidad de programar el modo de interacción entre un dispositivo de cómputo.

3. Interfaz de programación visual y reconocimiento de gestos

Para la investigación de este proyecto, se ha creado un prototipo de una aplicación web, la cual consiste en que el usuario final tenga las posibilidades de programar sus propios gestos en 3D, asociarlos a eventos e implementarlos dentro de un entorno gráfico, mientras se utiliza programación basada en bloques para la creación de aplicaciones. El principal objetivo es desarrollar el pensamiento computacional del individuo y que este se familiarice con conceptos de interfaces humano-máquina naturales, programación algorítmica y pensamiento espacial, etc. La figura 1 nos muestra un diagrama de los componentes del prototipo y las librerías utilizadas.

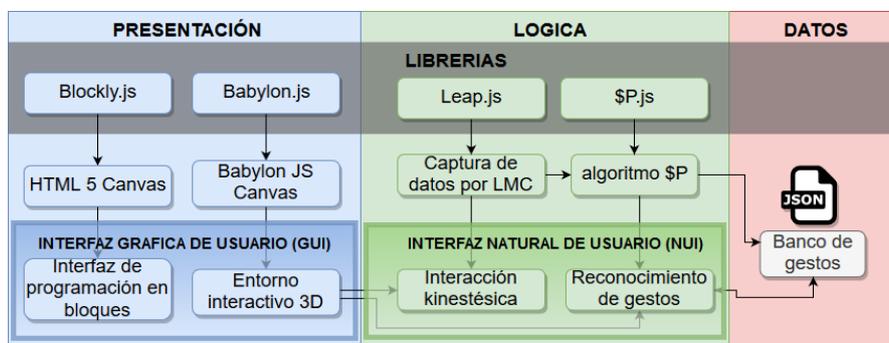


Fig. 1. Diagrama de componentes del prototipo interactivo.

El prototipo posee un entorno de interacción 3D con un sistema de físicas y tiene como propósito que dentro de un escenario virtual el jugador cree e interactúe con elementos 3D. La aplicación muestra al usuario una interfaz de programación visual para que genere elementos dentro de un escenario, creando cuerpos geométricos tridimensionales, los cuales pueden poseer una masa y ser afectados por simulación de efectos de gravedad, así como aplicación de fuerzas, velocidades e impulsos tanto angulares como lineales. Además, el usuario final también tiene la posibilidad de programar eventos, acciones y la forma en que sus manos interactúan con la escena 3D y sus objetos.

El entorno interactivo está ubicado en un escenario 3D (ver figura 7 en sección 4.2). Su presentación más básica consiste en un plano 3D que funciona como terreno, sin embargo, se tiene la posibilidad de cargar diferentes escenarios. Este incluye un avatar, un personaje caricaturesco redondo que posee un par de manos flotantes, las cuales son utilizadas para representar los gestos de manos. El avatar también representa la posición en la que se encuentra el jugador y ofrece un punto de visión de las manos dentro del entorno. El usuario a partir de bloques, programa la creación de elementos dentro del entorno gráfico y las funciones de interacción que implementan el reconocimiento de gestos.

3.1. Reconocimiento de gestos con Leap Motion

Leap Motion es un periférico USB principalmente diseñado para la detección de la posición de dedos y gestos de una mano para software de aplicaciones interactivas. Posee una API de desarrollo la cual provee acceso a un modelo de las características

vectoriales del esqueleto de una mano detectada y está diseñado para ser situado en un escritorio, mirando verticalmente. El campo de visión tiene una forma cónica, la cual se extiende entre 2.5 a 60 cm por encima del dispositivo. El Controlador de Leap Motion (LMC) posee dos cámaras infrarrojas monocromáticas y tres LEDs infrarrojos. Esto permite al software del controlador generar un patrón 3D a partir de los datos 2D obtenidos por ambas cámaras (ver figura 2).

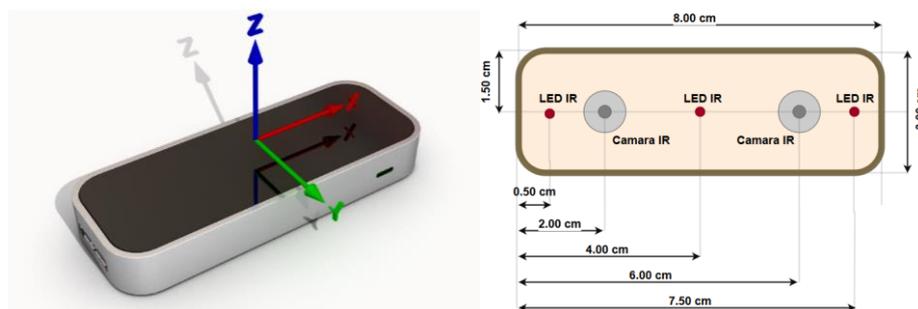


Fig. 2. Dimensiones del controlador Leap Motion.

3.2. Reconocimiento de gestos en entornos 2D y 3D y aprendizaje máquina

Este proyecto se enfoca principalmente en realizar reconocimiento de gestos interpretando a este como el trayecto realizado por las puntas de los dedos de una mano, simulando la información que se recibe al utilizar algún dispositivo táctil, pero aplicado a un entorno 3D. Esto último implica que debe de existir alguna forma de registrar la trayectoria de una mano no solo en una perspectiva 2D, sino también 3D.

Para el problema de localización de la posición de una mano de acuerdo a la profundidad de un entorno este proyecto se ha enfocado en dar a conocer 2 propuestas: utilizar un algoritmo que pueda reconocer gestos realizados tanto en 2D como 3D y determinar campos de profundidad de un entorno a partir de áreas de interacción.

“SP” es un algoritmo que tiene como propósito ofrecer una solución rápida de implementación de reconocimiento de gestos a proyectos que requieran interacciones humana-máquina naturales [15]. Este representa a un gesto como una nube de puntos que, al unirlos entre sí, forman un trazo o trayectoria geométrica que después es rotada, escalada y normalizada, para que sea comparada con una base de datos que almacene gestos (ver figura 3). En términos de aprendizaje máquina, SP es un clasificador basado en KNN con una función de distancia euclidiana 2D, es decir, un igualador de plantillas geométricas.

Para el reconocimiento de gestos se ha adaptado el algoritmo SP para su uso en 3D, representando a un gesto como el desplazamiento de las puntas de los dedos de una mano, formando un trazo o una serie de puntos en un plano. A partir del dispositivo LMC, se obtienen características de las manos humanas que se encuentren dentro de su campo de visión. La información es almacenada en forma de series de fotogramas o frames, parecidos a tomar video, pero en su lugar se registran datos vectoriales de la posición, orientación, grados de inclinación de cada parte que compone a la mano humana, desde los huesos de la muñeca, las articulaciones y huesos de los dedos.

Esta información es procesada y registrada en un archivo en formato JSON, el cual contiene los datos vectoriales de cada frame, con su respectivo tiempo de captura. Esto permite que se creen animaciones de acuerdo a la duración del movimiento o gesto.

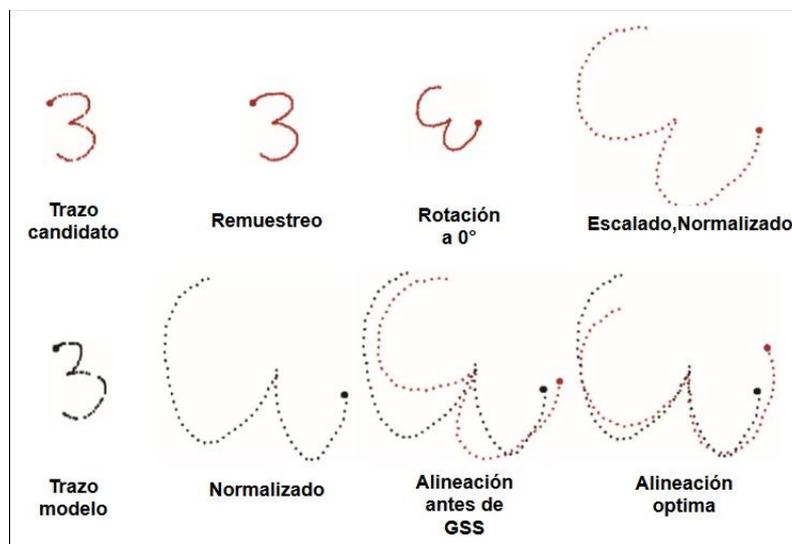


Fig. 3. Pasos del proceso de reconocimiento en \$P recognizer. GSS significa Golden Section Search [15].

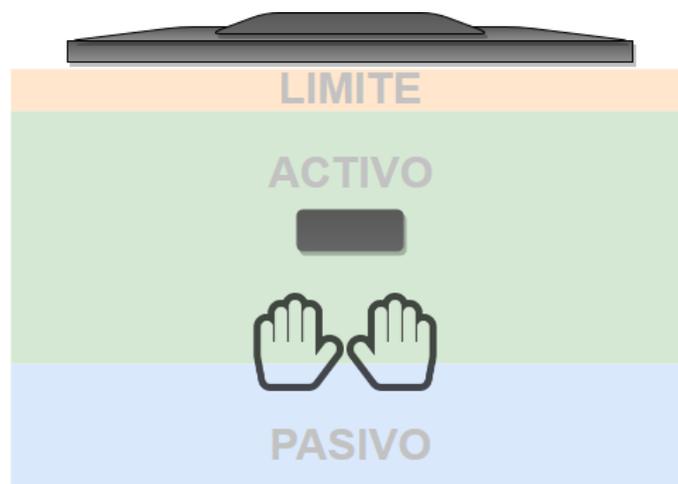


Fig. 4. Representación de las áreas de interacción con respecto a la profundidad entre el usuario y el LMC.

Para el reconocimiento de los gestos 3D, se ha optado por separar el gesto realizado por las puntas de los dedos de una mano en sus diferentes perspectivas de acuerdo a los planos ortogonales XY y XZ. Sin embargo, implementar un sistema de reconocimiento de gestos dentro de entornos 3D implica también reconocimiento de la posición de la mano dentro de la misma en coordenadas 3D. Por lo que es necesario delimitar en qué momento un gesto es usado. Se definen 3 estados dentro del área de visión del sensor, un estado donde las manos son representadas, pero la interactividad y gestos están desactivados (PASIVO), el estado en la que la interactividad esta activada, pero el reconocimiento de gestos se mantiene desactivado (ACTIVO) y el estado cuando se

toca el límite frontal del LMC para activar reconocimiento de gestos (LIMITE) (ver figura 4).

3.3. Interactividad kinestésica en entornos 3D

Si bien, el reconocimiento de gestos forma parte de una interfaz humana-máquina natural, este no proporciona capacidades de interacción kinestésicas dentro del entorno. Es decir, que el usuario pueda interactuar con los elementos que se encuentre dentro de un escenario virtual de la misma manera que se realiza en la vida real, como levantar un objeto con sus manos, moverlo de un lugar a otro, lanzarlo, etc.

Esta es una característica que un sistema de reconocimiento de gestos no puede atender, puesto que este no utiliza información relacionada a por ejemplo verificar, si las manos dentro de un entorno virtual colisionan o se encuentran cerca de un objeto interactivo. Por lo cual, crear un componente de interacción para esta clase de entornos también forma parte de la interfaz de este proyecto.

Para que el usuario final dentro de la aplicación pueda interactuar directamente con sus manos, se utiliza una representación tridimensional de las manos, que consiste de un modelo esquelético para el entorno gráfico 3D. Además, la interfaz cuenta con un componente dedicado a la interacción mediante eventos a partir de ciertas posturas y acciones de las manos.

Este componente primero utiliza la información recibida por el sensor para evaluar la distancia entre el dedo pulgar y el dedo índice (picking). Después, evalúa que tanto la palma de la mano está cerrada o abierta (grabbing) y si el dedo índice está apuntando a un objeto a distancia (pointing) mientras se determina si la posición de la mano virtual se encuentra cerca de un elemento interactivo con el que se pueda realizar alguna acción. Esto ofrece formas en la que el usuario puede interactuar con los elementos dentro del entorno y también están disponibles para ser utilizados para programar eventos personalizados.

3.4. Interfaz de programación visual Blockly

El usuario final utiliza bloques que representan una serie de comandos para la creación de código, lo cual permite que el ejercicio de codificación se concentre más en la lógica del problema, lo que apoya el proceso de aprendizaje de los principiantes.

En la aplicación, Blockly [16]. Se encarga de proporcionar bloques que representan constructores de figuras geométricas tridimensionales, funciones básicas de personalización, configuración de posición, así como funciones orientadas al sistema de simulación de físicas, donde se aplican propiedades como masa, velocidad, impulsos, fuerza, gravedad, etc. Esto con el fin de presentar las funciones necesarias para la creación de objetos dentro del escenario. Además, se presentan bloques de ciclos, lógica y variables que tiene por defecto Blockly. (ver más adelante figura 6 en sección 4.2).

También se proporciona bloques relacionados a la programación de eventos de interacción y de reconocimiento de gestos, donde el usuario final puede programar en qué momento o bajo que circunstancia ocurre alguna acción relacionada al interactuar con los elementos del escenario. Esto significa que, a partir de la programación de eventos interactivos mediante gestos, se puede cambiar de color un objeto, cambiar de tamaño, moverlo, rotarlo, etc. También se pueden programar eventos de interacción kinestésicos, donde al sostener un objeto y después soltarlo, que el objeto sea lanzado,

o programar eventos de áreas de interacción, donde si una mano se encuentra fuera o dentro del campo de visión del sensor se controle la gravedad de toda la escena, etc.

4. Uso de la aplicación web

El usuario tiene la posibilidad de utilizar 2 secciones dentro de la aplicación. La primera llamada CREA está enfocada a la importación o exportación de un corpus de gestos. La segunda llamada JUEGA que consiste en el entorno de interacción que utiliza una interfaz visual basada en bloques para la programación de escenarios interactivos y que utiliza la carga de corpus de gestos creados por el propio usuario para la implementación de reconocimiento de gestos dentro de la interfaz de interacción.

4.1. CREA tus propios gestos

CREA consiste en la creación de un corpus de reconocimiento de gestos, donde a partir de la técnica de *programación por ejemplos* [17], se crea un archivo JSON que representa el corpus de gestos con su respectiva etiquetación, la cual puede ser cargado en los entornos del apartado JUEGA. Para crear gestos, el usuario usa sus manos sobre el sensor LMC, mientras que el software automáticamente procede a capturar datos y etiquetar por tiempo (en milisegundos) los fotogramas (frames) de lo que dura el gesto. La figura 5 muestra la interfaz del entorno de desarrollo CREA.

CREA permite al usuario visualizar los gestos producidos por medio de un reproductor que muestra una representación animada del gesto en 3D, y que incluye la visualización de las trayectorias realizadas por las puntas de sus dedos en perspectivas XY y XZ. El visualizador también apoya al usuario con un verificador incorporado que mide la exactitud de los gestos, comparándolo con el contenido del corpus generado.

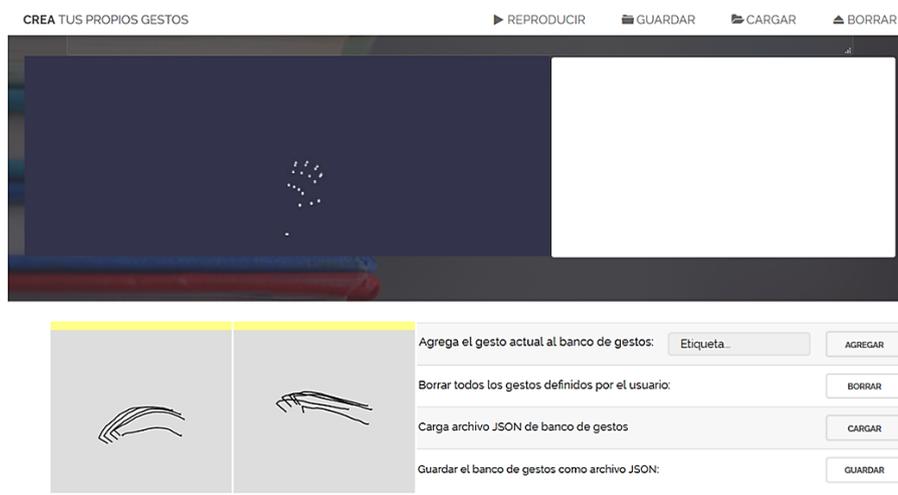


Fig. 5. Entorno de Desarrollo de CREA

Para crear un archivo JSON que represente el corpus de gestos, el usuario deberá de generar un gesto y después etiquetarlo con un nombre. Se puede repetir el registro del gesto y el proceso de etiquetación las veces que se desee. El archivo JSON resultante

se guarda en la computadora y este puede ser utilizado dentro del entorno JUEGA como parte de la interfaz de interacción, la cual implementa el reconocimiento de los gestos.

4.2. JUEGA y diviértete

JUEGA es el entorno interactivo que implementa una interfaz de interacción mediante reconocimiento de gestos y una interfaz visual orientada a la programación mediante bloques (ver figura 6).

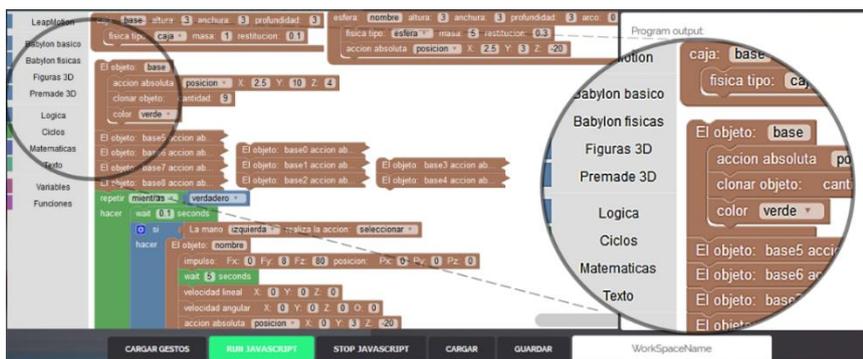


Fig. 6. Entorno de programación visual con Blockly.

A partir de los bloques, el usuario final construye elementos personalizados dentro del entorno, los cuales poseen físicas y pueden interactuar entre sí. También hay bloques para programación de la lógica y control de eventos que implementan la interfaz de interacción. Esto con el fin de que el usuario mediante gestos manipule e interactúe con lo que ocurre en el entorno 3D utilizando directamente sus manos de una forma más kinestésica, o bien utilizando el sistema de reconocimiento de gestos y relacionarlos a eventos o acciones en pantalla, a partir de la perspectiva de un avatar (ver figura 7).

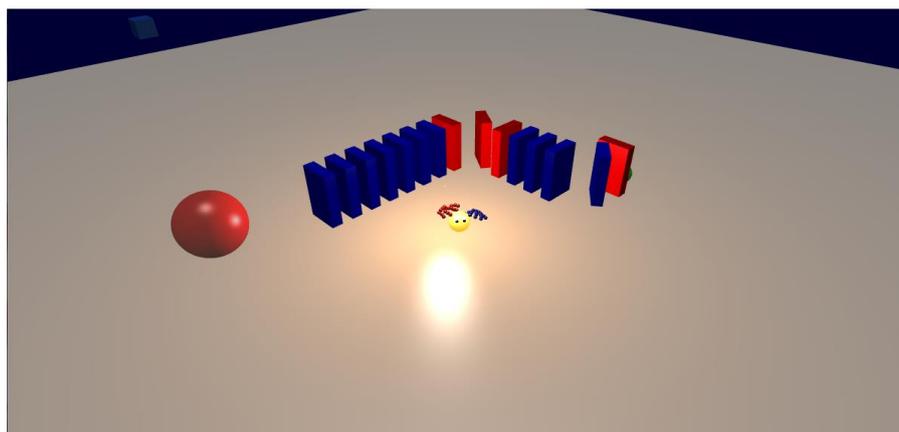


Fig. 7. Escena 3D con interacción kinestésica desde un avatar.

reconocimiento de gestos, por lo que CREA y JUEGA pueden ser utilizados en conjunto.

Respecto al ejercicio de programar un entorno interactivo, se puede dividir en 3 aspectos:

Creación de constructores de elemento 3D. Se utilizan bloques para la construcción de figuras geométricas con sus respectivas dimensiones. Se crea instancias de figuras 3D y se les asigna un nombre.

Personalización de elementos 3D y del entorno. Ya sea durante o después del proceso de construcción, se usan bloques que permiten personalizar el aspecto y las propiedades físicas de una figura. Permiten definir el comportamiento del objeto a partir del sistema simulador de físicas. Para que la gravedad simulada haga efecto, por ejemplo, se necesita que el objeto se le asigne una masa.

Creación del comportamiento lógico de entorno mediante eventos. Se realiza la programación de la lógica de eventos y las interacciones que pueden ocurrir por las acciones del usuario. Se programa a partir de bloques de secuencias y funciones lógicas, permite crear animaciones en el entorno, o programar eventos que utilicen la interfaz de interacción e implementar reconocimiento de gestos.

5. Evaluación de diseño centrado en el humano

El diseño centrado en el humano (DCH) tiene como objetivo el diseñar productos interactivos que sean fáciles, efectivos y disfrutables de usar, así como óptimos de la interacción del usuario con un sistema y su entorno. La literatura en DCH ha propuesto abstracciones acerca de los aspectos o principios de diseño como guías de lo que se debe y no se debe hacer al diseñar un sistema [18,19].

Por ello, nuestra investigación evalúa las interfaces propuestas de acuerdo a los siguientes principios básicos de interacción a partir de una matriz ponderada. Este método de evaluación tiene una larga tradición en la ingeniería de software y en la literatura de sistemas de información [20].

Visibilidad. Hacer visibles las interacciones de los humanos con la computadora en la medida de lo posible. Mientras más visibles sean las funciones, más probable será que los humanos realicen la acción apropiada.

Retroalimentación. Proporcionar al usuario información inmediata acerca de la acción que se está ejecutando o que se acaba de ejecutar. En DCH existen diferentes tipos de retroalimentación que involucran el uso de sonidos, animaciones, vibraciones y combinaciones de dichos estímulos sensoriales.

Restricciones. Es limitar las opciones que los usuarios pueden elegir al ejecutar una acción con la finalidad de que el usuario no cometa errores, ofreciendo solo opciones relevantes.

Consistencia. Utilizar operaciones o elementos similares para tareas similares, ya que esto permite que los sistemas sean más fáciles de utilizar porque el usuario aprende un camino aplicable a varios objetos.

Asequibilidad. Indicar o dar pistas acerca de las acciones que se deben de realizar sobre un objeto. Este principio de diseño se ha utilizado mayormente en el diseño de objetos

físicos ya que el mapeo es natural e involucra el uso de metáforas basadas en interacciones reales.

Para la evaluación, se utiliza la siguiente escala de calificación a partir del cumplimiento de los aspectos o principios mencionados anteriormente.

Si cumple: +2 puntos.

Parcialmente cumple: +1 punto.

No cumple: 0 puntos.

Tabla 1. Resultados de evaluación de aspectos de las interfaces del prototipo.

Aspectos	Criterios			
	GUI		NUI	
	Entorno gráfico interactivo 3D	Interfaz de programación grafica por Bloques	Interfaz de interacción kinestésica 3D	Interfaz de interacción por reconocimiento de gestos.
Visibilidad	2	2	2	2
Retroalimentación	2	2	2	1
Restricciones	2	1	1	1
Consistencia	2	2	2	2
Asequibilidad	2	1	2	2

De acuerdo con los resultados de evaluación mostrados en la tabla 1, la **visibilidad** se cumple en las NUI y GUI de la aplicación, el uso de interfaces graficas 3D permiten al usuario observar con claridad el comportamiento de la aplicación con sus acciones. La representación de las manos en 3D y el avatar ayudan a la orientación visual de donde se encuentra el usuario dentro del entorno interactivo, brindan asistencia de cómo actuar en determinadas circunstancias (+2 puntos). La GUI de programación por bloques cumple este aspecto, presenta bloques para la codificación clasificados por tipo en una lista de forma organizada (+2 puntos).

El aspecto de la **retroalimentación** se cumple en la mayoría de las interfaces de la aplicación. En conjunto las NUIs y el entorno interactivo 3D retroalimentan al usuario de sus acciones, a partir de la representación de un modelo 3D de manos en tiempo real (+2 puntos). Por otra parte, la interfaz de reconocimiento de gestos no posee ninguna forma de representación visual, depende de la activación de eventos para producir retroalimentación al usuario (+1 punto). La GUI de programación por bloques, reproduce efectos de audio y retroalimentación visual cuando se conectan los bloques, e indicaciones cuando se está ejecutando código o existe algún error, por lo que cumple con este aspecto (+2 puntos).

Las **restricciones** son limitadas en las interfaces naturales, puesto que su objetivo es el de ofrecer libertad creativa para crear e interactuar, esto ocasiona que no se definan con exactitud las posibilidades de uso de la aplicación (+1 punto). Sin embargo, existen excepciones en la GUI del entorno interactivo 3D, donde su propósito es el de delimitar las dimensiones del escenario y las áreas en las que el usuario puede llegar a interactuar desde la perspectiva de un avatar (+2 puntos).

La **consistencia** se cumple dentro de las GUI, donde el mismo proceso de codificación de escenarios y los eventos es realizado sin importar el tipo de ejercicio, utilizando botones y funciones vistas en otras GUI fáciles de familiarizar como guardar, reproducir, editar y borrar. El usuario puede personalizar como interactúa con el entorno mediante las NUI, pero el proceso de creación e implementación de kinestésica y gestos es natural y siempre es el mismo (+2 puntos).

Las NUI cumplen con el aspecto de la **asequibilidad**, ya que la implantación de reconocimientos de gestos, así como la representación de la mano dentro del entorno interactivo, son por naturaleza, mecanismos de representación de como el humano interactúa con el entorno (+2 puntos). La GUI de programación por bloques, por otra parte, representa metafóricamente a la programación como la unión lógica de distintos elementos relacionados entre sí, los bloques poseen señalamientos de que se pueden unir o ser insertados uno sobre el otro. Sin embargo, no todos los usuarios poseen la capacidad de abstracción para este tipo de conceptos, ocasionando ciertas dificultades de uso (+1 punto).

6. Conclusiones y trabajos futuros

El desarrollo del pensamiento computacional y la búsqueda de mejores formas de educar a las nuevas generaciones sigue siendo una tarea demandante. En este artículo, se presenta un enfoque de las implementaciones de interfaces humano-máquina naturales para el aprendizaje de lógica algorítmica y programación. Aplicando evaluaciones de diseño centrado en el humano, se demuestra que las NUI y GUI presentadas en este proyecto cumplen con los principios básicos de interacción natural, sin embargo, el proceso de DCH es altamente empírico, donde la toma de decisiones es basada en el conocimiento que se tiene de los usuarios y del contexto en el que se utilizará un producto. Como trabajo futuro se planea realizar evaluaciones con grupos de estudiantes de nivel secundaria, para analizar mejor el impacto de la herramienta y medir aspectos de usabilidad, motivación y satisfacción que se experimenta con el uso de este tipo de interfaces.

La aplicación fue desarrollada en JavaScript, utilizando la API de Leap Motion para la recepción de los datos de entrada. La interfaz de programación visual se implementó con Blockly. Para la presentación del entorno gráfico, se utilizó un framework de gráficos 3D para aplicaciones web llamada Babylon JS [21], componente principal para la creación de escenarios tridimensionales, así como la representación de manos humanas dentro del entorno.

Referencias

1. Grover, S., Pea, R.: Computational thinking in K-12: A review of the state of the field. *Educ. Res.* 42(1), 38-43 (2013)
2. Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S., Carey, T.: *Human-Computer Interaction*. Essex, UK, UK: Addison-Wesley Longman Ltd. (1994)
3. Lin, W., Du, L., Harris-adamson, C.: *Human-Computer Interaction. User Interface Design, Development and Multimodality*, *Human-Computer Interact. User Interface Des. Dev. Multimodality*, vol. 10271, no. December, pp. 584-592 (2017)
4. Wigdor, D., Wixon, D.: *Brave NUI world: designing natural user interfaces for touch and gesture*. Elsevier (2011)
5. Cabero, J.: Bases pedagógicas del e-learning. *Rev. Univ. y Soc. del Conoc.*, vol. 3, pp. 1-10 (2006)
6. Wenger, E.: Artificial intelligence and tutoring systems: computational and cognitive approaches to the communication of knowledge. *Morgan Kaufmann* (2014)

7. Lin, H.-C. K., Wu, C.-H., Hsueh, Y.-P.: The influence of using affective tutoring system in accounting remedial instruction on learning performance and usability. *Comput. Human Behav.*, vol. 41, pp. 514–522 (2014)
8. Shneiderman, B., Plaisant, C.: *Diseño de interfaces de usuario, Estrategias para una Interacción Pers. Ef. Cuarta Edición* ed Pearson Educ. (2006)
9. Resnick, M. et al.: Scratch: Programming for all. *Commun. Acm.* 52(11), 60–67 (2009)
10. Werner, L., Campe, S., Denner, J.: Children learning computer science concepts via Alice game-programming, p. 427 (2012)
11. Bowe, S.J., Antoniou, M.N., Garrett, C., Huber, B., Kaufman, J., Tarasuik, J.: Young children's transfer of learning from a touchscreen device. *Comput. Human Behav.*, vol. 56, pp. 56–64 (2015)
12. Ebner, M., Spot, N.: Game-Based Learning with the Leap Motion Controller. No. October, pp. 555–565 (2015)
13. Zhu, G., Cai, S., Ma, Y., Liu, E.: A Series of leap motion-based matching games for enhancing the fine motor skills of children with autism. In: *Proc. - IEEE 15th Int. Conf. Adv. Learn. Technol. Adv. Technol. Support. Open Access to Form. Informal Learn. ICALT 2015*, pp. 430–431 (2015)
14. Tran, V.T., Lee, J., Kim, D., Jeong, Y.S.: Easy-to-use virtual brick manipulation techniques using hand gestures. *J. Supercomput.* 72(7), 2752–2766 (2016)
15. Vatavu, R.-D., Anthony, L., Wobbrock, J. O.: Gestures as point clouds, p. 273 (2012)
16. Fraser, N. (et. al.): Blockly: A visual programming editor, URL <https://code.google.com/p/blockly>, (2013). Último acceso: 2019/05/11
17. Lieberman, H.: *Your wish is my command: Programming by example*. Morgan Kaufmann, (2001)
18. Norman, D.: *The design of everyday things: Revised and expanded edition*. Basic books (2013)
19. Castro, L.A., Rodríguez, M.D.: *A. M. D. E. Computación, Interacción Humano-Computadora y Aplicaciones en México*, Primera ed. Academia Mexicana de Computación, A.C. (2018)
20. Wieringa, R., Maiden, N., Mead, N., Rolland, C.: Requirements engineering paper classification and evaluation criteria: A proposal and a discussion. *Requir. Eng.* 11(1), 102–107 (2006)
21. Catuhe, D., Rousseau, M., Lagarde, P., Rousset, D.: *Babylon.js a 3D engine based on WebGL and JavaScript* (2014)

Pronóstico de la velocidad del viento mediante técnicas de Inteligencia Artificial

Juan Ramón V. Barraza¹, Hector Rodriguez R.¹, Víctor González Huitrón¹,
Carlos A. Lara Alvarez³, Juan J. Flores²

¹ Instituto Tecnológico de Culiacán, División de Estudios de Posgrado e Investigación, Culiacán, Sinaloa, México

² Universidad Michoacana de San Nicolás de Hidalgo, Facultad de Ingeniería Eléctrica, División de Estudios de Posgrado, Morelia, Michoacán, México

³ Centro de Investigación en Matemáticas, A.C. Unidad Zacatecas, Zacatecas, México
juan.valenzuela@itculiacan.edu.mx, hrodriguez@itculiacan.edu.mx,
victor.gonzalez@conacyt.mx, carlos.lara@cimat.mx, juanf@umich.mx

Resumen. Realizar pronósticos de múltiples pasos en el tiempo no es una tarea sencilla; por lo tanto, es importante desarrollar y usar técnicas que simplifiquen esta tarea. Este trabajo prueba modelos de Inteligencia Artificial para tres diferentes estrategias de pronósticos a múltiples pasos: *Iterativa*, *Directa*, y *Multiple Input - Multiple Output (MIMO)*. El primer paso consiste en preprocesar las series de tiempo para *limpiar* (eliminar datos atípicos y ruido) y *transformar* los datos en secuencias o imágenes; después, se usan tres tipos de *Artificial Neural Networks (ANN)* para construir modelos de pronóstico: *Multilayer Perceptron (MLP)*, *Long Short Term Memory (LSTM)* y *Convolutional Neural Networks (CNN)*. Se muestran experimentos con series de tiempo obtenidas en estaciones climáticas en Michoacán, México, donde se puede observar el desempeño que se obtuvo con cada modelo y estrategia.

Palabras clave: inteligencia artificial, pronóstico, velocidad del viento.

Wind Speed Forecast Using Artificial Intelligence Techniques

Abstract. *N*-steps forecasting is not an easy task; it is important to develop and use techniques that simplify this task. This paper tests Artificial Intelligence models for three different multi-step forecasting strategies: *Iterative*, *Direct*, and *Multiple Input - Multiple Output (MIMO)*. The first step preprocesses the time-series to *clean* (remove outliers and noise) and *transform* the data into sequences or images; after that, three types of Artificial Neural Networks (ANN) are used to construct forecasting models: *Multilayer Perceptron (MLP)*, *Long Short Term Memory (LSTM)* and *Convolutional Neural Networks (CNN)*. Experiments that use data generated from climate stations in Michoacan, Mexico are shown, where the performance obtained with each model and strategy can be observed.

Keywords: artificial intelligence, forecasting, wind speed.

1. Introducción

En la actualidad la energía eléctrica es la base para muchas de las actividades realizadas por la humanidad; sin embargo, su producción puede utilizar recursos no renovables y deteriorar el medio ambiente. Durante el año 2017 la generación de energías limpias en México fue del 21.08 % del cual solo el 3.23 % es energía eólica [1].

Aunque la energía eólica tiene muchas ventajas, la mayor dificultad es que la velocidad del viento no es constante a través del tiempo. Por este motivo, es muy importante pronosticar la velocidad del viento —y en consecuencia, conocer la cantidad de energía limpia que se puede producir. Sin embargo, la tarea de pronosticar la velocidad del viento no es una tarea sencilla, especialmente cuando se desea pronosticar un horizonte amplio.

Los modelos de pronóstico que usan Inteligencia Artificial se han popularizado en los últimos años porque tienen mejor desempeño que los métodos estadísticos clásicos [2]. El diseño de estos modelos es importante porque afecta drásticamente su velocidad y eficiencia [4].

El objetivo de este trabajo es comparar y presentar las fortalezas y desventajas de tres tipos de *Artificial Neural Networks* (ANN) —*Multilayer Perceptron*, *Long Short Term Memory* (LSTM) y *Convolutional Neural Network* (CNN)— para pronóstico directo, recursivo y MIMO. Además se explica el preprocesamiento (limpieza y transformación) de series de tiempo para su uso en modelos de pronóstico. En especial, se describe la técnica *Gramian Angular Field* que se usa para transformar series de tiempo a imágenes, lo que permite analizar las series con una CNN.

El resto de este trabajo se estructura de la siguiente forma: la Sección 2, presenta trabajos relacionados e implementaciones similares, la Sección 3 explica los modelos de IA utilizados, la Sección 4 detalla la metodología utilizada, la Sección 5 muestra los resultados obtenidos y Sección 6 concluye este trabajo.

2. Trabajos relacionados

El pronóstico de series de tiempo es de creciente interés en muchas áreas de la ciencia e ingeniería —v.g., economía, finanzas, meteorología y telecomunicaciones. El pronóstico de viento a múltiples pasos consiste en pronosticar un rango más amplio de tiempo con intervalo definido. Los problemas de pronóstico a múltiples pasos se pueden clasificar en: super-corto, corto, mediano, y largo plazo según sea su horizonte de tiempo [2]. A diferencia de pronosticar el paso siguiente en el tiempo, los pronósticos con un horizonte más amplio son más difíciles porque el error y la incertidumbre se acumulan, lo que reduce la precisión [4].

Además, la estrategia que use un pronosticador es de vital importancia, ya que afecta el costo computacional y precisión del pronosticador.

Los modelos de aprendizaje de máquina se han popularizado en las últimas décadas como serios contendientes de los modelos de pronóstico estadísticos clásicos [5]. Entre los modelos de aprendizaje de máquina se encuentran los modelos de aprendizaje profundo, como lo son las *Artificial Neural Networks: Multilayer Perceptron* (MLP), *Long Short Term Memory* (LSTM) y *Convolutional Neural Network* (CNN).

Pasero & Ruffa [6] implementan un enfoque de pronóstico a largo plazo con un modelo MLP y una estrategia recursiva; sin embargo, sus resultados muestran que un modelo de *Support Vector Machine* (SVM) con estrategia directa consigue mejores resultados. Por otro lado, las LSTM tienen la característica de poder procesar grandes cantidades de datos y poseen la capacidad de generalización; es decir, se adaptan a datos no conocidos. Con mejores resultados que los modelos basados en SVM [7].

Las CNN son redes ampliamente conocidas por su uso en clasificación de imágenes. Estas al igual que las LSTM tienen la capacidad de procesar grandes cantidades de información, Anwen, et al. [8] utiliza CNN para realizar pronósticos de viento para múltiples pasos.

Los pronósticos a múltiples pasos que se obtienen con la estrategia iterativa tienen la desventaja de deteriorarse al acumular errores, ya que solamente se entrena un modelo a un paso; sin embargo, su resultado es comparable al trabajar un horizonte limitado [9].

Una solución para evitar la deterioración al acumular errores tal como en la estrategia iterativa es entrenar un modelo por cada paso en el horizonte de tiempo como se propone en la estrategia directa [4]. Quin, et al. [10] muestran el uso de la estrategia directa en conjunto para pronósticos de la velocidad del viento a cuatro horas.

Otra alternativa es la estrategia MIMO que se caracteriza por ser un solo modelo que predice múltiples pasos en el tiempo [4]. Li, et al. [11] comparan las estrategias iterativa y MIMO, para obtener el pronóstico de velocidad de viento a corto plazo donde MIMO obtiene mejores resultados.

3. Modelos de Inteligencia Artificial

Esta sección revisa los conceptos fundamentales de algunos modelos de inteligencia artificial que se utilizan en este trabajo para realizar pronósticos de la velocidad del viento.

3.1. Artificial Neural Network

El *Multilayer Perceptron* (MLP) es uno de los tipos de redes más utilizados, el cual típicamente se conforma de múltiples capas y nodos [3]. La figura 1 muestra la arquitectura de un MLP, donde sus entradas son valores históricos, y como salida el pronóstico del viento.

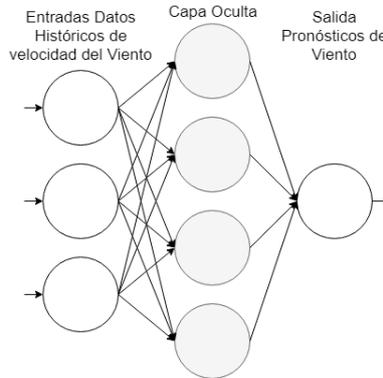


Fig. 1. Diagrama representación de Red Neuronal Artificial MLP

3.2. Long Short Term Memory

Las Redes Neuronales LSTM son muy similares a las MLP en su estructura, tienen capas de entrada, capas ocultas y capa de salida. Sin embargo, LSTM en su capa oculta poseen una unidad de memoria [7].

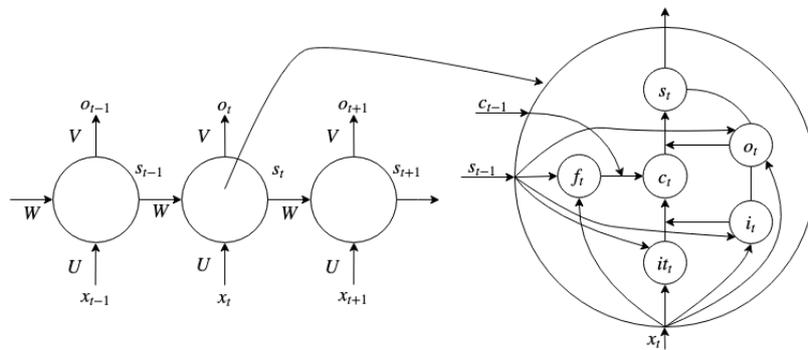


Fig. 2. Diagrama de funcionamiento de Long Short Term Memory

La unidad de memoria consiste de tres compuertas (*Input Gate* (i_t), *Forget Gate* (f_t) y *Output Gate* (o_t)) y una conexión recurrente. La unidad tiene una entrada x_t y dos retroalimentaciones de estados previos de la unidad que son la salida del estado previo s_{t-1} y la variable de estado c_{t-1} . Las compuertas utilizan una función de activación sigmoide g , mientras que los estados utilizan una función \tanh . La unidad de memoria de una LSTM se puede definir por las siguientes expresiones:

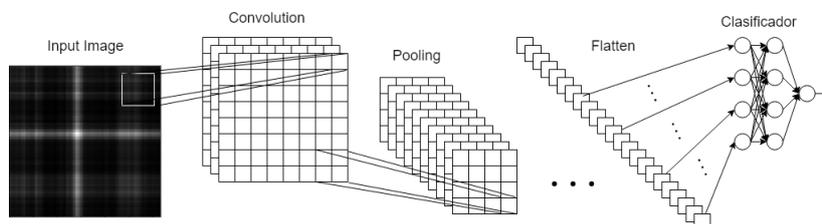


Fig. 3. Diagrama de funcionamiento de una Red Neuronal Convolucionada

$$\begin{aligned}
 \text{Input Gate} & \quad i_t = g(W_{xi}x_t + W_{hi}s_{t-1} + b_i), \\
 \text{Forget Gate} & \quad f_t = g(W_{xf}x_t + W_{hf}s_{t-1} + b_f), \\
 \text{Output Gate} & \quad o_t = g(W_{xo}x_t + W_{ho}s_{t-1} + b_o), \\
 \text{Input Transform} & \quad it_t = \tanh(W_{xc}x_t + W_{hc}s_{t-1} + b_{it_t}), \\
 \text{State Update} & \quad c_t = f_t \cdot c_{t-1} + i_t \cdot it_t, \\
 \text{Output} & \quad s_t = o_t \cdot \tanh(c_t),
 \end{aligned} \tag{1}$$

donde W son parámetros y b es el sesgo [7].

3.3. Redes neuronales convolucionales

Las Redes Neuronales Convolucionales son capaces de extraer características de imágenes de dos dimensiones [8]. Estas redes tienen cuatro etapas principales: Convolución, agrupación (pooling), aplanado de datos (flatten) y clasificador. En la figura 3 se muestra el proceso generalizado que realiza una CNN.

La etapa de *Convolución* consiste en extraer características a partir de recorridos en la imagen, mientras que el *pooling* realiza un sub-muestreo para reducir el costo computacional requerido al disminuir las dimensiones y parámetros previamente extraídos durante una convolución. Posteriormente se tiene la etapa de *flatten* que aplanan los datos a un vector para finalmente alimentar las características extraídas a un clasificador.

4. Metodología

La figura 4 muestra el proceso general utilizado en este trabajo, la entrada es una serie de tiempo y la salida es un modelo de predicción del viento para n pasos futuros. El proceso consiste de cuatro etapas principales: (i) selección de estrategia, (ii) selección de modelo de IA, (iii) preprocesamiento de Datos, y (iv) entrenamiento y ajuste de modelo.

4.1. Estrategias de predicción múltiples pasos en el tiempo

A diferencia del pronóstico del siguiente valor de una serie de tiempo, el pronóstico de múltiples pasos en el tiempo es considerablemente más difícil

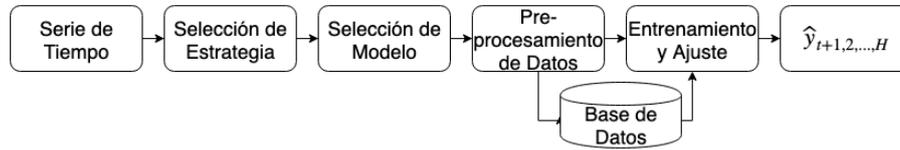


Fig. 4. Diagrama del proceso general de Serie de Tiempo a modelo predictivo con n pasos a futuro

debido a la acumulación de errores, reducción de precisión, e incremento de incertidumbre [3]. Por este motivo, se han propuesto diferentes estrategias para mejorar la aptitud de los pronósticos. Taieb et al. [4] hacen una revisión de las técnicas conocidas. En este trabajo se utilizan tres estrategias: Iterativa, Directa y MIMO.

Estrategia Iterativa. La Estrategia Iterativa consiste en entrenar un modelo f para un solo paso en el tiempo [4]; es decir,

$$y_{t+1} = f(y_t, \dots, y_{t-d+1}) + r, \quad (2)$$

donde $t \in \{d, \dots, N - 1\}$. Cuando se busca pronosticar H pasos en el futuro, primero se pronostica un paso aplicando el modelo f . El valor obtenido en el primer pronóstico se alimenta al modelo para pronosticar el siguiente paso en el tiempo (figura 5), repitiendo el proceso hasta predecir el horizonte H completo [3].

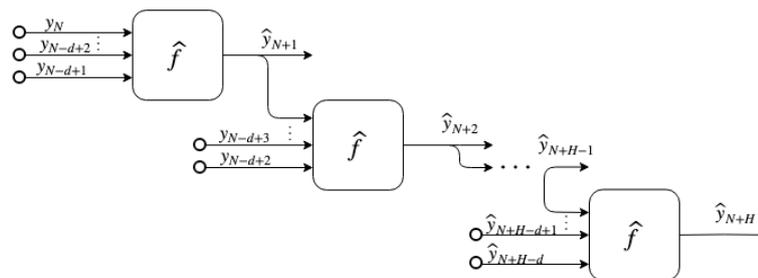


Fig. 5. Estrategia Iterativa para Pronóstico de Series de Tiempo para H pasos a futuro.

En (3) se observa que \hat{y} se estima a partir de entradas de la serie y valores estimados. Cuando $h = 1$, el modelo solo usa valores de la serie para pronosticar un solo paso. Si $2 \geq h \geq d$, el modelo se alimenta de valores estimados y de la serie de tiempo; finalmente, cuando $h > d$, el modelo se alimenta solo de valores pronosticados.

Algoritmo 1 Algoritmo para Estrategia Iterativa

```

1: procedure ITERATIVE_STRATEGY(model, data, horizon, window_length)
2:   queue = data
3:   predictions = []
4:   for h = 0; 0 ≤ horizon; h = h + 1 do
5:     current = queue[-window_length :]
6:     predicted = model.predict(current)
7:     queue = queue.append(predicted)
8:     predictions.add(predicted)
9:   end for
   return predictions
10: end procedure

```

$$\hat{y}_{N+h} = \begin{cases} \text{If } h = 1 & \hat{f}(y_N, \dots, y_{N-d}), \\ \text{If } h \in (2, \dots, d) & \hat{f}(\hat{y}_{N+h-1}, \dots, \hat{y}_{N+1}, \dots, y_{N-d+h}), \\ \text{If } h \in (d+1, \dots, H) & \hat{f}(\hat{y}_{N+h-1}, \dots, \hat{y}_{N+h-d}). \end{cases} \quad (3)$$

La estrategia iterativa sufre de la acumulación de errores por cada paso que recorre para $h > 1$ hasta llegar al horizonte máximo H . Esto es especialmente cierto cuando $h > d$, como plantea el tercer caso de (3), donde el modelo comienza a ser alimentado con solamente valores previamente pronosticados [4].

El Alg. 1 muestra la secuencia lógica utilizada en este trabajo para realizar pronósticos con la estrategia iterativa. El algoritmo recibe el modelo (*model*), los datos de entrada (*data*), el horizonte H (*horizon*) y el tamaño de ventana de datos (*window_length*). El proceso consiste en una cola (*queue*) iniciando con los datos (*data*) como punto inicial. Posteriormente, se inicia un ciclo For desde $h = 0$ hasta el horizonte H , avanzando un solo paso por ciclo. Finalmente, se realiza pronóstico con datos en cola y se actualiza.

Estrategia Directa. Se caracteriza por pronosticar cada paso del horizonte en tiempo de manera independiente [4]. Es decir, si se tiene un horizonte H , se debe entrenar la misma cantidad de modelos para cada paso de tiempo que conforma el horizonte completo. Sea su representación dada por:

$$y_{t+h} = f_h(y_t, \dots, y_{t-d+1}) + r, \quad (4)$$

donde $t \in (d, \dots, N - H)$, $h \in (1, \dots, H)$ y la constante r representa el ruido externo que pueda tener la serie de tiempo.

La Estrategia Directa (figura 6) presenta la ventaja de no utilizar valores aproximados ya que no retroalimenta datos anteriormente pronosticados, además de poseer la capacidad de ajustar cada modelo pronosticador individualmente.

En Alg. 2 se puede observar la lógica de programación utilizada para realizar pronósticos con la estrategia directa. La lógica utilizada es muy similar a la

Algoritmo 2 Algoritmo para Estrategia Directa

```

1: procedure DIRECT_STRATEGY(models, data, horizon)
2:   predictions = []
3:   for  $h = 0; 0 \leq \textit{horizon}; h = h + 1$  do
4:     model = models[ $h$ ]
5:     predicted = model.predict(data)
6:     predictions.add(predicted)
7:   end for
   return predictions
8: end procedure

```

utilizada por la estrategia iterativa, siendo la mayor diferencia un recorrido realizado a una colección que contiene los modelos para realizar cada pronóstico.

A pesar de las ventajas que presenta la estrategia directa, el costo computacional requerido para el entrenamiento individual y el ajuste de los parámetros de cada modelo es una gran desventaja, por lo que se debe considerar al comparar con otras estrategias.

Estrategia MIMO. La Estrategia MIMO (Multiple Input - Multiple Output) posee múltiples salidas, cada una de las cuales corresponde a un horizonte distinto. Por lo anterior, la estrategia MIMO emplea solamente un modelo F para las series de tiempo $[y_t, \dots, y_N]$ presentes en $[y_H, \dots, y_{t+1}] = F(y_t, \dots, y_{t-d+1}) + r$ donde $t \in \{d, \dots, N - H\}$, $F : \mathbb{R}^d \rightarrow \mathbb{R}^H$ es un modelo vectorial y $r \in \mathbb{R}^H$ es un vector de ruido, como define Taieb et al. [4]. El pronóstico obtenido por un modelo utilizando la Estrategia MIMO se expresa como:

$$[\hat{y}_{t+H}, \dots, \hat{y}_{t+1}] = \hat{F}(y_N, \dots, y_{N-d+1}), \quad (5)$$

donde N es el paso de tiempo del dato y d es el tamaño de ventana utilizado.

La figura 7 se observa la representación estructural para el modelo MIMO, donde se muestran las entradas y salidas como se mostraron en (5).

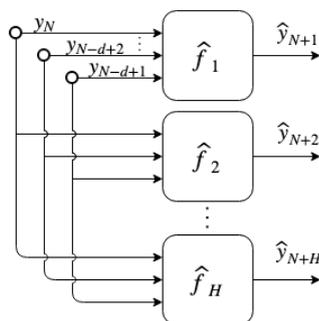


Fig. 6. Estrategia Directa para pronósticos de series de tiempo

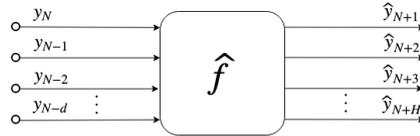


Fig. 7. Estructura de Estrategia MIMO para pronósticos de series de tiempo para H horizonte

4.2. Selección de modelo

Durante la selección de modelo se utilizaron los modelos de *Artificial Neural Networks*: MLP, CNN y LSTM como modelos base, buscando combinar las fortalezas de cada uno de ellos. Las combinaciones realizadas para este trabajo son: MLP, LSTM, LSTM-MLP, CNN-MLP, CNN-LSTM y CNN-LSTM-MLP.

4.3. Preprocesamiento de datos

El objetivo del preprocesamiento es acondicionar los datos para su uso. Dos procesos se usan en esta etapa: **limpieza** y **transformación** de datos.

Limpieza de Datos. El proceso de Limpieza de Datos consiste en explorar la serie de tiempo para buscar datos atípicos, los cuales pueden impactar de forma negativa el proceso y obtener resultados con un error mayor [14]. Este proceso tiene como finalidad reducir el ruido en los datos, e identificar y remover datos atípicos, para este trabajo se utilizan las técnicas de Normalización de Datos y Media Móvil.

Normalización de Datos. El objetivo de la Normalización de Datos es garantizar que los rangos de los valores existentes dentro de la serie de tiempo no sea muy amplio. Esto ayuda a que el proceso de entrenamiento sea más rápido, además de ayudar a reducir el posible sesgo [13]. Para obtener una normalización en el rango $[0, 1]$ se puede usar la expresión

$$N_i = \frac{n_i - \min_n}{\max_n - \min_n}, \quad (6)$$

donde N_i es el valor normalizado obtenido, n_i es el valor original de la serie a normalizar, \min_n es el valor mínimo dentro de la serie y \max_n es el valor máximo.

Media Móvil. La Media Móvil es una función para suavizar o filtrar los datos, ya que reemplaza cada observación y_t con una combinación lineal de los otros datos que están cercanos en el tiempo, tal como lo explica Douglas, et al [12]. Una Media Móvil simple de rango N asigna pesos $1/N$ para las N observaciones mas recientes $Y_T, Y_{T-1}, \dots, Y_{T-N+1}$ y asigna cero a todas las demás observaciones. Lo anterior se puede definir por la expresión:

$$M_T = \frac{Y_T, Y_{T-1}, \dots, Y_{T-N+1}}{N} = \frac{1}{N} \sum_{t=T-N+1}^T y_t, \tag{7}$$

donde M_T es la observación nueva obtenida, Y las observaciones cercanas y N el rango de las observaciones.

Transformación de datos. En el proceso de transformación se obtienen los datos en el formato adecuado para el proceso de entrenamiento del modelo pronosticador. En este trabajo se usan dos formatos, secuencias e imágenes.

Transformación de Serie de Tiempo a Secuencias. La serie de tiempo se transforma en múltiples secuencias de entrenamiento. Como ilustra la figura 8, cada secuencia de entrenamiento tiene longitud $w + H$, con w datos de entrada y H datos esperados. La Tabla 1 muestra la representación de secuencias de una serie de tiempo $X = (x_1, x_2, \dots, x_n)$ y su correspondiente serie normalizada $\tilde{X} = (\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_n)$.

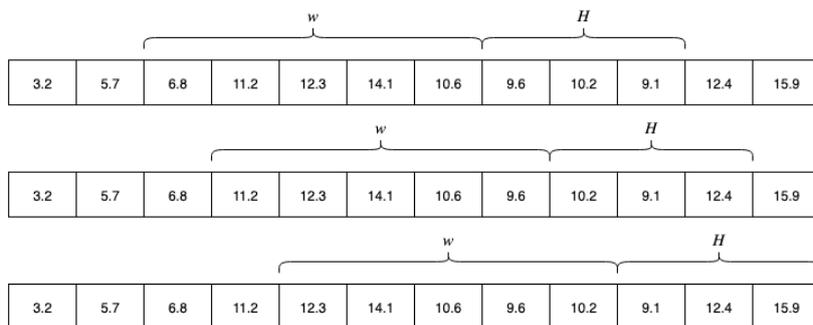


Fig. 8. Ejemplo de generación de secuencias a partir de una serie de tiempo

Tabla 1. Ejemplo de generación de secuencias

# Secuencia	w	Horizonte Esperado
1	$\tilde{x}_1, \tilde{x}_2, \tilde{x}_3, \tilde{x}_4, \dots, \tilde{x}_w$	$\tilde{x}_{w+1}, \dots, \tilde{x}_{w+H}$
2	$\tilde{x}_2, \tilde{x}_3, \tilde{x}_4, \tilde{x}_5, \dots, \tilde{x}_{w+1}$	$\tilde{x}_{w+2}, \dots, \tilde{x}_{w+H+1}$
3	$\tilde{x}_3, \tilde{x}_4, \tilde{x}_5, \tilde{x}_6, \dots, \tilde{x}_{w+2}$	$\tilde{x}_{w+3}, \dots, \tilde{x}_{w+H+2}$
4	$\tilde{x}_4, \tilde{x}_5, \tilde{x}_6, \tilde{x}_7, \dots, \tilde{x}_{w+3}$	$\tilde{x}_{w+4}, \dots, \tilde{x}_{w+H+3}$
5	$\tilde{x}_5, \tilde{x}_6, \tilde{x}_7, \tilde{x}_8, \dots, \tilde{x}_{w+4}$	$\tilde{x}_{w+5}, \dots, \tilde{x}_{w+H+4}$
\vdots	\vdots	\vdots
n	$\tilde{x}_n, \tilde{x}_{n+1}, \tilde{x}_{n+2}, \tilde{x}_{n+3}, \dots, \tilde{x}_{w+n}$	$\tilde{x}_{w+n+1}, \dots, \tilde{x}_{w+n+H+1}$

Transformación de de Serie de Tiempo a Imagen. Para obtener una imagen a partir de la serie de tiempo, se utilizó la técnica de *Gramian Angular Field* (GAF) [15]. Cada entrada de la serie se convierte en coordenadas polares con

$$\phi_i = \arccos(\tilde{x}_i), \tag{8}$$

$$r_i = \frac{t_i}{N}. \tag{9}$$

Para identificar la correlación temporal dentro de diferentes intervalos de tiempo se considera la suma trigonométrica entre cada punto [15]. Esto se codifica en la matriz de Gram:

$$G = \begin{bmatrix} \cos(\phi_1 + \phi_1) & \dots & \cos(\phi_1 + \phi_n) \\ \cos(\phi_2 + \phi_1) & \dots & \cos(\phi_2 + \phi_n) \\ \vdots & \ddots & \vdots \\ \cos(\phi_n + \phi_1) & \dots & \cos(\phi_n + \phi_n) \end{bmatrix}. \tag{10}$$

Entre otras propiedades, la matriz de Gram conserva la dependencia temporal de la serie de tiempo. Finalmente, la matriz se convierte en imagen en escala de gris con un escalamiento entre [0, 255]. La figura 9 muestra un ejemplo de una serie de tiempo de 64 datos que se convierte a imagen.

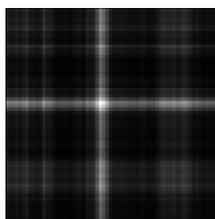


Fig. 9. Ejemplo de Serie de Tiempo a Imagen tamaño de ventana 64

Así como la transformación de series de tiempo a secuencias genera múltiples secuencias, la transformación de series de tiempo a imágenes genera múltiples imágenes. Cada imagen tiene asignado su horizonte (h) esperado. En la Tabla 2 se muestra un ejemplo de lógica que se obtiene al generar un banco de imágenes.

5. Resultados

Los datos utilizados para este trabajo fueron obtenidos por estaciones meteorológicas en distintos puntos de Michoacán, México. Para este trabajo se utilizaron datos de dos localidades diferentes: Aristeo Mercado, y Corrales. Estas series de tiempo contaban con un total de 20,000 datos en intervalos de una hora, de los cuales se utilizó el 30 % para validar el modelo.

Tabla 2. Tabla de ejemplo de generación de Imágenes a partir de secuencias de serie de tiempo

# Imagen	w					<i>Horizonte Esperado</i>		
1	\tilde{x}_1	\tilde{x}_2	\tilde{x}_3	\tilde{x}_4	\dots, \tilde{x}_w	\tilde{x}_{w+1}	\dots	\tilde{x}_{w+H}
2	\tilde{x}_2	\tilde{x}_3	\tilde{x}_4	\tilde{x}_5	\dots, \tilde{x}_{w+1}	\tilde{x}_{w+2}	\dots	\tilde{x}_{w+H+1}
3	\tilde{x}_3	\tilde{x}_4	\tilde{x}_5	\tilde{x}_6	\dots, \tilde{x}_{w+2}	\tilde{x}_{w+3}	\dots	\tilde{x}_{w+H+2}
4	\tilde{x}_4	\tilde{x}_5	\tilde{x}_6	\tilde{x}_7	\dots, \tilde{x}_{w+3}	\tilde{x}_{w+4}	\dots	\tilde{x}_{w+H+3}
5	\tilde{x}_5	\tilde{x}_6	\tilde{x}_7	\tilde{x}_8	\dots, \tilde{x}_{w+4}	\tilde{x}_{w+5}	\dots	\tilde{x}_{w+H+4}
\vdots			\vdots				\vdots	
n	\tilde{x}_n	\tilde{x}_{n+1}	\tilde{x}_{n+2}	\tilde{x}_{n+3}	\dots, \tilde{x}_{w+n}	\tilde{x}_{w+n+1}	\dots	$\tilde{x}_{w+n+H+1}$

Las métricas de error utilizadas son *Mean Squared Error* (MSE), *Mean Absolute Error* (MAE) y *Root Mean Squared Error* (RMSE):

$$\begin{aligned}
 MSE &= \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2, \\
 MAE &= \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|, \\
 RMSE &= \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}.
 \end{aligned}
 \tag{11}$$

La Tabla 3 muestra los resultados obtenidos con la serie de tiempo “Aristeo Mercado”. El modelo de IA que obtuvo mejores resultados fue LSTM y la mejor estrategia fue MIMO. En este caso, las LSTM muestran mejor desempeño; sin embargo, se debe considerar que pueden tomar mas tiempo para entrenar. Este factor se debe considerar para estrategias como la Directa donde afecta más el costo computacional.

Tabla 3. Resultados obtenidos con serie de tiempo Aristeo Mercado

Modelo	Estrategia Iterativa				Estrategia Directa				Estrategia MIMO			
	MSE	MAE	RMSE	Tiempo	MSE	MAE	RMSE	Tiempo	MSE	MAE	RMSE	Tiempo
MLP	0.004957	0.053168	0.07040	30.8s	0.004121	0.04845	0.06419	181.14s	0.003716	0.0446845	0.060962	43.9s
LSTM	0.004351	0.050203	0.065965	65.4s	0.003820	0.04543	0.06180	922.53s	0.003702	0.043297	0.060844	84.4s
LSTM-MLP	0.004664	0.051165	0.068295	48.05s	0.003983	0.04652	0.06311	1326.5s	0.003808	0.045964	0.061716	102.1s
CNN-MLP	0.005957	0.059158	0.077182	6.48s	0.004705	0.05031	0.06859	1187.3s	0.004705	0.050317	0.0686	87.9s
CNN-LSTM	0.006317	0.056700	0.079480	78.1s	0.004341	0.04781	0.06588	1438s	0.004443	0.048514	0.06666	134.3s
CNN-LSTM-MLP	0.005795	0.056109	0.076128	60.2s	0.005247	0.05430	0.07243	1328s	0.003989	0.043932	0.063151	219.8s

La Tabla 4 muestra los resultados obtenidos para la serie de tiempo “Corrales”. LSTM con la estrategia MIMO obtienen mejor resultado, seguidos por MLP con estrategia Directa.

En la figura 10 se muestran las gráficas para pronóstico de 24 horas utilizando el mejor modelo de cada serie respectivamente.

Tabla 4. Resultados obtenidos con serie de tiempo Corrales

Modelo	Estrategia Iterativa				Estrategia Directa				Estrategia MIMO			
	MSE	MAE	RMSE	Tiempo	MSE	MAE	RMSE	Tiempo	MSE	MAE	RMSE	Tiempo
MLP	0.004264	0.050060	0.065298	43.62s	0.003570	0.044231	0.059746	124.46s	0.003584	0.043696	0.059865	21.1s
LSTM	0.003963	0.047519	0.062952	36.76s	0.003574	0.044190	0.059782	727.48s	0.0032523	0.04004	0.05673	47.5s
LSTM-MLP	0.004433	0.050148	0.066583	58.34s	0.004392	0.049743	0.066275	1136.07s	0.00326	0.039836	0.05709	75.42s
CNN-MLP	0.005228	0.054755	0.072303	82.57s	0.006229	0.068313	0.078924	1282.26s	0.0038311	0.045463	0.061895	189.89s
CNN-LSTM	0.005272	0.055817	0.072612	156.77s	0.005935	0.068320	0.077042	946.42s	0.0038147	0.046528	0.061763	242.27s
CNN-LSTM-MLP	0.005144	0.054997	0.071724	64.99s	0.006137	0.069617	0.078342	918.9s	0.00542	0.057264	0.07362	137.91s

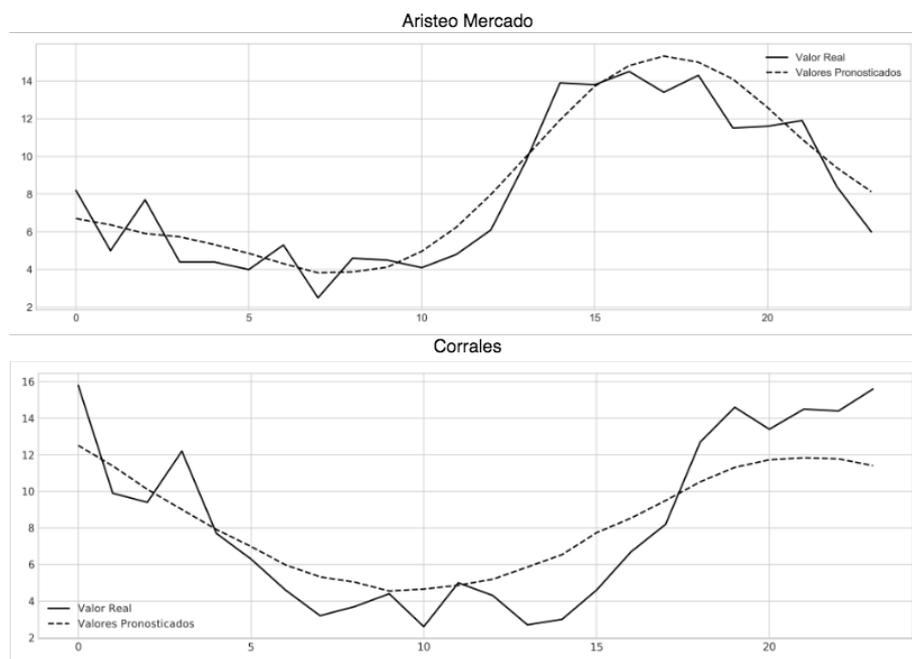


Fig. 10. Gráficas de pronóstico de 24 horas para los mejores resultados en “Aristeo Mercado” y “Corrales”)

La mayoría de los resultados con aptitud deseable fueron aquellos que utilizaron estrategia MIMO y modelo LSTM. Sin embargo, la estrategia Directa y los modelos combinados son más complejos y requieren ajustar más parámetros. Al tener más parámetros se pueden ajustar a diferentes situaciones; sin embargo, del ajuste de parámetros de forma manual requiere mucho tiempo.

6. Conclusiones

El pronóstico de la velocidad de viento a múltiples pasos es una tarea compleja. Sin embargo, utilizar las estrategias y modelos correctos puede facilitar esta tarea. En este trabajo se exploraron tres modelos de Inteligencia Artificial,

además de sus combinaciones y tres estrategias para pronósticos de múltiples pasos. Se observó que el modelo y estrategia con mejor aptitud fueron el modelo LSTM y MIMO, respectivamente. No obstante, usar modelos y estrategias más complejas —tales como la estrategia Directa y los modelos combinados— presenta más dificultad para afinar los modelos porque tienen una gran cantidad de parámetros. Como trabajo futuro se propone combinar el uso de técnicas para optimizar estos parámetros complementando las técnicas presentadas en este trabajo.

Referencias

1. Beltrán Rodríguez, L., Rodríguez Aguilar, J.S. (et. al.): Reporte de Avance de Energías Limpias. Secretaría de Energía (2017)
2. Soman, S.S., Zareipour, H., Malik, O., Mandal, P.: A review of wind power and wind speed forecasting methods with different time horizons. In: North American Power Symposium 2010, pp. 1–8 (2010)
3. An, N.H., Anh, D.T.: Comparison of strategies for multi-step-ahead prediction of time series using neural network. In: 2015 International Conference on Advanced Computing and Applications (ACOMP), pp. 142–149 (2015)
4. Taieb, S.B., Bontempi, G., Atiya, A.F., Sorjamaa, A.: A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition. *Expert systems with applications* 39(8), pp. 7067–7083 (2012)
5. Ahmed, N.K., Atiya, A.F., Gayar, N.E., El-Shishiny, H.: An empirical comparison of machine learning models for time series forecasting. *Econometric Reviews*, vol. 29, pp. 594–621 (2010)
6. Pasero, E., Raimondo, G., Ruffa, S.: MLP: a multi-layer perceptron application to long-term, out-of-sample time series prediction. In: International Symposium on Neural Networks, pp. 566–575. Springer, Berlin, Heidelberg (2010)
7. Xiaoyun, Q., Xiaoning, K., Chao, Z., Shuai, J., Xiuda, M.: Short-term prediction of wind power based on deep long short-term memory. In: 2016 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC), pp. 1148–1152 (2016)
8. Zhu, A., Li, X., Mo, Z., Wu, R.: Wind power prediction based on a convolutional neural network. In: 2017 International Conference on Circuits, Devices and Systems (ICCDs), pp. 131–135, IEEE (2017)
9. Ahmed, A., Khalid, M.: An intelligent framework for short-term multi-step wind speed forecasting based on Functional Networks. *Applied energy*, vol. 225, pp. 902–911 (2018)
10. Qin, Q., Lai, X., Zou, J.: Direct Multistep Wind Speed Forecasting Using LSTM Neural Network Combining EEMD and Fuzzy Entropy. *Applied Sciences*, vol. 9, pp. 126 (2019)
11. Li, Y., Shi, H., Han, F., Duan, Z., Liu, H.: Smart wind speed forecasting approach using various boosting algorithms, big multi-step forecasting strategy. *Renewable Energy*, vol. 135, pp. 540–553 (2019)
12. Montgomery, D.C., Jennings, C.L., Kulahci, M.: Introduction to time series analysis and forecasting. 2nd Ed. John Wiley & Sons (2015)
13. Nayak, S.C., Misra, B.B., Behera, H.S.: Impact of data normalization on stock index forecasting. *Int. J. Comp. Inf. Syst. Ind. Manag. Appl*, vol. 6, pp. 357–369 (2014)

14. Tsay, R.S.: Outliers, level shifts, and variance changes in time series. *Journal of forecasting*, vol. 7, pp. 1–20 (1988)
15. Wang, Z., Oates, T.: Encoding time series as images for visual inspection and classification using tiled convolutional neural networks. In: *Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence* (2015)

Diseño de una red neuronal distribuida entre dispositivos Raspberry Pi conectados a Internet por medio de XMPP

Alberto Martínez Contreras¹, David Tinoco Varela², Fernando Gudiño Peñaloza²

¹ Instituto Politécnico Nacional, ESIME Zacatenco, México

² Universidad Nacional Autónoma de México, Facultad de Estudios Superiores Cuautitlán, Departamento de Ingeniería, ITSE, México
phama_contra26@hotmail.com, dativa19@hotmail.com,
ilciarmin@gmail.com

Resumen. Actualmente se tienen diferentes esquemas de comunicación que buscan la interconexión de la mayor cantidad de dispositivos posibles, tal como el Internet de las Cosas, para así poder controlar actuadores u obtener datos desde diferentes puntos geográficos. Cuando se tienen situaciones tales como el estudio de los comportamientos ambientales, análisis de los comportamientos de consumo de individuos, o análisis de comportamientos poblacionales generales, es necesario obtener información de diferentes zonas y analizarlas en conjunto para obtener el resultado adecuado. Por este motivo, la generación de sistemas que puedan trabajar en conjunto desde distintas latitudes se vuelve un tema de estudio importante. En este trabajo se describirán diferentes modelos para la generación de una red neuronal distribuida, o segmentada, entre diferentes dispositivos conectados a Internet comunicándose mediante el protocolo de mensajería XMPP. Estos modelos logran identificar patrones de entrenamiento con datos provenientes desde distintas fuentes. El modelo utiliza tarjetas de desarrollo Raspberry Pi 3B+ para procesar y entrenar la red neuronal.

Palabras clave: internet de las cosas, redes neuronales distribuidas, cómputo distribuido, Raspberry Pi, XMPP.

Design of a Distributed Neural Network between Raspberry Pi Devices Connected to the Internet through XMPP

Abstract. Currently there are different communication schemes that seek the interconnection of any kind of devices, such as the Internet of Things, in order to control actuators or obtain data from different geographical points. When there are situations such as the study of environmental behaviors, or analysis of general population behaviors, it is necessary to obtain information from different areas and analyze such data together to obtain the correct result. For this reason, the generation of systems that can work together from different latitudes becomes an important topic of study. In this paper we will describe different models for the generation of a distributed or segmented neural network between different de-

vices connected to the Internet through the XMPP messaging protocol. Such models are able to identify training patterns with data from different sources. The model uses Raspberry Pi 3B + development boards to process and train the neural network.

Keywords: internet of things, distributed neural networks, Raspberry Pi, XMPP.

1. Introducción

Actualmente la adquisición, procesamiento, transmisión y análisis de datos se ha vuelto de gran valor científico, tecnológico, e incluso social, debido a la cantidad de aplicaciones que los resultados de tales análisis puedan generar. El *Internet de las cosas* (IoT, por sus siglas en inglés) es un esquema de comunicaciones que busca la recolección y el análisis de la mayor cantidad posible de datos.

El IoT ha ido tomando relevancia tecnológica y científica en los últimos años, este esquema busca la conexión de todo tipo de dispositivos electrónicos a Internet, y que estos se puedan comunicar, de forma automática, más aún, este esquema busca que estos dispositivos puedan tomar decisiones o actuar de forma “inteligente”. Esto con la finalidad de lograr un entorno más cómodo, se pueden ver las tecnologías, y las direcciones que se espera que este esquema llevara en el futuro en [1, 2].

Muchos dispositivos comerciales o sistemas embebidos conectados a estos esquemas informáticos adquieren y analizan información proveniente de un solo usuario o zona geográfica, pero ¿Y si los datos de entrada no estuvieran dados en la misma máquina o ubicación, sino separadas una gran distancia, como por ejemplo el nivel de tráfico en distintos puntos de una ciudad, la cantidad de personas que entran a en distintos puntos de un determinado transporte?

Si obtenemos información desde distintos puntos, se podría crear una serie de patrones de comportamiento avanzado y preciso que tendrían un sinnúmero de aplicaciones. Dicho de otra forma, se tienen grandes cantidades de datos provenientes de distintos lugares, pero actuando como datos de un mismo comportamiento a analizar o un mismo problema a resolver. Para estas posibilidades, es necesario generar no solo modelos de comunicación, sino también dispositivos tecnológicos que analicen datos distribuidos, es decir, que se transmitan información entre ellos y puedan analizar la información desde puntos diferentes pero que actúen como si fueran un solo dispositivo.

Es precisamente para analizar la información proveniente de diferentes zonas geográficas, entornos y usuarios, que en este trabajo se proponen diferentes modelos neuronales para adquirir, procesar y analizar datos a través de diferentes tarjetas de desarrollo tipo *Raspberry Pi*. Se han propuesto modelos basados en redes neuronales, que se distribuyen en diferentes dispositivos electrónicos para analizar de forma inteligente los datos que cada dispositivo adquiere de forma distribuida, y obtener una respuesta como si de un solo dispositivo se tratara. Con estos modelos se ha logrado identificar patrones de entrenamiento con datos provenientes desde diferentes puntos, demostrando así que se pueden identificar patrones por medio de redes neuronales seccionadas en dispositivos de baja potencia.

2. Preliminares

2.1. Raspberry Pi

Raspberry Pi es considerada una computadora de placa reducida, todos sus componentes, incluidos el procesador, la memoria y los puertos de entrada y salida, están integrados en una sola placa. Cuenta con 1GB de memoria RAM, un procesador Quad Core ARM Cortex-A53 corriendo a 1.2 Ghz, cuenta con puertos Ethernet y Wi-Fi para poder conectarse a Internet, todo esto a un costo reducido de aproximadamente \$40 USD. Uno de los aspectos más importantes por los cuales se decidió utilizar esta placa de desarrollo, es que esta se ha analizado y probado como un elemento que se puede vincular rápidamente dentro de la red IoT [3]. También puede ser una herramienta de alto impacto en los procesos de enseñanza y aprendizaje [4].

Este dispositivo ha sido utilizado en distintos proyectos tecnológicos y de desarrollo, tales como sistemas de seguridad basados en la detección de rostros [5], o sistemas de alarmas que envía videos y fotografías cuando un movimiento es detectado en el inmueble a proteger [6].

Pero lo más importante para fines de este proyecto, la Raspberry Pi ha sido el centro de operaciones de muchas propuestas tecnológicas que buscan la conexión de dispositivos dentro del IoT, como por ejemplo la generación de sistemas de monitoreo ambiental [7], así como el control y monitoreo de sistemas domóticos tal como se ve en [8] y [9], incluso se ha propuesto el monitoreo del entorno urbano [10].

Las características mencionadas, así como las aplicaciones dentro y fuera del IoT, hacen de Raspberry Pi, la mejor opción para el desarrollo de este proyecto.

2.2. XMPP

XMPP (*Extensible Messaging and Presence Protocol*) es un protocolo abierto que se basa en XML (*eXtensible Markup Language*), diseñado para proporcionar soluciones relacionadas con las comunicaciones en tiempo real.

Algunas de las ventajas más notables de XMPP son: descentralización, seguridad, extensibilidad, estándar abierto y gratuito. Debido a sus ventajas, el protocolo XMPP es útil en el desarrollo de interfaces hombre-máquina que pueden controlar dispositivos tecnológicos de forma remota [11].

Cada entidad XMPP requiere un identificador Jabber (JID), que contiene el nombre de usuario y el dominio en el que se conecta el recurso. JID utiliza un formato que es similar a la dirección de correo electrónico "usuario @ dominio / recurso".

XMPP es multilenguaje y multiplataforma pudiendo implementarse en Windows, en Linux, Android y casi cualquier sistema operativo, también puede implementarse en diferentes lenguajes de programación.

Adicionalmente, cuenta con algoritmos de encriptación que permite proteger los datos transmitidos.

2.3. Redes neuronales y aplicaciones distribuidas

Tratar de imitar la inteligencia de los seres vivos utilizando circuitos y maquinas sin alma es actualmente uno de los objetivos más ambiciosos de los investigadores y aunque se han logrado grandes avances e innovaciones, pensar que en un futuro próximo tendremos maquinas más inteligentes que nosotros es aún un sueño incierto.

Uno de los paradigmas de la computación que trata de asemejar el funcionamiento del cerebro de los seres vivos son las redes neuronales artificiales (RNA), que están avanzando rápidamente gracias a las nuevas tecnologías, y el desarrollo de equipos de cómputo con mayores capacidades de procesamiento y almacenamiento; aún más, se cuenta con el desarrollo de sistemas especiales para ejecutar procesos en paralelo como en el caso de las GPU's (*Graphics Processing Unit*).

Las redes neuronales son ampliamente utilizadas para la clasificación y detección de patrones, en donde se da una serie de entradas a la red neuronal y está dependiendo del entrenamiento previo arrojará un resultado, el esquema básico de una red neuronal puede verse en la figura 1.

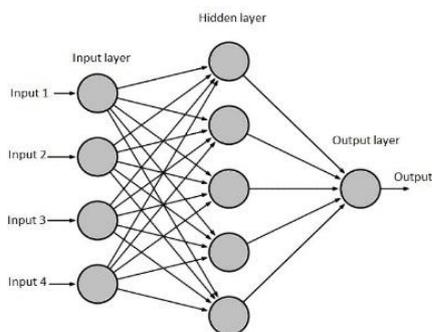


Fig. 1. Esquema básico de una red neuronal que contiene 4 entradas, una salida y una capa oculta.

Las RNA son sistemas que necesitan gran cantidad de recursos computacionales, aunque pueden ser implementadas en sistemas de baja potencia computacional tal como Raspberry [12, 13]. Sin embargo, cuando se tienen grandes cantidades de datos de entrada, un solo dispositivo puede no tener la capacidad necesaria para procesar tales cantidades de información, por este motivo, las RNA podrían ejecutarse en diferentes dispositivos de forma distribuida para aminorar la carga de cómputo en cada uno, logrando así una red distribuida de procesamiento de información. Para lograr esto de manera práctica, debemos considerar un protocolo que nos permita distribuir la información entre los diferentes nodos de la red, tal como XMPP.

En la literatura científica, existen experimentos en donde se han seccionado RNA en diferentes dispositivos electrónicos. En [14], *Krizhevsky* y otros autores mostraron la forma de entrenar una red neuronal con grandes cantidades de datos de entrada, para este fin, ellos distribuyeron la red en dos GPU.

Un ejemplo similar al anterior es el experimento llevado a cabo por *Dean* y otros autores en [15], donde se entrenó una red neuronal profunda con un billón de parámetros de entrenamiento utilizando 16000 CPU en solo 3 días.

En [16] los autores utilizan un clúster de GPU's para entrenar RNA con millones de parámetros de entrenamiento. Según los autores, se puede entrenar una red con un billón de parámetros usando solo tres GPUs en 3 días.

Los experimentos mencionados han sido realizados sobre unidades de gran nivel de procesamiento, sin embargo, en el proyecto presentado, se muestra la idea de generar RNA distribuidas sobre elementos de bajo procesamiento, en este caso la tarjeta Raspberry. En la literatura se tienen pocos experimentos relacionados a esta tarjeta de desarrollo, en [17], los autores afirman que Raspberry Pi tiene recursos limitados, por lo que la RNA se procesa y se entrena en la Nube, y la Raspberry sólo se encarga de obtener los datos relacionados al reconocimiento facial.

En [18] se genera una RNA distribuida, pero esta se distribuye en dos secciones, por un lado, se genera una RNA en una Raspberry que se entrena para casos prioritarios, y por otro lado se genera una red neuronal en la nube para casos más específicos, en este esquema, las RNA son entrenadas de forma independiente.

Caso similar sucede en [19], donde se genera una RNA distribuida también dividida en dos secciones, la red comienza a procesarse en el dispositivo restringido como la Raspberry y solo se basa en la parte remota cuando la parte local no proporciona un resultado lo suficientemente preciso.

Como se puede ver en los experimentos mencionados, solo se han desarrollado casos particulares en donde la Raspberry solo ha sido utilizada para un entrenamiento parcial, y trabaja de forma independiente al elemento de mayor procesamiento. Sin embargo, en la presente propuesta, se definen modelos en los cuales las tarjetas de desarrollo trabajan conjuntamente para solucionar un mismo problema. A conocimiento de los autores, no existe otro experimento que utilice estas placas de desarrollo para generar un procesamiento neuronal distribuido.

3. Implementación

En esta sección se presentarán los modelos propuestos para el procesamiento de los datos mediante una red neuronal artificial distribuida entre placas tipo Raspberry, así como los experimentos realizados para evaluar su operación.

3.1. Modelos de redes neuronales distribuidas

La idea principal para este proyecto es proponer un modelo para implementar una red neuronal distribuida a través de Internet, que pueda analizar datos que son recolectados por diferentes dispositivos desde diferentes posiciones geográficas. Actualmente los modelos existentes se basan en arquitecturas centralizadas o descentralizadas (Figura 2-a y 2-b), en el que los dispositivos periféricos envían los datos a uno o varios servidores y estos se encargan de aplicar los algoritmos necesarios para el análisis de los datos. En este trabajo se propone un modelo que no dependerá de servidores para su funcionamiento, sino en la cooperación e interacción de todos los nodos en la red.

El modelo que se pondrá a prueba en este trabajo es distribuir una red neuronal entre diferentes dispositivos físicos separados entre sí y conectados a través de Internet, como se muestra en la figura 2-c. Algunas de las ventajas de este modelo son las siguientes:

- a) Escalabilidad. Se pueden añadir dispositivos a la red incrementando su alcance y potencia.
- b) Confiabilidad. Al estar distribuida la carga de trabajo entre todos los dispositivos, la falla o desconexión de unos pocos no afecta a las demás y el sistema sobrevive como un todo.
- c) Economía. Es mucho más barato añadir dispositivos clientes que servidores.
- d) Independencia. Cada dispositivo tiene un sistema de procesamiento independiente, pero entre todos realizan acciones en conjunto.
- e) Recursos compartidos. Se satisfacen las necesidades de muchos usuarios a la vez al compartir la información y los recursos de cada dispositivo.

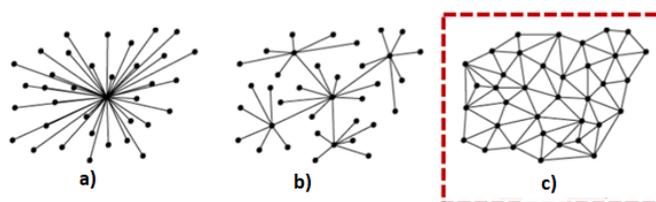


Fig. 2. Distintos tipos de arquitecturas de conexión: a) centralizada, b) descentralizada y c) distribuida.

Actualmente el modelo distribuido está tomando mucha importancia en campos como el IoT.

Para este tipo de distribución se proponen tres modelos de procesamiento de información, cada modelo tiene un flujo de información diferente. A continuación, se describen los tres modelos y la forma en la que las RNA son distribuidas entre los dispositivos electrónicos. Hay que recordar que se considera que cada dispositivo se encuentra en puntos geográficos diferentes, y obtienen la información de elementos sensoriales independientes entre sí.

Modelo 1. Cada dispositivo tendrá pocas neuronas y solo en conjunto se formará la red. Cuenta con una corrección de errores colectiva entre todos los dispositivos, el intercambio de información es masivo. Todos los dispositivos pueden tener entradas de datos y recolectar información de otros dispositivos, pero solo los finales tendrán una salida válida para la red. Para el entrenamiento de la red se intercambia información sobre los pesos de los enlaces, justo como si fuera una red neuronal clásica. Este modelo puede apreciarse en la figura 3.

Modelo 2. Cada dispositivo tendrá una red neuronal completa que se entrenará localmente, y analizará los datos recolectados en el mismo dispositivo, pero también enviará sus resultados a otros dispositivos en la red neuronal. Cuenta con corrección de errores interna, la red neuronal del dispositivo corregirá errores y también tendrá una corrección de errores colectiva pues todos los dispositivos corregirán errores entre sí. El intercambio de información es moderado, necesita pocos dispositivos para completarse. En este modelo cada dispositivo tiene un resultado particular correspondiente al análisis de sus entradas, pero la salida de la red puede ser consultada en cada dispositivo. Para la salida total, solo se podrá consultar en los dispositivos finales. En la figura 4, puede visualizarse la estructura de este modelo.

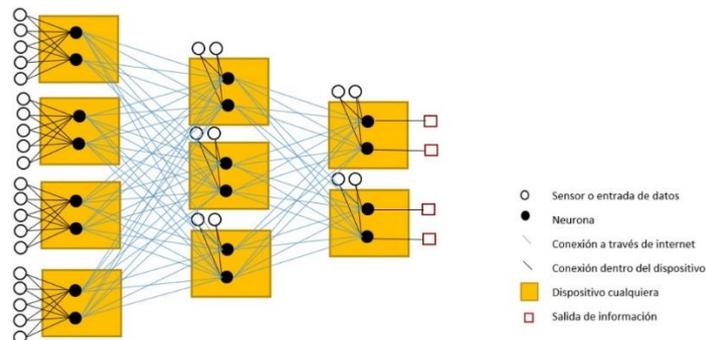


Fig. 3. Modelo 1 de procesamiento distribuido de información sobre una cantidad indefinida de dispositivos electrónicos.

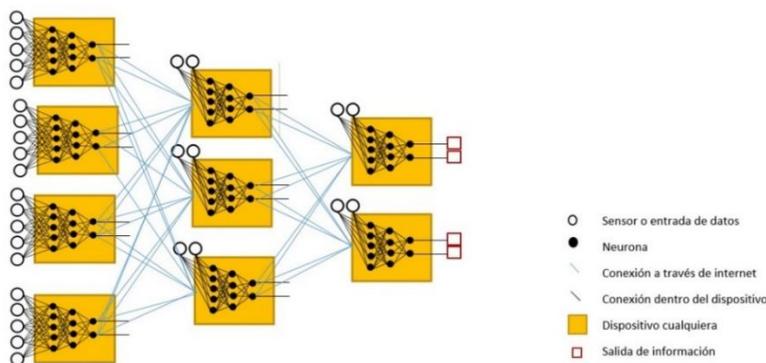


Fig. 4. Modelo 2 de procesamiento distribuido de información sobre una cantidad indefinida de dispositivos electrónicos.

Modelo 3. Cada dispositivo tendrá una red neuronal completa interna que analiza los datos de sus entradas, pero también analiza los resultados de la salida de otros dispositivos. Cuenta con corrección de errores interna, no se hace corrección de errores colectiva, cada dispositivo corrige sus propios errores. El intercambio de información es poco, basta un dispositivo para estar completa, pero entre más dispositivos, su capacidad de procesamiento será mejor. Esta red no tiene jerarquía como los modelos anteriores, solo se envía la información necesaria a los dispositivos que la soliciten no como en los modelos anteriores que la salida de unos es enviada a todos los de la siguiente capa. La arquitectura de este modelo se puede ver en la figura 5.

De los tres modelos este último sería el más adecuado de implementar en un escenario real, pues limita el flujo de datos, y presenta simplicidad en su implementación, pero los tres modelos pueden ser implementados dependiendo el escenario particular.

Sería especialmente útil en escenarios donde un nodo recolecta información de diferentes tipos, cada tipo de información puede ser útil para una red neuronal diferente, y cada una de estas redes neuronales se estarían ejecutando en el mismo dispositivo, pero en diferentes instancias, y podría o no haber comunicación entre estas, como se muestra en la figura 6.

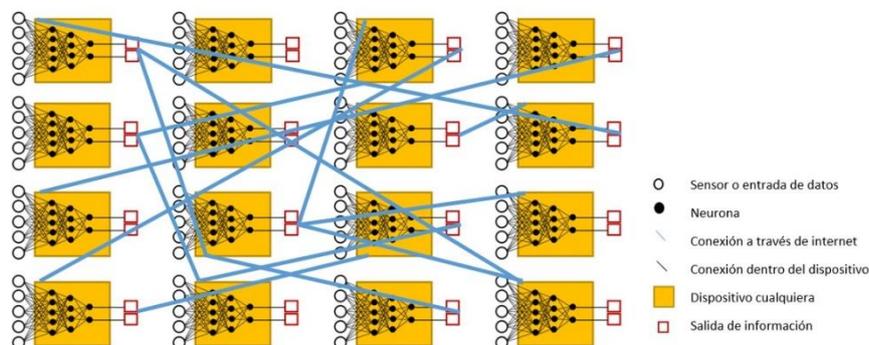


Fig. 5. Modelo 3 de procesamiento distribuido de información sobre una cantidad indefinida de dispositivos electrónicos.

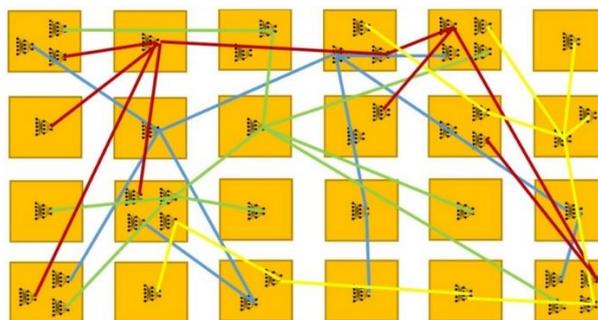


Fig. 6. Ejemplo de un dispositivo con múltiples instancias neuronales, donde cada línea de color representa distintas redes.

En la figura 6 se puede observar que dentro del mismo dispositivo se pueden tener múltiples instancias de RNA, cada una conectándose a través de Internet con los demás dispositivos de su red, un dispositivo puede formar parte de varias redes que analicen diferentes flujos de datos cada una.

El objetivo de estos modelos es analizar información recolectada por diferentes dispositivos que estén conectados a través de Internet.

Un ejemplo es analizar datos recolectados por los *smartphones* de las personas, los datos pueden ser imágenes, datos de los sensores, datos de comportamiento, etc. Estos datos al ser analizados en una red neuronal podrían dar patrones del comportamiento de los usuarios.

Con los ejemplos anteriores queda en claro las aportaciones y el uso que se le podría dar a un sistema como el descrito en este artículo, principalmente para IoT.

3.2. Pruebas

Para realizar las pruebas de los modelos planteados se tomaron los tres escenarios descritos anteriormente y se toma en cuenta una red de pocas neuronas y pocos dispositivos para hacer la demostración, el modelo de neurona propuesto en este trabajo fue

programado en Python, las pruebas se llevaron a cabo en Raspberry Pi 3B+, un dispositivo móvil con *Android* y una PC con *Windows* para demostrar la compatibilidad del lenguaje y protocolo de mensajería en diferentes dispositivos. El protocolo para establecer comunicación entre los dispositivos será XMPP, pues nos permite conectar usando servidores gratuitos y en cualquier parte del mundo que se tenga una conexión a Internet. El modelo básico será el mostrado en la figura 7.



Fig. 7. Modelo básico de prueba de las modelos de redes neuronales distribuidas.

Cada modelo de prueba tendrá neuronas individuales que funcionan con el siguiente algoritmo base, las entradas serán diferentes dependiendo del modelo.

Evento 0. Creación de la neurona. Este proceso lo lleva a cabo únicamente una vez cuando se comienza la ejecución.

1. Poner con valor de cero la salida y la señal de error.
2. Inicializa con un valor aleatorio o precargado los pesos y el offset.
3. Inicializa con un valor aleatorio o precargado el factor de aprendizaje.
4. Queda a la espera de percibir un cambio en las entradas.

Evento 1. Cambio en la entrada. Este proceso se ejecuta de forma concurrente, cada vez que la neurona detecta un cambio en sus entradas.

1. Multiplica cada entrada por su respectivo peso.
2. Realiza la sumatoria de todas las entradas y suma el valor del offset.
3. Aplica la función programada, puede ser rampa, sigmoidea o escalón.
4. Cambia el valor de la salida.

Evento 2. Cambio en la señal de error. Este proceso se ejecuta de forma concurrente cada vez que la neurona detecta un cambio en la señal de error.

1. Calcula el valor esperado considerando todas las neuronas que tiene a su salida.
2. Calcula el valor de error comparando el valor esperado contra el obtenido.
3. Hace la corrección de los pesos, ya sea aumentándolos o disminuyéndolos en proporción al error que aporte cada entrada y el factor de aprendizaje.
4. Hace la corrección del offset.
5. En base al error obtenido, hace el cálculo del valor ideal en cada entrada y lo pasa a la neurona en esa entrada.

El sistema se ha probado con un patrón binario sencillo de 4x4 casillas donde cada posición podrá tener valor de (0,1). Las neuronas después de ser entrenadas deberán identificar un patrón entre los múltiples patrones que se les presentaran, se contarán las veces que el patrón es identificado correctamente y los errores que se cometan.

También se contarán los paquetes intercambiados en la red en el tiempo de entrenamiento más el tiempo de reconocimiento, el número de paquetes es un promedio no el número exacto de cada reconocimiento.

Para el porcentaje de éxito se toma una base de 100 repeticiones con patrones similares.

Modelo 1. El patrón se reconoce por medio de dos dispositivos, la mitad de un patrón es entrenada en un dispositivo, la otra mitad en un dispositivo diferente y el final en el tercer dispositivo. Los dos dispositivos que leen el patrón no pueden reconocer su parte. El resultado puede consultarse en el tercer dispositivo únicamente, el esquema de este modelo se muestra en la figura 8. En la tabla 1, pueden verse los resultados experimentales del modelo representado en la figura 8.

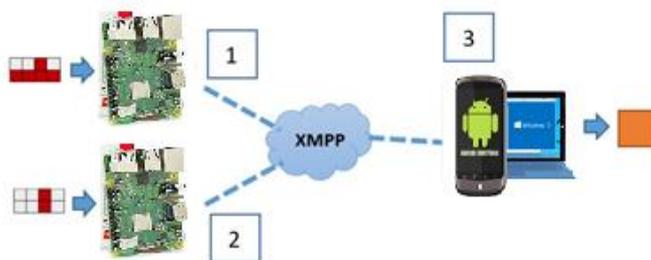
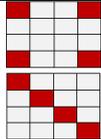
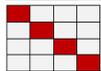


Fig. 8. Dos dispositivos reconocen la mitad de un patrón definido, solo el tercer dispositivo logrará identificar el patrón final.

En las tablas 1, 2 y 3, se muestran los patrones a entrenar, la cantidad de paquetes XMPP enviados a través de Internet entre los dispositivos durante el entrenamiento, y la cantidad de mensajes recibidos, para los modelos 1, 2 y 3, respectivamente. Para cada modelo se muestra la cantidad de veces que cada dispositivo logró identificar el patrón correcto.

Tabla 1. Comportamiento del modelo representado en la figura 8. Se puede visualizar que solo el dispositivo 3 logra identificar con diferentes porcentajes los patrones de entrenamiento.

Patrón	Enviados	Recibidos	Dispositivo 1	Dispositivo 2	Dispositivo 3
	360	352	NA	NA	87%
	341	402	NA	NA	83%
	328	298	NA	NA	93%
	268	354	NA	NA	98%
	413	352	NA	NA	84%

Patrón	Enviados	Recibidos	Dispositivo 1	Dispositivo 2	Dispositivo 3
	278	368	NA	NA	72%
	401	323	NA	NA	79%

Modelo 2. El patrón se reconoce entre los tres dispositivos, cada dispositivo puede reconocer su parte del patrón, pero el resultado final se puede ver únicamente en el tercer dispositivo, la estructura de este modelo puede verse en la figura 9. En la tabla 2, pueden verse los resultados experimentales del modelo representado en la figura 9.

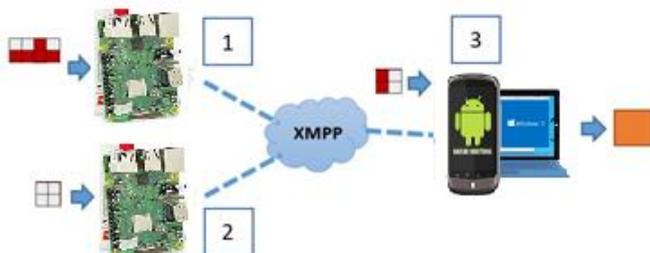
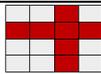
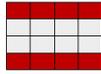
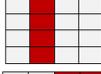
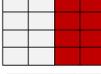
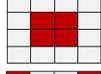
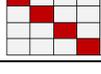


Fig. 9. Cada dispositivo reconoce el patrón a identificar, pero solo el dispositivo 3 lo puede mostrar.

Tabla 2. Comportamiento del modelo representado en la figura 9. Se puede visualizar que los 3 dispositivos logran identificar con diferentes porcentajes los patrones de entrenamiento.

Patrón	Enviados	Recibidos	Dispositivo 1	Dispositivo 2	Dispositivo 3
	114	119	88%	89%	85%
	121	124	87%	85%	84%
	106	129	92%	93%	90%
	114	122	99%	97%	95%
	144	131	87%	95%	86%
	160	120	74%	79%	71%
	149	128	81%	82%	77%

Modelo 3. El patrón completo está repartido entre los tres dispositivos, cada dispositivo puede reconocer su parte del patrón y preguntar a los otros dispositivos por la parte faltante del patrón (a los tres se les da el mismo patrón al mismo tiempo), la estructura de este modelo puede verse en la figura 10. En la tabla 3, pueden verse los resultados experimentales del modelo representado en la figura 10.

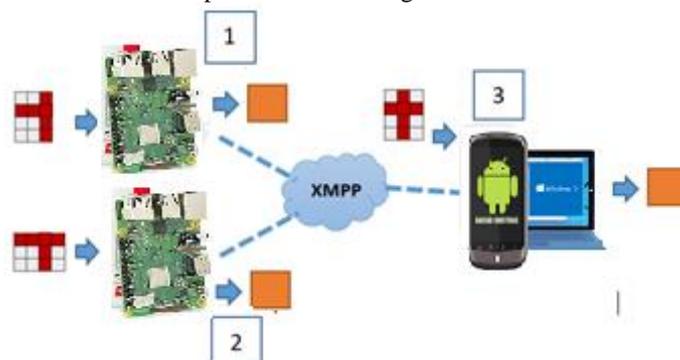
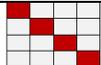


Fig 10. Cada dispositivo puede reconocer su parte del patrón e interactuar con los demás dispositivos para identificar cada sección del patrón.

Una vez analizadas las tablas 1 a 3 con las pruebas experimentales, se puede notar que los tres modelos tienen un porcentaje de reconocimiento similar, la diferencia es la complejidad de los patrones reconocidos y el tamaño de la red neuronal implementada, así como la cantidad de mensajes intercambiados por los dispositivos en el entrenamiento y reconocimiento. Las pruebas son únicamente para ver el intercambio de información entre los dispositivos, en otro entorno se implementará una red neuronal con pesos ya probados y con un porcentaje de éxito mayor.

Tabla 3. Comportamiento del modelo representado en la figura 10. Se puede visualizar que los 3 dispositivos logran identificar con diferentes porcentajes los patrones de entrenamiento.

Patrón	Enviados	Recibidos	Dispositivo 1	Dispositivo 2	Dispositivo 3
	40	12	84%	85%	86%
	33	15	83%	87%	85%
	50	8	96%	97%	96%
	38	16	95%	99%	97%
	48	8	80%	80%	82%
	43	13	73%	70%	71%

Patrón	Enviados	Recibidos	Dispositivo 1	Dispositivo 2	Dispositivo 3
	39	8	77%	79%	78%

Para contar los mensajes intercambiados únicamente se activa un contador que incrementa con cada mensaje XMPP enviado o recibido, no se hace distinción sobre el tipo de mensaje, pues se envían mensajes de configuración propios del protocolo además de los de la red neuronal, pero en un entorno real estos mensajes de configuración también se intercambiarían.

4. Conclusiones

Distribuir una red neuronal en diferentes dispositivos es rentable únicamente si la información que circulara entre los dispositivos no excede la capacidad del canal que los conecta. En el caso de Internet se debe tener en cuenta que la información de muchos otros servicios viaja por el mismo canal, por lo que la cantidad de información que puede fluir entre los dispositivos es limitada. La red neuronal debe configurarse de tal manera que los nodos solo intercambien la información más relevante, como el modelo 3 presentado en este trabajo.

Esta propuesta permite obtener información desde diferentes puntos o usuarios, y trabajar con toda esa información como si se estuviera ejecutando en un solo dispositivo, esta parte es de suma importancia, porque podemos generar redes de reconocimiento de patrones que se entrenaran con datos de todo el mundo, esto claro, permite generar una red neuronal que pueda interactuar directamente dentro del esquema del Internet de las cosas, comunicando diversos dispositivos entre sí, y procesando información de manera distribuida a través de él.

Se propone una comunicación basada en XMPP por su simplicidad de implementación y su compatibilidad con múltiples lenguajes y dispositivos, pero se puede implementar este modelo usando cualquier otro protocolo de comunicaciones instantáneo.

La ventaja que nos da el uso de XMPP, es que se vuelve una comunicación gratuita, y al tener protocolos criptográficos inherentes, permite que los datos transmitidos estén seguros.

Agradecimientos. Proyecto apoyado por PAPIIT IN 105219, PAPIME PE111519, PIAPIME 4.31.05.19, PIAPI 1824.

Referencias

1. Atzori, L., Iera, A., Morabito, G.: The Internet of Things: A survey, *Computer networks*, 54(15), 2787–2805 (2010)
2. Gubbi, M., Buyya, J., Marusic, R., Palaniswami, S.: Internet of Things (IoT): A vision, architectural elements, and future directions. *Future generation computer systems* 29(7), 1645–1660 (2013)
3. Maksimović, M., Vujović, V., Davidović, N., Milošević, V., Perišić, B.: Raspberry Pi as Internet of Things hardware: Performances and Constraints. In: *Proceedings of 1st International*

- Conference on Electrical, Electronic and Computing Engineering IcETRAN 2014, pp. ELI1.6.1–6 Vrnjačka Banja, Serbia (2014)
4. Zhong, X., Liang, Y.: Raspberry Pi: An Effective Vehicle in Teaching the Internet of Things in Computer Science and Engineering. *Electronics*, 5(3), p. 56 (2016)
 5. Gunawan, T.S., Gani, M.H.H., Rahman, F.D.A., Kartiwi, M.: Development of face recognition on raspberry pi for security enhancement of smart home system. *Electri. Eng. Inf.(IJEEL)*, 5, pp. 317–325 (2017)
 6. Ansari, A.N., Sedky, M., Sharma, N., Tyagi, A.: An Internet of things approach for motion detection using Raspberry Pi. In: *Proceedings of 2015 International Conference on Intelligent Computing and Internet of Things*, pp. 131–134. IEEE (2015)
 7. Ibrahim, A., Elgamri, M., Babiker, A., Mohamed, S.: Internet of things based smart environmental monitoring using the Raspberry-Pi computer. In: *2015 Fifth International Conference on Digital Information Processing and Communications (ICDIPC)*, pp. 159–164 (2015)
 8. Patchava, V., Kandala, H.B., Babu, P.R.: A Smart Home Automation technique with Raspberry Pi using IoT. In: *2015 International Conference on Smart Sensors and Systems, IC-SSS 2015*, pp. 1–4, IEEE (2017)
 9. Sandeep, V., Gopal, K.L., Naveen, S., Amudhan, A., Kumar, L.S.: Globally accessible machine automation using Raspberry pi based on Internet of Things. In: *2015 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2015*, pp. 1144–1147, IEEE (2015)
 10. Shete, R., Agrawal, S.: IoT based urban climate monitoring using Raspberry Pi. In: *International Conference on Communication and Signal Processing, ICCSP 2016*, pp. 2008–2012. IEEE (2016)
 11. Ramírez, D., Franco, J.A.R., Tinoco Varela, E.M.: Fuzzification of facial movements to generate human-machine interfaces in order to control robots by XMPP internet protocol, In: *MATEC Web of Conferences*, vol. 125, p. 04020. EDP Sciences (2017)
 12. Bechtel, H., McEllhiney, M.G., Kim, E., Yun, M.: Deeppicar: A low-cost deep neural network-based autonomous car. In: *2018 IEEE 24th Int. Conf. Embed. Real-Time Comput. Syst. Appl.*, pp. 11–21, IEEE (2018)
 13. Kumar, V.S., Gogul, I., Raj, M.D., Pragadesh, S.K., Sebastin, J.S.: Smart Autonomous Gardening Rover with Plant Recognition Using Neural Networks. In: *Procedia Computer Science*, 93, pp. 975–981 (2016)
 14. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet Classification with Deep Convolutional Neural Networks. In: *ImageNet Classification with Deep Convolutional Neural Networks*, pp. 1097–1105 (2012)
 15. Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Mao, M., Senior, A., Tucker, P., Yang, K., V Le, Q.: Large scale distributed deep networks. *Advances in neural information processing systems*, pp. 1223–1231 (2012)
 16. Coates, A., Huval, B., Wang, T., Wu, D.J., Ng, A.Y.: Deep learning with COTS HPC systems. In: *International conference on machine learning*, pp. 1337–1345 (2012)
 17. Sajjad, M., Nasir, M., Muhammad, K., Khan, S., Jan, Z., Sangaiah, A.K., Elhoseny, M., Baik, S.W.: Raspberry Pi assisted face recognition framework for enhanced law-enforcement services in smart cities. *Future Generation Computer Systems* (2017)
 18. De Coninck, E., Verbelen, T., Vankeirsbilck, B., Bohez, S., Simoens, P., Demeester, P., Dhoedt, B.: Distributed neural networks for internet of things: The big-little approach. In: *International Internet of Things Summit*, pp. 484–492. Springer, Cham (2015)
 19. Leroux, S., Bohez, S., De Coninck, E., Verbelen, T., Vankeirsbilck, B., Simoens, P., Dhoedt, B.: The cascading neural network: building the Internet of Smart Things. *Knowl. Inf. Syst.* 52(3), 791–814 (2017)

Diseño de una red neuronal distribuida entre dispositivos Raspberry Pi conectados a Internet...

Optimización Distribuida de Redes Convolucionales para la Clasificación de Imágenes

Emmanuel F. Ramírez Hernández¹, Héctor Rodríguez Rangel¹,
Víctor González Huitrón¹, Juan J. Flores², Vicenç Puig³

¹ Instituto Tecnológico de Culiacán,
División de Estudios de Posgrado e Investigación, Culiacán, Sinaloa, México

² Universidad Michoacana de San Nicolás de Hidalgo,
Facultad de Ingeniería Eléctrica, División de Estudios de Postgrado,
Morelia, Michoacán, México

³ Institut de Robòtica i Informàtica Industrial, Barcelona, España
eramirezhdz@gmail.com, hrodriguez@itculiacan.edu.mx,
victor.gonzalez@conacyt.mx, juanf@umich.mx, vicenc.puig@upc.edu

Resumen. Este artículo propone una mejora en los tiempos de ejecución en la optimización de los hiperparámetros de redes convolucionales en tareas de clasificación de imágenes. Para esto se realizó una modificación, distribuyendo la función de evaluación del algoritmo genético compacto, de esta forma se mejoraron los tiempos de ejecución de la evaluación de los individuos de cada generación. Esta estrategia se utilizó para optimizar los tiempos de búsqueda esperando encontrar una mejor configuración de hiperparámetros de una Red Neuronal Convolutiva (RNC). Además, se propuso una arquitectura para soportar la ejecución de procesos distribuidos utilizando Rabbit MQ; esta arquitectura es capaz de proporcionar propiedades deseables para sistemas distribuidos como alta disponibilidad, escalabilidad y tolerancia a fallos. Este trabajo fue probado utilizando la base de datos MNIST encontrando una mejora en los tiempos de ejecución en comparación con los algoritmos genéticos simples. Obteniendo una precisión comparable con los mostrados en la literatura.

Palabras clave: optimización de hiperparámetros, cómputo distribuido, red neuronal convolutiva.

Distributed Optimization of Convolutional Networks for Image Classification

Abstract. This article proposes an improvement in the execution times in the optimization of the hyperparameters of Convolutional Neural Networks in image classification tasks. For this purpose, an improvement was made by distributing the evaluation function of the compact genetic algorithm, improving the time in the execution of the evaluation of the

individuals of each generation. This strategy was used to optimize scanning times and hope to find a better configuration of hyperparameters of a convolutional neural network (RNC). In addition, an architecture was proposed to support the execution of distributed processes using Rabbit MQ, this architecture is capable of providing desirable properties for distributed systems such as high availability, scalability and fault tolerance. This work was tested using the MNIST database and found an improvement in execution times compared to simple genetic algorithms. Obtaining an accuracy comparable to those shown in the literature.

Keywords: hyperparameters optimization, distributed computing, convolutional neural network.

1. Introducción

En la actualidad las redes de aprendizaje profundo han estado revolucionando el ámbito de la inteligencia artificial, gracias a los resultados obtenidos mediante su uso. Las redes de aprendizaje profundo han empezado a utilizarse debido a que la capacidad de cómputo ha incrementado. Dicho incremento en la capacidad de cómputo ha hecho posible implementar estos modelos que son costosos en tiempo y procesamiento.

Normalmente las redes de aprendizaje profundo cuentan con millones de parámetros y requieren grandes cantidades de información para la optimización de estos y obtener el resultado esperado. Si incrementamos el tamaño de la red de aprendizaje profundo y el conjunto de datos podremos obtener modelos con mejores resultados pero a costa de un tiempo computacional considerablemente alto [1].

En la actualidad los tiempos en el entrenamiento de una red convolucional dependen del tamaño del conjunto de datos de entrenamiento y de las configuraciones asignadas. Si el conjunto de datos para el entrenamiento es demasiado extenso (para tener un modelo de calidad necesitamos que el conjunto de datos sea extenso, variado y balanceado) costará más tiempo en finalizar su entrenamiento.

En este artículo se propone una metodología basada en distribución de la evaluación de los individuos de un algoritmo genético compacto para la optimización de una red neuronal convolucional; de esta manera se podrá explorar un número mayor de posibles soluciones en un menor tiempo.

Este trabajo está estructurado de la siguiente manera: En la Sección 2 se encuentra una breve descripción de algunos algoritmos y metodologías de optimización para modelos de aprendizaje profundo. En la Sección 3 y 4 se muestran la definición, estructura y funcionamiento de una red convolucional y de los algoritmos genéticos compactos, respectivamente. En la Sección 5 se detalla la arquitectura propuesta para la distribución de los procesos del algoritmo genético compacto. La descripción de la implementación de la arquitectura propuesta se

encuentra en la Sección 6. Para finalizar en las Secciones 7 y 8 tenemos los resultados de los experimentos realizados y conclusiones, respectivamente.

2. Trabajos relacionados

La búsqueda de soluciones mejores y mas rápidas ha sido un gran potenciador para un un gran número de trabajos para la búsqueda en la optimización de los procesos de entrenamiento de redes de aprendizaje profundo.

Las redes de aprendizaje profundo son notablemente eficientes descubriendo estructuras de correlación en datos sin supervisión. Por lo tanto son ampliamente utilizadas en el análisis de procesamiento de lenguaje natural y visión computacional. En el artículo presentado por Hegde et al. [1] se han realizado diferentes formas de paralelizar el aprendizaje en entornos multinúcleo y distribuidos. También se han analizado de forma empírica la aceleración del entrenamiento de una RNC utilizando CPU y GPU.

En el trabajo de Brownlee [2] se ha considerado el problema de entrenar una red de aprendizaje profundo con miles de millones de parámetros utilizando miles de núcleos de un CPU. En este artículo se presenta un marco de trabajo llamado *DistBelief* que permite utilizar clústeres informáticos con miles de máquinas para entrenar modelos grandes. Por último en este trabajo se han desarrollado dos algoritmos para el entrenamiento distribuido a gran escala tales como: SGD y Sandblaster.

El trabajo presentado por Travis Densell [4] expone un nuevo algoritmo que es llamado *Evolutionary exploration of augmenting convolutional topologies (EXACT)*, el cual es capaz de evolucionar la estructura de la red neuronal convolucional. *EXACT* está modelado con base al algoritmo *Neuroevolution of augmenting topologies (NEAT)* que permite escalar en ambientes computacionales de manera distribuida y evolucionar las redes neuronales convolucionales. A diferencia de Travis Densell se realizó la aplicación del *Algoritmo genético compacto*.

En el trabajo presentado por Castillo, Pedro A [3] se utiliza el protocolo transferencia de estado representacional (REST) basado en HTTP como protocolo de comunicación. Además la función de evaluación de los individuos es la ejecución de una red neuronal multi-capas. Por último Castillo, Pedro A. et al. no describen algún mecanismo en caso de que suceda una falla en la ejecución de la función de aptitud por lo cual es posible tener la pérdida de individuos.

3. Redes neuronales convolucionales

El objetivo principal de las redes neuronales convolucionales (RNC) es aprender características de orden superior que se encuentran en los datos a través de convoluciones. La eficacia de las RNC en el reconocimiento de imágenes es una de las principales razones por las cuales el mundo reconoce el poder del aprendizaje profundo [5].

Las RNC extraen la información mediante convoluciones y estas características sirven de entrada a la capa de clasificación. En términos generales, se puede decir que las RNC básicamente están conformadas por tres grupos de capas:

1. Capa de entrada.
2. Capa de extracción de características (Convolución).
3. Capa de clasificación (aprendizaje).

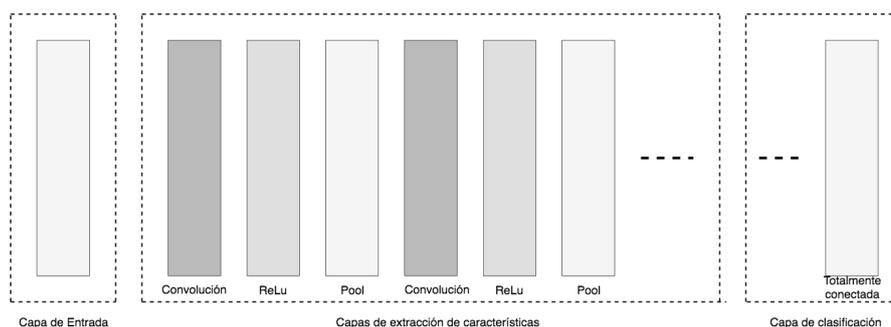


Fig. 1. Arquitectura general de una red convolucional.

3.1. Capas de entrada

La capa de entrada es donde ingresan los datos crudos de una imagen para ser procesados en la red. Normalmente los datos de entrada especifican sus dimensiones, como el ancho y alto. Cuando la imagen se encuentra a color se maneja una dimensión adicional denominada canal. Típicamente el número de canales son tres, para el valor de RGB de cada pixel.

3.2. Capa de extracción de características

La capa de extracción de características tiene un patrón general repetitivo de la siguiente secuencia:

1. Capa de convolución.
2. Capa de pooling (Agrupamiento).

Las capas de convolución. Son consideradas el núcleo más importante en la construcción de la arquitectura para una RNC. Estas capas transforman los datos de entrada usando un parche de neuronas conectado localmente desde una capa anterior. La capa calculará el producto punto entre la región de las neuronas de la capa de entrada y los pesos a los que se encuentran conectadas localmente en la capa de salida.

La salida resultante generalmente contará con las mismas dimensiones espaciales (o dimensiones espaciales más pequeñas) pero a veces aumenta el número de elementos en la tercera dimensión de la salida (dimensión de profundidad).

Las capa de pooling (Agrupamiento) Encuentran una serie de características en las imágenes y construyen progresivamente características de un orden superior.

Estas capas se insertan comúnmente entre capas convolucionales sucesivas. Se busca seguir capas agrupadas para reducir progresivamente el tamaño espacial (ancho y alto) de la representación de datos. La agrupación de capas reduce la representación de datos progresivamente en la red y ayuda a controlar el sobre ajuste.

3.3. Capa de clasificación

Estas capas son en las que se encuentran una o más capas completamente conectadas para tomar las características de orden superior y producir probabilidades o puntajes de clase. Estas capas están completamente conectadas a todas las neuronas de la capa anterior. La salida de estas capas produce normalmente una salida bidimensional de las dimensiones $[b \times N]$, donde b es el número de ejemplos en el mini lote y N es el número de clases que estamos interesados en calificar.

La Figura 2 muestra las diferentes capas trabajando en conjunto para realizar una clasificación desde la capa de entrada, pasando por un conjunto de capas de extracción de características (convolución y agrupamiento) y por último pasando por la capa de clasificación y otorgando un resultado.

4. Algoritmo genético compacto

El Algoritmo Genético Compacto (AGc) es una técnica de búsqueda u optimización probabilística, la cual está relacionada con el algoritmo genético y otros algoritmos evolutivos que están inspirados en la teoría de la evolución por medio de la selección natural. El objetivo AGc es simular el comportamiento del algoritmo genético con menores requerimientos (sin requerir que una población sea mantenida en memoria) [6]. Esto se logra utilizando un vector de probabilidades en lugar de la población entera. Las soluciones candidatas son probabilísticamente generadas desde la cadena de características. Las características que proveen una mejor solución se utilizan para realizar cambios en un vector de probabilidades [2].

El Algoritmo genético inicia definiendo el vector de probabilidad, a partir del dicho vector, dos individuos se generan y evalúan. El proceso de evaluación de los individuos prueba los hiperparámetros propuestos por el AGc. Los individuos generados compiten y de acuerdo al ganador se actualiza el vector de probabilidad. Este proceso se repite hasta que el criterio de convergencia se cumple.

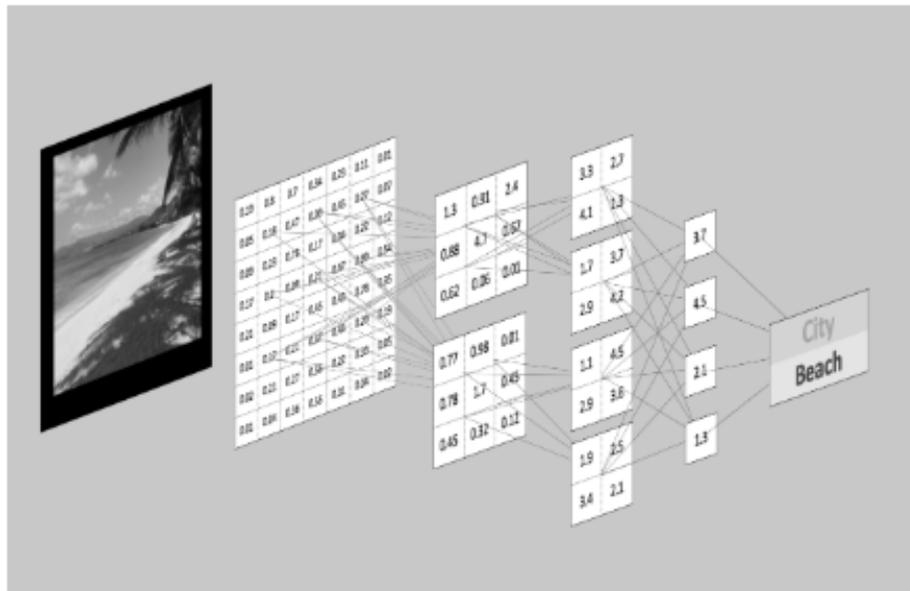


Fig. 2. Ejemplo de las diferentes capas para la obtención de una clasificación [5].

5. Arquitectura propuesta

Esta sección describe la arquitectura propuesta para llevar a cabo este trabajo. En la Figura 3 se muestra de forma general la arquitectura utilizada.

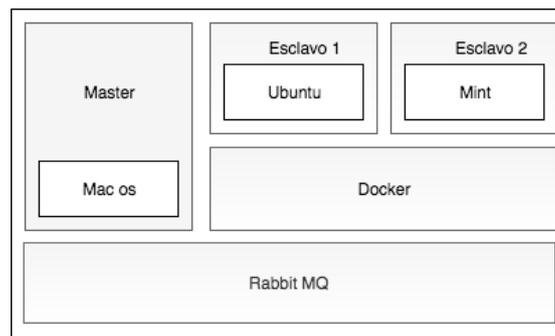


Fig. 3. Arquietctura general

El componente *esclavo* es el encargado de realizar la ejecución de la función de aptitud del individuo, esto quiere decir que en este caso el componente esclavo realizará el entrenamiento de la RNC según la configuración especificada por el

individuo. Al finalizar la ejecución del entrenamiento, este componente retorna el valor de aptitud especificado.

El componente *Maestro* es el encargado de coordinar los procesos que se realizan en el algoritmo genético compacto. Este componente realiza la generación de los individuos, la recepción de individuos entrenados y la competencia de los individuos. Además es el encargado de realizar la asignación del individuo que será procesado por el esclavo [7].

El componente llamado Docker es el encargado de la virtualización de los ambientes requeridos para realizar la ejecución de los procesos necesarios para el funcionamiento correcto del sistema. Docker nos permite utilizar entornos de aplicación aislados llamados contenedores. Esto es diseñado para el apoyo de los desarrolladores para poder replicar el ambiente utilizado en los esclavos sin mayor complejidad.

El último componente *RabbitMQ*, un “intermediario de mensajes” de código abierto que implementa el Protocolo Avanzado de Message Queue Server (AMQP), tiene como función el garantizar que los datos (los mensajes) vayan de un productor a los consumidores, para que estos mensajes sean procesados. El principal destinatario de los mensajes es la “cola”, un acumulador de datos potencialmente ilimitado, que reside dentro de RabbitMQ. Si el productor y los consumidores están conectados a la misma cola, pueden comunicarse sin que realmente se conozcan entre sí. Esto convierte a RabbitMQ en una poderosa herramienta para la distribución escalable y confiable para realizar la segmentación de los procesos.

6. Implementación del algoritmo genético compacto distribuido

En esta sección se presenta la implementación de la arquitectura propuesta, la distribución de los procesos y la coordinación entre los componentes descritos anteriormente. Además revisaremos los cambios que se necesitaron realizar al algoritmo genético compacto para realizar la distribución [8].

El nodo maestro inicializará los parámetros configurados tales como el tamaño de cromosoma y el número de generaciones que creara el AGCD. Para la definición del cromosoma se creara con base al tamaño del cromosoma especificado y se tomarán en cuenta los siguientes hiperparámetros.

1. Generales

- a) **Épocas:** Es un número entero positivo que limita el número de pasos que el conjunto de validación es evaluado.
- b) **Aprendizaje:** Es un número de punto flotante que representa la magnitud de la actualización por cada época de entrenamiento.
- c) **Entrenamiento:** Es un número que representa la proporción de datos utilizados en el conjunto de entrenamiento.
- d) **Optimizador:** Son funciones que calculan gradientes para una medida de pérdida y aplican gradiente a variables. Los tres posibles valores

de este parámetro son Stochastic gradient descent(SGD), AdamOptimizer(ADAM) y RMSPropOptimizer (RMSProp).

- e) **Activación:** Funciones que proveen la no linealidad. Este parámetro puede tomar dos valores: una función rectificadora (relu) y una función para suavizar la no linealidad (elu).

2. Convolucionales

- a) **Tamaño del Filtro:** Es un número que representa el tamaño del filtro que recorrerá la matriz que representa la imagen.
- b) **Strides:** Una lista de números enteros que representa el número de características que serán recorridas de izquierda a derecha en la matriz, moviendo el filtro por cada dimensión en el vector de entrada.
- c) **Padding:** Indica si el filtro puede ir más allá de los límites de la matriz. Los valores que puede tomar este parámetro son dos *same* y *valid*.
- d) **Pooling:** Reduce la dimensionalidad de la entrada permitiendo hacer suposiciones de características contenidas en una región de la entrada. Los valores que puede tomar este parámetro son *maxpooling* y *avgpooling*.
- e) **Dropout:** Es una técnica de regularización para reducir el sobre entrenamiento en redes neuronales. Los valores que puede tomar son: 0.3, 0.4, 0.5, y 0.6.

Para la obtención de cada uno los modelos se realizó la búsqueda de los diferentes hiperparámetros con los cuales se obtuviera el modelo que mejor se apegue a los datos de entrenamiento utilizando un AGC para la búsqueda de estos. Las Tablas 1 y 2 muestran los hiperparámetros tomados en cuenta, clasificados en Generales y Convolucionales, respectivamente.

Tabla 1. Hiperparámetros generales y sus posibles valores.

Parámetro	Rango
Épocas	[20, 40, 60, 80, 100, 120, 160]
Aprendizaje	[0.0001, 0.0006, 0.0011, 0.0016, 0.0021, 0.0026, 0.0031]
Entrenamiento	[0.70, 0.80, 0.90, 1.00]
Optimizador	[SGD, ADAM, RMSprop]
Activación	[relu, elu]

Tabla 2. Hiperparámetros convolucionales y sus posibles valores.

Parámetro	Rango
Tamaño del Filtro	[3, 4, 5, 6]
Strides	[2, 3, 4, 5]
Padding	[valid, same]
Pooling	[MaxPooling2D, AveragePooling2D]
Dropout	[0.3, 0.4, 0.5, 0.6]

La Figura 4 muestra un ejemplo de que posiciones del cromosoma proporcionado por el AGc corresponden a los diferentes hiperparámetros. El cromosoma es un vector de valores binarios, el cual es decodificado. e.g. para obtener el hiperparámetro Épocas se toman como referencia las 3 primeras posiciones, se codifica a decimal y así se determina que valor corresponde a dicha variable.

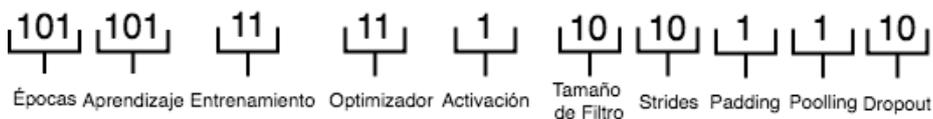


Fig. 4. Codificación de las diferentes hiperparámetros en el cromosoma del AGc.

Una vez preparado el nodo maestro, los nodos esclavos serán inicializados con la arquitectura seleccionada de la RNC (se muestra en la Figura 5) y realizando el cargado de datos que se le ha otorgado. El nodo esclavo una vez cargado e inicializado se suscriben a la cola de Rabbit MQ llamada *individuos* y a partir de ese momento los nodos esclavos estarán listos para procesar los individuos que el nodo maestro les proporcione.

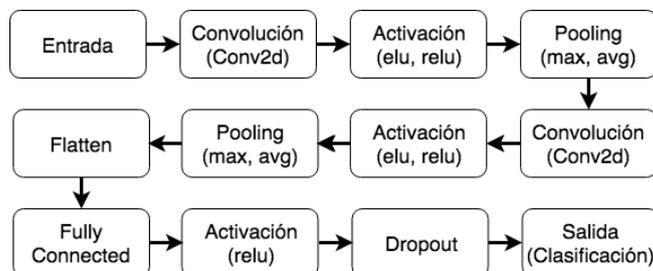


Fig. 5. Arquitectura de la CNN.

Una vez inicializado el nodo maestro y los nodos esclavos, el nodo maestro procederá a iniciar el proceso del AGc como se indica en la línea 11 y 12 del algoritmo 1 en las cuales indica que se empezará con la creación de la primera generación. En la línea 13 y 14 del algoritmo 1 se indica que los individuos serán enviados a la cola de Rabbit MQ llamada *individuos* a la cual los nodos esclavos ya han sido registrados previamente y tendrá que esperar mientras los nodos esclavos finalizan su proceso como se indica en la línea 15 del mismo algoritmo.

Algoritmo 1 Algoritmo Genético Compacto Distribuido.

```

1: procedure AGcD( $n, l, g$ )           ▷  $n$  como tamaño de población,  $l$  longitud del
   cromosoma
2:   Global individualTrainedOne
3:   Global individualTrainedTwo
4:   Global individualsTraining
5:   Se suscribe a la cola de Rabbit MQ
6:   subscribeQueueIndividualTrained(recvieveIndividualTrained);
7:   Inicializar el vector de probabilidad
8:   for  $i := 1$  to  $l$  do  $p[i] := 0.5$ ; do
9:     individualsTraining = True;
10:    Generar 2 individuos desde el vector
11:     $a := generate(p)$ ;
12:     $b := generate(p)$ ;
13:    sendIndividualToQueueIndividualGenerated(a);
14:    sendIndividualToQueueIndividualGenerated(b);
15:    while individualsTrained do
16:      wait();
17:    end while
18:    //Finalizado el entrenamiento se compiten los individuos
19:    winner, loser := compete(a, b);
20:    //Actualizar el vector utilizando el mejor
21:    for  $i := 1$  to  $l$  do
22:      if  $winner[i] \neq loser[i]$  then
23:        if  $winner[i] = 1$  then
24:           $p[i] := p[i] + 1/n$ 
25:        else
26:           $p[i] := p[i] - 1/n$ ;
27:        end if
28:      end if
29:    end for
30:    // Revisar si el vector ha convergido
31:    for  $i := 1$  to  $l$  do
32:      if  $p[i] > 0$  &  $p[i] < 1$  then
33:        return to step 4;
34:      end if
35:    end for
36:  end for
37:  return p
38: end procedure

```

Los nodos esclavos toman un individuo de la cola y empiezan la ejecución de la función de aptitud, en el caso de la RNC es la ejecución del entrenamiento tomando como métrica la *precisión*. Una vez finalizado el proceso de entrenamiento en los nodos esclavos, estos mandan un mensaje a la cola de *individuosEntrenados* de la cual el nodo maestro está escuchando; una vez que los dos individuos hayan finalizado su trabajo, libera los recursos para iniciar la competencia de los

individuos entrenados como lo indica la línea 12 del algoritmo 2. Seleccionado el individuo con mayor aptitud (individuo que tenga mayor precisión) como lo indica la línea 19 de algoritmo 2, se procede a la actualización del vector de probabilidades con base al individuo ganador y una vez actualizado el vector de probabilidades se verifica si el problema ha convergido. En caso de que no haya convergencia el nodo maestro procede a la siguiente generación; este proceso se repite hasta que el problema haya convergido o haya finalizado el número de generaciones especificadas.

Algoritmo 2 Algoritmo de control de la siguiente generación.

```

1: procedure RECIEVEINDIVIDUALS(individualRecieved) ▷ individualRecieved es el
   individuo ya entrenado
2:   Global individualTrainedOne
3:   Global individualTrainedTwo
4:   Global individualsTraining
5:   Global individualsTrainingCounter
6:   Se recibe el individuo desde la cola
7:   individual = json.load(individualRecieved);
8:   contadorIndividuosEntrenados = contadorIndividuosEntrenados + 1
9:   if individualsTrainingCounter == 2 then
10:    individualTrainedTwo = individualRecieved;
11:    individualsTrainingCounter = 0
12:    individualsTrained = False
13:   else
14:    individualTrainedOne = individualRecieved;
15:   end if
16: end procedure

```

7. Resultados

Se realizaron una serie de experimentos para demostrar que el algoritmo genético compacto distribuido nos otorga una mejora en tiempos de procesamiento. La configuración de los experimentos llevados a cabo fue la siguiente:

1. Un nodo maestro, un nodo esclavo.
2. Un nodo maestro, dos nodos esclavos.

La Figura 6 especifica las dos configuraciones con las cuales se realizaron los experimentos para comprobar que la metodología logró reducir los tiempos de procesamiento obteniendo la misma precisión. El nodo maestro es un equipo con un procesador Intel Core i5 de 2.9 GHz, con 8 GB de memoria RAM DDR3 con una frecuencia de 1600 MHz y una tarjeta GPU NVIDIA GeForce GTX 660M de 512 MB. El esclavo número uno es un equipo con un procesador Intel core i5-7500 de 3.40 GHz, con 8 GB de memoria RAM DDR3 con una frecuencia 1600 MHz y una tarjeta de video GPU NVIDIA GeForce GTX 1080 con 8 Gb

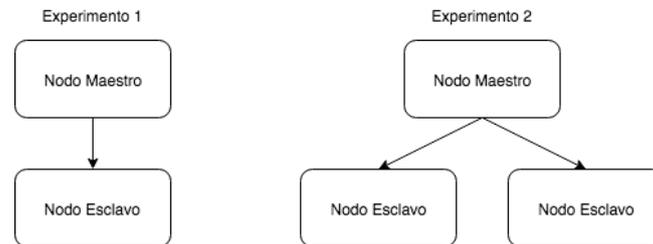


Fig. 6. Representación de la configuración de los experimentos realizados.

en RAM. El esclavo número dos es un equipo con un procesador Intel Xenon v4 de 1.70 GHz, y 32 GB de memoria RAM DDR3 a una frecuencia 1600 MHz y una tarjeta GPU NVIDIA GeForce GTX 1080 con 8 Gb en RAM.

La base de datos utilizada en los experimentos es el conocido “MNIST Database of Handwritten Digits” que contiene 10 clases posibles [9]. Esta base de datos contiene 60,000 imágenes de entrenamiento y 10,000 para realizar pruebas. Todas las imágenes están en escala de grises, sus tamaños son de 28 x 28 píxeles.

A continuación se presentan los resultados obtenidos con la configuración de un esclavo y un nodo maestro. En la tabla 3 se observa la representación del cromosoma con los tiempos de ejecución y la precisión obtenida. Podemos concluir que el tiempo promedio de ejecución es de 9 horas con 42 minutos y 59 segundos con una precisión promedio de 99.43%.

Tabla 3. Resultados obtenidos en experimentos con un nodo maestro y un nodo esclavo.

Representación del cromosomas	Tiempo	Precisión
011 001 00 10 0 10 11 0 0 11	09:11:32	99.50 %
110 010 10 11 0 11 00 1 0 10	08:57:34	99.32 %
110 110 11 01 0 11 10 0 0 10	09:26:05	99.48 %
010 001 11 11 0 10 10 0 0 11	09:06:12	99.42 %
001 011 00 10 1 10 11 0 1 11	09:03:15	99.12 %
111 110 10 01 0 10 11 0 0 11	10:45:35	99.48 %
111 000 10 10 0 11 10 0 0 00	10:24:25	99.52 %
100 111 10 10 0 11 10 0 0 00	10:37:09	99.50 %
111 100 00 10 0 11 11 0 0 11	09:29:49	99.48 %
011 010 10 10 0 10 00 1 0 11	10:08:16	99.51 %
Promedio	09:42:59	99.433 %

Los resultados obtenidos con la configuración de dos esclavos y un nodo maestro. En la tabla 4 podemos observar una mejoría en tanto a tiempo de ejecución como de precisión. El promedio de tiempo de ejecución es de 6 horas 24 minutos con 11 segundos consiguiendo una mejora significativa en comparación a los métodos simples.

Tabla 4. Resultados obtenidos en experimentos con un nodo maestro y dos nodo esclavo

Representación del cromosomas	Tiempo	Precisión
100 010 11 11 0 10 01 0 0 00	06:48:52	99.51 %
100 001 01 01 0 11 01 0 0 00	06:28:07	99.49 %
110 101 00 11 0 01 10 0 0 00	06:23:53	99.48 %
011 010 10 11 0 11 00 0 0 11	06:33:21	99.47 %
111 010 11 11 0 01 11 0 0 11	05:48:26	99.47 %
100 011 01 01 0 11 01 0 0 00	06:48:38	99.45 %
101 011 00 01 0 10 01 1 0 01	06:20:52	99.46 %
110 100 10 11 0 10 10 0 0 01	06:02:53	99.51 %
110 000 01 10 0 10 01 1 0 10	06:22:39	99.49 %
100 010 01 10 0 10 01 1 0 10	06:32:29	99.41 %
Promedio	06:24:11	99.474 %

8. Conclusiones

Las principales contribuciones de este trabajo son el planteamiento del algoritmo genético compacto distribuido así como la arquitectura necesaria para su implementación, con lo cual se redujeron significativamente los tiempos de búsqueda de una configuración de hiperparámetros óptimos. También como observación para obtener un resultado en tiempos óptimo los nodos esclavos necesitan tener un hardware muy similar (capacidad de procesamiento), debido a que el AGCD esperará la finalización de los dos individuos, esto quiere decir que el tiempo de ejecución de una generación dependerá del equipo con menor capacidad de cómputo.

Tabla 5. Comparación de promedios de los experimentos realizados.

Configuración	Tiempo	Precisión
AGC simple	09:25:37	99.458 %
AGCD con un esclavo	09:42:59	99.433 %
AGCD con dos esclavos	06:24:11	99.474 %

Los resultados promedio obtenidos (ver la Tabla 5) se han comparado con otros trabajos tales como el de Tabik y et al. [10] en los cuales podemos observar que la precisión obtenida en este trabajo utilizando el MNIST original es muy similar e incluso superada, entonces podemos afirmar que en la metodología propuesta es comparable con el estado del arte en términos de precisión.

Es importante considerar la asignación de la configuración (individuo), ya que esta puede ser mas costosa en tiempo de ejecución, y al ser entregada al equipo con menor capacidad de procesamiento producirá que el equipo con mayor capacidad culminará la evaluación del individuo y esperará la finalización del

equipo con menor capacidad y mas carga de trabajo. También es necesario que los nodos esclavos y el nodo maestro se encuentren interconectados a través de una red y se recomienda que sea una conexión alámbrica.

Para finalizar podemos concluir que la implementación del algoritmo genético compacto distribuido podemos obtener una mejora significativa en los tiempos de ejecución comparado con el algoritmo genético compacto simple.

Referencias

1. Hegde, V., Usmani, S.: Parallel and distributed deep learning. Stanford University (2016)
2. Brownlee, J.: *Clever algorithms: nature-inspired programming recipes*. First Edition, Jason Brownlee, Australia (2011)
3. Desell, T.: Large scale evolution of convolutional neural networks using volunteer computing. In: *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, ACM (2017)
4. Castillo, P.A., Arenas, M.G., Mora, A.M., Laredo, J.L.J., Romero, G., Rivas, V.M., Merelo, J.J.: Distributed Evolutionary Computation using REST. arXiv preprint arXiv:1105.497 (2011)
5. Patterson, J., Gibson, A.: *Deep learning: A practitioner's approach*. O'Reilly Media, Inc. (2017)
6. Harik, G.R., Lobo, F.G., Goldberg, D.E.: The compact genetic algorithm. *IEEE transactions on evolutionary computation* 2(5), 287–297 (1999)
7. Flynn, M.J.: Some computer organizations and their effectiveness. *IEEE Transactions on Computers*, pp. 94–96 (1972)
8. Kshemkalyani, A.D., Singhal, M.: *Distributed computing: principles, algorithms, and systems*. Cambridge University Press, Inc. (2011)
9. Deng, Li.: The MNIST database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, pp. 141–142 (2012)
10. Tabik, S., Peralta, D., Herrera-Poyatos, A., Herrera, F.: A snapshot of image pre-processing for convolutional neural networks: case study of MNIST. *International Journal of Computational Intelligence Systems*, pp. 555–568 (2017)

Emotion Mining Mechanism over Texts in Social Media

Luis Casillas, Alejandro Ramirez

University of Guadalajara, Computer Science Department, Mexico

`luis.casillas@cucei.udg.mx`

`alejandro_ramirez_munoz@protonmail.com`

Abstract. Nowadays societies are clearly bound to social networking through the Internet. It is common to find out people posting remarks, quotes or moods in social media. Leaders, politicians, celebrities, and ordinary people have posted on social media as a regular in their expression range. Even organizations generate postings on social networks. Humans' manifestation is always linked to the emotions they have. Authors believe that it is possible to discover the sentiment expression from texting. This proposal consists of a model to gather, classify, and emotionally-assess the texting in social networks. The model collects text posts from social media, processes the postings, and generates an emotional assessment. Such evaluation could be bound to the mood of postings' authors. The emotional-assessment consist of inference mechanisms based on knowledge coming from affective dictionaries and automated reasoning.

Keywords: emotion mining, emotions classification, affective dictionary, affective awareness, fuzzy classification.

1 Introduction

Social networks have become a popular mean for informal communication. Along the time, diverse social media have appeared and disappeared. Twitter¹ is one of the most popular social networks, with huge traffic on a daily basis. In social networks, members share large volumes of diverse opinions related to events occurring in their environment during real time.

Society is clearly bound to social media through the Internet. Leaders, politicians, celebrities, and common people have posted on social networks as a regular in their expression range. Even organizations generate postings on social networks. Humans' manifestation is always linked to the emotions they have.

According to Power [21] in their Social Media Benchmark Study:

“67% of consumers have used a company's social media site for servicing, compared with 33% for social marketing. Younger consumers (18-29 years old) are more likely to use brands' social media sites for servicing interactions (43%) than for marketing (23%).”

A second edition of the Social Media Benchmark Study from Power [22] express: “Nearly one-third (29%) of social media users get recommendations about a product or service from friends and family exclusively through social media. The most frequently

¹ "Twitter" is a registered trademark of Twitter, Inc.

used social media marketing channel is Facebook (29%), followed by YouTube (19%) and Twitter (11%) The most frequently used social media servicing channel is Facebook (84%), followed by Twitter (34%) and YouTube (25%)²."

The contributions and messages available in social media tend to follow informal patterns regarding their organization, sequence, and structure. Humans usually express freely, aiming at a human public. People convey ideas without hoping a machine should understand them. Nevertheless, in order to achieve an automated assessment of the human's expression, a machine should understand the messages as a human would do. Such a goal implies binding an emotional awareness of humans' discourse.

Humans' expression is deeply bound to emotional machinery. As Ekman [9] asserts when trying to organize his 20 years-old research, by presenting the emotions' understanding through families and features. Emotions are not isolated states, but families organizing common responses. Any emotion can be measured by nine features as universality, presence in other primates, psychological distinction, events succession, response coherence, onset speed, duration, appraisal, and unbidden occurrence.

Emotions have clearly defined evolutionary goals. As Plutchik [20] establishes in his long-term research, which involves the nature of emotions and is based on defining evolutionary profits to every response in order to achieve survival effects. This approach requires understanding emotions as a "complex chain of loosely connected events", instead of a simple feeling state.

In this context, a feeling could be assumed as a psychological state derived from an opinion or emotion, for example, fear would be a feeling while panic would be an emotion, love is a feeling and passion is an emotion.

According to Yadollahi, Shahraki, & Zaiane [25], "text sentiment analysis has been an attractive topic of study since the mid-1990s". Based on Medhat, Hassan, & Korashy [14], Sentiment Analysis (SA) or Opinion Mining (OM) refers to a computational study of people's opinions, as well as their attitudes and emotions toward an entity. That entity could represent individuals, events or topics.

The work [25] provides a taxonomical structure to Sentiment Analysis (SA). First, the SA is split into Opinion-mining (OM) and Emotion-mining (EM). The OM implies Subjectivity Detection, Opinion Polarity Classification, Opinion Spam Detection, Opinion Summarization, and Argument Expression Detection. EM involves Emotion Detection, Emotion Polarity Classification, Emotion Classification, and Emotion Cause Detection.

The SA is a broad area of study, in which there is no universal analysis technique for all the fields it covers. The diversity of languages can involve greater or lesser complexity from language to language. Depending on the type of problem to address, they must be reviewed the advantages and disadvantages of techniques to decide which strategy suits the requirements of the problem.

Medhat et al. [14] have classified the existing SA techniques under types of approach: Machine Learning (ML), Lexicon Based (LB), and Hybrid. The ML approach involves the already defined machine learning algorithms by supervising the linguistic features in a text. The LB approach is based on a sentiment lexicon, which is

² "Twitter" is a registered trademark of Twitter, Inc., "YouTube" is a registered trademark of Google Inc., and "Facebook" is a registered trademark of Facebook, Inc.

a collection of known and precompiled sentiment terms, these techniques might involve a dictionary or corpus. Finally, the hybrid approach involves some creative mixture of ML and LB.

Our proposal consists of an emergent and hybrid strategy to identify and determine the emotional perception that users have about specific events, situations, or people around them. The proposed model performs semantic analysis to determine the presence of emotions in the texts extracted from a social network, chats, emails, etc. In most cases, these texts will not follow grammar and spelling rules, the suggested model could preprocess the information to discriminate data that does not contribute value to the classification process.

The actual study has used data coming from social media, which are highly accessed due to the available information and their services. However, the proposed model could easily be operated over text coming from any social media, text-based chats, web-based blogs, emails, etcetera. Hence, this model must not be bound to a specific social network.

This proposal consists of a scheme for gathering, classify and emotionally-assess text postings in social networks. Although the system can be used on emails, blogs, chats, etc. The system collects text and generates an emotional assessment to the posts. Such an assessment might be bound to the mood of the post's author. The emotional-assessing is provided by an inference mechanism, which is based on knowledge coming from affective dictionaries and automated reasoning.

The paper is organized as follows: "Introduction" presents the study object as some bases from the theoretical frame, "Related work" focuses on the most remarkable projects in sentiment and emotion analyses, "Methodological approach" section presents the analysis strategy for this study, "Applying the model" relates the experience of seizing the strategy over some famous quotes. Finally, the "Conclusion" and "References" sections close to the current proposal.

2 Related Work

The sentiment and emotion analyses have been done for over 15 years. One of the first successful efforts was made by Pang, Lee, & Vaithyanathan [18] through a proposal to classify documents by sentiment instead of the topic. This challenge was interestingly managed through machine learning algorithms, mainly Naive Bayes, Maximum Entropy Classification, and Support Vector Machines. In the same perspective, involving thumbs up or down to phrases, and at the same time Turney [23] presents a simple unsupervised learning algorithm for classifying reviews. That reviews' classification was made by predicting the average semantic orientation of phrases in reviews, involving adjectives or adverbs. Phrases had positive semantic-orientation when they included "good" associations, and negative semantic-orientation when comprised "bad" associations. These preliminary efforts are somehow the base of the impressive opinion-mining available nowadays.

Bifet & Frank [2] developed an approach involving novel algorithms and the use of data-mining tools. These authors used Weka³ and MOA⁴ to support online learning-software based on examples. The training datasets were collected from a popular social network. The goal is to extract features using text filters. Those features are used as analysis-classes to produce sentiment awareness. This model would face challenging scenarios when unbalanced data streams are presented, due to the presence of a scarce list of classes. Such an issue is handled by authors through the use of sliding windows. Unfortunately, this approach has a deep dependency on specific tools, including the social network selected for the study.

Liang & Dai [12] created a system with an architecture that is able to automatically analyze the sentiments of collected messages. These authors' strategy consisted of collecting a set of messages and cataloged them. Those cataloged messages provided the system with the ability to filter, extract and figure out the sentiment direction, as positive or negative. Unfortunately, their approach is tied up to preconceptions for stored messages, implying a lack of flexibility.

The efforts from Feidakis & Daradoumis [10] implied a literature review regarding emotional learning and emotion assessment. This study allowed a deeper understanding of emotions and their impact on the learning processes. Throughout this review, the authors involve an approach to classifying sentiments along with rings as neuro-bio-cognitive, emotional, and socio-cultural. That classifying machinery deals with emotions from biological, cognitive, social, and cultural bases. The present study includes some classifying effort to sentiments and emotions gathered from social-networks texting.

Musto, Semeraro, & Polignano [16] performed a lexicon-based classification. Their experiment was based on two datasets, SemEval-2013⁵ and Stanford Twitter Sentiment (STS)⁶. The SemEval-2013 dataset consists of 11,435 tweets already divided into training (8,180 Tweets) and test data (3,255). The tweets have been manually processed. They were tagged and classified as positive, neutral, and negative. The STS dataset has more than 1,600,000 Tweets, already divided into training and testing, but the set of tests is smaller than the training (only 359 Tweets). The approach from these authors needed a preliminary training step to define analysis thresholds. This implies an undesirable consumption of machine resources.

Mukherjee & Bala [15] proposed a practical approach to detect sarcasm in costumers' posts. This is a highly focused field to research. There are few open studies, although the corporative marketing-areas would have plenty of related studies. These authors offer a solution divided into several stages, the first consists of obtaining test data. They gathered 15 thousand tweets having the hashtag #sarcasm. All the retweets were eliminated. Using filters, authors manually classified the Tweets. There were found 2,600 sarcastic texts and 2,400 non-sarcastic posts. Both datasets were used as training packages using cross-validation. The second stage is described as the extraction

³ Waikato Environment for Knowledge Analysis (Weka) is a suite of machine learning software written in Java, developed at the University of Waikato, New Zealand.

⁴ Massive Online Analysis (MOA) is a free open-source software project specific for data stream mining with concept drift. It is written in Java and developed at the University of Waikato, New Zealand.

⁵ SemEval-2013 is a challenging task aimed at Sentiment Analysis on Twitter.

⁶ The Stanford Twitter sentiment corpus consists of two different sets, training, and test. The training set has 1.6 million tweets automatically labeled as positive or negative based on emotion.

of characteristics. Features extraction becomes essential when dealing with classification problems. Finally, these authors applied two classification algorithms: Naive Bayesian and Maximum Entropy. Unfortunately, their approach has deep connections to specific hashtags and needs non-automated (manual) classification.

Ohmura, Kakusho, & Okadome [17] developed an intensive strategy to assess massive volumes of tweets to produce awareness about public moods. The analysis consisted of Latent Dirichlet Allocation (LDA), which assumes a categorical and multinomial distribution to words. LDA deals with the universe of words, involving diverse sources. This perspective allows the discovery of global trends. These trends are assessed through a six-dimensional mechanism involving daily time series of public mood (“calm,” “alert,” “sure,” “vital,” “kind,” and “happy”). Our proposal handles every text-asset independently, enabling specific sentiment-analysis.

Another interesting approach to give an affective-analysis has been developed by Cycil, Perry, & Laurier [5], which focuses on conflict resolution in special conditions. Specifically, in the confined scope of passengers in a car. These authors examine the efforts of parents while managing multiple stresses during the driving experience, along with the challenges of distractions from media use and disputes in the car. This kind of emotional scenario can be analyzed through an ethnographic study, involving conversations analysis. Automobiles have become a communication platform. Besides, mobile devices could improve the social interactions among passengers, by presenting relaxing expressions.

The reviewed projects included an important selection of data-mining techniques. Most of the results are accurate and fulfill the purposes. However, some negative aspects can be highlighted. These projects are restricted to certain scopes or they are not open-source. In most of the cases, systems were implemented under specific guidelines and constraints. Thus, they are strongly coupled systems. Besides, it is common that the architecture of the system is not fully described or available.

Our proposal is based on a methodology that provides flexibility to the system. The model is weakly coupled and is enabled to exchange its components in future versions. This proposal follows some principles and techniques that gave good results in reviewed projects. Applications are not limited to specific scopes or situations. Any communication scenario involving words can be measured by this affective analyzer.

3 Methodological Approach

Most of the emotion-classification systems provide results in three categories: negative, neutral and positive. The present project involves the use of fuzzy logic to classify among 27 possible emotions. This approach assumes that emotions cannot be classified with specifically defined limits from qualitative and quantitative variables such as age or maturity.

To capture some elements about the semantic dimension, it is performed a grammatical tagging over the messages that were gathered from social media. But as mentioned before, the proposed model can be applied to diverse text constructions as emails, chats, blogs, etcetera.

As regards social networks, the information produced by people could be hard to analyze since users will likely disregard correct grammatical structures. The

grammatical categorization of words is a helpful technique to discriminate valuable information from the noise that does not contribute anything and keeps vainly occupying processing time.

Hence, a Part of Speech (PoS) tagging is performed over each word and punctuation mark in the text. Along with this process, text tokens will be bind to morphosyntactic labels. Different labeling systems use different sets of labels, but usually, a label describes a word class and some specific characteristics of that word class, for instance, number and gender. The number of labels varies between dozens and hundreds.

Carlberger & Kann [6] highlight two problems that should be addressed when developing a grammar tagger:

1. Finding out all the possible labels for each word. The complexity of this task will increase when the words are unknown to the tagger since it must infer a possible label, otherwise, the tagger must fail on tagging and report a not-found result. When all the words to tag are known, the complexity decreases.
2. Choosing a single label for the word in the specific position of the sentence. This is known as syntactic disambiguation, and it must be solved on every ambiguous word in the sentence. Ambiguous words are frequently present in most languages.

Due to the large scale to develop a grammatical tagger, the present project conferred the tagging tasks to the Stanford CoreNLP [13]. A powerful, flexible, and extensible natural language processing tool.

Another methodological aspect involved is the use of affective dictionaries: packages of words evoking feelings, affective response, emotions, emotive actions, etc. According to Warriner et al. [24], "information about the affective meanings of words is used by researchers working on emotions and moods, word recognition and memory, and text-based sentiment analysis. Three components of emotions are traditionally distinguished: valence (the pleasantness of a stimulus), arousal (the intensity of emotion provoked by a stimulus), and dominance (the degree of control exerted by a stimulus)". Some psychologists conceptualize the emotion along two dimensions: "valence" and "arousal"; excluding the "dominance". Lewis et al. [11] established that "arousal invokes a single axis of intensity increasing from neutral to maximally arousing. Valence can be described variously as a bipolar continuum, as independent positive and negative dimensions, or as hedonic value (distance from neutral)."

The present project has involved the three affective components: valence, arousal, and dominance to perform the proposed sentiment analysis over social media.

It is very common that this type of research makes use of the affective dictionary ANEW from Bradley & Lang [4], which consists of 1034 words. Based on the diversity of messages, the number of words in ANEW may be sufficient to obtain acceptable results. In the present project, it is assumed that there is an important diversity of messages since different topics can be exposed, as well as the presence of expressions regarding temporary nature. Besides, the scope of social networks implies diverse social and cultural situations. Hence, the affective dictionary from [24] was selected. This extended dictionary consists of 13915 words, which has been assessed under the same principles of ANEW. The ANEW is inside in the extended version from [24].

The third main stage in the proposed strategy is fuzzy logic. This classification technique is one of the most widely used inference methods in many areas. Fuzzy logic allows simulating diverse procedures of human reasoning in knowledge-based systems.

It deploys models that allow dealing with the uncertainty happening in cognitive human-processes performed by computers. Although fuzzy logic was already used in the projects [1, 7], the actual approach has considered a completely new classification understanding over the input variables, as well as for the implied emotion categories in the output. This new, experimental, and from-scratch organization has implied a simpler, dynamic, and accurate processing of data.

The input variables are valence, arousal, and dominance, which are the analysis dimension produced by the affective dictionary from [24]. Every input variable is fed with data coming from the text analysis. This procedure consists in selecting from the analyzed text those words found in the affective dictionary. All the emotion ratings per word are summarized and averaged. Averaged values are settled in the corresponding input variables. Now, every input variable is fuzzified to produce a qualitative representation of the stored average. As shown in figures 1 and 2.

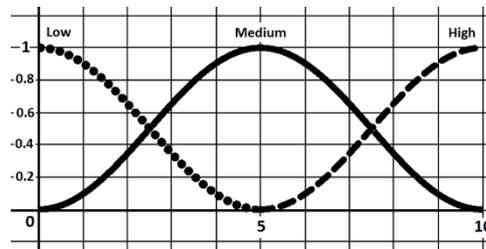


Fig. 1. Membership functions to produce fuzzy representation for input variables arousal and dominance: Low, Medium, and High. The shown curves and their actual dimensions are presented in this figure for demonstration purposes. The classifying system uses specific and floating limits, based on the indicators for centrality and dispersion of the data from the affective dictionaries.

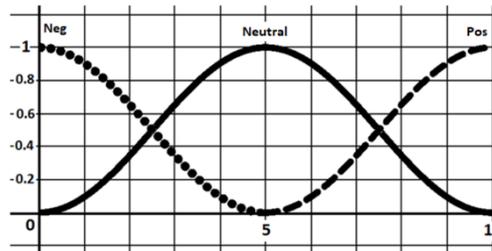


Fig. 2. Membership functions to produce fuzzy representation for the input valence: Negative, Neutral, and Positive. The shown curves and their actual dimensions are presented in this figure for demonstration purposes. The classifying system uses specific and floating limits, based on the indicators for centrality and dispersion of the data from the affective dictionaries.

Membership functions in figures 1 and 2 are produced by the formulas (1), (2), and (3). Formula (1) is known as Z function, formula (2) is known as soft-lambda ($s\Lambda$) function, and formula (3) is known as S function. The Z function has been bound to the linguistic-value "Low" in figure 1, and the linguistic-value "Negative" in figure 2. The $s\Lambda$ function has been bound to the linguistic-value "Medium" in figure 1, and the linguistic-value "Neutral" in figure 2. The S function has been bound to the linguistic-

value "High" in figure 1, and the linguistic-value "Positive" in figure 2. These formulas are based on the proposal from [8].

$$\begin{aligned} Z &= 1, \text{ if } x < \text{left}; \\ Z &= (1 + \cos(((x - \text{left}) / (\text{right} - \text{left})) * \pi)) / 2, \text{ if } \text{left} \leq x \leq \text{right}; \\ Z &= 0, \text{ if } x > \text{right} \end{aligned} \tag{1}$$

$$\begin{aligned} s\Lambda &= 0, \text{ if } x < \text{left} \text{ or } x > \text{right}; \\ s\Lambda &= (1 + \cos(((x - \text{center}) / (\text{center} - \text{left})) * \pi)) / 2, \text{ if } \text{left} \leq x < \text{center}; \\ s\Lambda &= (1 + \cos(((x - \text{center}) / (\text{right} - \text{center})) * \pi)) / 2, \text{ if } \text{center} \leq x \leq \text{right} \end{aligned} \tag{2}$$

$$\begin{aligned} S &= 0, \text{ if } x < \text{left}; \\ S &= (1 + \cos(((x - \text{right}) / (\text{right} - \text{left})) * \pi)) / 2, \text{ if } \text{left} \leq x \leq \text{right}; \\ S &= 1, \text{ if } x > \text{right} \end{aligned} \tag{3}$$

The borders "left", "center" and "right" in formulas (1), (2), and (3) refer to specific limits that can be settled dynamically in the model, according to actual centrality and dispersion data from the general average and the standard deviation shown by the dataset to classify.

Once the input variables are fuzzified, the resulting linguistic-values are used to produce the corresponding fuzzy output. Figure 3 shows the 3D crossing structure for input variables. The 27 inner regions produced by this crossing process are bound to the qualitative results for output variable: emotion.

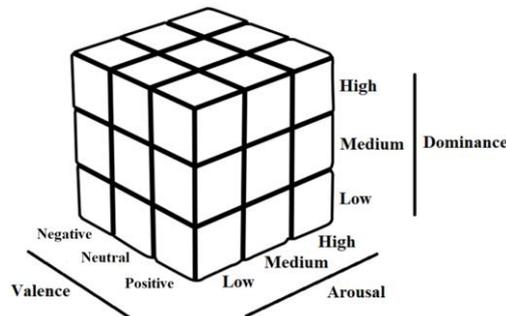


Fig. 3. This is the 3D classifying structure. It involves the three input variables as dimensions. Every variable has three segments. Segments refer to linguistic values.

Regarding the classification process, the input variables are fed through the sum and averaging of accumulated values. These values are collected from the identified words in the affective dictionary. These calculations produce sharp and precise numbers. Due to computational complexity to deal with these numerical representations, data are fuzzified. The fuzzy version allows a simplified decision scheme when predicting and classifying the emotional state. Instead of an unlimited continuum of emotions, the actual emotional state is selected from a limited set with 27 alternatives. These alternatives are the emotions set up by Boulic et al. [3]: Scared, Afraid, Empathic, Embarrassed, Surprised, Delighted, Furious, Vigilant/Alert, Excited, Depressed/Sad, Doubtful/Pessimistic, Compassionate, Unhappy, Bored/Sad, Tired, Peaceful, Anxious, Happy, Neutral, Sleepy, Glad, Angry, Alert/Willing, Sulky, Annoyed, Malicious, and Vicious.

It is true that humans have emotional manifestations as an infinite continuum. However, it is also true that computers have real problems when detecting and handling humans' emotions. The present proposal gives a simplified scheme to predict emotional profiles for social-networking users, based on their text postings. Nevertheless, any specific diagnosis for the emotional situation would need professional assessing performed by humans. The automated predictions made by the proposed system aims at producing some guidelines to measure and, in some cases, feedback users.

The project [3] is aimed at producing facial expressions for avatars and they directly use the actual values from valence, arousal, and dominance to produce adjustments on the facial components of avatars. The proposal in our study is aimed to produce a qualitative response to fuzzified values from the input, by processing the text according to an affective dictionary. Tables 1, 2, and 3 explain the classification rules from linguistic representations of input variables. These tables represent the 3D classification layers as 2D ensembles. The sliced variable is dominance, its organization is also inspired by the distribution set up at [3], although some reorganization was made over the original proposal.

Table 1. Classification rules among Valence, Arousal, and Dominance. Low Dominance.

Dominance: Low		Arousal		
		Low	Medium	High
Valence	Negative	Bored/Sad	Depressed/Sad	Afraid
	Neutral	Tired	Doubtful/Pessimistic	Scared
	Positive	Peaceful	Compassionate	Empathic

Table 2. Classification rules among Valence, Arousal, and Dominance. Medium Dominance.

Dominance: Medium		Arousal		
		Low	Medium	High
Valence	Negative	Anxious	Unhappy	Embarrassed
	Neutral	Sleepy	Neutral	Surprised
	Positive	Glad	Happy	Delighted

4 Applying the Model

The analysis performed by the proposed model is fed from social media elements. Authors have decided to apply the emotional-classification to classical quotes. These quotes have been written or spoken by famous authors or personages. Social media is currently under scrutiny, due to the privacy-risks that can be implied to the users of these services. Hence, authors have decided to apply the emotion mining analysis over famous quotes. The very same emotional classification can be applied to any text from

social-networks postings, mini-blogs, emails, etc. when the texts' authors have agreed on the emotion-mining.

Table 3. Classification rules among Valence, Arousal, and Dominance. High Dominance.

Dominance: High		Arousal		
		Low	Medium	High
Valence	Negative	Sulky	Angry	Furious
	Neutral	Annoyed	Alert/Willing	Vigilant/Alert
	Positive	Vicious	Malicious	Excited

In table 4 is presented a detailed analysis of one quote. Quotes are entered into the CoreNLP, which proceeds to tokenize and tag every phrase. The CoreNLP tagging is based on the treebank approach from [19]. All words are searched in the affective-dictionary, but only some of them use to be there. Affective-dictionaries have limited corpus, due to the massive effort to produce affective-data for words. Those found words give data to the input variables. From the amounts collected in summarized input variables, the fuzzy classifier produces qualitative representations.

Finally, these qualitative inputs imply a specific emotion bounded to the quote. The automated process for emotion-implying is decorated by a certainty level, which is based on membership satisfaction during the classifying stage.

As shown in table 4, a sentence is automatically processed to produce a prediction of the emotion beating in the quote's author when the phrase was created. As well as the certainty bound to such prediction. Thus, the quote "you know you're in love when you can't fall asleep because reality is finally better than your dreams" from Dr. Seuss, was written under a HAPPY emotional state. In addition, this prediction has a 70.8% certainty. This incomplete-certainty can be explained by the fuzzy classification process when values had no full membership to the involved sets.

Unfortunately, the analysis does not include all the words in the sentence, but as explained before: affective-dictionaries have limited sets of words. A similar analysis was performed over some other famous quotes. Table 5 holds the results from these analyses.

Input variables are now fuzzified into three values. This approach simplifies dealing with data and decision making. Nevertheless, there is a significant loss of information. A highly precise value is fuzzified into one of three alternatives. A future approach for this solution would be based on a more precise splitting for variables arousal and dominance. If the semantic differential scale has five levels as "very low", "low", "medium", "high", and "very high", 75 regions would result by crossing five values for arousal, five values for dominance, and three values for valence.

The 27 emotions defined at [3] would be the same, but now they would be spread along the 75 regions. There would be a repetition of emotions in the regions, but the system could provide improved performance with precise responses. Future efforts in this research line would produce an assertive distribution of emotions in these 75 regions.

Table 4. An example of system’s operation over a quote.

Quote	
"You know you're in love when you can't fall asleep because reality is finally better than your dreams." Dr. Seuss	
Words found in the affective dictionary:	Results
Word: "know", PartOfSpeech: VERB_SINGULAR_PRESENT_NONTHIRD_PERSON :: Arousal: 3.24, Valence: 6.82, Dominance: 5.78 Word: "love", PartOfSpeech: NOUN :: Arousal: 5.36, Valence: 8, Dominance: 5.92 Word: "fall", PartOfSpeech: VERB :: Arousal: 4.24, Valence: 3.89, Dominance: 3.83 Word: "asleep", PartOfSpeech: ADVERB :: Arousal: 2, Valence: 6.5, Dominance: 4.33 Word: "reality", PartOfSpeech: NOUN :: Arousal: 4.1, Valence: 5.73, Dominance: 7.26	Numerical results: Averaged Arousal: 3.78 Averaged Valence: 6.19 Averaged Dominance: 5.42 Fuzzy classification: Fuzzy Arousal: Medium Fuzzy Valence: Positive Fuzzy Dominance: Medium ***** Implied Emotion: HAPPY Certainty level: 70.8%

Table 5. More examples of emotion-mining on quotes.

Quote	Results
“You cannot escape the responsibility of tomorrow by evading it today.” Abraham Lincoln	<i>Implied Emotion: WILLING</i> <i>Certainty level: 55.86%</i>
“A revolution is a struggle to the death between the future and the past.” Fidel Castro	<i>Implied Emotion: AFRAID</i> <i>Certainty level: 81.09%</i>
“The keenest sorrow is to recognize ourselves as the sole cause of all our adversities.” Sophocles	<i>Implied Emotion: SLEEPY</i> <i>Certainty level: 67.3%</i>
“Jealousy is that pain which a man feels from the apprehension that he is not equally beloved by the person whom he entirely loves.” Joseph Addison	<i>Implied Emotion: EMBARRASSED</i> <i>Certainty level: 54.22%</i>
“The true soldier fights not because he hates what is in front of him, but because he loves what is behind him.” G.K. Chesterton	<i>Implied Emotion: ALERT</i> <i>Certainty level: 74.12%</i>

Regarding the current use of famous quotes as the target of the system’s operation. The reader is invited to consider that these quotes were expressed by quote author in social media, even though some of those authors would not have agreed with the experience of using social networks as most people do. Nevertheless, the system is currently enabled to extract, process, and emotionally classify assertions, conveys,

arguments, postings, blogging, etcetera. All of them regular nowadays expression mechanisms under the ceiling of modern Information and Communications Technologies (ICT).

5 Conclusion

Humans' expression is deeply bound to emotional machinery. Emotions have clearly-defined evolutionary goals. Nowadays social-media has become a standard expression mechanism. The academic/education environments, the collaborative projects, the business arena, and almost any scope with human interaction are clearly bound to the emotional states of people.

Authors of this proposal believe that any awareness about the current emotional state of people might be useful. There are diverse sources of emotional awareness, such as human' faces, handwriting, sweating, etcetera. This proposal has focused on the word choice during persons' text-based assertions. This analysis is based on the grammatical categorization and the affective assignment for words in text messages.

Affective indicators define sentiment loads to words and expressions. The system developed as part of this proposal is enabled to recover, process, tag, and affectively assess text-based arguments. Nevertheless, the authors of this study have considered that sentiment analysis over social-media implies an important responsibility. Text elements project more than people may want to reveal. Hence, any analysis over expressions in social networks must be carefully handled.

There must be always a formal permission-request to analyzed people, as well as a formal consent for that analysis. That is the reason this proposal has processed famous-quotes in the public domain, instead of specific extracts from social-media. Any analysis of specific text from social-media would run the same way as shown for famous-quotes. The selected quotes for this study do not represent any specific claim from the authors of this study. They are only test cases.

The current analysis gives an awareness about emotions that could be laying-beneath the arguments in quotes. Although the system is currently enabled to extract, process, and emotionally classify assertions, conveys, arguments, postings, blogging, etcetera.

Besides, some other components might be discovered through a syntax analysis. Authors believe that adverbs and adjectives could involve some active modification over the already defined numbers for the input variables. This active response, as well as considering the 75 regions previously mentioned will lead the future efforts of current research.

References

1. Arguedas, M., et al.: A model for providing emotion awareness and feedback using fuzzy logic in online learning. *Soft Computing* 22(3), 963–977 (2018)
2. Bifet, A., Eibe, F.: Sentiment knowledge discovery in twitter streaming data. *International conference on discovery science*. Springer, Berlin, Heidelberg (2010)
3. Boulic, R., et al.: Towards the Instantaneous Expression of Emotions with Avatars. *Cyberemotions*. Springer, Cham, 255–278 (2017)

4. Bradley, M.M., Lang, P.J.: Affective norms for English words (ANEW): Instruction manual and affective ratings 30(1), Technical report C-1, the center for research in psychophysiology, University of Florida (1999)
5. Chandrika, C., Perry, M., Laurier, E.: Designing for frustration and Disputes in the family Car. *International Journal of Mobile Human Computer Interaction (IJMHCI)* 6(2), 46–60 (2014)
6. Carlberger, J., Kann, V.: Implementing an efficient part-of-speech tagger. *Software: Practice and Experience* 29(9), 815–832 (1999)
7. Casillas, L., Peña, A., Gutierrez, A.: Towards an Automated Model to Evaluate Collaboration through Non-Verbal Interaction in Collaborative Virtual Environments. *Intelligent Systems: Concepts, Methodologies, Tools, and Applications*. IGI Global, pp. 1570–1586 (2018)
8. del Brío, B.M., Sanz Molina, A.: *Redes neuronales y sistemas difusos*. Alfaomega Ra-Ma (2002)
9. Ekman, P.: An argument for basic emotions. *Cognition & emotion* 6(3-4), 169–200 (1992)
10. Feidakis, M., Thanasis, D.: A framework for designing computer supported learning systems with sensibility. *International Journal of e-Collaboration (IJEC)* 9(1), 57–70 (2013)
11. Lewis, P.A., et al.: Neural correlates of processing valence and arousal in affective words. *Cerebral cortex* 17(3), 742–748 (2006)
12. Liang, P.W., Bi-Ru, D.: Opinion mining on social media data. In: 2013 IEEE 14th International Conference on Mobile Data Management Vol. 2 (2013)
13. Manning, C. et al.: The Stanford CoreNLP natural language processing toolkit. In: Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations. (2014)
14. Medhat, W., Hassan, A., Korashy, H.: Sentiment analysis algorithms and applications: A survey. *Ain Shams engineering journal* 5(4), 1093–1113 (2014)
15. Mukherjee, S., Pradip, K.B.: Detecting sarcasm in customer tweets: an NLP based approach. *Industrial Management & Data Systems* 117(6), 1109–1126 (2017)
16. Cataldo, M., Semeraro, G., Polignano, M.: A comparison of lexicon-based approaches for sentiment analysis of microblog posts. *Information Filtering and Retrieval* 59 (2014)
17. Ohmura, Masahiro, Koh Kakusho, Okadome, T.: Tweet sentiment analysis with latent dirichlet allocation. *International Journal of Information Retrieval Research (IJIRR)* 4(3) 66–79 (2014)
18. Pang, Bo, Lillian Lee, Vaithyanathan, S.: Thumbs up? Sentiment classification using machine learning techniques. Proceedings of the ACL-02 conference on Empirical methods in Natural Language Processing - Volume 10. Association for Computational Linguistics (2002)
19. Prasad, R. et al.: The Penn Discourse TreeBank 2.0. In: LREC (2008)
20. Plutchik, R.: The nature of emotions: Human emotions have deep evolutionary roots, a fact that may explain their complexity and provide tools for clinical practice. *American scientist* 89(4), 344–350 (2001)
21. Power, J.D.: Poor social media practices can negatively impact a businesses' bottom line and brand image (2013)
22. Power, J.D.: Positive Automotive Social Media Experience Impacts Purchase Decisions across All Generations. April 10 - September. <http://www.jdpower.com/press-releases/2014-social-media-benchmark-study-auto> (2014)
23. Turney, P.D.: Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In: Proceedings of the 40th annual meeting on association for computational linguistics. Association for Computational Linguistics (2002)
24. Warriner, A.B., Kuperman, V., Brysbaert, M.: Norms of valence, arousal, and dominance for 13,915 English lemmas. *Behavior research methods* 45(4), 1191–1207 (2013)

Luis Casillas, Alejandro Ramirez

25. Yadollahi, A., Shahraki, A.G., Zaiane, O.R.: Current state of text sentiment analysis from opinion to emotion mining. *ACM Computing Surveys (CSUR)* 50(2), p. 25 (2017)

Evaluación de resúmenes automáticos con y sin resúmenes de referencia para el idioma inglés

Selene Arai Vilchis Sepúlveda, Yulia Ledeneva

Universidad Autónoma del Estado de México,
Estado de México
selene.vilchis96@hotmail.com, yledeneva@yahoo.com

Resumen. Durante casi 60 años se han realizado diversas investigaciones sobre diferentes métodos y herramientas comerciales para generación automática de resúmenes de texto, en consecuencia, también se ha indagado en los diferentes métodos de evaluación de resúmenes de texto automáticos. Entre los más utilizados en el estado de arte encontramos los evaluadores ROUGE, mismos que hacen la comparación de un resumen generado de manera automática contra uno generado por un humano. Por otro lado, encontramos evaluadores más robustos como Jensen-Shannon y ROUGE-C que evalúan el resumen generado de manera automática contra el documento original (debido a la ausencia de resúmenes generados por humanos). En este artículo, se evaluaron los resúmenes generados automáticamente con diferentes evaluadores, donde se observó que cada evaluador toma una parcialidad diferente.

Palabras clave: evaluación de resúmenes automáticos, resumen de referencia, medidas, Jensen Shannon, ROUGE-C, ROUGE-1, ROUGE-2.

Evaluation of Automatic Summaries with and without Reference Summaries for the English Language

Abstract. During almost 60 years, several investigations have been carried out on different methods and commercial tools for the automatic generation of text summaries, consequently, it also has been researched in the different methods of evaluation of automatic text summaries. Among the most used in the state-of-the-art, we find ROUGE evaluators, which make the comparison of a summary generated automatically against one generated by a human. On the other hand, we find more robust evaluators like Jensen-Shannon and ROUGE-C that evaluate the automatically generated summary against the original document (due to the absence of summaries generated by humans). In this paper, the automatically generated abstracts were evaluated with different evaluators, where it was observed that each evaluator takes a different partiality.

Keywords: evaluation of automatic summaries, reference summary, measures, Jensen Shannon, ROUGE-C, ROUGE-1, ROUGE-2.

1. Introducción

El ser humano ha buscado a lo largo del tiempo transmitir el conocimiento que ha adquirido en su paso por la tierra, desde las pinturas rupestres que fueron creadas hace miles de años, con el fin de transmitir los conocimientos adquiridos a sus descendientes, posterior a ello la información documentada [13] ha ido evolucionando en diferentes partes del mundo como las tablillas de arcilla en Mesopotamia, el papiro en Egipto, el pergamino en Grecia y Roma, la invención del papel en China, hasta la época actual en donde encontramos grandes volúmenes de información digitalizada en todo el mundo [1]. Con la revolución tecnológica, la ayuda de internet y de grandes albergadores de documentos, el ser humano actualmente cuenta con una gran cantidad de información. La Fundación Nacional de la Ciencia, muestra los indicadores sobre el crecimiento de la información publicada de artículos científicos y técnicos, misma en la que se percibe un crecimiento exponencial en un periodo de 2003 a 2016, con un crecimiento acelerado a nivel mundial, en 2003 se reportaron 1.12 millones de artículos publicados y para 2016 se registraron 2.37 millones de artículos publicados [5].

El crecimiento acelerado de información ha dificultado la búsqueda de temas específicos, por lo que las ciencias computacionales han intervenido a través del Procesamiento del Lenguaje Natural (PLN), al que se le conoce como una disciplina en la inteligencia artificial que se encarga de la investigación de mecanismos computacionales para la comunicación entre humanos y computadoras mediante el uso de Lenguajes Naturales (LN) [10,20,21].

El PLN ha creado técnicas, herramientas y sistemas para el manejo de la información tales como la Generación Automática de Resúmenes de Texto (GART) [8,33], los cuales se describen como la extracción de ideas principales del contenido de un texto original [28,34], donde el contenido es una cuestión tanto de información como de expresión, la extracción de ideas principales es una cuestión de lo que es esencial y lo que es relevante [15]. Los métodos de GART se clasifican, de acuerdo con el tipo de resumen de salida, resúmenes extractivos o abstractivos [6,30]. Los resúmenes abstractivos, son textos que describen el contenido y el sentido de un documento original, esto consiste en entender el contexto de un documento y después reescribirlo sin perder el sentido del texto utilizando nuevos conceptos en un número menor de palabras, sin perder el contexto del documento [28,31]. Mientras que los resúmenes extractivos consisten en, reducir el contenido textual de un documento o un grupo de documentos mediante la selección de un conjunto de frases u oraciones del texto original [14,31,32]. Generalmente, las personas realizan resúmenes de tipo abstractivo, mientras que la mayoría de las herramientas comerciales y los métodos del estado del arte generan resúmenes de forma extractiva. Actualmente existen herramientas gratuitas disponibles en internet para la generación automática de resúmenes en diferentes idiomas. Algunas de las herramientas instalables son Copernic summarizer [7] y Microsoft Office Word [23]. Algunas de las herramientas en línea son Open Text Summarizer [22], Tool4noobs Summarizer [27], Text Compactor [26], t-Conspectus [25], Summarizer [4] y SweSum [24], entre otras.

Para conocer la calidad de los resúmenes generados por las herramientas comerciales, en trabajos anteriores [11,12,29] se han realizado las evaluaciones mediante el evaluador ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [16], propuesto por Lin y Hovy, Lin & Och [17,18,19]. Este sistema calcula la calidad

del resumen generado de manera automática mediante la comparación con resúmenes generados por un humano como menciona Ledeneva [28].

Para la utilización de estos evaluadores son importantes las referencias humanas, es decir, los seres humanos deben crear resúmenes de alta calidad. Por otro lado, existen métodos de evaluación de resúmenes como Jensen Shannon y ROUGE-C. Estos métodos de evaluación automáticos no requieren la creación de resúmenes modelo humano, se evalúan tomando un conjunto de resúmenes generados de manera automática y se comparan con los textos originales [2,3], ya que se tiene la idea que los textos originales contienen la información completa.

En este artículo se realiza la evaluación de los resúmenes generados de manera automática por las herramientas comerciales con los evaluadores que utilizan un resumen modelo humano y los que utilizan el texto original.

El resto del trabajo se organiza de la siguiente manera: En la sección 2, se describe el trabajo de estado del arte que ha realizado la evaluación de diferentes herramientas comerciales. En la sección 3, se describen los diferentes evaluadores para los resúmenes automáticos, por una parte, se describen los evaluadores con el uso de resúmenes de referencia y por otra parte se mencionan los evaluadores sin resumen de referencia. En la sección 4, se describe la colección de documentos, la experimentación y los resultados obtenidos. La sección 5, se describen las conclusiones del trabajo realizado.

2. Trabajo relacionado

Durante los últimos años se han realizado diversas investigaciones sobre la Generación Automática de Resúmenes de Texto (GART) de las herramientas comerciales y de métodos del estado del arte. Sin embargo, la mayoría de estos estudios realizados se evalúan a través de una comparación entre un resumen generado de manera automática contra con un resumen generado por un humano. El problema de investigación consiste en cómo realizar la evaluación de resúmenes automáticos generados por herramientas comerciales sin resúmenes modelo. En esta sección, se describe el trabajo realizado por Matías [12], en el que realiza la evaluación de herramientas comerciales y métodos del estado del arte con la colección de datos DUC02. Esta evaluación se realizó con el sistema evaluador ROUGE.

2.1. Herramientas comerciales

En el trabajo de Matías [12] se emplearon diversas herramientas comerciales algunas en línea como:

- Shvoong: Es una herramienta que permite generar resúmenes automáticos en 21 idiomas diferentes (checo, neerlandés, danés, inglés, finlandés, francés, alemán, griego, hebreo, húngaro, indonesio, italiano, malayo, noruego, polaco, portugués, rumano, ruso, español, sueco y turco). A diferencia de otras herramientas Shvoong no devuelve el resumen, sino subraya el texto que considera más importante del documento original.
- Pertinence Summarizer: Desarrollada por la empresa francesa Pertinence Mining. Pertinence es una herramienta en línea que permite generar resúmenes en 12 idiomas

(alemán, inglés, árabe, chino, coreano, español, francés, italiano, japonés, portugués, ruso y neerlandés) de los documentos de texto en formatos diversos (html, pdf, doc, rtf y txt).

- Tools4Noobs: Es una herramienta en línea que permite generar resúmenes desde 1 al 100 % del texto original. Para la generación de un resumen Tools4Noobs tiene 3 etapas: extracción de las oraciones, identificación de las palabras claves del texto contando la relevancia de cada palabra e identificación de las oraciones de acuerdo con las palabras claves identificadas.
- Open Text Summariser: Es una herramienta de código abierto para resumir textos, que puede ser descargada de Internet de forma gratuita. Sin embargo, también puede encontrarse la interfaz de ésta en línea. OTS genera resúmenes automáticos en diferentes porcentajes, genera resúmenes en 37 idiomas.

2.2. Métodos del estado del arte

Además del uso de herramientas comerciales Matías [12] realizó la comparación con los siguientes métodos del estado del arte:

- TextRank: Es un método que consiste en gráficos ponderados, en que se construye el gráfico para representar el texto, por lo que los nodos son palabras interconectadas por vértices con relaciones significativas. Para la tarea de extraer oraciones, el objetivo es calificar oraciones completas y ordenarlas de mayor a menor importancia. Por lo tanto, los vértices se agregan al gráfico para cada oración en el texto.
- Baseline: Esta heurística garantiza que la información más importante de un documento se encuentra en las primeras secciones para las que se toman las primeras n oraciones del documento para formar el resumen. Se ha demostrado que esta simple heurística genera muy buenos resúmenes en el campo de los documentos de noticias.
- SFMs: Una secuencia que se repite en el texto es frecuente y es maximal si dicha secuencia no es subsecuencia de otra secuencia frecuente.
- Baseline-Aleatorio: Esta heurística obtiene los mejores resúmenes para determinar los resúmenes cuando solo se toma un conjunto de oraciones al azar. La idea es determinar los resultados que se pueden lograr.

3. Evaluadores de resúmenes automáticos

Los evaluadores de resúmenes automáticos han sido de gran ayuda para los investigadores de GART. A través de estos sistemas se califica la calidad de los resúmenes generados de manera automática. Generalmente los investigadores utilizan los evaluadores ROUGE, porque éstos realizan la comparación entre un resumen automático y un resumen modelo generado por un humano. Sin embargo, no todas las colecciones de datos cuentan con un resumen modelo (o también le llaman estándar de oro), lo que dificulta la evaluación. Es por ello que es un problema de investigación importante que se ha indagado en los evaluadores que no ocupan un resumen modelo como ROUGE-C y Jensen Shannon. Estas evaluaciones se realizan al comparar el

resumen automático contra el texto original, por lo que se tiene la idea que los textos originales cuentan con las ideas completas.

3.1. ROUGE

ROUGE es un sistema para calcular la calidad de los resúmenes generados de forma automática mediante la comparación de los resúmenes automáticos con resúmenes creados por humanos, fue propuesto por (Lin & Hovy, 2003; Lin & Och, 2004a,b) [17, 18, 19]. Este sistema trabaja mediante la recuperación de n-gramas entre un resumen candidato y un resumen ideal creado por un humano y se calcula de la siguiente manera:

$$ROUGE - N = \frac{\sum_{S \in \{ReferenceSummaries\}} \sum_{gram_n \in S} \sum_{\epsilon \in S} Count_{match}(gram_n)}{\sum_{S \in \{ReferenceSummaries\}} \sum_{gram_n \in S} \sum_{\epsilon \in S} Count(gram_n)}, \quad (1)$$

donde n es la longitud del n-grama y $Count_{match}(gram_n)$ el número máximo de n-gramas que co-ocurren en el resumen candidato y el conjunto de resúmenes de referencia.

3.2. ROUGE-C

ROUGE-C es un método de evaluación automático para la evaluación de resúmenes de texto automáticos, fue propuesto por (He et al., 2008) [9]. Este método es una variación al sistema de evaluación de resúmenes de ROUGE-1.5.5, en el que se plantea la sustitución de los resúmenes de oro en el texto fuente y además se plantea el intercambio entre documentos, es decir, el documento fuente es colocado como resumen de evaluación y el resumen generado es colocado como un resumen de evaluación, mientras que el resumen regentado por el sistema es colocado como un resumen de referencia como se muestra en la figura 1. La evaluación de resúmenes con ROUGE-C está basado bajo la siguiente expresión:

$$ROUGE - C - N = \frac{\sum_{S \in \{PeerSummary\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{SourceDocument\}} \sum_{gram_n \in S} Count(gram_n)}, \quad (2)$$

donde n determina el uso del n-grama a evaluar a través de su longitud, $Count_{match}(gram_n)$ es el máximo número de co-ocurrencia de n-gramas entre el resumen generado por el sistema y el documento fuente.

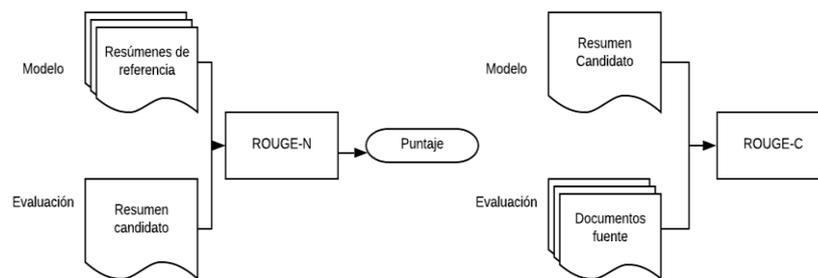


Fig. 1. Evaluación de ROUGE (Lin, 2004) y ROUGE-C (He et al., 2008) [9].

3.3. Jensen Shannon

Jensen Shannon divergence es un método de evaluación para la generación automática de resúmenes de texto, fue propuesto por (Louis & Nenkova, 2008), (Louis & Nenkova, 2009) [2, 3]. Este método evalúa un resumen generado de manera automática con el documento original, incorpora la idea de la distancia entre dos distribuciones no puede ser muy diferente del promedio de las distancias de su distribución media. La evaluación está basada bajo la siguiente expresión.

$$J(P||Q) = \frac{1}{2} [D(P||A) + D(Q||A)], \quad (3)$$

donde $A = \frac{P+Q}{2}$ es la distribución media de P y Q . La distancia JS es simétrica y siempre está definida.

4. Experimentación y resultados

Para la primera evaluación de los resúmenes generados automáticamente por las herramientas comerciales, se van a utilizar los evaluadores ROUGE-1 y ROUGE-2. La evaluación consiste en estimar el parecido de los resúmenes generados automáticamente con los resúmenes realizados por expertos humanos. Los resúmenes generados por las herramientas en línea e instalables fueron generados con un mínimo de 100 palabras, por lo que anterior a ello, se analizaron las herramientas para realizar dicha tarea.

Por otro lado, realizamos la segunda evaluación de los resúmenes automáticos generados por las herramientas comerciales, con el uso de los evaluadores ROUGE-C y Jensen Shannon. Estas evaluaciones constan de una comparación entre los resúmenes automáticos y los textos originales, ya que los autores de dichas evaluaciones plantean la idea de decir si realmente los humanos generan buenos resúmenes.

5. Corpus

DUC (Document Understanding Conference) es una serie de talleres creados por NIST (National Institute of Standards and Technology) para el desarrollo de la tarea de GART. Las colecciones de documentos elaboradas por DUC son las más usadas por investigadores en la tarea de GART de multi-documentos y para un solo documento, destacando las colecciones de datos DUC01 y DUC02.

Para este artículo se utilizó la colección de datos DUC01, la cual consta de 309 documentos en idioma inglés, agrupados en 30 colecciones, cada colección contiene en promedio de 10 a 11 documentos de noticias periodísticas, que abordan temas sobre desastres naturales, información biográfica de un individuo, temas políticos, etc. Esta colección de datos también contiene los resúmenes de oro generados por humanos.

6. Herramientas comerciales

En el artículo se utilizaron las herramientas comerciales de mayor auge en las investigaciones realizadas. La herramienta instalable en sus diferentes versiones es:

- Microsoft Office Word: Es una herramienta que se encuentra incluida en Microsoft Office Word en las versiones 2003 y 2007, permite generar resúmenes de 10 o 20 oraciones, 100 o 500 palabras. También cuenta con la opción de realizar resúmenes con porcentajes: 10%, 25%, 50% y 75% de palabras del documento original.

Las herramientas en línea que se consideran en este artículo son:

- Open Text Summarizer (OTS): Es una herramienta de código abierto analiza automáticamente textos. OTS genera resúmenes automáticos en diferentes porcentajes de 5% a un 80%. Se pueden generar resúmenes en 37 idiomas.
- Summarizer: es una herramienta que permite generar resúmenes automáticos. Está disponible como componente de Intellexer API y como aplicación de escritorio. Genera resúmenes en diferentes porcentajes de 1% a un 99%. Recibe un documento de origen, extrae texto sin formato, proporciona el procesamiento sintáctico y semántico, extrae la información para la generación de resumen del documento, finalmente, asigna un valor determinado por oraciones del texto original. Este valor define la importancia de la oración en lo que respecta a la idea del texto, tiene como opción crear diferentes tipos de resúmenes.
- Summarizing: Es una herramienta en línea para la generación automática de resúmenes de manera rápida y eficaz. La herramienta tiene las opciones de 100, 150, 200 y 300 palabras para obtener un resumen.
- SweSum: Es un generador de resúmenes multilingüe, propuesto para sueco e inglés. Utiliza múltiples aspectos para valorar las oraciones como: la posición o valor numérico de las oraciones.
- T-Conspectus: Es una aplicación web para resumir textos de un solo documento en inglés, alemán y ruso. Permite realizar resúmenes por porcentajes (va desde el valor 5% hasta el valor 70 %). El proceso se realiza en tres etapas: preprocesamiento, puntuación de resúmenes y generación del resumen.
- Text Compactor: Es una herramienta en línea para la generación automática de resúmenes creada para procesar cantidades grandes de información (Edyburn, 10), está basada en la herramienta Open Text Summarizer. Esta herramienta genera resúmenes de 1% a 100 % del texto.

7. Resultados

Como primer aspecto se realizó la generación de los resúmenes automáticos por las herramientas comerciales. Después de haber obtenido los resúmenes automáticos, se realizó la primera evaluación con el evaluador ROUGE-1. Los resultados obtenidos mostraron que la herramienta que obtuvo los resultados con mayor similitud contra uno realizado por un humano fue la herramienta SweSum, ya que obtuvo un puntaje de acuerdo con el F-measure de 0.43459. Esto significa que el resumen generado por la herramienta coincide en un 43% con un resumen generado por un humano experto.

En la segunda evaluación de los resúmenes automáticos que se realizó con el evaluador ROUGE-2, los resultados obtenidos demostraron que SweSum sigue siendo la herramienta en línea que genera resúmenes automáticos con mayor similitud a los generados por un humano, sin embargo, el puntaje obtenido con este evaluador

disminuyó en un 0.18203 de puntos, dejando a la herramienta con un puntaje final de 0.18203.

Tabla 1. Evaluación de diferentes herramientas comerciales usando ROUGE-1.

Herramienta	Recuerdo	Precisión	F-measure
SweSum	0.43626	0.43313	0.43459
OTS	0.42577	0.42386	0.42458
Text-Compactor	0.42243	0.41954	0.42084
T-Conspectus	0.41595	0.41070	0.41321
Summarizing	0.41050	0.40726	0.40874
Summarizer	0.40947	0.40646	0.40786
Word Office 2007	0.39597	0.39730	0.39651
Word Office 2003	0.39469	0.39618	0.39533

En la figura 3 se observa que la herramienta con mayor puntaje fue SweSum.

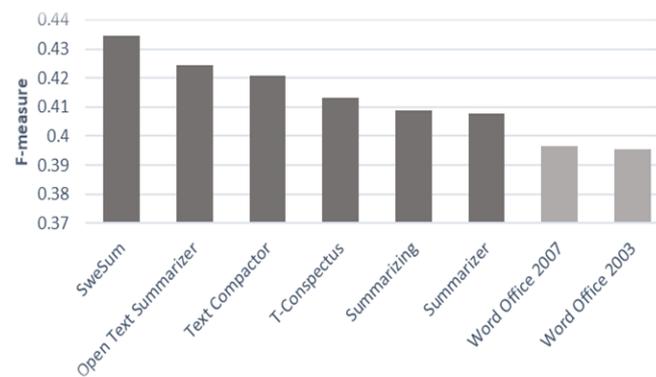


Fig. 2. Comparación del puntaje obtenido de las herramientas comerciales por el evaluador ROUGE-1.

Para las herramientas instalables Office Word 2007 sigue siendo mejor que Office Word 2003 con un puntaje de 0.14050, pero sigue teniendo la posición 8 en el ranking de las herramientas probadas en este artículo.

Tabla 2. Evaluación de diferentes herramientas comerciales usando ROUGE-2.

Herramienta	Recuerdo	Precisión	F-Measure
SweSum	0.18264	0.18153	0.18203
OTS	0.17242	0.17129	0.17177
Text-Compactor	0.16888	0.16759	0.16819
T-Conspectus	0.15718	0.15506	0.15607
Summarizing	0.14658	0.14545	0.14597
Summarizer	0.14526	0.14404	0.14462
Word Office 2007	0.14030	0.14086	0.14054
Word Office 2003	0.13915	0.13971	0.13940

En la tercera evaluación realizada se compara el resumen generado automáticamente contra el texto original y según los datos obtenidos Text Compactor es la mejor herramienta en línea para la GART, con este evaluador la herramienta Word 2007 ahora se posiciona en el lugar número 3.

Tabla 3. Evaluación de diferentes herramientas comerciales usando Jensen Shannon.

Herramienta	SMT	WSMT
Text-Compactor	0.73630	0.71277
Summarizer	0.72611	0.69972
Word Office 2007	0.72378	0.69679
OTS	0.72226	0.69674
SweSum	0.71815	0.69063
Word Office 2003	0.71165	0.68180
Conspectus	0.70925	0.68131
Summarizing	0.70185	0.67105

Las evaluaciones que se realizaron con el evaluador ROUGE-C demuestran que Text-Compactor es la herramienta con mayor similitud al texto original mientras que SweSum para esta evaluación, resulta ser una herramienta de poca competencia.

En la figura 3 se observa que la herramienta SweSum se posicionó en el quinto lugar, mientras que en las evaluaciones pasadas con resumen de referencia era que la se posicionaba en primer lugar.

Tabla 4. Evaluación de diferentes herramientas comerciales usando ROUGE-C.

Herramienta	C-1	C-2	C-3	C-L	C-SU4
Text-Compactor	0.25654	0.25029	0.24575	0.25630	0.24526
Summarizer	0.22946	0.20385	0.17852	0.22939	0.17537
Word 2007	0.22571	0.21244	0.19972	0.22549	0.19714
OTS	0.22570	0.21356	0.20340	0.22491	0.20176
SweSum	0.21080	0.19991	0.18922	0.21055	0.18730
Conspectus	0.20913	0.18442	0.15951	0.20764	0.15549
Word 2003	0.20353	0.19149	0.17966	0.20335	0.17690
Summarizing	0.19141	0.17795	0.16629	0.18711	0.16427

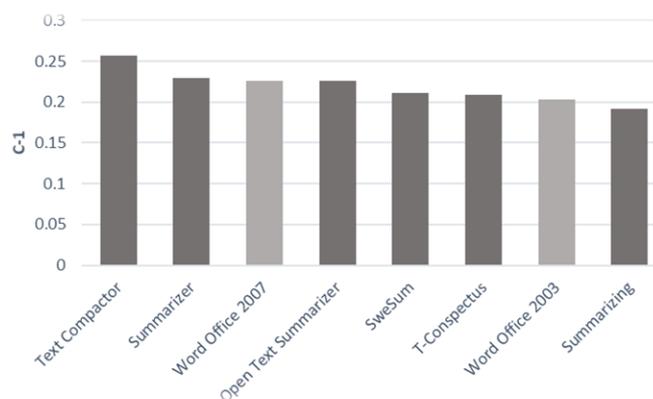


Fig. 3. Comparación de puntaje obtenido por el evaluador ROUGE-C sin resumen de referencia.

8. Conclusiones

En las evaluaciones realizadas con resúmenes de referencia y sin resúmenes de referencia, pudimos observar que la diferencia de puntajes entre evaluadores varía significativamente. Esto es debido a que los evaluadores evalúan aspectos diferentes, mientras los evaluadores con resumen de referencia evalúan la similitud que hay entre un resumen automático y uno generado por un humano. Los evaluadores sin resumen de referencia evalúan la divergencia que existe entre el texto original y el resumen automático. Por lo que se puede decir que el evaluador realiza una evaluación parcial, toma aspectos diferentes a evaluar.

De acuerdo con la evaluación realizada para los evaluadores con resumen de referencia y sin resumen de referencia, podemos apreciar que, mientras el evaluador ROUGE-C arroja puntajes mayores a 0.7, los evaluadores como ROUGE-2 y Jensen Shannon se encuentran por debajo del 0.3 y el evaluador ROUGE-1 se encuentra en un rango de 0.4 y 0.5.

De acuerdo con los puntajes obtenidos por los diferentes evaluadores, se concluye diciendo que las mejores herramientas para la generación automática de resúmenes son las herramientas: Text-Compactor y Open Text Summarizer. Esto es debido a que en 4 experimentos, estas herramientas lograron posicionarse dentro de las 4 mejores lugares, por lo que se determina que estas herramientas cumplen con el mayor número de características evaluadas por los diferentes evaluadores, en comparación con otras herramientas que se posicionaron en las primeras evaluaciones en los últimos lugares y para las últimas evaluaciones se posicionaron dentro de los primeros lugares.

Referencias

1. Sánchez-Arteche, A., Ortiz-López, H.A, Pérez-Rodríguez, N.C., Jiménez-Mercado. I.E., Cue-Vidal, J.C.: Cronología de la escritura y el libro, primera edición, SEP, Fondo editorial del Estado de México (2012)
2. Louis, A., Nenkova, A.: Automatic Summary Evaluation without Human Models. In: Notebook Papers and Results, Text Analysis Conference (TAC-2008), Gaithersburg, Maryland, USA (2008)
3. Louis, A., Nenkova, A.: Automatically Evaluating Content Selection in Summarization without Human Models. Proceedings of the Conference on Empirical Methods in Natural Language Processing: vol. 1, pp. 306–314 (2009)
4. La página de inicio de Article Summarizer, <https://www.summarizing.biz/best-summarizing-strategies/article-summarizer-online/>, ultimo acceso 2019/02/20
5. La página de inicio de Banco Mundial de Datos. <https://datos.bancomundial.org/indicador/ip.jrn.artc.sc>, ultimo acceso 2019/02/30
6. Ledeneva, Y., García-Hernández, R., Gelbukh, A.: Graph Ranking on Maximal Frequent Sequences for Single Extractive Text Summarization. LNCS Springer-Verlag, ISSN 0302-9743, vol. 8404, pp. 466–480 (2014)
7. La página de inicio de Copernic, www.copernic.com/en/products/summarizer, ultimo acceso 2019/02/20
8. Vázquez, E., García Hernández, R.A., Ledeneva, Y.: Learning Relevant Models using Symbolic Regression for Automatic Text Summarization. Computación y Sistemas 23(1), 127–141 (2018)

9. Loret, E., Plaza, L., Aker, A.: The Challenging Task of Summary Evaluation: An overview, Language Resources and Evaluation, vol. 52, pp. 101–148 (2018).
10. Martín Mateos, F.J., Ruiz Reina, J.L.: Procesamiento del Lenguaje natural. Departamento de la Computación e Inteligencia Artificial, Universidad de Sevilla (2013)
11. Matias Mendoza, G.A.: Generación automática de resúmenes independientes del lenguaje. Tesis de Maestría en Ciencias de la Computación, Universidad Autónoma del Estado de México (2016)
12. Matias Mendoza, G.A., Ledeneva, Y., García Hernández, R.A., Sidorov, G.: Evaluación de las Herramientas comerciales y métodos del estado del arte para la generación de resúmenes extractivos individuales. Research in Computing Science, vol. 70, pp. 265–274 (2013)
13. Hoffmann, E.: Defining Information: An Analysis of the information content of documents. Information Processing and Management, vol. 18, pp. 133–139 (1982)
14. Rojas Simón, J., Ledeneva, Y., García, R.: Calculating the Upper Bounds for Multi-Document Summarization using Genetic Algorithms. Computación y Sistemas 22(1), 11–26 (2018)
15. Rojas Simon, J., Ledeneva, Y., García, R.: Calculating the Significance of Automatic Extractive Text Summarization using a Genetic Algorithm. Journal of Intelligent & Fuzzy Systems. Applications in Engineering and Technology 35(1), 293–304 (2018)
16. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: Text summarization branches out: Proceedings of the ACL-04 workshop Vol. 8 (2004)
17. Lin, C.Y., Hovy, E.: Automatic evaluation of summaries using n-gram co-occurrence statistics. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, vol. 1, pp. 71–78 (2003)
18. Lin, C.Y., Och, F.J.: Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, p. 605 (2004)
19. Lin, C.Y., Och, F.J.: Orange: a method for evaluating automatic evaluation metrics for machine translation. In: Proceedings of the 20th international conference on Computational Linguistics. Association for Computational Linguistics, p. 501 (2004)
20. Cortez Vásquez, A., Vega Huerta, H., Pariona Quispe, J.: Procesamiento del Lenguaje Natural. Ingeniería de Sistemas e Informática vol. 6. Facultad de Ingeniería de Sistemas e Informática, julio-diciembre (2009)
21. Alonso Pardo, M.A., Alonso Ramos, M., Gómez Rodríguez, C., Gaña Gil, J., Villares Ferro, J.: La enseñanza del Procesamiento del Lenguaje Natural en la Facultad de Informática y Filología, Investigaciones sobre docencia universitaria y nuevas metodologías, Chapter: 1, pp.13–18 (2012)
22. La página de inicio de Open Text Summarizer <https://open-text-summarizer.soft112.com/>, último acceso 2019/02/20
23. La página de inicio de Microsoft Office Word <https://products.office.com/es-mx/word>, ultimo acceso 2019/02/20
24. La página de inicio de SweSum <http://swesum.nada.kth.se/index-eng.html>, ultimo acceso 2019/02/20
25. La página de inicio de T-CONSPECTUS <https://tconspectus.pythonanywhere.com/>, ultimo acceso 2019/02/20
26. La página de inicio de Text Compactor <http://textcompactor.com/>, ultimo acceso 2019/02/20
27. La página de inicio de Tools4noobs Summarizer, <https://www.tools4noobs.com/summarize/>, ultimo acceso 2019/02/20
28. Ledeneva, Y., García Hernández: Generación Automática de resúmenes. Retos, propuestas y experimentos. Automatic Generation of text Summaries Challenges Proposals and Experiments. Universidad Autónoma del Estado de México (Ed.)/Cigome (2017)

29. Rojas Sánchez, J.M.: Evaluación de herramientas comerciales y métodos del estado del arte para la generación de resúmenes en idioma ruso. Tesis para obtener el título de ingeniero en software. Universidad Autónoma del Estado de México (2016)
30. Hahn, U., Mani, I.: The challenges of automatic summarization. *Computer* 33, 29–36 (2000)
31. Hovy, E., Lin, C.Y.: Automated text summarization and the SUMMARIST system. In: *Proceedings of a Workshop held at Baltimore, Maryland: October 13-15, 1998, Association for Computational Linguistics*, pp. 197–214 (1998)
32. Reddy, P., Balabantaray, R.C.: Improvisation of the document summarization by combining the IR techniques with code-quantity memory and attention linguistic principles. *Procedia Technology* 6, pp 118–125 (2012)
33. García Hernández, R., Montiel, R., Ledeneva, Y., Rendón, E., Gelbukh, A., Cruz, R.: *Text Summarization by Sentence Extraction Using Unsupervised Learning*. Springer Berlin, Heidelberg, pp. 133–143 (2008)
34. Gillick, D., Favre, B., Hakkani-Tur, D., Bohnet, B., Liuand, Y., Xie, S.: *The ICSI/UTD Summarization System at TAC2009* (2009)

Modelos neuronales pulsantes adaptados para el mejoramiento de luminosidad de imágenes cerebrales de gran resolución

Manuel Mejía-Lavalle¹, Kevin Aguilar¹, Humberto Sossa², Dante Mujica¹,
Andrea Magadan¹

¹Tecnológico Nacional de México / CENIDET, Cuernavaca, Morelos, México

²Instituto Politécnico Nacional / CIC, Ciudad de México, México

{mlavalle, kevin.aguilar17ca, dantemv, magadan}@cenidet.edu.mx,
hsossa@cic.ipn.mx

Resumen. En este artículo se proponen, implementan y experimenta con dos Redes Neuronales Pulsantes basadas en el Modelo de Intersección Cortical para mejorar la luminosidad en imágenes médicas del cerebro humano. Las imágenes digitales son ampliamente utilizadas en el área de medicina, pero éstas pueden ser degradadas por diversos factores. La degradación en su luminosidad genera un problema para su correcto análisis, ya que tienen un rango dinámico corto y bajo contraste. La necesidad de obtener imágenes de buena calidad y la tendencia del aumento en la resolución de éstas, exigen nuevas técnicas para mejorar la calidad en menor tiempo. Por eso es necesario buscar paradigmas que puedan aprovechar el cómputo en paralelo como los son las Redes Neuronales Artificiales Pulsantes o de Tercera Generación. La experimentación realizada muestra que los métodos propuestos son altamente competitivos comparados contra otras técnicas mencionadas en la literatura especializada.

Palabras clave: redes neuronales artificiales, redes neuronales pulsantes, modelo de intersección cortical, mejoramiento de imágenes cerebrales, imágenes de gran resolución.

Pulsed Neural Models Adapted for the Enhancement of High Resolution Brain Images Luminosity

Abstract. In this paper it is propose, implement and experiment with two Pulsed Neural Networks based on the Intersection Cortical Model for the enhancement of human brain medical images luminosity. Digital images are widely used in the medicine area, but these can be degraded by several factors. The degradation in its luminosity generates a problem for its correct analysis, since they have a short dynamic range and low contrast. The actual necessity to obtain good quality images joined to the resolution increase tendency, demand new techniques to improve the quality, but in less time. That is why it is necessary to search for paradigms that can take advantage of parallel computing such as Pulse-Coupled Artificial Neural Networks. Experimentation shows that the proposed methods are highly competitive compared versus other techniques mentioned in the specialized literature.

Keywords: artificial neural networks, pulsed neural networks, intersection cortical model, brain images enhancement, high resolution images.

1. Introducción

1.1. Motivación - problemática

Las imágenes digitales son utilizadas en diversas áreas, una de ellas es el área de la medicina, que utiliza una gran variedad de estudios por imágenes como lo son los estudios de resonancia magnética. Sin embargo, la calidad de las imágenes puede verse afectada por muchos factores. Estas afectaciones dificultan el análisis o cualquier otro proceso que se requiera realizar con la imagen. La calidad de una imagen es descrita por las siguientes características: nitidez, contraste, color, claridad y artefactos [1]. La mala luminosidad es un claro ejemplo de los factores que afectan a la imagen; ésta generalmente afecta el contraste y la claridad de la imagen, ya que presentan acumulaciones de píxeles con niveles cromáticos altos o bajos.

Además de lo anterior, se tiene el problema de que cada vez la resolución de las imágenes es mayor (Fig. 1), por lo que el tiempo de procesamiento para mejorar este tipo de imágenes tiende a crecer, lo cual no es satisfactorio cuando hablamos de procesar cientos de miles de imágenes, como es el caso del área médica.

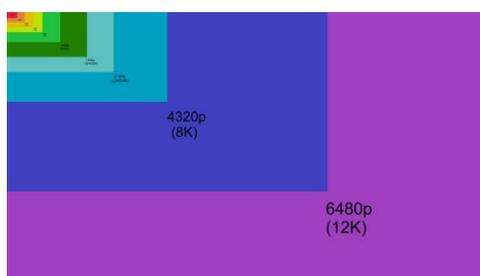


Fig. 1. Evolución de la resolución en imágenes digitales [2].

1.2. Intentos anteriores – estado del arte

Actualmente existe una gran variedad de técnicas que tratan este problema; uno de los más utilizados es la Ecuación del Histograma (EH), por su fácil implementación y buenos resultados. El histograma de una imagen es una función discreta que representa el número de píxeles en la imagen en función al nivel de intensidad [3]. El objetivo de la Ecuación del Histograma es distribuir de forma uniforme el valor que toma cada uno de los píxeles a lo largo del rango dinámico, obteniendo como resultado una imagen con mejor contraste. Al ser secuencial, el tiempo de procesamiento puede ser alto o prohibitivo para algunas aplicaciones, como es el caso médico.

Otra técnica tradicionalmente utilizada es *Power-Law Transform* (PLT) [4], siendo una técnica rápida y sencilla, pero sigue siendo relativamente ineficiente; sin embargo, siguen surgiendo nuevas técnicas y variaciones de las mismas, tal es el caso de *Contrast*

Limited Adaptive Histogram Equalization (CLAHE), siendo una variación de EH, usado en [5], para mejorar imágenes submarinas.

A pesar de la existencia de diferentes técnicas para abordar este problema, se siguen investigando y desarrollando nuevas, ya que existen diferentes problemáticas a resolver como lo es la definición de las imágenes, siendo éste un factor a considerar en la calidad de una imagen; se requiere mayor resolución para obtener la mejor calidad posible. Actualmente existen distintos formatos y resolución [6]. Estas nuevas resoluciones aumentan el tamaño de las imágenes y con ello su tiempo de procesamiento, por lo que existe la necesidad de procesar en un tiempo aceptable toda esta información en la imagen que tiende a ir en aumento. Esto exige nuevas técnicas capaces de procesar la mayor cantidad de información en el menor tiempo posible y obteniendo buenos resultados.

En este sentido, otras técnicas no tradicionales utilizadas en el procesamiento de imágenes son las Redes Neuronales Artificiales (RNAs), que han entregado buenos resultados [7]. Las Redes Neuronales Pulsantes o Pulso-Acopladas (*Pulse-Coupled Neural Network PCNN*) son una técnica que se ha utilizado en el área de Visión Artificial; este paradigma se basa en gran medida en el modelo de Eckhorn, que parte de la frecuencia de activación de las neuronas de la corteza visual [8]. Esto les permite cumplir de cierto modo algunas características visuales humanas [9], a diferencia de otras técnicas más matemáticas que pueden encontrarse en [10].

Del modelo PCNN surgen otras variantes como lo es el Modelo de Intersección Cortical (*Intersection Cortical Model ICM*) que es una versión más simple y está especialmente diseñado para el procesamiento de imágenes de manera eficiente. Fue presentado como una herramienta para el procesamiento de imágenes que es computacionalmente más rápido que el modelo PCNN completo. La característica de la auto-onda le permite procesar una vecindad de píxeles de manera más cercana a como lo hace la visión biológica. Algunas de sus aplicaciones han sido en la detección de objetos [11], la detección de ruido aditivo y gaussiano [12] y la detección de bordes [13]. Este modelo ICM, al igual que el modelo PCNN, entrega imágenes binarias que poseen una especie de meta-información o meta-conocimiento de la imagen y que puede ser utilizada para mejorarla. La introducción de la Matriz de Tiempo en un modelo de Red Neuronal Pulsante se ha utilizado para mejorar distintas características de la imagen como se muestra en [12,13], ya que ésta da información sobre el tiempo de disparo de las neuronas, obteniendo así información adicional de la imagen.

1.3. Propuesta de solución - aportación

En este trabajo se emplea una ICM pero con una Matriz de Tiempo (MT) para obtener meta-información o meta-conocimiento de la intensidad luminosa de los píxeles y ciertas zonas en la imagen, logrando así modificar y mejorar los valores cromáticos de la imagen. Con esta meta-información se alcanza una mejor distribución a lo largo del rango dinámico, sin perder zonas de interés y evitando introducir ruido a la imagen, logrando de esta forma una imagen con un mejor contraste y luminosidad.

Por otra parte, se presenta una configuración en los parámetros del modelo ICM para obtener un mejor desempeño con la Matriz de Tiempo, así mismo se presenta una modificación en la Matriz de Pesos para reducir la propagación de la auto-onda. Finalmente se presentan los diferentes resultados obtenidos con dos modelos

propuestos basados en ICM, que incluyen algunas variaciones en la normalización y pesos.

Este artículo está organizado de la siguiente manera: en la Sección dos se presenta el modelo utilizado y la Matriz de Tiempo; en la Sección tres se describen los algoritmos propuestos; en la Sección cuatro se presenta la experimentación y resultados obtenidos, en particular se describen las métricas de evaluación utilizadas y se muestran los resultados de la evaluación de los experimentos, que arrojan que lo propuesto es altamente competitivo al ser comparado contra otras técnicas actuales presentadas en la literatura especializada; en la quinta Sección se presenta una discusión y finalmente las conclusiones y trabajo futuro es mencionado en la Sección seis.

2. Descripción del Modelo de Intersección Cortical (ICM)

2.1. Modelo básico ICM

El modelo neuronal ICM minimiza el costo del cálculo, manteniendo a la vez la efectividad del modelo cortical cuando se aplica a imágenes. Se basa en los elementos comunes de varios modelos biológicos [14]. Es un modelo simplificado del modelo PCNN, siendo la Red Neuronal ICM un caso especial de la Red PCNN en la cual no hay neuronas de enlace (*linking*). La ICM sólo utiliza entradas de alimentación (*feeding*). Es decir, si la fuerza de enlace de la PCNN se establece en ceros, se convertirá en la ICM. En la Fig. 2, se muestra el diagrama de la ICM.

La señal F_{ij} se obtiene mediante el producto del parámetro f y la señal F_{ij} dada en un estado anterior; más la ponderación de la última salida del vecindario de neuronas Y_{kl} por una matriz w_{ijkl} (en este caso i, j se refiere a las coordenadas del pixel que se está analizando y que corresponde a una neurona de ICM, mientras que k, l se refieren a los pixeles vecinos de i, j , por lo que la matriz W propiamente es de tamaño k, l) y por el estímulo externo de entrada S_{ij} . El umbral dinámico T_{ij} crece obedeciendo a h cuando su salida se activa y mantiene su estado previo con una atenuación dada por g , lo cual da origen a la formación de pulsos. El modelo ICM se describe con las ecuaciones (1), (2) y (3).

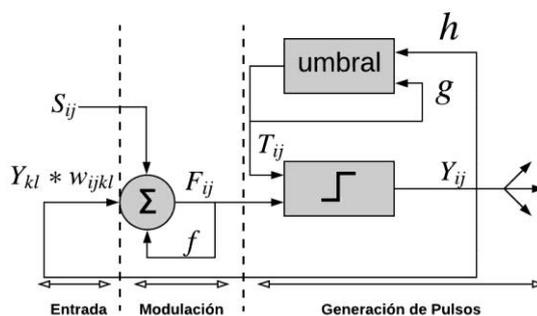


Fig. 2. Diagrama ICM [12].

$$F_{ij}[n] = fF_{ij}[n - 1] + \sum w_{ijkl}Y_{kl}[n - 1] + S_{ij}, \quad (1)$$

$$T_{ij}[n] = gT_{ij}[n - 1] + hY_{ij}[n - 1], \quad (2)$$

$$Y_{ij}[n] = \begin{cases} 1 & \text{si } F_{ij}[n] > T_{ij}[n], \\ 0 & \text{en otro caso} \end{cases}, \quad (3)$$

donde n es la iteración actual, w es la matriz de pesos sinápticos generalmente gaussiana de 3×3 , como se muestra en la ecuación (4), que liga una neurona con sus vecinas y finalmente f, g y h son coeficientes de ajuste, donde $g < 1.0$, $f < g$ y h es un valor cercano a 20, según se recomienda en [14]:

$$w_{kl} = \begin{bmatrix} 0.25 & 0.5 & 0.25 \\ 0.5 & 0 & 0.5 \\ 0.25 & 0.5 & 0.25 \end{bmatrix}. \quad (4)$$

2.2. ICM con Matriz de Tiempo o Meta-conocimiento

En este artículo se emplea una ICM con una Matriz de Tiempo (MT) para obtener meta-información o meta-conocimiento de la intensidad luminosa de los pixeles además de algunas áreas específicas en la imagen. La MT se define para registrar la iteración o tiempo de activación para todas las neuronas Y_{ij} , como se muestra en la Fig. 3.

Si la neurona se dispara se registrará el tiempo en que se disparó [9]. Por tanto, un pequeño valor en la Matriz de Tiempo indica que el píxel correspondiente tiene una alta intensidad y pertenece a una región más brillante. De modo contrario, un gran valor indica que el píxel correspondiente tiene una intensidad baja y pertenece a una región más oscura. La Matriz de Tiempo se define con la ecuación (5).

$$M_{ij}[n] = \begin{cases} n & \text{si } Y_{ij} = 1, \\ M_{ij}[n - 1] & \text{en otro caso.} \end{cases} \quad (5)$$

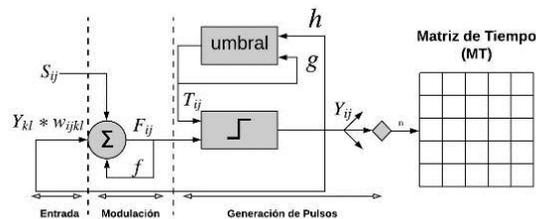


Fig. 3. Diagrama del modelo ICM-MT [12].

3. Aportación: Modelos neuronales para mejorar imágenes médicas de gran resolución

A continuación, se proponen y presentan dos modelos neuronales pulsantes diseñados para procesar eficientemente imágenes médicas de gran resolución que están, ó muy luminosas, o muy oscuras.

3.1. Intersection Cortical Model – Luminance Time Matrix ICM-LTM

El proceso del algoritmo propuesto denominado Modelo de Intersección Cortical con Matriz de Tiempo Luminosa (*Intersection Cortical Model - Luminance Time Matrix ICM-LTM*) se inspira en ICM, sólo que se anexa la Matriz de Tiempo donde se registrará el número de iteración n de activación de cada una de las neuronas de salida Y_{ij} . El proceso del algoritmo se muestra en Fig. 4.

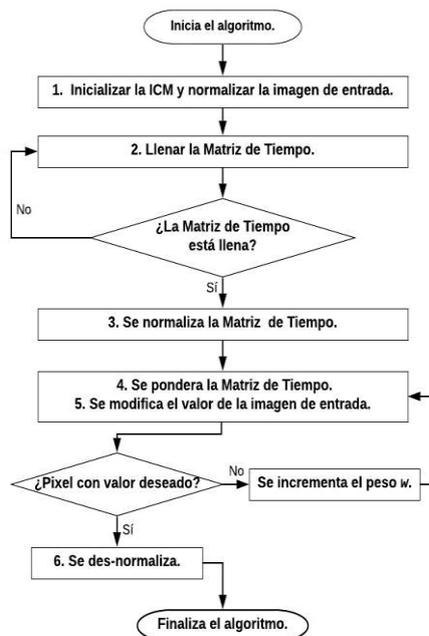


Fig. 4. Diagrama del modelo neuronal propuesto ICM-LTM.

Los seis pasos del algoritmo ICM-LTM se describen a continuación con un ejemplo de caso de imagen oscura.

Paso 1. El proceso del algoritmo inicia estableciendo los parámetros de la ICM, f , g y h , del mismo modo se inicializan las matrices F_{ij} , Y_{ij} , T_{ij} y S_{ij} , donde S_{ij} se inicializa con los valores de la imagen de entrada normalizados, con respecto al valor máximo que puede tomar el pixel, en este caso al utilizar imágenes de 8 bits (255). Es decir, su

valor original dividido entre 255, obteniendo así una matriz con valores entre cero y uno como se muestra en Fig. 5.

Paso 2. El proceso para obtener la Matriz de Tiempo M_{ij} es mostrado en la Fig. 6. Cuando la salida Y_{ij} de la ICM es activada por primera vez se registra en la MT la iteración de activación. La ICM itera hasta conseguir que se disparen todas las neuronas. Esto da como resultado una MT con valores entre uno y el número de iteraciones requeridas para activar todas las neuronas (11 en este ejemplo), donde uno corresponde a las neuronas que se dispararon en la primera iteración y 11 corresponden a las últimas neuronas en activarse. Las primeras neuronas en activarse corresponden a los píxeles más luminosos y las últimas corresponden a los píxeles más oscuros.

Paso 3. La MT se normaliza dividiendo el valor en la MT y las iteraciones requeridas, restándole a ambas partes la primera época de activación; así la MT normalizada (MTN) queda con valores entre cero y uno (Fig. 7); con imágenes oscuras se modifica el valor restando uno al valor de la MT normalizada (Fig. 8).

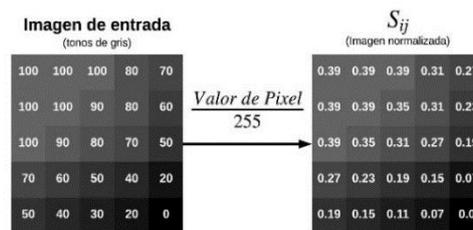


Fig. 5. Normalización de la imagen de entrada.

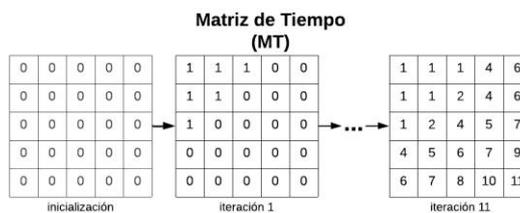


Fig. 6. Llenado de la Matriz de Tiempo.

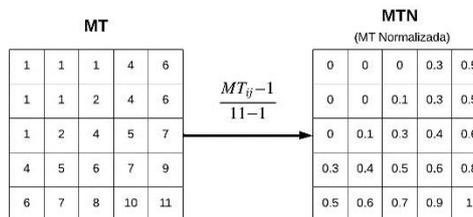


Fig. 7. Normalización de MT.

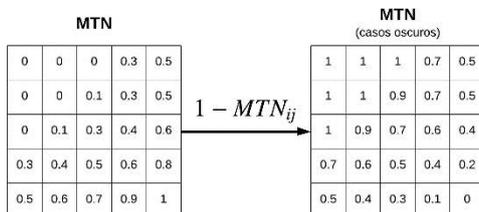


Fig. 8. MT normalizada (MTN) para imágenes oscuras.

Paso 4. Posteriormente, los valores de la Matriz de Tiempo normalizados se multiplican por un peso w inicialmente de 0.1 para todos los pixeles con la finalidad de no modificar demasiado los valores de la imagen de entrada, obteniendo una Matriz de Tiempo ponderada (MTP) como se muestra en la Fig. 9.

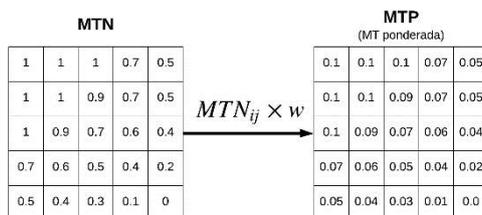


Fig. 9. Obtención de la MT ponderada (MTP).

Paso 5. El valor obtenido de esta multiplicación se utiliza para disminuir los valores de los pixeles de la imagen de entrada normalizada, en el caso de las imágenes luminosas, o incrementar en caso de las imágenes oscuras, es decir, a la matriz S_{ij} se le resta o suma la MT ponderada, obteniendo como resultado el nuevo valor normalizado del pixel, la Fig. 10 muestra el resultado de una imagen oscura.

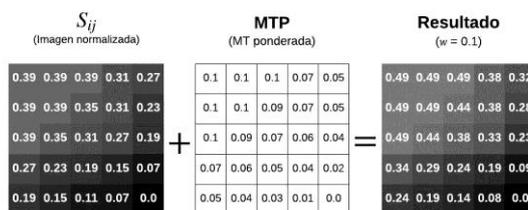


Fig. 10. Resultado $w = 0.1$.

El proceso de obtención de la Matriz Ponderada se repite modificando el valor del peso w hasta conseguir que por lo menos el valor de un pixel llegue al valor deseado, en este caso al ser una imagen oscura que cuenta con valores en cero se busca el uno (Fig. 11). Esto es con la finalidad de abarcar todo el rango dinámico de la imagen, siendo uno equivalente a 255 al des-normalizar.

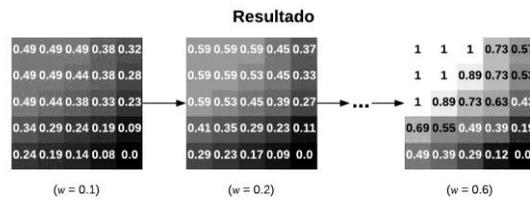


Fig. 11. Mejoramiento gradual del resultado.

Paso 6. Finalmente, el valor obtenido como resultado se des-normaliza multiplicándolo por 255, dando como resultado la imagen mejorada final (Fig. 12). La comparación de la imagen mejorada y la resultante de este ejemplo es mostrada en la Fig. 13.

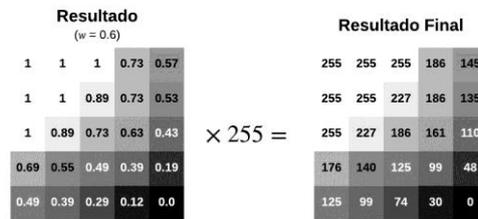


Fig. 12. Des-normalización de la imagen.

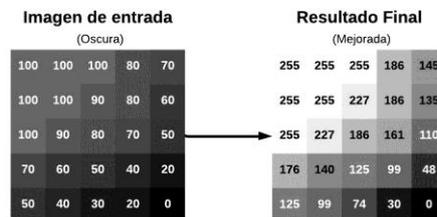


Fig. 13. Comparación de la entrada oscura y la salida mejorada.

3.2. Intersection Cortical Model – Luminance Time Matrix2 ICM-LTM2

El modelo que se denominó ICM-LTM2 es una modificación al algoritmo ICM-LTM, con la finalidad de mejorar en tiempo y resultados. Se realizaron tres modificaciones: dos en el Paso uno, específicamente en la matriz de pesos de ICM y la normalización de la imagen de entrada y una modificación más en el Paso 4, donde se eliminaron las iteraciones para encontrar el peso adecuado. La primera modificación se realizó en la matriz de pesos, con el fin de reducir la onda expansiva de la ICM, ya que esto provocaba un destello en los bordes lo que afecta su calidad.

La segunda modificación se emplea en la normalización de la imagen de entrada normalizándola de acuerdo al rango dinámico de la imagen, es decir de acuerdo con el valor máximo que aparece en la imagen y el valor mínimo en la escala de 0 a 255,

logrando así reducir el tiempo requerido para obtener la MT. Finalmente se calculó directamente el peso w requerido, estableciéndolo igual al valor mínimo en la imagen de entrada para imágenes luminosas y como la diferencia entre uno y el valor máximo de la imagen de entrada para imágenes oscuras; esto se realizó ya que el valor máximo que toma la MT ponderada es igual a uno y esto afectará más a los valores máximos para obtener un 255 a la salida en imágenes oscuras, o a los mínimos para obtener un cero en la salida en imágenes luminosas según sea el caso. Logrando de esta manera eliminar las iteraciones realizadas para encontrarlo y con ello reducir el tiempo de procesamiento.

4. Experimentación y resultados

Para la experimentación se tomaron tres estudios de resonancia magnética con 150 imágenes cada uno en tres modalidades: PD, T1 y T2, simuladas en *BrainWeb* [15-17]. Estas imágenes fueron desplazadas utilizando la técnica de Desplazamiento del Histograma (DH) [3], en: 20, 40, 60, 80, 100, 120 y 140 del valor cromático del píxel, tanto positivo como negativo, obteniendo así un total de 6,300 imágenes para la experimentación. Se establecieron los valores de M_{ij} , Y_{ij} y F_{ij} , en cero, del tamaño de las imágenes de experimentación. La matriz T_{ij} se inicializó en 1, del mismo tamaño que las imágenes de entrada. El parámetro f se estableció en 0.1, g en 0.9 y h en 20 tanto para ICM-LTM como para ICM-LTM2. Los parámetros fueron seleccionados de experimentaciones previas considerando la información que entregaba cada iteración de la ICM con respecto a la intensidad de los píxeles, el tiempo y distribución de los valores de la Matriz de Tiempo. La experimentación y los resultados se dividieron en tres etapas: la etapa 1), donde se realizaron experimentos con imágenes en modalidad PD; la etapa 2), donde se realizaron experimentos con la modalidad T1; y finalmente la etapa 3), donde se experimentó con imágenes en modalidad T2.

Los métodos utilizados para el mejoramiento de la imagen dependen generalmente de las necesidades especiales para algunas aplicaciones. En la mayoría de los casos, los efectos de mejora son evaluados por la percepción visual, difícilmente se llegan a evaluar de forma objetiva por medio de una métrica, a pesar de ello, existen diferentes métricas para evaluar la calidad de una imagen tomando en cuenta nitidez, contraste, color, claridad y artefactos. Se han usado diferentes modelos multidimensionales para evaluar la calidad a partir de las medidas obtenidas sobre los atributos; la calidad en este tipo de modelos se puede aproximar como una función que calcula la distancia euclídea de las imágenes degradadas con respecto a una imagen ideal en un espacio n dimensional, como se muestra en la ecuación (6) [18]:

$$C(x,y) = \sqrt{\sum_{i=1}^n ((x_i - y_i) * p_i)^2}, \quad (6)$$

donde la calidad C es entendida como la distancia euclidiana entre una imagen digital x y su imagen referente ideal y , a partir de sus atributos i , ponderados mediante sus coeficientes de ponderación p .

Otro método utilizado es la Métrica de Similitud Estructural (*SSIM*) que se usa para medir la similitud entre dos imágenes considerando la luminancia, contraste y

estructura, combinando las tres y simplificando se obtiene la ecuación (7)[19]. Es una métrica de referencia completa por lo que toma una imagen de referencia siendo ésta la imagen sin distorsionar:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (7)$$

donde μ_x y μ_y son el promedio de cada una de las imágenes, σ_x^2 y σ_y^2 es la varianza de las imágenes, σ_{xy} es la covarianza, $C_1 = (k_1L)^2$ y $C_2 = (k_2L)^2$, donde $L = 255$, $k_1 = 0.01$ y $k_2 = 0.03$ por defecto.

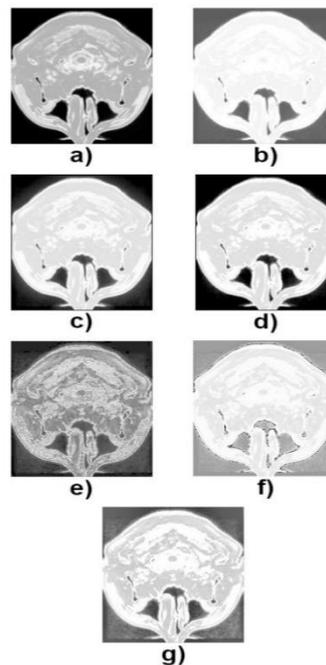


Fig. 14. Resultados PD luminosa, a) imagen original, b) imagen desplazada en +60, resultados con los algoritmos, c) ICM-LTM, d) ICM-LTM2, e) EH, f) PLT, g) CLAHE.

Se iniciaron los experimentos con imágenes médicas en la modalidad PD, comenzando con las imágenes luminosas; se aplicaron los 7 desplazamientos antes mencionados de forma positiva en las 150 imágenes del estudio. Las imágenes fueron procesadas por los algoritmos: ICM-LTM, ICM-LTM2, EH [3], PLT[4], CLAHE[5]. Uno de los resultados obtenidos en esta experimentación se muestra en la Fig. 14, donde se observa que las imágenes obtenidas por los modelos propuestos visualmente se asemejan más a la imagen original, al igual que CLAHE, sin embargo, a pesar de entregar un buen contraste, éste presenta más ruido en la imagen resultante. Algo similar se hizo para las modalidades T1 y T2.

Los resultados de la evaluación de las tres etapas de experimentación antes mencionadas se muestran a continuación, considerando el promedio de las 150 imágenes de cada estudio en cada uno de los siete diferentes desplazamientos del

histograma (DH), con las cinco técnicas empleadas, ICM-LTM, ICM-LTM2, EH, PLT y CLAHE evaluadas con la métrica *C*, que presenta un mejor resultado mientras más cercano a cero (Tabla 1) y *SSIM*, que se entiende como mejor resultado el más cercano a uno (Tabla 2); la evaluación general considera el promedio de los tres estudios en cada uno de los desplazamientos con las dos métricas utilizadas. Los mejores resultados evaluados según las métricas son marcados en negritas subrayadas, y los peores en cursivas subrayadas. Las Figs. 15 y 16 muestran esto de manera gráfica.

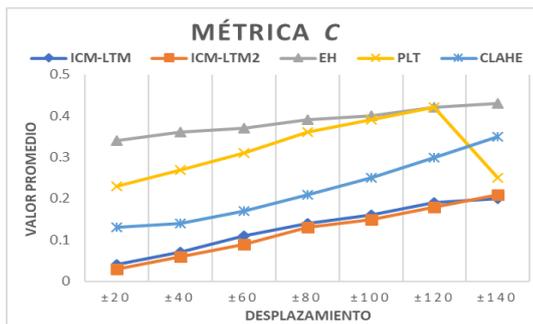


Fig. 15. Gráfica evaluación *C*.

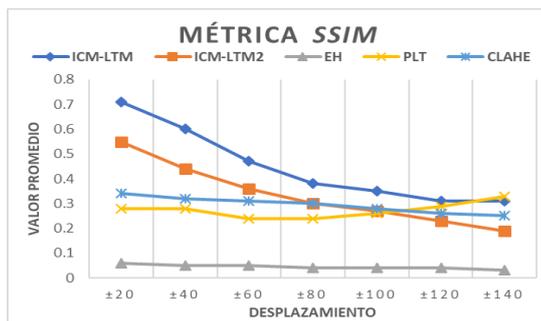


Fig. 16. Gráfica evaluación *SSIM*.

Tabla 1. Evaluación general con la métrica *C*.

DH	ICM-LTM	ICM-LTM2	EH	PLT	CLAHE
20	0.04	<u>0.03</u>	<i><u>0.34</u></i>	0.23	0.13
40	0.07	<u>0.06</u>	<i><u>0.36</u></i>	0.27	0.14
60	0.11	<u>0.09</u>	<i><u>0.37</u></i>	0.31	0.17
80	0.14	<u>0.13</u>	<i><u>0.39</u></i>	0.36	0.21
100	0.16	<u>0.15</u>	<i><u>0.40</u></i>	0.39	0.25
120	0.19	<u>0.18</u>	<i><u>0.42</u></i>	<i><u>0.42</u></i>	0.30
140	<u>0.20</u>	0.21	<i><u>0.43</u></i>	0.25	0.35
Prom.	0.13	<u>0.12</u>	<i><u>0.39</u></i>	0.32	0.22

Tabla 2. Evaluación general con la métrica *SSIM*.

DH	ICM-LTM	ICM-LTM2	EH	PLT	CLAHE
20	0.71	0.55	<u>0.06</u>	0.28	0.34
40	0.60	0.44	<u>0.05</u>	0.28	0.32
60	0.47	0.36	<u>0.05</u>	0.24	0.31
80	0.38	0.30	<u>0.04</u>	0.24	0.30
100	0.35	0.27	<u>0.04</u>	0.26	0.28
120	0.31	0.23	<u>0.04</u>	0.29	0.26
140	0.31	0.19	<u>0.03</u>	0.33	0.25
Prom.	0.45	0.33	<u>0.04</u>	0.27	0.29

5. Conclusiones y trabajo futuro

Se obtuvo en promedio un 10% de mejora con la métrica *C* y un 4% de mejora con *SSIM* en comparación con CLAHE, siendo esta la técnica que entregó mejores resultados en promedio. Se observó que las técnicas entregan un mejor desempeño en imágenes luminosas que en imágenes oscuras lo que podría deberse a las modificaciones realizadas para tratar con ellas.

La modificación realizada en ICM-LTM2 ayudó a obtener los resultados en menor tiempo hasta en un 25% más rápido y entregar mejores resultados con respecto a la métrica *C*.

Como trabajo futuro podrían optimizarse los parámetros de la ICM al igual que la Matriz de Pesos w , ya que han sido establecidas de acuerdo a las propuestas en la literatura y experimentos realizados a prueba y error. Así mismo, habrá que realizar experimentos estableciendo el valor deseado mínimo y máximo del pixel de acuerdo al tipo de estudio, ya que el exceso de brillo o falta de él, podría deberse a que la imagen naturalmente no toma valores en 0 ó 255; del mismo modo, trabajar la implementación del algoritmo en paralelo para aprovechar las ventajas que presenta su naturaleza paralela, estimándose ahorros en tiempo en aproximadamente n unidades siendo éste igual al número de núcleos utilizados y menor o igual al número de pixeles en la imagen a procesar. En todo caso, lo presentado y propuesto resulta una alternativa muy competitiva para procesar imágenes de gran resolución.

Referencias

1. Pedersen, M. Bonnier, N. Hardeberg, J.Y. Albrechtsen, F.: Attributes of image quality for color prints. *J. Electron. Imaging* 19(1), 011016.doi:10.1117/1.3277145 (2010)
2. Wikimedia Commons (s.f.): https://commons.wikimedia.org/wiki/File:12K_Screen_Resolution_Column.jpg Último acceso: 2019/04/01
3. Pajares, G., Cruz, J.M. de la.: Ejercicios resueltos de visión por computador. Madrid: Alfaomega (2008)

4. Janani, P., Premaladha, J., Ravichandran, K.S.: Image Enhancement Techniques: A Study. *Indian Journal of Science and Technology* 8 (22), 12. doi:10.17485/ijst/2015/v8i22/79318 (2015)
5. Ma, J., Fan, X., Yang, S.X., Zhang, X., Zhu, X.: Contrast Limited Adaptive Histogram Equalization-Based Fusion in YIQ and HSI Color Spaces for Underwater Image Enhancement. *International Journal of Pattern Recognition and Artificial Intelligence* 32(07), 1854018. doi:10.1142/S0218001418540186 (2018)
6. Castillo, J.M.: El futuro de la televisión. *Comunicación y hombre: revista interdisciplinar de ciencias de la comunicación y humanidades* (11), 83–100 (2015)
7. Jena, M., Mishra, S.: Review of Neural Network Techniques in the Verge of Image Processing. In: *International Proceedings on Advances in soft computing, Intelligent Systems and Applications*, pp. 345–361. Singapore: Springer, Singapore. doi:10.1007/978-981-10-5272-9_33 (2018)
8. Eckhorn, R., Reitboeck, H.J. Arndt, M., Dicke, P.: Feature Linking via Synchronization among Distributed Assemblies: Simulations of Results from Cat Visual Cortex. *Neural Computation* 2(3), 293–307. doi:10.1162/neco.1990.2.3.293 (1990)
9. Ma, Y., Zhan, K., Wang, Z.: *Applications of pulse-coupled neural networks*. Berlin: Springer (2010)
10. Rodríguez Morales, R., Sossa Azuela, J.H.: *Procesamiento y análisis digital de imágenes*. México: Alfaomega (2011)
11. Ekblad, U., Kinser, J.M., Atmer, J., Zetterlund, N.: The intersecting cortical model in image processing. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 525(1–2), 392–396. doi:10.1016/J.NIMA.2004.03.102 (2004)
12. Ortiz, E., Mejía-Lavalle, M., Sossa, H.: Using Pulse Coupled Neural Networks to Improve Image Filtering Contaminated with Gaussian Noise. *Computación y Sistemas* 21(2), pp. 381–395. doi: 10.13053/CyS-21-2-2742 (2017)
13. Cardenas, S.Y., Mejía-Lavalle, M., Sossa, H., Pardo, E.C.: Pulse-Coupled Neural Networks applied to Human Brain Image Processing. In: *2014 International Conference on mechatronics, electronics and automotive engineering*, pp. 60–65. IEEE. doi:10.1109/ICMEAE.2014.46 (2014)
14. Lindblad, T., Kinser, J.M.: *Image processing using pulse-coupled neural networks*. Heidelberg New York Dordrecht London: Springer (2005)
15. Kwan, R.K.-S., Evans, A.C., Pike, G.B.: An Extensible MRI Simulator for Post-Processing Evaluation. *Visualization in Biomedical Computing* 1131, 135–140 (1996)
16. Kwan, R.K.-S., Evans, A.C., Pike, G.B.: MRI simulation-based evaluation of image-processing and classification methods. *IEEE Transactions on medical imaging* 18(11), 1085–97 (1999)
17. Collins, D.L., Zijdenbos, A.P., Kollokian, V., Sled, J.G., Kabani, N.J., Holmes, C.J., Evans, A.C.: Design and Construction of a Realistic Digital Brain Phantom. *IEEE Transactions on Medical Imaging* 17(3), 463–468 (1998)
18. Robledano-Arillo, J., Moreno-Pelayo, V., Pereira-Uzal, J.M.: Aproximación experimental al uso de métricas objetivas para la estimación de calidad cromática en la digitalización de patrimonio documental gráfico. *Revista Española de Documentación Científica* 39(2), e128. doi:10.3989/redc.2016.2.1249 (2016)
19. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on image processing* 13(4), 1–14 (2004)

Modelo neurodifuso para el control de un exoesqueleto para rehabilitación de brazo en pacientes con EVC

Santos M. Orozco-Soto¹, Alberto I. Pérez-Sanpablo^{1,2}, Elisa Romero-Ávila³, Catherine Disselhorst-Klug³, Juan M. Ibarra-Zannatha¹

¹ CINVESTAV, Departamento de Control Automático, Ciudad de México, México

² Instituto Nacional de Rehabilitación, Ciudad de México, México

³ Universidad Técnica de Aachen, Departamento de Ingeniería de Rehabilitación y Prevención, Aachen, Alemania

sorozco@ctrl.cinvestav.mx, albperez@inr.gob.mx,
{jibarra,alberto.perez}@cinvestav.mx,
{romero,disselhorst-klug}@ame.rwth-aachen.de

Resumen. En este trabajo se presenta un modelo neurodifuso para el comportamiento muscular de los movimientos de flexión y extensión del codo, utilizando electromiografía de superficie. Adicional a la electromiografía, se considera la dirección del movimiento para determinar el tipo de actividad y la correcta activación muscular. El modelo neurodifuso presentado funciona como un activador de autoridad para un exoesqueleto robótico que pretende usarse en la rehabilitación del brazo en pacientes con EVC. Se utilizaron datos de personas sanas y de pacientes para desarrollar y evaluar el modelo neurodifuso, mostrando resultados satisfactorios, ya que la activación solo se presenta al utilizar las señales de pacientes que realizaron extensión de codo.

Palabras clave: modelado y control neurodifuso, exoesqueleto robótico, robótica de rehabilitación.

Neural-Fuzzy Model for Control of an Exoskeleton for Arm Rehabilitation in Stroke

Abstract. This paper presents a neural-fuzzy model for health behavior of elbow flexion and extension motions by means of surface electromyography. In addition to electromyography, the direction of the motion is considered to determine the type of motion and the correct muscular activation. The presented neural-fuzzy model works as the trigger for the authority of a robotic exoskeleton for arm rehabilitation of patients with stroke. Data of healthy persons and stroke patient were used for developing and testing the neural-fuzzy model, showing successful results, since the triggering control is present only with patient signals while performing elbow extension.

Keywords: neural-fuzzy modelling and control, robotic exoskeleton, rehabilitation robotics.

1. Introducción

El evento vascular cerebral (EVC) es un problema de salud mundial que puede causar discapacidad de los miembros superiores, los cuales pueden requerir rehabilitación para recuperar la movilidad parcial [1]. Para abordar dicho problema, se han implementado tecnologías robóticas, tales como los dispositivos hápticos o exoesqueletos, con el fin de mejorar y asistir la rehabilitación de miembros superiores [2,3].

Durante el proceso de rehabilitación, es importante detectar la intención de la persona de ejecutar un movimiento, ya sea para motivar o mejorar dicha acción [4]. La electromiografía de superficie (sEMG) es un método conveniente para valorar la intención de movimiento de un sujeto, no obstante, las señales de sEMG que provienen de los sujetos sanos difieren significativamente de aquellas que se leen en los pacientes con EVC; esto se debe a la falta de fuerza, falta de control muscular, coactivación muscular patológica, incremento de actividad muscular de fondo, alteración del tono muscular o espasticidad posteriores al EVC. Lo mencionado anteriormente se puede apreciar en la Fig. 1 y Fig. 2, en las cuales se presentan señales de sEMG tanto de personas sanas, como de pacientes, respectivamente. Se han propuesto diversas técnicas para la extracción de información a partir de señales de sEMG [5]–[13], sin embargo, la utilidad de los resultados es todavía controversial [4].

En este trabajo se propone un modelo neurodifuso para control de un exoesqueleto para brazo, el cual pretende ser utilizado para la rehabilitación de sujetos que han sufrido EVC. Dicho exoesqueleto, por el momento está pensado para la articulación del codo como una prueba de concepto, la cual, eventualmente se puede extender para otras articulaciones. La articulación del codo se puede representar como una articulación de un grado de libertad (GDL) rotacional, la cual es accionada por dos músculos cuyo movimiento puede ser evaluado utilizando sEMG. La flexión del codo es llevada a cabo por el bíceps brachii y la extensión por el tríceps brachii. En consecuencia, el modelo neurodifuso presentado utiliza datos grabados de personas sanas y de pacientes con EVC para establecer cuando el exoesqueleto debe asistir el movimiento del codo. Se decidió utilizar esta técnica de modelado debido a que es una metodología probada para la clasificación de señales biológicas [3,4].

Se analizaron datos de 14 sujetos sanos y de 4 pacientes con EVC, quienes realizaron movimientos de flexión y extensión para otro estudio realizado previamente [14]. Los datos se obtuvieron utilizando sEMG bipolar, de acuerdo con las recomendaciones de SENIAM. Dichas señales fueron filtradas (Butterworth, 10–450 Hz), rectificadas, suavizadas (media móvil con ventana de 80 ms) y normalizadas para obtener los perfiles de activación muscular. La posición angular del codo fue medida por medio de un sistema óptico seguidor de movimiento (VICON MX), utilizando el modelo biomecánico de miembro superior

introducido por Williams et al. [15]. Asimismo, la posición del codo fue filtrada (Butterworth de 4° orden 6.6 Hz). La Fig. 1 presenta las señales de sEMG de una persona sana mientras realiza movimientos de flexión y extensión del codo, correspondientes a la activación de bíceps y tríceps, respectivamente. Además, se puede observar la señal de la posición angular suministrada por el seguidor de movimiento. Las mismas variables mencionadas, pero con movimientos de los pacientes con EVC, se pueden observar en la Fig 2. A continuación, se detalla el desarrollo del modelo neurodifuso.

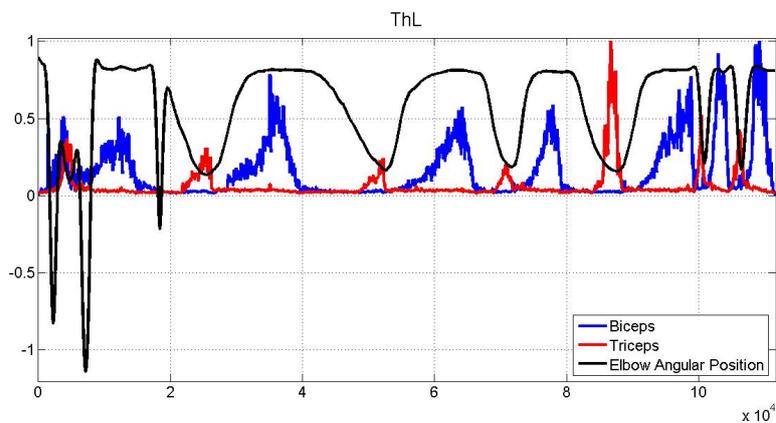


Fig. 1. Ejemplo de señales sEMG de una persona sana.

2. Modelo neurodifuso del comportamiento muscular sano

El objetivo del modelo neurodifuso es detectar activación muscular incorrecta mientras se realiza un movimiento, entonces, dicho modelo se desarrolló utilizando datos de personas sanas, de esta manera, si al modelo ingresan datos de pacientes con EVC, se activará la asistencia de la órtesis robótica. Los datos musculares fueron adquiridos utilizando sensores de sEMG colocados en bíceps y tríceps, de esta manera, de acuerdo con los niveles de amplitud normalizados, se puede determinar la intensidad de la actividad muscular. Entonces, considerar $U_t \in [0, 1]$ como el universo de discurso para los datos normalizados de señales sEMG de tríceps y $U_b \in [0, 1]$ para bíceps, ambos asociados a las siguientes variables lingüísticas:

- 1 : *Intenso*
- 2 : *Débil*
- 3 : *Ausente*

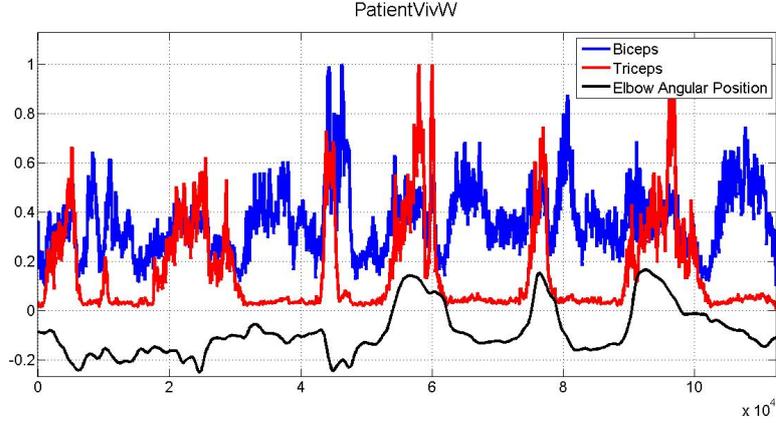


Fig. 2. Ejemplo de señales sEMG de un paciente con EVC.

De esta forma, los conjuntos difusos que representan la actividad muscular están dados por:

$$\tilde{A}_{b_i} = \left\{ \left(x_{b_i}, \mu_{\tilde{A}_{b_i}}(x_{b_i}) \mid x_{b_i} \in U_b \right) \right\}, \quad i = 1, 2, 3 \quad (1)$$

Adicionalmente, se determinaron los siguientes conjuntos difusos para la actividad muscular del tríceps:

$$\tilde{A}_{t_i} = \left\{ \left(x_{t_i}, \mu_{\tilde{A}_{t_i}}(x_{t_i}) \mid x_{t_i} \in U_t \right) \right\}, \quad i = 1, 2, 3 \quad (2)$$

El valor de pertenencia instantáneo para la señal del bíceps dentro del i -ésimo conjunto difuso se calcula utilizando:

$$\mu_{\tilde{A}_{b_i}}(x_{b_i}) = \max \left(\min \left(\frac{x_{b_i} - a_b(i)}{b_b(i) - a_b(i)}, 1, \frac{d_b(i) - x_{b_i}}{d_b(i) - c_b(i)} \right), 0 \right). \quad (3)$$

Donde $a_b, b_b, c_b, d_b \in \mathbb{R}^3$ son vectores que contienen los parámetros del i -ésimo conjunto difuso. De forma similar, el valor de pertenencia instantáneo para la señal del tríceps para el i -ésimo conjunto difuso está dado por:

$$\mu_{\tilde{A}_{t_i}}(x_{t_i}) = \max \left(\min \left(\frac{x_{t_i} - a_t(i)}{b_t(i) - a_t(i)}, 1, \frac{d_t(i) - x_{t_i}}{d_t(i) - c_t(i)} \right), 0 \right). \quad (4)$$

También $a_t, b_t, c_t, d_t \in \mathbb{R}^3$ contienen los parámetros de los conjuntos difusos correspondientes. De esta forma, sea $\bar{\mu}_b \in \mathbb{R}^3$ el vector que contiene el valor de pertenencia instantáneo de las señales sEMG de bíceps compuesto por:

$$\bar{\mu}_b = \left[\mu_{\tilde{A}_{b_1}}(x_{b_1}), \mu_{\tilde{A}_{b_2}}(x_{b_2}), \mu_{\tilde{A}_{b_3}}(x_{b_3}) \right]^T. \quad (5)$$

Adicionalmente, sea $\bar{\mu}_t \in \mathbb{R}^3$ otro vector que contiene el valor de pertenencia instantáneo de las señales sEMG de tríceps dado por:

$$\bar{\mu}_t = \left[\mu_{\bar{A}_{t_1}}(x_{t_1}), \mu_{\bar{A}_{t_2}}(x_{t_2}), \mu_{\bar{A}_{t_3}}(x_{t_3}) \right]^T. \quad (6)$$

Considerar también los siguientes vectores de salida para la representación de los movimientos de flexión y extensión respectivamente:

$$\bar{u}_f = [1, 0, 0]^T, \quad (7)$$

$$\bar{u}_e = [1, 0, 0]^T. \quad (8)$$

Entonces, la función de salida $y_f : U_b \rightarrow \mathbb{R} | y_f \in [0, 1]$ que determina el coeficiente de movimiento tipo flexión, de acuerdo con los datos sEMG de bíceps es:

$$y_f = \bar{\mu}_b^T \bar{u}_f. \quad (9)$$

Y sea la función $y_e : U_t \rightarrow \mathbb{R} | y_e \in [0, 1]$ que proporciona el coeficiente de movimiento de extensión:

$$y_e = \bar{\mu}_t^T \bar{u}_e. \quad (10)$$

Nótese que tanto (9) y (10) son posibles debido a la configuración tipo *hard fuzzy* de los conjuntos difusos. Para determinar la activación de la asistencia por medio del exoesqueleto robótico, se implementó una capa de perceptrones, cuyo objetivo es detectar la dirección del movimiento y la presencia de activación muscular incorrecta (como coactivación de bíceps y tríceps), para que la autoridad del exoesqueleto sea activada. Primero, la dirección del movimiento se calcula utilizando:

$$\bar{q} = \tanh(100\dot{q}). \quad (11)$$

Donde q es la medición de la posición angular. La red de perceptrones puede ser expresada matemáticamente como:

$$y = 0,5 [0,5(y_f + y_e) + \bar{q}]. \quad (12)$$

Donde $y \in [0, 1]$ es la autoridad del exoesqueleto robótico. El sistema neurodifuso completo se ilustra en la Fig. 3, donde se pueden apreciar los datos de entrada y la estructura del sistema de clasificación. Nótese que hay dos sistemas en la Fig. 3; la estructura que contiene los bloques *P.VimW...* está alimentada con datos de pacientes; asimismo, la que contiene bloques *H.ThL...* utiliza datos de personas con músculos sanos.

3. Resultados

El sistema propuesto fue evaluado utilizando señales de 4 personas sanas y de 4 pacientes con EVC, las cuales fueron adquiridas mientras se realizaban movimientos de extensión con resistencia de 5 Kg y flexión sin carga. Los coeficientes

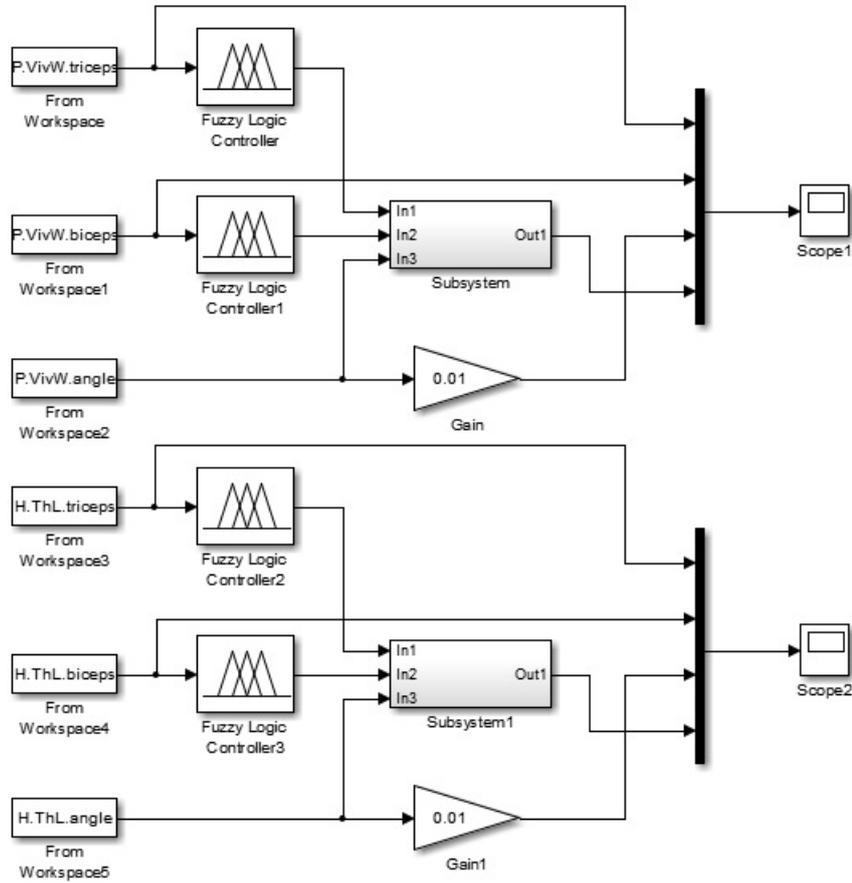


Fig. 3. Sistema neurodifuso para la clasificación de las señales.

de los conjuntos difusos utilizados en este trabajo para bíceps son los siguientes:

$$a_b = \begin{bmatrix} -2 \\ 0,15 \\ 0,18 \end{bmatrix} \quad b_b = \begin{bmatrix} -1 \\ 0,18 \\ 0,2 \end{bmatrix} \quad c_b = \begin{bmatrix} 0,15 \\ 0,18 \\ 2 \end{bmatrix} \quad d_b = \begin{bmatrix} 0,18 \\ 0,2 \\ 2,5 \end{bmatrix}. \quad (13)$$

Y los coeficientes para la fusificación de las señales de tríceps son:

$$a_t = \begin{bmatrix} -2 \\ 0,05 \\ 0,1 \end{bmatrix} \quad b_t = \begin{bmatrix} -1 \\ 0,1 \\ 0,15 \end{bmatrix} \quad c_t = \begin{bmatrix} 0,05 \\ 0,1 \\ 2 \end{bmatrix} \quad d_t = \begin{bmatrix} 0,1 \\ 0,15 \\ 2,5 \end{bmatrix}. \quad (14)$$

Dichos coeficientes fueron determinados de acuerdo a los niveles normalizados de sEMG que representan movimientos de flexión o extensión correspondientes a la activación de cada músculo. La Fig. 4 muestra las entradas de sEMG y de

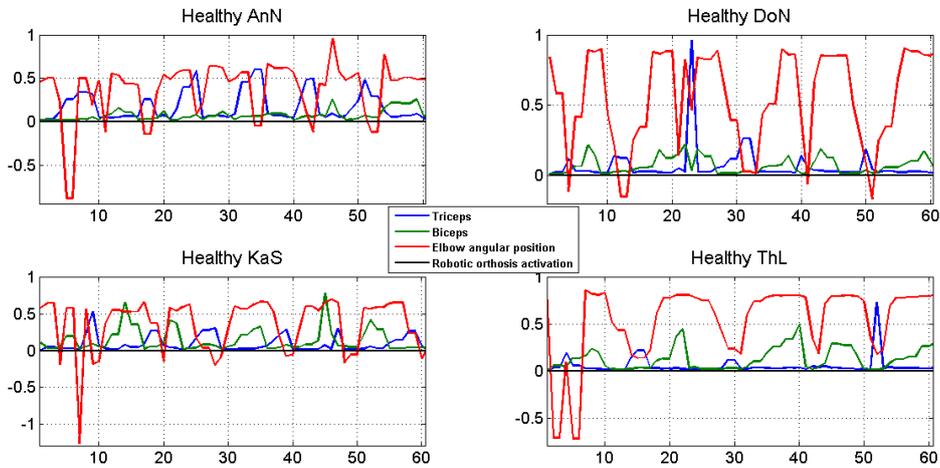


Fig. 4. Comportamiento del sistema neurodifuso con señales de personas sanas.

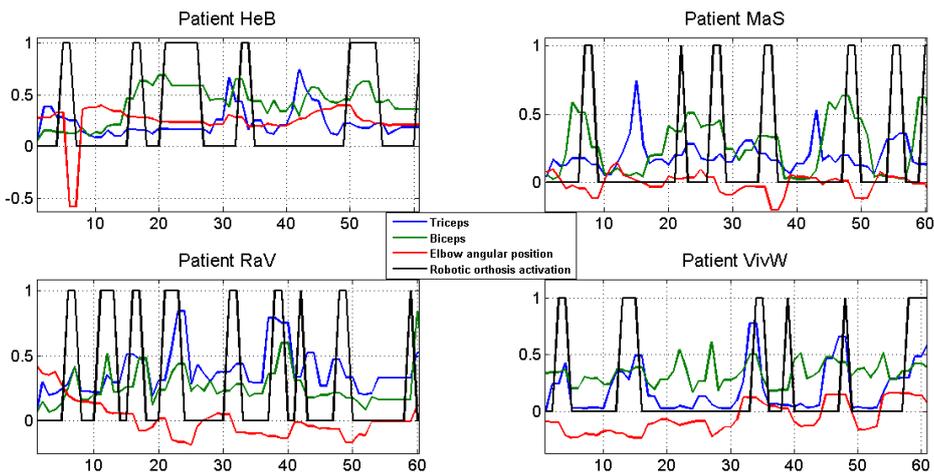


Fig. 5. Comportamiento del sistema neurodifuso con señales de pacientes con EVC.

posición angular del codo de 4 personas sanas; notar que el modelo neurodifuso detecta adecuadamente la actividad muscular correcta, por lo que la salida es 0, es decir, exoesqueleto no se activaría. La Fig. 5 ilustra las entradas de 4 pacientes con EVC y a su vez, la salida correspondiente a la activación el exoesqueleto. Se puede observar que en todos los casos, cuando el paciente trata de extender el codo, se activan tanto bíceps como tríceps, lo cual significa que la actividad muscular es incorrecta, y a su vez corresponde con la presencia de la señal de salida, lo cual significa que la asistencia de movimiento con la órtesis robótica se activaría.

4. Conclusiones

En este trabajo se presentó el desarrollo de un modelo neurodifuso para la clasificación de señales de sEMG y de posición angular para movimientos de codo con resistencia a la extensión y flexión sin carga. El sistema difuso se utilizó para clasificar la actividad muscular con entradas de datos de sEMG, y la red de perceptrones determinó la activación de un exoesqueleto robótico que ayuda al movimiento de extensión, utilizando la salida del modelo difuso y la dirección del movimiento. El sistema neurodifuso fue evaluado utilizando datos de 4 sujetos sanos y 4 pacientes con EVC, mostrando resultados satisfactorios en la clasificación de las señales y para la activación de la órtesis robótica cuando ocurre coactivación muscular de tríceps y bíceps mientras se ejecuta el movimiento de extensión. Asimismo, cuando se detecta actividad muscular sana, no se activa la autoridad del exoesqueleto robótico.

Sin embargo, estudios recientes muestran que la espasticidad no está relacionada directamente con la coactivación de tríceps y bíceps, sino está asociada a otros fenómenos como la velocidad del movimiento. Entonces, este trabajo es útil como primera aproximación para el modelado y control de un exoesqueleto robótico para asistencia a pacientes espásticos, pero se pretende migrar esta técnica utilizando datos de velocidad, tal como lo sugiere la literatura reciente [14]. A pesar de lo mencionado anteriormente, los resultados de este trabajo motivan a implementar el modelo neurodifuso desarrollado en una órtesis robótica real como un sistema preliminar de activación utilizando sEMG.

Agradecimientos. Este proyecto de investigación está financiado por el apoyo CONACyT-BMBF-FONCICYT-267734.

Referencias

1. Suarez-Escobar, M., Rendon-Velez, E.: An overview of robotic/mechanical devices for post-stroke thumb rehabilitation. *Disabil. Rehabil. Assist. Technol.* 13(7), 683–703 (2018)
2. Orozco-Soto, S.M. et al.: Fuzzy Sliding Mode Control for Trajectory Tracking and Force Compensation of a Robotic Haptic Interface. In: *Proc. of 14th IEEE Int. Carpathian Control Conf.*, pp. 267–273 (2013)
3. Weber, L.M., Stein, J.: The use of robots in stroke rehabilitation: A narrative review. *NeuroRehabilitation* 43(1), 99–110 (2018)
4. Li, M., Xu, G., Xie, J., Chen, C.: A review: Motor rehabilitation after stroke with control based on human intent. *Proc. Inst. Mech. Eng. Part H J. Eng. Med.* 232(4), pp. 344–360 (2018)
5. Leonardis, D. et al.: An EMG-controlled robotic hand exoskeleton for bilateral rehabilitation. *IEEE Trans. Haptics* 8(2), 140–151 (2015)
6. Nurhanim, K., Elamvazuthi, I., Vasant, P., Ganesan, T., Parasuraman, S., Ahamed Khan, M.K. A.: Joint torque estimation model of surface electromyography (sEMG) based on swarm intelligence algorithm for robotic assistive device. *Procedia Comput. Sci.*, vol. 42, no. C, pp. 175–182 (2014)

7. Genna, C., Dosen, S., Paredes, L., Turolla, A., Graimann, B., Farina, D.: Replace, Repair, Restore, Relieve – Bridging Clinical and Engineering Solutions in Neurorehabilitation. In: Proceedings of the 2nd International Conference on NeuroRehabilitation (ICNR2014), Aalborg, 24-26 June, vol. 7, pp. 383–391 (2014)
8. Hu, X.L., Tong, R.K.Y., Ho, N.S.K., Xue, J.J., Rong, W., Li, L.S.W.: Wrist Rehabilitation Assisted by an Electromyography-Driven Neuromuscular Electrical Stimulation Robot after Stroke, *Neurorehabil. Neural Repair* 29(8), 767–776 (2015)
9. Chen, X., Zeng, Y., Yin, Y.: Improving the Transparency of an Exoskeleton Knee Joint Based on the Understanding of Motor Intent Using Energy Kernel Method of EMG. *IEEE Trans. Neural Syst. Rehabil. Eng.* 25(6), 577–588 (2017)
10. Irastorza-Landa, N. et al.: Design of Continuous EMG Classification approaches towards the Control of a Robotic Exoskeleton in Reaching Movements. In: *IEEE Int. Conf. Rehabil. Robot.*, pp. 128–133 (2017)
11. Zhang, Q., Liu, R., Chen, W., Xiong, C.: Simultaneous and continuous estimation of shoulder and elbow kinematics from surface EMG signals- *Front. Neurosci.*, vol. 11, May, pp. 1–12 (2017)
12. Kilic, E., Dogan, E.: Design and fuzzy logic control of an active wrist orthosis. *Proc. Inst. Mech. Eng. Part H J. Eng. Med.*, p. 728–746 (2017)
13. Liu, J., Kang, S. H., Xu, D., Ren, Y., Lee, S. J., Zhang, L. Q.: EMG-Based continuous and simultaneous estimation of arm kinematics in able-bodied individuals and stroke survivors. *Front. Neurosci.*, vol. 11, August, pp. 1–11 (2017)
14. Becker, S., von Werder, S.C.F.A., Lassek, A.K., Disselhorst-Klug, C.: Time-frequency coherence of categorized sEMG data during dynamic contractions of biceps, triceps, and brachioradialis as an approach for spasticity detection. *Med. Biol. Eng. Comput.* 57(3), 703–713 (2019)
15. Williams, S., Schmidt, R., Disselhorst-Klug, C., Rau, G.: An upper body model for the kinematical analysis of the joint chain of the human arm. *J. Biomech.* 39(13), 2419–2429 (2006)

Algoritmo basado en reglas de asociación para la extracción de relaciones no taxonómicas en corpus de dominio

Irvin Yair Cabrera Moreno, Mireya Tovar Vidal, José de Jesús Lavalle Matínez,
Meliza Contreras Gonzalez

Benemérita Universidad Autónoma de Puebla,
Facultad de Ciencias de la Computación,
Puebla, México

yair.cb@gmail.com, mtovar@cs.buap.mx, jlavallentor@gmail.com,
mcontreras@cs.buap.mx

Resumen. La identificación de relaciones no taxonómicas es una tarea que se realiza con el aprendizaje y la creación de ontologías. Además, la construcción manual de ontologías para expertos e ingenieros de conocimiento es una tarea costosa y lenta, por lo que es necesario crear algoritmos automáticos y/o semiautomáticos que agilicen el procedimiento. En esta investigación se propone un algoritmo para la extracción de relaciones no taxonómicas en una ontología de Inteligencia Artificial (IA), las cuales son evaluadas a través de una técnica de minería de datos: *reglas de asociación*, que cuenta con medidas estadísticas que determinan la probabilidad de ocurrencia entre los conceptos y el verbo conector relacionados. Los resultados experimentales indican que el 72% de las relaciones obtenidas en el algoritmo existen en la ontología de IA.

Palabras clave: relaciones no taxonómicas, extracción de información, ontología.

An Algorithm Based on Association Rules for the Extraction of Non-Taxonomic Relationships in the Domain Corpus

Abstract. The identification of non-taxonomic relationships is a task that is carried out with learning and the creation of ontologies. Also, the manual construction of ontologies for experts and knowledge engineers is a costly and slow task, which is why it is necessary to create automatic or semi-automatic algorithms that speed up the procedure. In this research we propose an algorithm for the extraction of non-taxonomic relationships in an ontology of Artificial Intelligence (AI), evaluated through a data mining technique: *association rules*, which has statistical measures that determine the probability of occurrence between the concepts and the related connector verb. The experimental results indicate that 72% of the relationships obtained in the algorithm exist in the ontology of AI.

Keywords: non-taxonomic relationships, information extraction, ontology.

1. Introducción

La creación automática de ontologías y su representación en un lenguaje formal comúnmente se realiza por ingenieros de conocimiento junto con expertos de dominio. Este proceso implica la extracción, conceptualización, evaluación y formalización del conocimiento del dominio. La construcción manual de ontologías se ha identificado en gran medida como una tarea costosa, tediosa y con tendencia a errores [1]. Además de dificultades técnicas, como la falta de estándares para reutilizar las ontologías existentes y la ausencia de métodos de extracción automática de conocimientos, son problemas que dificultan la creación de ontologías [2].

La definición más popular de ontología en computación es brindada por Gruber [3], quien define una ontología como: “*una especificación explícita de una conceptualización*”, es decir, que proporciona una estructura y contenidos de forma explícita que codifica las reglas implícitas de una parte de la realidad; estas declaraciones explícitas son independientes del fin y del dominio de la aplicación en el que se usarán o se reutilizarán sus definiciones.

Actualmente, la creación de ontologías a partir de textos utilizando métodos de aprendizaje automático y de minería de datos se ha propuesto como un método que facilita el proceso de ingeniería ontológica. En este contexto, en [4] el aprendizaje ontológico se ha identificado como un campo que apunta a ayudar a los ingenieros del conocimiento, así como a los usuarios finales en la creación de ontologías. Puede verse como un campo multidisciplinario, con disciplinas como la ingeniería ontológica, el aprendizaje automático y el procesamiento del lenguaje natural. El uso de estas tecnologías se distribuye en tres tareas principales, extracción de entrada léxica, extracción de taxonomía y extracción de relaciones no taxonómicas [4]; todos juntos permiten construir una ontología desde cero o mejorar una ontología existente utilizando diferentes fuentes de información.

El aprendizaje automático de ontologías a partir de textos constituye un medio prometedor para que los ingenieros ontológicos aceleren la creación manual de ontologías, de tal forma que se han propuesto varios enfoques para cubrir las diferentes tareas. En este proceso, la fase de extracción de relaciones no taxonómicas ha sido reconocida como uno de los problemas con más dificultad [4] y menos cubiertos [5]. Esta fase se puede dividir en dos problemas diferentes: descubrir la existencia de una relación entre un par de conceptos y luego etiquetar esta relación de acuerdo con su significado semántico. La asignación de etiquetas a las relaciones también es difícil ya que son posibles varias relaciones entre instancias de los mismos conceptos generales [4].

En particular en esta investigación se usa un método de minería de datos llamado, *reglas de asociación*, en el cual gracias a medidas estadísticas se puede detectar la relación entre un par de conceptos de acuerdo a la ocurrencia de

ellos en las oraciones del texto. El objetivo de esta investigación es crear un algoritmo que extraiga de forma automática las relaciones no taxonómicas y evaluar su desempeño con respecto a las relaciones no taxonómicas existentes en una ontología del dominio de Inteligencia Artificial.

Este artículo está distribuido de la siguiente forma: en la Sección 2 se presentan algunos trabajos relacionados con la extracción automática de relaciones no taxonómicas. En la sección 3 se expone la teoría de las reglas de asociación; en la Sección 4 se presenta el algoritmo propuesto, en la Sección 5 se exponen los resultados obtenidos de la investigación y finalmente en la Sección 6 se presentan las conclusiones y el trabajo a futuro de esta investigación.

2. Trabajos relacionados

En esta sección se presentan trabajos de otros autores, estos trabajos están relacionados con la extracción de relaciones no taxonómicas, su evaluación y descubrimiento, así como también el aprendizaje automático de ontologías.

Serra y Girardi en [6] proponen un proceso semiautomático para extracción de relaciones no taxonómicas de fuentes de texto, haciendo uso de técnicas de procesamiento de lenguaje natural (PLN) para identificar las relaciones no taxonómicas y técnicas de minería de datos para sugerir un nivel alto dentro de la jerarquía de la ontología en inglés. El proceso se divide en tres fases. La primera fase consta de la extracción de las relaciones candidatas utilizando PLN. El objetivo es encontrar el verbo que indique la relación no taxonómica. En la segunda fase se hace un análisis del nivel jerárquico y por último, en la tercera fase se realiza una selección manual de relaciones por un experto.

Mäedche y Staab [7] establecen un nuevo enfoque para el descubrimiento de relaciones no taxonómicas en textos y facilitar la ingeniería de estas relaciones no taxonómicas. Usan reglas de asociación que no solo detectan relaciones entre conceptos. También, el apropiado nivel de abstracción el cual determina la relación. Se hace uso de métodos de procesamiento de texto plano para identificar pares de palabras relacionadas, esto comprende un etiquetador, lo que quiere decir, que crea abreviaciones, también hace uso de análisis léxico para el reconocimiento de entidades, la recuperación información en dominios específicos y el algoritmo de reglas de asociación. La salida de las reglas de asociación son pares de conceptos que son entregados al ingeniero de conocimiento para que las incluya en la ontología.

Weichselbraun, Scharl, Granitzer, Neidhart y Juffinger presentan [8] un enfoque automático para sugerencias en tipos de relaciones en ontologías. En él, crean un vector de verbos con las relaciones encontradas en el corpus, agrega centroides conocidos de relaciones ya identificadas que se encuentran en una base de conocimiento. Los vectores de verbos desconocidos son comparados con vectores ya conocidos y se crean sugerencias de relaciones. Las relaciones no taxonómicas son descubiertas en tres pasos a través de un análisis de los principales sustantivos, crear sinónimos a estos sustantivos y enriquecen WordNet. El

último paso es un análisis de subsunción, en el cual se asume que el documento está compuesto por oraciones de un conjunto de otros documentos.

Sánchez y Moreno [9] en su trabajo presentan un método automático que extrae conceptos de relaciones no taxonómicas y etiquetado de relaciones usando toda la web como corpus. Para el descubrimiento y extracción de las relaciones no taxonómicas en la red hacen uso de: *técnicas de análisis ligero* que se usa para tener una mejor escalabilidad en un entorno como la web, *análisis estadístico* que es aplicado en las tareas de adquisición del conocimiento. También se usan *patrones lingüísticos* que es una técnica muy efectiva para no consultar a expertos y para descubrir las relaciones no taxonómicas. Por último se usa *bootstrapping* que es utilizado para restringir las consultas hechas por el motor de búsqueda web para obtener corpora de documentos de dominios específicos.

Kavalec, Mäedche y Svátek presentan en su trabajo [10] una combinación entre análisis de texto plano, minería de datos y modelado de conocimiento. Además, para la extracción de relaciones no taxonómicas usan una técnica que se basa en el método de Text-to-Onto. El método usado para el descubrimiento de las relaciones está basado también en las reglas de asociación, donde dos o más léxicos pertenecen a una transacción si estos se encuentran juntos en un documento o en un texto definido, las transacciones más frecuentes son salidas como asociaciones entre sus objetos. Las relaciones no taxonómicas candidatas entre dos conceptos son aquellos verbos en el que el número de transacciones que se mantienen entre el verbo v , el concepto c_1 y el concepto c_2 aparecen en una ventana de n palabras a partir de una aparición de v .

Mäedche y Staab describen [11] un enfoque para extracción de relaciones no taxonómicas de un corpus creado con técnicas de procesamiento de texto plano, este enfoque está basado en el algoritmo de reglas de asociación para encontrar las relaciones y también para definir un nivel de abstracción en ellas. El algoritmo encuentra las relaciones que ocurren en un par de elementos. La regla de asociación básica consiste en un conjunto de transacciones, donde cada transacción tiene un conjunto de elementos y cada uno de los elementos forma parte de un conjunto de conceptos. Para descubrir las relaciones, se construye un esquema de aprendizaje donde modifican el conjunto de transacciones y las métricas para generar las reglas de asociación.

Nabila, Basir y Mamat en [12], presentan un método para extraer relaciones no taxonómicas usando la similitud entre relaciones que existen en más de una oración. El propósito del método es mejorar el proceso de recuperación de relaciones no taxonómicas en dominios específicos de texto. Consiste en extraer conceptos mediante preprocesamiento y se realiza un análisis estadístico para extraer los términos más importantes. Con eso, los conceptos se clasifican en dos listas una de temas y otra de objetos, después, se generan pares de conceptos usando producto cartesiano entre las dos listas y para evitar la existencia de relaciones taxonómicas se usan algunas restricciones. Finalmente, se realiza la extracción y asignación de buenas relaciones entre los pares de conceptos.

Villaverde en su documento [13] propone una técnica para el descubrimiento de relaciones no taxonómicas y la extracción de elementos léxicos que sirven

como conectores entre los conceptos relacionados. Su enfoque está basado en el análisis de estructuras sintácticas y dependencias entre conceptos. Para el descubrimiento de relaciones no taxonómicas su técnica toma como entrada un corpus de textos específicos de dominio y una jerarquía de conceptos que describe las relaciones taxonómicas entre conceptos y busca otras posibles relaciones en el texto. Se usa el etiquetador POS para asignar etiquetas a las palabras de contenido e identificar frases de verbos y nominales en cada oración. Este análisis es aplicado a todas las oraciones encontradas que contienen ambos conceptos candidatos que cumplan con el patrón: $\langle \text{término} \rangle \langle \text{verbo} \rangle \langle \text{término} \rangle$. Una vez que las relaciones candidatas estén disponibles, son validadas bajo el criterio de las reglas de asociación.

En esta investigación se propone un algoritmo en el cual por medio de preprocesamiento de texto y a través de la técnica de minería de datos: reglas de asociación, se realiza la extracción automática de relaciones no taxonómicas en un corpus de dominio. En los resultados experimentales las relaciones no taxonómicas obtenidas son evaluadas bajo las medidas estadísticas llamadas soporte y confianza [14].

3. Reglas de asociación

Las reglas de asociación describen la relación entre los elementos de un conjunto de datos. Estas reglas nacen de la investigación de Agrawal, Imielinski y Swami [14] donde consideran la colección de datos que generan las compras en un supermercado, sirviendo como apoyo para saber que conjunto de productos se compran y generar por medio de las reglas de asociación promociones. En [14] definen las reglas de asociación como:

Dado C como el conjunto de conceptos y $T := \{t_i \mid i = 1 \dots n\}$ como la base de datos de transacciones, donde n es el número total de transacciones y cada transacción t_i consiste en un conjunto de elementos:

$$t_i = \{a_{i,j} \mid j = 1 \dots m_i, a_{i,j} \in C\},$$

y cada elemento $a_{i,j}$ es un elemento del conjunto C y m el número total de elementos en t_i . El algoritmo calcula las *reglas de asociación* presentadas en la ecuación (1):

$$X_k \Rightarrow Y_k (X_k, Y_k \subset C, X_k \cap Y_k = \{\}). \quad (1)$$

Una regla de asociación es una implicación de la forma $X_k \Rightarrow Y_k$ donde X_k es un conjunto de algunos elementos de C también llamado *antecedente* y Y_k es un sólo elemento en C que no está presente en X , también llamado *consecuente*. La regla se satisface si las medidas de *soporte*, ecuación (2), y *confianza*, ecuación (3), sean iguales o mayores a las deseadas. El soporte de una regla $X_k \Rightarrow Y_k$ es el porcentaje de transacciones que contiene $X_k \cup Y_k$ como un subconjunto, y la confianza de una regla $X_k \Rightarrow Y_k$ está definida como el porcentaje de transacciones donde Y_k aparece si X_k se encuentra en una transacción:

$$\text{Soporte}(X_k \Rightarrow Y_k) = \frac{|\{t_i \mid X_k \cup Y_k \subseteq t_i\}|}{n}, \quad (2)$$

$$\text{Confianza}(X_k \Rightarrow Y_k) = \frac{|\{t_i \mid X_k \cup Y_k \subseteq t_i\}|}{|\{t_i \mid X_k \subseteq t_i\}|}. \quad (3)$$

Las medidas se pueden interpretar como: una regla con bajo soporte indicaría que habrá aparecido por casualidad. Sin embargo, una regla con baja confianza indicaría que no existe relación entre el antecedente y el consecuente. Además, existe una diferencia entre $X_k \Rightarrow Y_k$ y $Y_k \Rightarrow X_k$, debido a que las reglas comparten el mismo soporte pero su confianza tiende a ser distinta.

4. Algoritmo propuesto

En esta sección se presenta el algoritmo propuesto para la extracción automática de relaciones no taxonómicas. El funcionamiento general del algoritmo se detalla en el Algoritmo 1.1. En el cual se identifican los acrónimos existentes en el corpus mediante la función *Acronimos*, la cuál por medio de una expresión regular, son seleccionados aquellos tokens de las oraciones que se deseen, especificando un patrón con la forma en que se desean encontrar las palabras. En este caso, se seleccionan aquellas palabras que cumplan con tener dos o más letras mayúsculas consecutivas sin que la palabra tenga minúsculas en ella. Después el corpus es dividido en oraciones en la función *Oracion*, esta división está dada por cada punto que aparezca en el corpus.

Algoritmo 1.1: Algoritmo de extracción de términos propuesto.

Datos: Corpus de dominio

Resultado: Corpus de dominio pre-procesado

```

1 a ← Acronimos(corpus)
2 sentences ← Oracion(corpus)
3 sentences, dic ← prepro(sentences, a)
4 resul ← pos_tag(sentences) inds ← buscarNPS(resul)
5 cand ← relacionesCandidatas
6 triple ← reglasAsociacion
    
```

Esto con la finalidad de que en la función *prepro* se realice una expansión de términos que es detallada en el Algoritmo 1.2. La función *prepro* recorre cada oración en busca de los acrónimos obtenidos en el paso anterior y los almacena en la variable *dic* que tiene función como diccionario de acrónimos. Cada oración es dividida en tokens para realizar una comparación entre los tokens de cada oración y los acrónimos obtenidos, para comprobar qué acrónimos tiene la oración. Si el resultado de la comparación tiene elementos, la función continúa con verificar cada elemento encontrado en la comparación. Si el acrónimo no está en el diccionario de acrónimos, se busca su significado tomando el tamaño del acrónimo para buscar hacia la izquierda las palabras que formen el significado, el acrónimo es agregado al diccionario y se elimina de la oración dejando solo su significado. Si el acrónimo ya se encuentra en el diccionario, sólo se realiza la sustitución del acrónimo con su significado en la oración.

Algoritmo 1.2: Función *prepro*.

Datos: Oraciones del corpus, acrónimos
Resultado: Oraciones con acrónimos expandidos, diccionario de acrónimos

```
1 Para i = longitud-oraciones hacer:
2   oracion ← oraciones[i]
3   comp ← Para elemento en acronimos Si elemento está en oracion
4   Si comp no está vacío hacer:
5     Para palabra en comp hacer:
6       indice ← oracion.index(oracion)
7       Si la palabra está en dic hacer::
8         Para letra en palabra:
9           Se toma el índice del acronimo, se elimina y se expande
10      FinPara
11    FinPara
12    Sino:
13      rango ← len(palabra)
14      Para j < rango:
15        Si oracion[ind-(rango-j)][0] == palabra[0] hacer:
16          acronimo = acronimo + oracion[ind-(rango-j)]
17        Sino
18          eliminar acronimo de acronimos
19      FinSi
20    FinPara
21  FinSi
22 FinSi
```

El siguiente paso es volver las oraciones en su forma minúscula para lograr una comparación entre las palabras más exacta. Después se realiza el etiquetado *Parte del discurso*, este etiquetado agrega una etiqueta al contenido y función de cada palabra logrando así identificar las clases de palabras en las oraciones. Mediante una gramática, las frases nominales de las oraciones se identifican. Esta gramática es muy parecida a una expresión regular, la diferencia es el uso de las etiquetas POS para crear el patrón de frase que se desea encontrar. Estas frases nominales solo son etiquetadas en el análisis y son devueltas como tipo árbol por cada oración, así que se hace una extracción de las frases nominales en la función *buscarNPS* que puede verse en el Algoritmo 1.3. En la función *buscarNPS* se recibe como entrada la variable *resul* la cuál lleva como contenido las oraciones en forma de árbol que fueron devueltas en el etiquetado POS. En *BuscarNPS* se recorre cada árbol generado en el paso anterior, con la intención de encontrar las frases nominales que se agruparon. Se considera que, al etiquetar una palabra con el analizador POS, cada palabra se vuelve una tupla¹, que consta de la palabra y su etiqueta de clase. Al crear la gramática, se agrupan ciertos tipos de palabras que conforman en grupo un mismo significado, dando

¹ Una tupla es un tipo de lista que solo consta de dos elementos

así la frase nominal. Para su extracción, en el algoritmo solo es necesario buscar conjuntos de tuplas, es decir, en el árbol resultado, se buscan nodos que sean de tipo árbol y se extrae cada palabra que lleven sus tuplas formando así la frase nominal.

Algoritmo 1.3: Función *buscarNPS*.

Datos: resul
Resultado: inds

```
1 Para oracion en resul hacer:
2   nps ← [] Para nodo en oracion hacer:
3     Si nodo es tipo árbol hacer:
4       palabra ← ''
5       Para i < len(nodo) hacer:
6         palabra ← palabra + nodo[i][0] + ''
7       FinPara
8       palabra ← palabra[:len(palabra)-1]
9       nps ← agrega(aux)
10    FinSi
11  FinPara
12  aux ← [index(oracion),nps]
13  inds ← aux
14 FinPara
```

Cuando se tiene la lista de frases nominales que ocurren en una oración, se guardan junto con el índice de la oración en el que se encuentran y se devuelve como salida esa nueva lista de índices con frases nominales. Una vez disponibles las frases nominales junto con el índice de la oración en la que se encuentran, se continúa con la extracción de las relaciones no taxonómicas. El algoritmo propuesto para la extracción de relaciones no taxonómicas consta de dos funciones: *relacionesCandidatas* y *reglasAsociacion*.

El procedimiento que sigue la función *relacionesCandidatas* se detalla en el Algoritmo 1.4. En esta función se observa que la entrada son las oraciones del corpus y la lista de frases nominales e índices de su oración de ocurrencia, la salida son las tripletas candidatas para la ontología. El algoritmo de la función *relacionesCandidatas* comienza recorriendo las oraciones del corpus. En este recorrido se pretende encontrar el texto entre dos frases nominales. En el texto entre ellos se puede hallar una frase verbal que conecte a dos frases nominales que ocurren en la oración.

De los pasos 3-5, se toman los índices donde comienzan cada frase nominal y así conocer el texto que hay entre ellas. Este texto es etiquetado por el análisis *POS* en la función *pos_tag* para conocer que tipo de palabras existen entre estas frases nominales. Nuevamente, se repite el procedimiento usado para encontrar las frases nominales, con la diferencia que aquí se buscan frases de verbos. De la misma forma, se crea una gramática en la función *parser* para identificar las frases verbales que ocurran en el texto obtenido especificando etiquetas de tipo verbo y adverbio.

Si es encontrada una frase verbal entre las frases nominales, en la lista *cand*, son almacenadas la frase nominal de la izquierda, el verbo que conecta y la frase nominal de la derecha, formando una tripleta. Después de obtener las tripletas candidatas son evaluadas bajo las medidas de *soporte y confianza* de las *reglas de asociación* [14].

En este contexto, una transacción en *T* representa la ocurrencia de un par de conceptos con algún verbo de enlace en el cuerpo del texto. La fuerza de la asociación de ambos conceptos con el verbo estará dada por la regla de confianza. Se debe encontrar las frases nominales que tienen soporte de transacciones por encima del soporte mínimo. El soporte para un conjunto de elementos es el número de transacciones que contienen el conjunto de elementos.

El algoritmo fue codificado con la biblioteca NLTK de python [15]. En la siguiente sección se presentan los resultados obtenidos.

Algoritmo 1.4: Función *relacionesCandidatas*.

```
Datos: oraciones, ind
Resultado: tripleta
1 Para i < len(oraciones) hacer:
2   Para j < len(ind[i][1]-1) hacer:
3     ind1 ← oraciones[i].index(ind[i][1][j])
4     ind2 ← oraciones[i].index(ind[i][1][j+1])
5     vp ← oraciones[i][len(ind[i][1][j])+ind1+1:ind2]
6     vp2 ← pos_tag(vp)
7     Si len(vp2) == 1 y vp2[0][1] == V hacer:
8       aux ← [ind[i][1][j], vp, ind[i][1][j+1]]
9       tripleta ← agrega(aux)
10    Sino Si 8 > len(vp2) > 1 hacer:
11      vp2 ← parse(vp2)
12      Para k < len(vp2) hacer:
13        Si vp2[k] es tipo árbol hacer:
14          Para v en vp2[k] hacer:
15            verbo ← verbo + v[0] + ' '
16            verbo ← verbo[:len(verbo)-1]
17            aux ← [ind[i][1][j], verbo, ind[i][1][j+1]]
18            cand ← agrega(aux)
19          FinPara
20        FinSi
21      FinPara
22    FinSi
23  FinPara
24 FinPara
```

5. Resultados experimentales

En esta sección se muestran los resultados obtenidos con el algoritmo propuesto para la extracción de relaciones no taxonómicas. Para la investigación se usó la ontología de inteligencia artificial propuesta en [16]. A continuación en la Tabla 1 se muestran los datos utilizados. La ontología consta de un número determinado de documentos, tokens o palabras, vocabulario y número de oraciones.

Tabla 1. Datos del corpus de dominio.

Ontología	Documentos	Tokens	Vocabulario	Oraciones
IA	8	10,805	1,510	460

Para la ontología IA, durante el procesamiento de sus datos, se obtuvieron los resultados que se presentan en la Tabla 2. En la Tabla 5 se muestran algunos de los resultados experimentales de la extracción automática de relaciones no taxonómicas usando las medidas de reglas de asociación. En nuestra investigación, el soporte y confianza de una regla candidata $C \Rightarrow v$, donde C indica ambos conceptos y v el verbo que los conecta. Una transacción en T representa la ocurrencia de un par de conceptos con algún verbo que los una.

Tabla 2. Total de conceptos y relaciones no taxonómicas en la ontología de dominio de IA.

Ontología	Conceptos	Relaciones no taxonómicas
IA	276	61

Tabla 3. Resultados obtenidos para pares de conceptos con medidas de Soporte y Confianza.

$Concepto_1$	$Concepto_2$	Verbo	Soporte	Confianza
neural network	agents	based	0.01949	1
temporal agent	time	may use	0.01949	1
intelligent agent	enviroment	perceives	0.01949	1

Gracias a estas medidas estadísticas se logra segmentar los resultados para obtener las relaciones que tengan un mayor grado de relación. Los resultados obtenidos se evaluaron con los de la ontología provista en [16] y se calculó la

exactitud, la cuál se refiere a la evaluación del sesgo de las predicciones, es decir, responde a la pregunta: *¿Cuál es el promedio de las predicciones correctas?* [17]. La fórmula de exactitud se presenta en la ecuación 4:

$$Exactitud = \frac{CantidadDeCasosCorrectos}{TotalDeCasos}. \quad (4)$$

donde nuestro sistema extrajo un total de casos correctos de 44 y el total de casos en la ontología es igual a 61, lo que representa un total de 72 % de exactitud.

6. Conclusiones

En esta investigación se implementó un algoritmo en Python utilizando la biblioteca de NLTK para el procesamiento del lenguaje natural y para la extracción de relaciones no taxonómicas en un corpus de dominio de inteligencia artificial mediante la técnica de reglas de asociación. Esta técnica describe la probabilidad de que exista una relación entre objetos, en nuestro caso entre un par de conceptos y un verbo que los conecta en una oración en el corpus de dominio. Posteriormente, los resultados fueron comparados con las relaciones no taxonomicas de una ontología de dominio de IA.

Con base en los resultados experimentales, se logró obtener el 72 % de relaciones no taxonómicas existentes en la ontología de IA [16]. Con base en estos resultados se observó que al contar con un corpus pequeño, el soporte que hay entre los conceptos y el verbo es muy bajo, es decir, menor al 2 %, ya que esta medida representa la probabilidad de encontrar al par de conceptos y el verbo que los conecta dentro del dominio. Sin embargo, esto provoca que la confianza sea mayor al 50 %, ya que describe la probabilidad de que esta relación sea verdadera, es decir, que al encontrar los dos conceptos, el verbo se encuentre en la misma oración. Además, cabe mencionar que algunas de las relaciones que no detectó el algoritmo, y existen en la ontología, son aquellas que la localización del verbo está al final de la oración y no de manera intermedia.

Como trabajo a futuro se propone implementar una propuesta de solución que identifique relaciones no taxonómicas en diferentes tipos de estructuras de la oración en inglés. Así mismo, aplicar el enfoque a otras ontologías y comparar los resultados.

Agradecimientos. Esta investigación es apoyada por el Fondo Sectorial de Investigación para la Educación, con el proyecto CONACyT CB/257357 bajo el número de becario 28617 y por el proyecto VIEP-BUAP 100409344-VIEP2019.

Referencias

1. Shamsfard, M., Abdollahzadeh Barforoush, A.: The state of the art in ontology learning: a framework for comparison. *The Knowledge Engineering Review* 18(4), 293–316. <https://doi.org/10.1017/S0269888903000687> (2003)

2. Shamsfard, M., Abdollahzadeh Barforoush, A.: Learning ontologies from natural language texts. *International Journal of Human-Computer Studies*, No. 60, pp. 17–63. <https://doi.org/10.1016/j.ijhcs.2003.08.001> (2004)
3. Gruber, T.R.: Toward Principles for the Design Ontologies Used for Knowledge Sharing. *International Journal of Human-Computer Studies* 43(5-6), 907–928 (1992)
4. Maedche, A., Staab, S.: Ontology learning for the semantic web. *IEEE Intelligent Systems* 16(2), 72–79. <https://doi.org/10.1109/5254.920602> (2001)
5. Sánchez, D., Moreno, A.: Learning non-taxonomic relationships from web documents for domain ontology construction. *Data and Knowledge Engineering* 64(3), 600–623. <https://doi.org/10.1016/j.datak.2007.10.001> (2008)
6. Serra, I., Girardi, R.: A Process for Extracting Non-Taxonomic Relationships of Ontologies from Text. *Intelligent Information Management* 3, pp.124–119. <https://doi.org/10.4236/iim.2011.34014> (2011)
7. Maedche, A. Staab, S.: Mining Non-Taxonomic Conceptual Relations from Text. In: *Ekaw-00 – European knowledge acquisition workshop*, Juan-les-pins, LNAI, Springer. <https://doi.org/10.1.1.41.4860> (2000)
8. Weichselbraun, A., Wohlgenannt, G., Scharl, A., Granitzer, M., Neidhart, T., Juffinger, A.: Discovery and evaluation of non-taxonomic relations in domain ontologies. *International Journal of Metadata, Semantics and Ontologies* 4(3), 212–222 (2009)
9. Sánchez, D., Moreno, A.: Discovering non-taxonomic relations from the Web. In: *7th International Conference on Intelligent Data Engineering and Automated Learning*. LNCS 4224. <https://doi.org/10.1.1.83.3546>, pp. 636–629 (2006)
10. Kavalec, M., Maedche, E., Svátek, V.: Discovery of Lexical Entries for Non-Taxonomic Relations in Ontology Learning. In: *Proceedings of SOFSEM 2004: Theory and Practice of Computer Science*, LNCS 2932. <https://doi.org/10.1.1.10.2718>, pp. 256–249 (2004)
11. Maedche, A., Staab, S.: Semi-Automatic Engineering of Ontologies from Text. In: *Proceedings of the 12th Internal Conference on Software and Knowledge Engineering*. <https://doi.org/10.1.1.453.2051>, pp. 239–231 (2000)
12. Nabila, N.F., Basir, N., Mamat, A.: Synonymous Non-Taxonomic Relations Extraction. *ARPN Journal of Engineering and Applied Sciences* 10(2) (2015)
13. Villaverde, J., Persson, A., Godoy, D., Amandi, A.: Supporting the discovery and labeling of non-taxonomic relationships in ontology learning. *Data and Knowledge Engineering* 36(7), 10288–10294. <https://doi.org/10.1016/j.eswa.2009.01.048> (2009)
14. Agrawal, R., Imielinski, T., Swami, A.: Mining Association Rules between Sets of Items in Large Databases. In: *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data* 22(2), 207–216, Washington DC (USA) <https://doi.org/10.1145/170035.170072> (1993)
15. Loper, E., Bird, S.: NLTK: The Natural Language Toolkit. In: *Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*, Philadelphia: Association for Computational Linguistics (2002)
16. Zouaq, A., Gasevic, D., Hatala, M.: Linguistic patterns for information extraction in ontocmaps. In: *Proceedings of the 3rd International Conference on Ontology Patterns* 929, pp. 61–72. CEUR-WS.org (2012)
17. Vazques, K.: *Monitoreo de opiniones en redes sociales sobre la calidad del servicio*. Tesis Licenciatura, Benemérita Universidad Autónoma de Puebla (2017)

Estudio comparativo de arquitecturas de CNNs en hojas de Pimiento Morrón infectadas con virus PHYVV o PEPGMV

Eduardo Díaz-Gaxiola, Zuriel E. Morales-Casas, Oscar Castro-López, Gerardo Beltrán-Gutiérrez, Ines F. Vega-López, Arturo Yee-Rendón

Universidad Autónoma de Sinaloa, Culiacán, Sinaloa, México
 {eduardogaxiola, zurielernesto, oscarcastro, gerardo, ifvega, arturo.yee}@uas.edu.mx

Resumen. En este trabajo, presentamos un estudio comparativo de dos arquitecturas de Redes Neuronales Convolucionales (CNNs por sus siglas en inglés): MobileNet y VGG19. Estas arquitecturas de CNNs son utilizadas para construir modelos predictivos para clasificar hojas de pimiento morrón en tres clases: sana o infectada con virus PHYVV o PEPGMV a partir de imágenes digitales. La idea de este estudio es medir la *accuracy*, *precision* y *recall* de las arquitecturas de CNNs, más allá de las técnicas tradicionales de *Machine Learning*, para prevenir y detectar plagas y enfermedades en plantas de pimiento morrón. Los resultados obtenidos en los experimentos muestran una gran ventaja de las arquitecturas CNNs sobre las técnicas tradicionales de *Machine Learning*.

Palabras clave: arquitecturas de redes neuronales convolucionales (CNNs), MobileNet, VGG-19, técnicas tradicionales de machine learning, imágenes digitales de hojas de pimiento morrón.

A Comparative Study of CNNs Architectures in Bell Pepper Leaves Infected with PHYVV or PEPGMV

Abstract. In this paper, we present a comparative study of two Convolutional Neural Networks (CNNs) architectures: MobileNet and VGG-19. These CNNs architectures are used to build predictive models to classify bell pepper leaves into three different classes, healthy or infected with virus PHYVV or PEPGMV from digital images. The idea of this study is to measure the accuracy, precision and recall from CNNs architectures, beyond the traditional Machine Learning techniques, and to prevent and detect pests and diseases in bell peppers plants. The results obtained in the experiments show a great advantage of the CNNs over the traditional Machine Learning techniques.

Keywords: convolutional neural networks (CNNs) architectures, MobileNet, VGG-19, traditional machine learning techniques, bell pepper leaves.

1. Introducción

Las plagas y enfermedades de plantas son el principal problema en agricultura que podría desestabilizar la producción global de alimentos. El pimiento morrón es un cultivo de gran valor económico que se extiende globalmente. La producción y comercialización de pimiento morrón ha ido en crecimiento en los últimos años. Solo en México, la producción de 2016 alcanzó 104.4 miles de toneladas y se posiciona como el quinto producto agrícola más exportado [18].

En este trabajo, se lleva a cabo un estudio comparativo entre técnicas de aprendizaje profundo (*Deep Learning*) y técnicas tradicionales de *Machine Learning*. La idea es construir modelos predictivos para clasificar hojas de pimiento morrón en dos clases: sana o infectada. En este trabajo nos concentramos en cultivos de pimiento morrón que sean afectadas por el Virus Huasteco de Vena Amarilla del Chile (PHYVV) [13] o el Virus del Mosaico Dorado del Chile (PEPGMV) [14].

Los métodos tradicionales para análisis de cultivos de pimiento morrón en búsqueda de PHYVV y PEPGMV no son eficientes. Por ejemplo, los expertos en el área hacen uso de laboratorios especializados y el proceso es lento y costoso. Por otro lado, el análisis visual de plantas y hojas por parte de expertos es más rápido que los análisis de laboratorio. Sin embargo, incluso para los expertos, analizar grandes áreas de cultivos se vuelve impráctico. Por estos motivos proponemos automatizar la detección de los virus PHYVV y PEPGMV en hojas de pimiento morrón a partir de imágenes digitales, utilizando técnicas de *Deep Learning* y *Machine Learning*.

La identificación de especies de plantas a partir de imágenes es un reto muy grande en el área de la visión por computadora. En particular, es necesario tratar con formas y texturas irregulares, así como una gran cantidad de variabilidad intra-clase y pequeñas diferencias inter-clase [22]. La literatura reporta un gran número de enfoques para la identificación de plantas a partir de imágenes de sus hojas [8,10,26,27]. Todos estos enfoques tienen excelentes resultados con técnicas tradicionales de *Machine Learning*, pero utilizan conjunto de datos (*datasets*) limitados y con imágenes tomadas bajo condiciones bien controladas (fondo, iluminación, orientación, etc.)

Cuando utilizamos técnicas de *Machine Learning* para reconocer objetos a partir de imágenes, es necesario en primer lugar encontrar una representación formal y estructurada que caracterice los atributos importantes para ayudar al proceso de clasificación. Usualmente, esta caracterización se presenta como un vector, llamado vector de características (*feature vector*). Hasta hace unos años, el proceso de extracción de características era realizado por humanos expertos. Recientemente, este proceso se ha delegado a los algoritmos de *Machine Learning* para identificar las características relevantes para el aprendizaje. Las técnicas de *Machine Learning* que utilizan este método de caracterización se denominan *Deep Learning* [2]. Para diferenciar estos enfoques, llamaremos técnicas tradicionales de *Machine Learning* a aquellas que requieran de una fase de caracterización por parte de humanos expertos.

En este trabajo, presentamos un estudio de las arquitecturas de CNNs MobileNet y VGG-19, y las técnicas tradicionales de *Machine Learning*, Regresión Logística (LR), Red Neuronal Artificial (ANN por sus siglas en inglés) y Máquinas de Soporte Vectorial (SVMs por sus siglas en inglés) sobre un *dataset* de imágenes de hojas de pimiento morrón infectadas con virus PHYVVV o PEPGMV. Las imágenes de las hojas se encuentran etiquetadas como una de las tres clases: sanas o infectada con virus PHYVVV o PEPGMV.

El resto de este trabajo se encuentra organizado de la siguiente manera. En la sección 2, presentamos una revisión de la literatura de enfoques existentes en identificación de plantas a partir de imágenes basados en técnicas tradicionales de *Machine Learning*, así como técnicas de *Deep Learning*. En la sección 3, describimos las arquitecturas de CNNs: MobileNet y VGG-19. En la sección 4, presentamos la adquisición de datos y preprocesamiento de imágenes para la extracción de características para técnicas tradicionales de *Machine Learning*. La sección 5 presenta los resultados de una serie de experimentos detallados basados en simulación, los cuales muestran la precisión de los modelos predictivos. Por último, en la sección 6 presentamos nuestras conclusiones y dirección de investigación futura.

2. Antecedentes y trabajo relacionado

Existen muchos enfoques que se centran en la detección de plantas usando técnicas tradicionales de *Machine Learning*. Por ejemplo, el proyecto LeafSnap [10] identifica especies de árboles a partir de fotografías de sus hojas. Para hacer esto, utiliza técnicas de segmentación y extracción de características, apoyándose principalmente en medidas de la curvatura de las hojas. Así mismo, ApLeaf [27] hace uso de extracción de características (principalmente de hojas) para identificar especies vegetales del mediterráneo Francés.

Jamit et al. [8] reportan un análisis de sensibilidad de varios métodos de caracterización y concluyeron que, para la clasificación de plantas a partir de imágenes de las hojas, las características basadas en textura llevan a modelos más precisos, alcanzando valores de 94 % usando una base de datos controlada de imágenes de hojas cuidadosamente seleccionadas, maduras, y libres de enfermedades o parásitos. Las hojas son colocadas sobre un fondo plano y fotografiadas en un entorno controlado. El experimento incluye solamente siete especies. Los valores de precisión y el hecho de que las características de textura resultan en mejores modelos son consistentes con los resultados reportados por Zhong-Qiu Zhao et al. (2015) [26] con 4,870 imágenes de hojas extraídas de la base de datos del reto ImageCLEF en 2012, correspondiendo a 128 especies.

Sfar et al. (2015) [19] proponen un algoritmo basado en *confidence sets* para clasificar especies de plantas a partir de imágenes de sus hojas. Los resultados reportados corresponden a experimentos llevados a cabo con conjuntos de datos que varían en tamaño y calidad. En las imágenes, las hojas aparecen cuidadosamente colocadas en un fondo blanco. La precisión lograda con esta propuesta es muy buena para imágenes tomadas en condiciones ideales (97 %

para el conjunto de datos Swedish Leaf), pero es drásticamente afectada por dos factores: cuando las imágenes son tomadas en condiciones reales (condiciones de campo) y cuando el número de especies se incrementa (53% para el conjunto de datos ImageCLEF).

El reto ImageCLEF para la identificación de plantas a partir de imágenes es bastante más complejo que la identificación de especies usando solamente imágenes de hojas. Hasta 2011, los investigadores en esta área basaban sus enfoques en técnicas tradicionales de *Machine Learning*. Sin embargo, desde 2012 las técnicas de *Deep Learning* han mostrado ser superiores para identificar plantas a partir de imágenes [4,23]. De hecho, en el reto de ImageCLEF 2016, todos los participantes utilizaron técnicas basadas en *Deep Learning* [5]. El mejor enfoque se basó en Redes Neuronales Convolucionales (CNNs) [11], alcanzando una precisión de 82.7%.

Un enfoque presentado en el reto ImageCLEF 2015 por Reyes et al. (2015) [16] diseñaron un sistema de identificación de plantas utilizando *Deep Learning* que no requiere la caracterización explícita de las imágenes. El sistema propuesto se basa en observaciones de 3 a 5 imágenes del mismo espécimen y no en imágenes únicas para realizar la clasificación. El modelo es un proceso de dos fases. En la primera fase, llamada pre-entrenamiento, 1.2 millones de imágenes fueron utilizadas, donde aparecen objetos (muestras) de 1,000 categorías distintas. Estas categorías corresponden a objetos que un humano puede reconocer con facilidad, sin ser experto en algún área en particular. Estas imágenes se obtuvieron del reto ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [17], en su edición de 2012. En la segunda fase, refinamiento, se utilizaron 91,759 imágenes correspondientes a 13,887 plantas de 1,000 especies diferentes, incluyendo árboles, hierbas, helechos, entre otros. El objetivo de esta fase es aprender a distinguir las categorías de interés de un tema específico a partir de un modelo entrenado para reconocer objetos genéricos (a esto se le conoce como *transfer learning*). Con este modelo, Reyes et. al. obtuvieron una precisión en la identificación de plantas del 65%. Un año después, Lee et. al. (2016) [11] usaron Redes Neuronales Convolucionales (CNNs) para la propuesta que presentaron al reto ImageCLEF. Adicionalmente, utilizaron un proceso conocido como *data augmentation* para enriquecer en tamaño y variedad su conjunto de entrenamiento. Este proceso consiste en aplicar transformaciones aleatorias a las imágenes del *dataset* inicial, tales como cambios en luminosidad, orientación, fondo, oclusiones parciales, entre otras. Con este modelo, Lee et. al. lograron una precisión de 74.2%.

Las técnicas de *Deep Learning* han sido utilizadas exitosamente para detectar plagas en algunos cultivos a partir del análisis de fotografías de sus hojas. Por ejemplo, Sladojevic et. al. (2016) [21] utilizaron CNNs para clasificar 13 enfermedades que ocurren en cinco especies de cultivos pera, cereza, durazno, manzana y uva. En este trabajo, los autores utilizan *Deep Learning* para generar un modelo predictivo que está entrenado con 30,800 imágenes. La base de datos propuesta contiene solamente 4,483 imágenes, pero los autores usan *data augmentation* para incrementar el número de imágenes. La precisión de este enfoque es de 96% en Top-1 y 99.99% en Top-5. Una respuesta de Top-K retorna las K

categorías a las que el objeto es más probable que pertenezca de acuerdo al modelo predictivo. Si la categoría correcta se encuentra en cualquiera de las K propuestas, la respuesta se considera correcta.

Los enfoques más recientes en materia de identificación de plantas a partir de imágenes utilizando Redes Neuronales Convoluciones (CNNs) se describen a continuación.

Lee et. al. [11] utilizaron una arquitectura de red VGG16 modificada para el reto LifeCLEF 2016, con la meta de identificar 1000 especies de plantas y, al mismo tiempo, rechazar clases desconocidas, obteniendo resultados limitados. Hang et. al. [6] utilizaron una red VGG16 modificada (denominada Bluefield) para el reto LifeCLEF 2016. La red fue modificada removiendo la última capa de *Pooling* con una capa *Spatial Pyramid Pooling* y cambiando la función de activación estándar ReLU por la variante paramétrica, alcanzando un *Mean Average Precision* (MAP) de 0.827, el más alto de entre todos los participantes. Mehdipour et. al. [12] usaron dos arquitecturas de CNNs, en concreto GoogLeNet y VGGNet, pre-entrenadas con el *dataset* ILSVRC 2012 y refinadas con el *dataset* LifeCLEF 2015. Adicionalmente, entrenaron GoogLeNet para rechazar imágenes que no contuvieran plantas. El sistema alcanzó una calificación final de 0.738.

Toma et. al. [24] emplearon una red AlexNet pre-entrenada con el dataset PlantCLEF 2017, alcanzando un *Mean Reciprocal Rank* de 0.361 en su mejor corrida. Pawara et al. [15] utilizaron AlexNet y GoogleNet para clasificar clases de los *datasets* Folio, AgrilPlant y Swedish leaf, usando a su vez una serie de técnicas de *data augmentation* para incrementar la eficacia de los modelos. Barre et al. [1] diseñaron una arquitectura de CNN orientada a la clasificación de hojas, denominada LeafNet. Esta demostró tener un desempeño superior a métodos tradicionales para la clasificación de imágenes en los datasets Foliage, LeafSnap y Swedish Leaf.

Carpentier et al. [3] entrenaron una red con arquitectura ResNet para identificar especies de árboles nativos de Canadá a partir de imágenes de cortezas, logrando un nivel de confianza máximo de 97.81 %. Los resultados obtenidos por Waldchen et al. [25] en su revisión del estado del arte en materia de identificación de plantas demuestran que las CNNs se desempeñan consistentemente mejor que los métodos de clasificación tradicionales.

3. Descripción formal de arquitecturas de Redes Neuronales Convolucionales (CNNs)

La Red Neuronal Convolutiva (CNN) es una clase de red neuronal profunda que se utiliza en tareas de visión por computadora. Fue popularizada por Krizhevsky et al. en 2012 [9]; los autores presentaron la arquitectura de CNN AlexNet, la cual utiliza capas convolucionales consecutivas, y una función de activación ReLU. Esta arquitectura fue la ganadora del reto ImageNet en 2012.

En contraste con las técnicas tradicionales de *Machine Learning*, las CNNs poseen la ventaja de ser capaces de descubrir de forma autónoma las caracte-

terísticas necesarias para discriminar entre las clases que se le presenten. Esto elimina la necesidad de desarrollar vectores de características especializados y les permite ser utilizadas en diferentes tareas de clasificación.

El obstáculo principal cuando se utilizan las CNNs es la vasta cantidad de datos requeridos para su entrenamiento. Sin embargo, proyectos tales como ImageNet han contribuido de gran manera con datasets que cubren la alta diversidad de plantas en el mundo, mientras que proyectos como Pl@ntNet y iNaturalist han construido comunidades de voluntarios para obtener y clasificar imágenes de plantas alrededor del mundo.

3.1. Arquitectura MobileNet

La primer version de la arquitectura MobileNet [7] fue propuesta por Google, para aplicaciones de visión por computadora móviles y embebidas. MobileNet se basa en una arquitectura simplificada que utiliza convoluciones separables en profundidad, o *depthwise-separable convolutions*, para construir redes neuronales profundas y ligeras. La convolución separable en profundidad consiste de dos operaciones de convolución: la convolución en profundidad, o *depthwise convolution*, y la convolución por punto, o *pointwise convolution*. La convolución en profundidad aplica convoluciones por separado a cada canal del tensor de entrada, esto es, una convolución tradicional de $n \times m$ en una imagen a color. Posteriormente, los mapas de activación producto de las operaciones de convolución se concatenan sobre el eje de profundidad. Una convolución tradicional de 1×1 es entonces aplicada al tensor resultante, (*pointwise convolution*), lo cual combina los canales de los mapas de activación concatenados.

Por ejemplo, la convolución bidimensional tradicional opera sobre una ventana de $N \times M$. Sin embargo, también opera sobre todos los canales del tensor de entrada; esto significa que si el tensor de entrada es una imagen RGB (3 canales de profundidad), un kernel de convolución de 5×5 en realidad tendría la forma $5 \times 5 \times 3$. Así, cada operación de convolución realiza $5 \times 5 \times 3 = 75$ operaciones. Usando la *depthwise-separable convolution*, el número de operaciones se reduce a $5 \times 5 + 3 = 28$, lo cual es significativamente más eficiente.

La arquitectura de la red MobileNet se basa en las *depthwise-separable convolution*, con la excepción de su primera capa, la cual es una capa de convolución tradicional. Todas las capas son seguidas de *batch normalization* y cuentan con la función de activación ReLU, excepto la última capa *fully-connected*. La arquitectura completa se describe en la Tabla 1.

3.2. Arquitectura VGG-19

Las arquitectura VGG fue propuesta en 2014 por Simonyan y Zisserman [20]. VGG consiste de *stacks* lineales de bloques conformados por una cierta cantidad de capas convolucionales, una función de activación no lineal y una capa *Max Pooling*, seguidos por 3 capas *fully-connected* y una capa Softmax al final.

La arquitectura VGG19 posee cinco de estos bloques. Los dos primeros contienen dos capas convolucionales de 64 y 128 filtros respectivamente; el bloque

Tabla 1. Arquitectura MobileNet.

Tipo / Stride	Forma de filtro	Tamaño de input
Conv / s2	3 x 3 x 3 x 32	224 x 224 x 3
Conv dw / s1	3 x 3 x 32 dw	112 x 112 x 32
Conv / s1	1 x 1 x 32 x 64	112 x 112 x 32
Conv dw / s2	3 x 3 x 64 dw	112 x 112 x 64
Conv / s1	1 x 1 x 64 x 128	56 x 56 x 64
Conv dw / s1	3 x 3 x 128 dw	56 x 56 x 128
Conv / s1	1 x 1 x 128 x 128	56 x 56 x 128
Conv dw / s2	3 x 3 x 128 dw	56 x 56 x 128
Conv / s1	1 x 1 x 128 x 256	28 x 28 x 128
Conv dw / s1	3 x 3 x 256 dw	28 x 28 x 256
Conv / s1	1 x 1 x 256 x 256	28 x 28 x 256
Conv dw / s2	3 x 3 x 256 dw	28 x 28 x 256
Conv / s1	1 x 1 x 256 x 512	14 x 14 x 256
Conv dw / s1	3 x 3 x 512 dw	14 x 14 x 512
Conv / s1	1 x 1 x 512 x 512	14 x 14 x 512
Conv dw / s2	3 x 3 x 512 dw	14 x 14 x 512
Conv / s1	1 x 1 x 512 x 1024	7 x 7 x 512
Conv dw / s2	3 x 3 x 1024 dw	7 x 7 x 1024
Conv / s1	1 x 1 x 1024 x 1024	7 x 7 x 1024
Avg Pool / s1	Pool 7 x 7	7 x 7 x 1024
Fully-Conn / s1	1024 x 1000	1 x 1 x 1024
Softmax / s1	Clasificador	1 x 1 x 1000

intermedio contiene tres capas convolucionales de 256 filtros, y los dos últimos contienen ambos tres capas convolucionales de 512 filtros cada uno. El 19 representa la cantidad de capas entrenables contenidas en la arquitectura: 16 capas convolucionales y 3 capas *fully-connected*.

Las capas convolucionales cuentan con un campo receptivo de 3×3 , *stride* de 1×1 y *padding* de 1 pixel. Las operaciones de Max Pooling se realizan con un *kernel* de 2×2 y un *stride* de 2×2 . Cada capa oculta de la red cuenta con la función de activación ReLU. La arquitectura VGG-19 se resume en la Tabla 2.

4. Adquisición y pre-procesamiento de datos

Se recolectaron diferentes hojas de Pimiento Morrón (bell pepper), las cuales se encontraban dentro de un invernadero experimental. De acuerdo a las especificaciones del experto en fitopatología, una manera de distinguir una hoja sana de una hoja infectada es enfocándose en los niveles de colores verde y amarillo presentes en la hoja. En Fig. 1, mostramos algunos ejemplos de hojas utilizadas en este estudio.

Reunimos 103 imágenes de hojas sanas, 103 imágenes de hojas infectadas con el virus PHYVV y 104 imágenes de hojas infectadas con el virus PEPGMV.

Tabla 2. Arquitectura VGG-19.

Tipo	N filtros/parámetros
Conv2D	64
Conv2D	64
Max Pool	N/A
Conv2D	128
Conv2D	128
Max Pool	N/A
Conv2D	256
Max Pool	N/A
Conv2D	512
Max Pool	N/A
Conv2D	512
Max Pool	N/A
Fully-Conn	4096
Fully-Conn	4096
Fully-Conn	1000
Softmax	N

Siendo un total de 310 imágenes reunidas. Las imágenes fueron tomadas utilizando un iPhone 6s (dispositivo móvil) con una cámara de 12 MP. Con la idea de resaltar las características de las hojas y mantener enfocada y estable la imagen, se utilizó un dispositivo para estabilizar y enfocar el dispositivo móvil a una altura de 30cm. Cada hoja fue colocada mostrando su área frontal. Además, por cada hoja se realizaron dos fotografías, una de ellas con fondo blanco y la siguiente con fondo negro. El propósito de los fondos diferentes fue obtener las características de las hojas en diferentes entornos, obteniendo así un total de 620 imágenes.

4.1. Pre-procesamiento de datos para la extracción de características

Para las técnicas tradicionales de *Machine Learning*, necesitamos construir vectores de características. Un vector de características describe la información relevante sobre la forma de la hoja contenida en un patrón, así que para las imágenes de hojas de Pimiento Morrón necesitamos obtener la información más

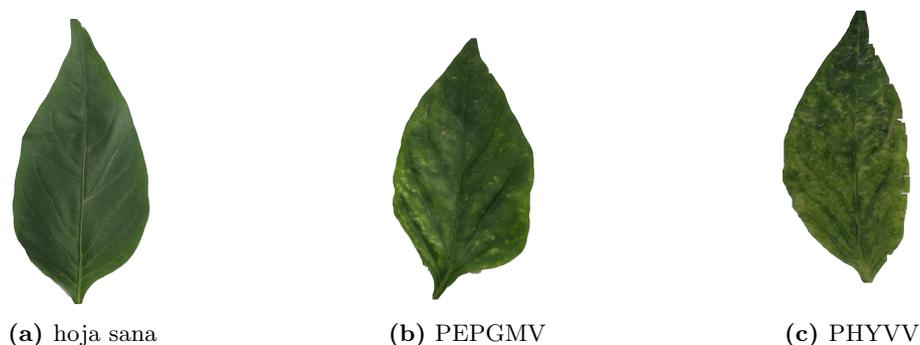


Fig. 1. Ejemplos de imágenes, (a) una hoja sana, (b) una hoja de PEPGMV y(c) una hoja de PHYVV.

relevante y representar dicha información en un espacio de baja dimensión. Utilizamos la biblioteca OpenCV para el procesamiento de las imágenes y resaltar los niveles de verde y amarillo en las hojas de Pimiento Morrón. La imagen original en formato RGB fue convertida a formato HSV. El formato HSV nos permite manipular el brillo y aislar los colores de una mejor manera para el fondo blanco y negro de las imágenes. A partir de la imagen HSV, se removió el fondo y se aplicó un proceso de binarización para resaltar los colores verde y amarillo (obteniendo dos imágenes, una para cada color). Se asignó el valor de 1 para el pixel detectado por color verde/amarillo y 0 al resto de los pixeles, ver Fig 2.

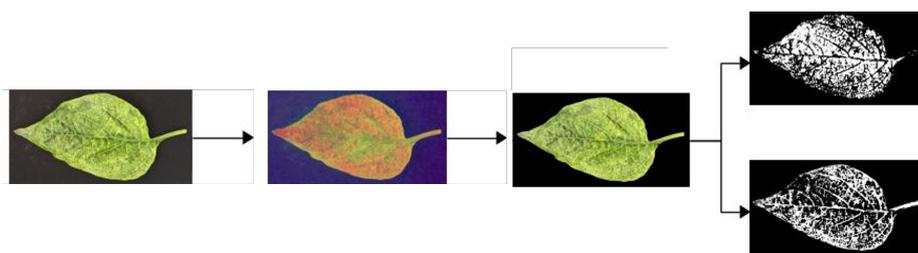


Fig. 2. Flujo de trabajo para el procesamiento de imágenes para resaltar los niveles verde y amarillo en las hojas.

Una vez generadas las imágenes en el pre-procesamiento, obtuvimos las siguientes características a partir de ellas (ver Tabla 3). Cada característica fue calculada tanto para color amarillo como para color verde. Un objeto es un conjunto de pixeles conectados, como resultado del pre-procesamiento de imágenes, estos pixeles adyacentes comparten un porcentaje de color similar.

Tabla 3. Características extraídas de las imágenes de hojas de Pimiento Morrón.

Feature	Description
PP	número de píxeles detectados en la hoja
NO	número de objetos detectados en la hoja
NOG	número de objetos detectados cuyo tamaño es mayor al 1 % de la imagen
TPO	tamaño promedio de los objetos, representado en píxeles
TPOG	tamaño promedio de los objetos filtrados, representado en píxeles.
STO	estadística que mide la variación en el tamaño de los objetos.
STOG	estadística que mide la variación en el tamaño de los objetos filtrados.

Para cada imagen, fue generado un vector de 14 variables (características) y fue etiquetado como clase sana / infectada para indicar la ausencia o presencia de virus (PHYVVV o PEPGMV). En total se generaron 620 vectores de características.

5. Experimentos y resultados

Los experimentos fueron realizados utilizando dos computadoras con las siguientes características en hardware/software. Procesadores Intel Xeon W-2133, 32 GB de RAM, y una tarjeta NVIDIA GTX 1080 cada una. Sistema operativo Ubuntu 18.04, CUDA toolkit 10.0.

Para las técnicas tradicionales de *Machine Learning*, se utilizaron el lenguaje R en la plataforma Rstudio. Para la regresión logística, se utilizaron las funciones *train* del paquete *caret* y el modelo *lm*. Se utilizó el paquete *nnet* para las redes neuronales artificiales y el paquete *e1071* con la función *svm* para las máquinas de soporte vectorial.

Las arquitecturas de CNNs, MobileNet y VGG-19 fueron diseñadas utilizando Keras version 2.2.4, con Tensorflow 1.13.1 como *backend*, y ejecutándose dentro de un entorno virtual de Anaconda 4.6.7 con Python version 3.7.8.

En las técnicas tradicionales de *Machine Learning* se consideraron como el conjunto de datos un total de 620 vectores de características. Para el caso de MobileNet y VGG-19, se consideraron un total de 121 imágenes sin fondo como el conjunto de datos. Para ambos casos, seleccionamos de manera aleatoria el 75 % de los datos para el entrenamiento y el resto fue usado para pruebas.

5.1. Clasificación en dos clases: sana e infectada

Los resultados obtenidos en los experimentos son reportados en términos de *accuracy*, *precision* y *recall*.

Verdadero sana (VS), Falso sana (FS), Falso infectado (FI) y Verdadero Infectado (VI). La *accuracy* es calculada a través de la ecuación (1). La *precision* para la clase sana (2) y para la infectada (3). El *recall* para la clase sana (4) y para la infectada (5).

$$Accuracy = \frac{VS + VI}{VS + FS + FI + VI}, \quad (1)$$

$$PrecisionClaseS = \frac{VS}{VS + FS}, \quad (2)$$

$$PrecisionClaseI = \frac{VI}{VI + FI}, \quad (3)$$

$$RecallClaseS = \frac{VS}{VS + FI}, \quad (4)$$

$$RecallClaseI = \frac{VI}{VI + FS}. \quad (5)$$

Tabla 4. Matriz de confusión de regresión logística.

	clase sana (actual)	clase infectada (actual)
clase sana	32 (VS)	5 (FS)
clase infectada	13 (FI)	71 (VI)
Acc	85.12 %	

Tabla 5. Matriz de confusión de redes neuronales artificiales.

	clase sana (actual)	clase infectada (actual)
clase sana	32 (VS)	1 (FS)
clase infectada	13 (FI)	75 (VI)
Acc	88.42 %	

Tabla 6. Matriz de confusión de máquinas de soporte vectorial.

	clase sana (actual)	clase infectada (actual)
clase sana	32 (VS)	5 (FS)
clase infectada	13 (FI)	71 (VI)
Acc	85.12 %	

Tabla 7. Matriz de confusión de la arquitectura VGG-19.

	clase sana (actual)	clase infectada (actual)
clase sana	37 (VS)	8 (FS)
clase infectada	6 (FI)	70 (VI)
Acc	88.42 %	

Tabla 8. Matriz de confusión de la arquitectura MobileNet.

	clase sana (actual)	clase infectada (actual)
clase sana	43 (VS)	2 (FS)
clase infectada	3 (FI)	73 (VI)
Acc	95.86 %	

Los resultados de las técnicas tradicionales de *Machine Learning* fueron los siguientes. En Tabla 4, presentamos los resultados de la regresión logística (LR), el **accuracy** de LR fue **85.12 %**, la *precision* de la clase sana fue 86.48 %, y para la infectada fue 84.52 %. El *recall* de la clase sana fue de 71.11 % y para la clase infectada fue de 93.42 %. En Tabla 5, presentamos los resultados de ANN, el

accuracy de ANN fue **88.42%**, la **precision** de la clase sana fue de 96.96%, y de la clase infectada fue de 85.22%. El **recall** de la clase sana fue 71.11% y de la clase infectada fue 98.68%. En Tabla 6, presentamos los resultados de SVMs, el **accuracy** de SVMs fue **85.12%**, la **precision** de la clase sana fue 86.48%, y de la clase infectada fue 84.52%. El **recall** de la clase sana fue 71.11% y de la clase infectada fue 93.42%.

Los resultados experimentales para las arquitecturas de CNNs fueron los siguientes. En Tabla 7, presentamos los resultados de VGG-19. El **accuracy** de VGG-19 fue **88.42%**, la **precision** de la clase sana fue 82.22%, y de la clase enferma fue 92.10%. El **recall** de la clase sana fue 86.04% y de la clase infectada fue 89.74%. En Tabla 8, presentamos los resultados de MobileNet. El **accuracy** de MobileNet es **95.86%**, la **precision** de la clase sana es 95.55%, y de la clase infectada es 96.05%. El **recall** de la clase sana es 93.47% y de la clase infectada es de 97.33%.

El mejor **accuracy** fue el obtenido por **MobileNet (95.86%)**, seguido por VGG-19 y RNA (88.42%).

5.2. Clasificación en tres clases: sana, PHYVV y PEPGMV

En los siguientes experimentos, solamente utilizamos las arquitecturas de CNNs. Incrementando el número de clases, la complejidad del proceso de clasificación también aumenta. Para eso, las técnicas tradicionales de *Machine Learning* no pueden reconocer adecuadamente entre los virus PHYVV y PEPGMV. Por ende, decidimos solamente realizar el estudio comparativo entre VGG-19 y MobileNet.

Tabla 9. Matriz de confusión de arquitectura VGG-19.

	clase sana (actual)	clase PHYVV (actual)	clase PEPGMV (actual)
clase sana	37	3	5
clase PHYVV	1	26	13
clase PEPGMV	5	3	26
Acc	75.20%		

Tabla 10. Matriz de confusión de arquitectura MobileNet

	clase sana (actual)	clase PHYVV (actual)	clase PEPGMV (actual)
clase sana	43	0	2
clase PHYVV	3	26	11
clase PEPGMV	0	2	34
Acc	85.12%		

En Tabla 9, presentamos los resultados de VGG-19. El **accuracy** de VGG-19 fue **75.20%** y en Tabla 10, presentamos los resultados de MobileNet, el **accuracy** de MobileNet es **85.12%**.

6. Conclusión

En este artículo se llevó a cabo un estudio comparativo entre las técnicas tradicionales de *Machine Learning* y *Deep Learning* redes neuronales convolucionales (CNNs) utilizadas en el reconocimiento de hojas de Pimiento Morrón infectadas con el virus PHYVV o PEPGMV en imágenes digitales. Tenemos dos etapas experimentales. En la primer etapa, el objetivo es el determinar si una hoja de Pimiento Morrón en una imagen está infectada o no, i.e., clasificar en dos clases: sana o infectada con virus PHYVV o PEPGMV. De acuerdo con nuestros resultados, la arquitectura de CNN, MobileNet, alcanzó el mejor **accuracy** con 95.86%. Para la segunda etapa, solamente usamos CNNs con tres diferentes clases, llamadas sana, PHYVV y PEPGMV. En este caso MobileNet alcanzó el mejor **accuracy** con 85.12%. Como reportamos en los resultados experimentales, las arquitecturas CNNs tienen una clara ventaja sobre las técnicas tradicionales de *Machine Learning* en el proceso de reconocimiento de virus PHYVV o PEPGMV en imágenes digitales de hojas de Pimiento Morrón. Como trabajo a futuro, tenemos la intención de usar otras arquitecturas de CNNs y realizar un estudio comparativo.

Agradecimientos. Los autores desean agradecer el financiamiento proporcionado para esta investigación al Consejo Nacional de Ciencia y Tecnología (CONACyT) y a la Universidad Autónoma de Sinaloa (UAS).

Referencias

1. Barré, P., Stöver, B.C., Müller, K., Steinhage, V.: LeafNet: A computer vision system for automatic plant species identification. *Ecological Informatics* 40, 50–56 (2017)
2. Bengio, Y., Courville, A., Vincent, P.: Representation Learning: A Review and New Perspectives. *IEEE Transactions on Software Engineering* 35(8), 1798–1828 (2013)
3. Carpentier, M., Giguere, P., Gaudreault, J.: Tree Species Identification from Bark Images Using Convolutional Neural Networks. arXiv preprint arXiv:1803.00949 (2018)
4. Goeau, H., Pierre, B., Alexis Joly.: LifeCLEF Plant Identification Task 2015. In: CLEF: Conference and Labs of the Evaluation forum, Toulouse, France, Working Notes of CLEF (2015)
5. Goeau, H., Pierre, B., Alexis Joly.: Plant Identification in an Open World. In: CLEF: Conference and Labs of the Evaluation forum, Evora, Portugal, Working Notes of CLEF (2016)
6. Hang, S.T., Tatsuma, A., Aono, M.: Bluefield (KDE TUT) at LifeCLEF 2016 Plant Identification Task. CLEF Working Notes (2016)

7. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
8. Jamil, N., Hussin, N., Nordin, S., Awang, K.: Automatic Plant Identification: Is Shape the Key Feature? *Procedia Computer Science* 76(5), 436–442 (2015)
9. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: 25th International Conference on Neural Information Processing Systems, pp. 1097–11052, Curran Associates Inc., Lake Tahoe, Nevada (2012)
10. Kumar, N., Belhumeur, P., Biswas, A., Jacobs, D., Kress, W., Lopez, I., Soares, J.: Leafsnap: A Computer Vision System for Automatic Plant Species Identification. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C.(eds.) *Computer Vision ECCV 2012*, LNCS, vol. 7573, pp. 502–516, Springer, Heidelberg. https://doi.org/10.1007/978-3-642-33709-3_36 (2012)
11. Lee, S.H., Chang, Y.L., Chan, C.S., Remagnino, P.: Plant Identification System based on a Convolutional Neural Network for the LifeClef 2016 Plant Classification Task. *CLEF Working Notes*, pp. 502–510 (2016)
12. Mehdipour Ghazi, M., Yanikoğlu, B., Aptoula, E.: Open-set plant identification using an ensemble of deep convolutional neural networks. *CLEF* (2016)
13. Melendrez-Bojorquez, N., Magallanes-Tapia, M.A., Armenta-Anaya, C., Camacho-Beltrán, E., Rodríguez-Negrete, E.A., Leyva-López, N.E., Méndez-Lozano, J.: Pepper huasteco yellow vein virus Associated to Sweet Pepper Disease in Sinaloa, Mexico. *Plant Disease* 100(11), 2338–2338 (2016)
14. Méndez-Lozano, J., Torres-Pacheco, I., Fauquet, C.M., Rivera-Bustamante, R.F.: Interactions Between Geminiviruses in a Naturally Occurring Mixture: Pepper huasteco virus and Pepper golden mosaic virus. *Phytopathology* 93(3), 270–277 (2013)
15. Pawara, P., Okafor, E., Schomaker, L., Wiering, M.: Data augmentation for plant classification. In: *International Conference on Advanced Concepts for Intelligent Vision Systems*, pp. 615–626, Springer (2017)
16. Reyes, A.K., Caicedo, J.C., Camargo, J.E.: Fine-tuning Deep Convolutional Networks for Plant Recognition. *CLEF Working Notes* (2015)
17. Russakovsky, O., Deng, J., Su, H., Krause, j., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision* 115(3), 211–252 (2015)
18. Secretaría de Agricultura y Desarrollo Rural: <https://www.gob.mx/sader/prensa/produccion-nacional-de-chile-alcanza-2-3-millones-de-toneladas>. Last accessed April 1, 2019
19. Sfar, A.R., Nozha, B., Donald G.: Confidence Sets for Fine-Grained Categorization and Plant Species Identification. *International Journal of Computer Vision* 111(3), 255–275 (2015)
20. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
21. Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., Stefanovic, D.: Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification. *Computational Intelligence and Neuroscience* (2016)
22. Sulc, M., Matas, J.: Texture-Based Leaf Identification. In: Agapito L., Bronstein M., Rother C. (eds.) *Computer Vision - ECCV 2014 Workshops*, LNCS, vol. 8928, pp. 185–200, Springer, Heidelberg. https://doi.org/10.1007/978-3-319-16220-1_14 (2015)

23. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston Ma, USA (2015)
24. Toma, A., Stefan, L.D., Ionescu, B.: UPB HES SO @ PlantCLEF 2017: Automatic Plant Image Identification using Transfer Learning via Convolutional Neural Networks. Working Notes of CLEF (2017)
25. Wäldchen, J., Rzanny, M., Seeland, M., Mäder, P.: Automated plant species identification—Trends and future directions. PLoS computational biology 14(4), 50–56 (2018)
26. Zhao, C., Chan, S.F., Cham, W.K., Chu, L.M.: Plant identification using leaf shapes—A pattern counting approach. Pattern Recognition 48(10), 3203–3215 (2015)
27. Zhao, Z.Q., Ma, L.H., Cheung, Y.M., Wu, X., Tang, Y., Chen, C.L.P.: ApLeaf: An efficient android-based plant leaf identification systems. Neurocomputing 151, 1112–1119 (2015)

Entrenamiento dinámico de redes convolucionales profundas para clasificación de imágenes

Oscar Frausto-Pérez, Alfonso Rojas-Domínguez,
Manuel Ornelas-Rodríguez, Héctor Puga, Martín Carpio

Instituto Tecnológico de León,
León, Guanajuato, México
ml2241437@itleon.edu.mx, alfonso.rojas@gmail.com,
mornelas67@yahoo.com.mx, pugahector@yahoo.com,
juanmartin.carpio@itleon.edu.mx

Resumen. En años recientes, el Aprendizaje Profundo se ha posicionado como uno de los paradigmas más exitosos para la solución de problemas de reconocimiento de patrones, tales como clasificación de imágenes. En particular, las CNNs (Convolutional Neural Networks) constituyen uno de los modelos más utilizados para esta tarea. Una CNN típica incorpora cientos de miles de parámetros que deben ser ajustados iterativamente, en lo que se conoce como el entrenamiento de la red, mediante el uso de grandes bancos de imágenes de referencia. Este entrenamiento representa un alto costo de recursos y tiempo computacional incluso cuando se cuenta con GPUs para realizar el procesamiento. En este trabajo se propone un esquema alternativo para el manejo de los datos de entrenamiento para CNNs, que consiste en un muestreo selectivo-adaptativo de los datos. Mediante experimentos con la base de datos CIFAR10 para clasificación de imágenes, se demuestra que este esquema consigue disminuir el tiempo de entrenamiento sin sacrificar significativamente el desempeño de las redes.

Palabras clave: redes neuronales convolucionales, clasificación de imágenes, tiempo de entrenamiento.

Dynamic Training of Deep Convolutional Networks for Image Classification

Abstract. In recent years, Deep Learning has positioned itself as one of the most successful paradigms for the solution of pattern recognition problems such as image classification. Particularly, CNNs (Convolutional Neural Networks) are one of the most used models for this task. A typical CNN incorporates hundreds of thousands of parameters that must be adjusted in what is known as the training of the network, through the use of large reference training sets. This training represents a high cost in computational time and resources even with the availability of GPUs to do the processing. In this work, an alternative scheme for management of the training data for CNNs is proposed which consists in the selective-adaptive sampling of the data. Through experiments performed on the CIFAR10 dataset for image classification, it is shown that the proposed scheme succeeds in decreasing the training time without significantly sacrificing the performance of the networks.

Keywords: convolutional neural networks, image classification, training time.

1. Introducción

Desde 2006, el aprendizaje estructurado profundo, o más comúnmente llamado aprendizaje profundo o aprendizaje jerárquico, se ha convertido en una nueva área de investigación en aprendizaje máquina [1]. Durante los últimos años las técnicas desarrolladas a partir de la investigación de aprendizaje profundo han impactado sobre una amplia gama de trabajos de procesamiento de información y en particular de imágenes, dentro de los ámbitos tradicionales y nuevos, incluidos los aspectos clave de aprendizaje automático e inteligencia artificial [2]. Específicamente, las redes neuronales convolucionales (CNN por sus siglas en inglés), han mostrado un desempeño que se ubica entre los mejores en la actualidad para las tareas de clasificación y detección de objetos en imágenes digitales [1] y también para detección de objetos en video y en tiempo real, donde dos algoritmos muy conocidos debido a su velocidad de procesamiento han destacado, estos son Yolo [3] y SSD [4].

Las CNNs tienen sus orígenes en el trabajo de Yann LeCun [5] y aunque su capacidad estuvo limitada por varios años, la utilización de GPUs para su implementación ha originado un despunte muy importante en el estado del arte a partir del 2011. Otra técnica que se utiliza con frecuencia para mejorar el porcentaje de clasificación de las CNNs es el *aumento de datos* (data augmentation en inglés). En términos de clasificación de imágenes o detección de objetos, la aumentación consiste en aplicar transformaciones a las imágenes de entrenamiento, dichas transformaciones pueden ser: rotación, espejo, recorte, ruido Gaussiano, entre otras.

El entrenamiento de una CNN requiere mucho tiempo y es computacionalmente costoso pues utiliza grandes conjuntos de imágenes de entrenamiento (típicamente se requieren varias decenas de miles de imágenes). Esto dificulta el empleo de otras técnicas que buscan mejorar el desempeño de CNNs o lograr el diseño automático de redes; por ejemplo, los trabajos [6] y [7] proponen llevar a cabo la optimización de la arquitectura de CNNs para las tareas de clasificación de imágenes (ver Fig. 1), mediante el uso de algoritmos evolutivos.

Esto es debido a que inicialmente la cantidad de capas convolucionales, así como la cantidad de mapas de activación por capa (ligados a la cantidad de filtros convolucionales), no se encuentran definidas previamente, sino que deben ajustarse según el problema por resolver, así como otras especificaciones del usuario, sumando al hecho de que para investigadores con poca experiencia es difícil diseñar y proponer nuevas arquitecturas que resulten exitosas.

Aunque dicha estrategia de optimización de la arquitectura de CNN mediante un algoritmo evolutivo ha resultado exitosa, su complejidad computacional es muy elevada y puede ser prohibitiva para muchos investigadores y practicantes que no cuentan con una infraestructura de cómputo suficientemente potente. Por ejemplo, en [7] se menciona que el tiempo requerido para la evolución de una CNN utilizando una variante de algoritmo genético y aplicada al conjunto de datos CIFAR10¹ fue de 35 días de cómputo en una GPU; mientras, en [8] se reporta que el método allí descrito consume

¹ <http://www.cs.toronto.edu/kriz/cifar.html>

2,750 días de cómputo en una GPU para evolucionar una red para el mismo problema de CIFAR10 (para más datos de sistemas del estado del arte, ver [7]).

Es por ello que en este trabajo se propone una metodología para el uso inteligente de las imágenes que son utilizadas para entrenamiento de las redes con finalidad de reducir el tiempo de entrenamiento facilitando así el uso otras técnicas de mayor costo computacional. Nuestra propuesta está inspirada en algoritmos de optimización metaheurísticos o de estimación de distribución, pero en vez de realizar un muestreo de las soluciones (en este caso arquitecturas de red), se aplica una selección y un muestreo dirigido sobre los datos de entrenamiento. A esta propuesta le hemos llamado Entrenamiento Dinámico y se describe en detalle en la Sección 3.

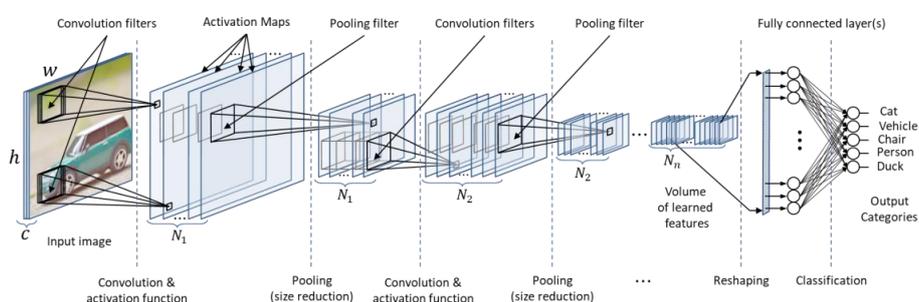


Fig. 1. Estructura de una Red Neuronal Convolutional para Clasificación de Imágenes.

2. Antecedentes

El gran costo computacional y las grandes cantidades de memoria requeridas por las CNNs es una constante en todas las tareas que esta técnica intenta resolver, ya sea clasificación de imágenes, detección de objetos o segmentación semántica. En esta última tarea, los autores de [9] proponen un esquema de muestreo adaptativo que utiliza mapas de error a-posteriori, generados a lo largo del entrenamiento, para enfocar el muestreo en regiones difíciles, lo cual resulta en un mejor aprendizaje.

En el trabajo [10] se propone un método para acelerar el entrenamiento de una red convolucional usando muestreo selectivo aplicado a la detección de hemorragias de retina sobre imágenes a color. En ese trabajo, los autores incrementan dinámicamente la probabilidad de que un muestreo mal clasificado sea seleccionado en la siguiente iteración de entrenamiento. Esto lo llevan a cabo asignando un peso dinámico a cada pixel de una imagen perteneciente al grupo de entrenamiento negativo (refiriéndose a los peores clasificados) y de esta manera indicando su nivel de *informatividad*. Después de cada época de entrenamiento, el peso de cada pixel de entrenamiento negativo es actualizado. Debido al tipo de imágenes que utiliza y a la tarea que pretende resolver, se observa que su propuesta no trata de manera global la imagen, sino que es más específica, afectando solo a ciertos puntos o píxeles en una imagen dada.

En cambio, la tarea seleccionada en el presente trabajo, sobre la cual se aplicará el método propuesto es la de clasificación de imágenes sobre el conjunto de datos CIFAR10. En esta tarea se busca clasificar de manera correcta cada imagen en un conjunto compuesto por 50,000 imágenes de entrenamiento y 10,000 imágenes de

prueba (también llamada validación) distribuidas en 10 categorías. La Fig. 2 presenta imágenes pertenecientes al repositorio CIFAR10. La elección de este conjunto de datos se realizó bajo criterios como que se trata de un conjunto de imágenes relativamente pequeño y bien conocido sobre el que se han reportado numerosos resultados exitosos en la literatura. En otras palabras, seleccionamos CIFAR10 porque se trata de un problema de diagnóstico, no con el objetivo de superar el desempeño de redes profundas existentes, sino con el de mostrar la efectividad de nuestra propuesta en lo que corresponde a lograr un entrenamiento más eficiente de dichas redes.

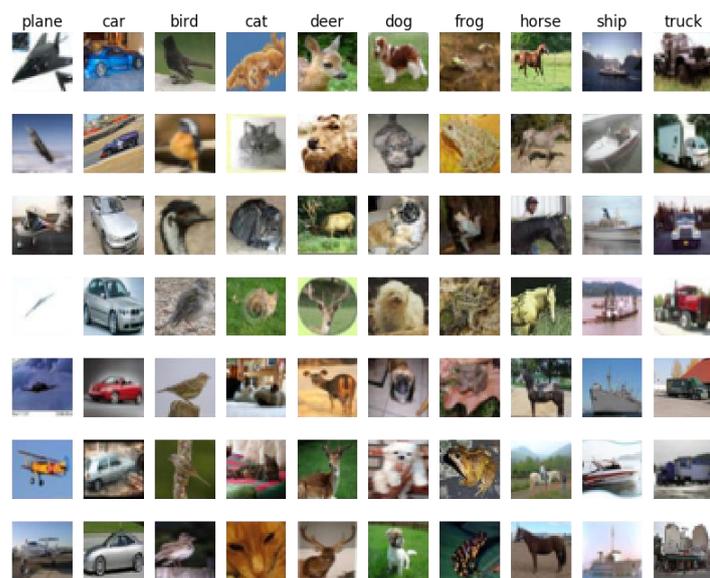


Fig. 2. Imágenes pertenecientes al repositorio CIFAR 10, cada columna muestra una selección de imágenes correspondientes a cada etiqueta.

3. Propuesta

En esta sección se describe cada uno de los pasos de nuestra propuesta. Ésta funciona bajo la suposición de que utilizando un muestreo selectivo-adaptativo de imágenes en la etapa de entrenamiento se puede disminuir el tiempo del mismo sin disminuir de manera significativa el desempeño de la red. Los parámetros utilizados por nuestro método son los valores de **selección** (α), el **porcentaje de remuestreo** (β) y el **umbral de rendimiento** (δ).

1. El primer paso es crear un subconjunto A del conjunto de imágenes de entrenamiento original, con base en un valor de **selección** α . Una consecuencia inmediata de esta selección es la creación del subconjunto B , formado por las imágenes que no se hayan seleccionado. En otras palabras, se realiza una partición del conjunto de imágenes original. Cabe mencionar que el desempeño de la red siempre se evaluará con base en un conjunto de validación que es independiente de cualquier conjunto

utilizado en el entrenamiento. Es decir, este último conjunto nunca se utiliza para el ajuste de los pesos de la red. La Fig. 3 muestra de manera gráfica este paso.

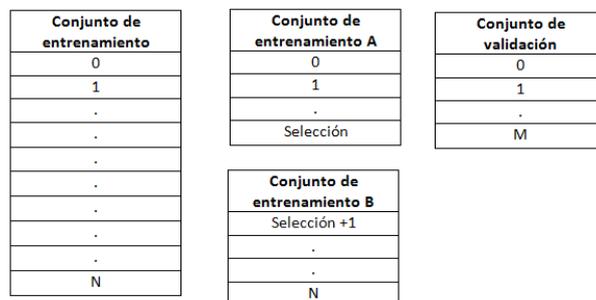


Fig. 3. Partición del conjunto de entrenamiento. Nótese que existe siempre un conjunto de imágenes independiente, usado para validación.

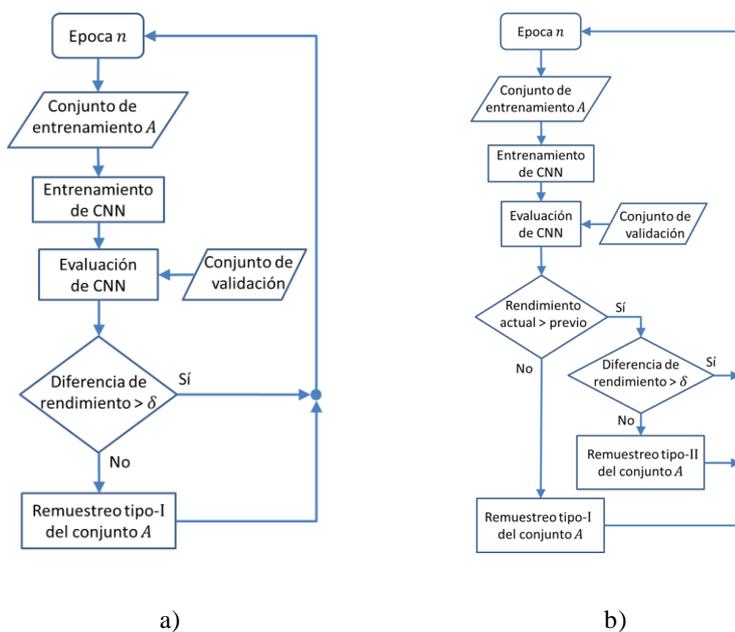


Fig. 4. Diagrama de flujo general del esquema de entrenamiento propuesto.

2. El siguiente paso se realiza dentro del proceso iterativo de entrenamiento de las redes. Cada iteración o ‘época’ se denota como n y una red se entrena durante un conjunto de iteraciones $n \in [1, 2, \dots, N]$. Por cada época se entrena la CNN con el conjunto A, para después ser evaluada con el conjunto de validación y encontrar un valor de desempeño de la red (porcentaje correcto en la clasificación de imágenes).
3. El tercer paso es calcular la diferencia entre dos valores de desempeño de iteraciones consecutivas para guiar la decisión de remuestrear o no el conjunto de

entrenamiento. En nuestro diseño base (Fig. 4-a) si la mejora del desempeño es menor al umbral δ el conjunto A es remuestreado según la Ec. (1).

En una variante (diagrama de flujo en la Fig. 4-b), primero se determina si el desempeño actual es mayor que el anterior. De no ser así, el conjunto A se remuestraa de acuerdo a la Ec. (1), llamado remuestreo tipo-I, usando el conjunto actual. Si el desempeño actual es superior al anterior, se determina si la diferencia supera al umbral de desempeño δ , de no ser así, el conjunto A se remuestraa según la Ec. (2), llamado remuestreo tipo-II, basado en el mejor conjunto de entrenamiento A^* . El remuestreo tipo-I del conjunto A consiste en reemplazar un porcentaje β de sus elementos (en este caso imágenes), seleccionados de manera aleatoria, con elementos del conjunto B , también seleccionados de manera aleatoria. Esto puede definirse como la unión de $\alpha(1 - \beta) = \alpha - \alpha\beta$ elementos seleccionados de A bajo distribución uniforme, con $\alpha\beta$ elementos seleccionados del conjunto B . El resultado de esta operación será el nuevo conjunto A . La Ec. (1) describe este proceso:

$$A \leftarrow \{A(u(\alpha - \alpha\beta))\} \cup \{B(u(\alpha\beta))\} \quad (1)$$

El remuestreo tipo-II es muy similar al remuestreo tipo-I, excepto que en vez de seleccionar elementos del conjunto A actual, se seleccionan del mejor conjunto de entrenamiento A^* , es decir, del que ha resultado en el mejor desempeño hasta la iteración actual. Lógicamente, con este método el mejor conjunto coincide con el conjunto de entrenamiento de la iteración anterior. Este proceso se describe con la Ec. (2). En este caso el valor del porcentaje β se rige por la Ec. (3), donde γ sirve como un factor multiplicativo y su valor se establece cercano a 1.0, mientras que n representa la época en la que se encuentra el entrenamiento y β_0 es el valor inicial de β . De esta manera se logra que el valor de β decrezca de manera potencial:

$$A \leftarrow \{A^*(u(\alpha - \alpha\beta))\} \cup \{B(u(\alpha\beta))\}, \quad (2)$$

$$\beta \leftarrow (\gamma^n) \beta_0. \quad (3)$$

4. Diseño de experimentos

La implementación de nuestra propuesta se realizó utilizando el lenguaje Python junto con la librería Pytorch [11] para la creación de la CNN. Para probar el efecto de los diferentes parámetros (α , β y δ) se realizó una serie de experimentos en los que se varían los valores dados a dichos parámetros, utilizando el diseño base en la Fig. 4-a.

Se experimentó con una selección de $\alpha \in \{15000, 20000, 25000\}$ imágenes para formar el conjunto A y con valores de $\beta \in \{0.1, 0.2, \dots, 0.7\}$ para la fracción o porcentaje de reemplazo y se estableció el valor de γ en 0.97. El valor del umbral de rendimiento se mantuvo fijo en $\delta = 0.01$ debido a que esta tasa de aprendizaje es un valor típico observado durante el entrenamiento regular de la red. De la combinación de estos valores de parámetros, se generan 21 experimentos que se ejecutaron con un GPU GeForce GTX 1070 de Nvidia, y se reportan los tiempos en la sección de resultados de este dispositivo.

El pseudocódigo para el funcionamiento general del sistema se muestra en la Fig. 5, aquí se muestran las llamadas a los métodos TrainEvalCNN donde la red realiza su entrenamiento a partir del currentSetA y su evaluación con el conjunto de validación

validationSet y al método Resample donde se actualiza el conjunto de entrenamiento currentSetA. Los pseudocódigos de los métodos ya descritos se muestran en la Fig. 6 y Fig. 7 respectivamente.

```



---


# Main program


---


For epoch in epochs do:
    currentAcc ← TrainEvalCNN(currentSetA)
    If abs(currentAcc – priorAcc) < δ do:
        currentSetA ← Resample(currentSetA, β)
    end If
end For
priorAcc ← currentAcc


---



```

Fig. 5. Pseudocódigo del programa principal.

```



---


# TrainEvalCNN


---


For [images, labels] in currentSetA do:
    predictions ← model(images)
    loss ← lossFunction(predictions, labels)
    UpdateCNN(loss, optimizer)
end For
currentAcc ← EvalCNN(validationSet)


---



```

Fig. 6. Pseudocódigo para el entrenamiento y validación de la CNN.

```



---


# Resample


---


currentSetB ← setDifference(setTotal, currentSetA)
SetA' ← randomPermutation(currentSetA)
SetB' ← randomPermutation(currentSetB)
currentSetA ← concatenate( SetA' [0: α(1 – β)], SetB' [0: αβ] )


---



```

Fig. 7. Pseudocódigo de remuestro del conjunto A.

La arquitectura CNN utilizada en nuestro estudio está basada en la descrita en [12], donde también se utiliza el conjunto de imágenes CIFAR10 para experimentación y se reportan buenos resultados. Dicha arquitectura es la expuesta en la Tabla 1. Los hiperparámetros son: Tamaño de los lotes de entrenamiento: 512, tasa de aprendizaje inicial igual a 0.001, el optimizador elegido es Adam debido a que se compara de manera favorable con otros optimizadores [13]. La métrica utilizada para la tarea de clasificación de imágenes (también empleada por trabajos como [12]) es la siguiente:

$$Accuracy = \frac{\# \text{ correct predictions}}{\# \text{ total predictions}}. \quad (4)$$

Tabla 1. Arquitectura CNN.

Cantidad	Capa	Tamaño	Canales de entrada	Canales de salida	Función de activación
1	Imagen	32 x 32 x 3	-	-	-

Cantidad	Capa	Tamaño	Canales de entrada	Canales de salida	Función de activación
1	3×3 conv.	32 x 32 x 32	3	32	ReLU
2	3×3 conv.	32 x 32 x 32	32	32	ReLU
1	Max. Pooling	16 x 16 x 32	32	32	-
1	3×3 conv.	16 x 16 x 64	32	64	ReLU
3	3×3 conv.	16 x 16 x 64	64	64	ReLU
1	Max. Pooling	8 x 8 x 64	64	64	-
1	3×3 conv.	4 x 4 x 128	64	128	ReLU
3	3×3 conv.	4 x 4 x 128	128	128	ReLU
1	Max Pooling	2 x 2 x 128	128	128	-
3	3×3 conv.	2 x 2 x 128	128	128	ReLU
1	Avg. Pooling	1 x 1 x 128	128	128	-
1	Fully Connected	128	128	10	Softmax

5. Resultados

Dado el conjunto de parámetros descritos en la sección anterior, se llevó a cabo múltiples veces el entrenamiento de redes profundas CNN con la arquitectura reportada en la Tabla 1 y tomando una combinación de parámetros distinta cada vez, realizándose un total de 21 experimentos. Para cada uno de estos experimentos se registró el máximo porcentaje de clasificación obtenido por la red correspondiente. También se registró el tiempo que le llevó a la red ejecutar el entrenamiento completo, considerando un máximo de 200 épocas. Los resultados se reportan en la Tabla 2 y respectivamente. Puesto que todas las redes alcanzan el 100% de clasificación sobre su respectivo conjunto de entrenamiento, y puesto que dicho conjunto es dinámico, sólo se reporta el porcentaje de clasificación sobre el conjunto de validación, ya que este es realmente el indicativo del rendimiento de las redes.

Tabla 2. Clasificación (mostrada en %) obtenida con distintas combinaciones de parámetros.

Selección, α	Fracción de reemplazo, β							
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	
25,000	82.1	82.5	83.8	83.4	83.3	82.6	84.4	
20,000	80.9	81.5	82.5	83.1	82.6	82.3	81.6	
15,000	79.4	79.3	80.0	80.7	80.1	82.1	82.2	

Como puede observarse en la Tabla 2, el porcentaje de clasificación más alto se logra utilizando la mayor cantidad de imágenes en la selección ($\alpha = 25,000$) y con la mayor fracción de reemplazo ($\beta = 0.7$). Este resultado (resaltado usando un tipo de fuente negrita) indica que el desempeño de la red se ve favorecido al disponer de una mayor cantidad de imágenes para su entrenamiento, pero también al incorporar la mayor diversidad en ese conjunto (lograda por medio del remuestreo).

Por otro lado, analizando la Tabla 3, puede observarse que entre mayores la cantidad de imágenes seleccionadas, mayor es el tiempo requerido para el entrenamiento. Esta observación tiene sentido y es de hecho uno de los supuestos principales por los que se diseñó la presente propuesta. Lo que resulta más interesante es que el parámetro de la fracción de reemplazo prácticamente no tiene un efecto sobre el tiempo de

entrenamiento, puesto que se puede notar cómo los tiempos se mantienen aproximadamente constantes para todos los valores de la Tabla 3 en un mismo renglón.

Tabla 3. Tiempo de entrenamiento requerido con las distintas combinaciones de parámetros.

Selección, α	Fracción de reemplazo, β						
	0.1	0.2	0.3	0.4	0.5	0.6	0.7
25,000	32m:28s	32m:14s	33m:14s	36m:44s	32m:25s	32m:44s	32m:38s
20,000	27m:14s	27m:15s	27m:14s	27m:15s	27m:19s	27m:20s	27m:21s
15,000	21m:22s	21m:21s	21m:26s	21m:30s	21m:24s	21m:25s	21m:32s

Una vez que la combinación de parámetros que genera el mejor rendimiento ha sido determinada, es necesario comparar esos resultados contra las alternativas por defecto, es decir, contra redes entrenadas sin el esquema de entrenamiento dinámico. En las Figs. 8-10 se presentan las gráficas de clasificación como función de las épocas de entrenamiento. La Fig. 8 muestra las gráficas de entrenamiento y validación de una red entrenada con la totalidad de las imágenes disponibles. La Fig. 9 contiene las gráficas correspondientes a una red entrenada con una selección de $\alpha = 25,000$ sin utilizar entrenamiento dinámico. La Fig. 10 muestra las gráficas de una red entrenada usando $\alpha = 25,000$ y $\beta = 0.7$, es decir, entrenada bajo la mejor configuración de entrenamiento dinámico. Los resultados correspondientes se reportan en la Tabla 4.

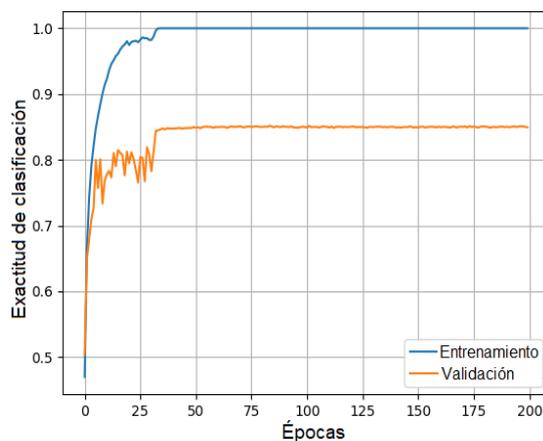


Fig. 8. Resultado de clasificación obtenido utilizando el conjunto completo (50,000 imágenes) de entrenamiento sin entrenamiento dinámico.

La principal observación sobre los resultados en la Tabla 4 es respecto de la exactitud de clasificación. Para la red entrenada con todo el conjunto de entrenamiento la exactitud (de validación) alcanzada es 85.17% y toma aprox. 50 min. completar el entrenamiento. Para la red que dispone solo de la mitad del conjunto de entrenamiento, el desempeño de clasificación es de 81.17% y se completa en 27 min. con 40 s. En otras palabras, al reducir el conjunto de entrenamiento en 50%, la exactitud de clasificación cae 4% pero se ahorra 45% de tiempo de cómputo (compárese Fig. 8 vs Fig. 9).

Tabla 4. Resumen de resultados comparativos, entrenamiento regular vs dinámico.

Gráfica	Imágenes de entrenamiento	Entrenamiento dinámico	Exactitud de clasificación	Tiempo consumido
Fig. 8	100%	No	85.17 %	50m:25s
Fig. 9	50%	No	81.17 %	27m:40s
Fig. 10	50%	SÍ	84.55 %	33m:14s

Ahora bien, al utilizar la mitad del conjunto de entrenamiento, pero introduciendo el entrenamiento dinámico, observamos que la red alcanza 84.55% de clasificación y se completa el entrenamiento en poco más de 33 min. Es decir, mediante el entrenamiento dinámico se puede ahorrar 34% de tiempo de cómputo (17 min. aproximadamente) y sólo se sacrifica un 0.62% en el desempeño de la red. Comparando la Fig. 8 vs la Fig. 10 puede observarse que las gráficas son muy parecidas entre sí.

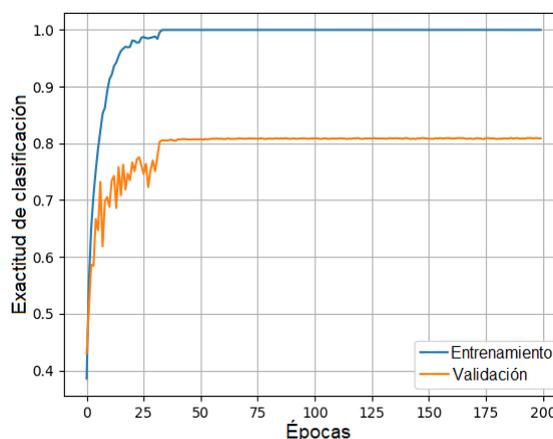


Fig. 9. Resultado de clasificación usando 25,000 imágenes, sin entrenamiento dinámico.

Finalmente, se repite el último experimento pero con la variante del diseño presentada en la Fig. 4-b y un número de épocas igual a 80 (puesto que en las Fig. 8 y Fig. 10 se observa que la red alcanza su potencial máximo en un número pequeño de épocas). El resultado se muestra en la Fig. 11, donde además el desempeño se grafica con elitismo; esto no afecta el resultado, sólo tiene el efecto estético de eliminar oscilaciones.

Cuantitativamente, la variante de nuestra propuesta alcanza un porcentaje correcto de clasificación en validación del 84.85%, es decir, sólo 0.76% abajo del desempeño de la red entrenada con el conjunto de imágenes completo. Esta variante de entrenamiento dinámico requiere sólo un 45.8% del tiempo que toma entrenar la red con el conjunto de entrenamiento completo. En otras palabras, prácticamente se iguala el mejor desempeño de la red y se consigue un ahorro de más de la mitad del tiempo requerido para su entrenamiento. Para mostrar la robustez del método, la técnica se ejecutó por un total de 20 repeticiones, obteniendo un promedio de los valores máximos de clasificación de cada repetición de 85.20%. Fig. 12 presenta el promedio de cada experimento por época, con sus desviaciones estándar correspondientes.

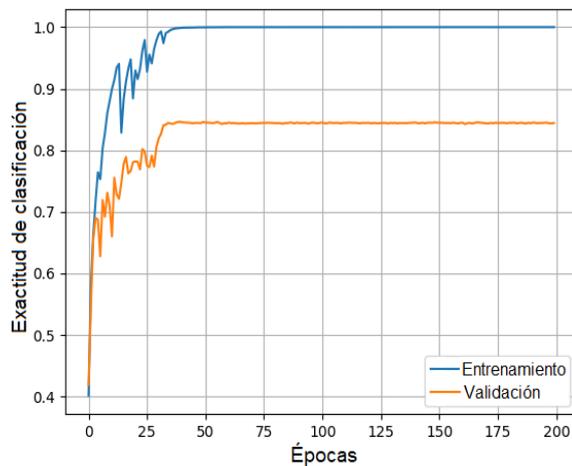


Fig. 10. Resultado de clasificación usando 25,000 imágenes y con entrenamiento dinámico.

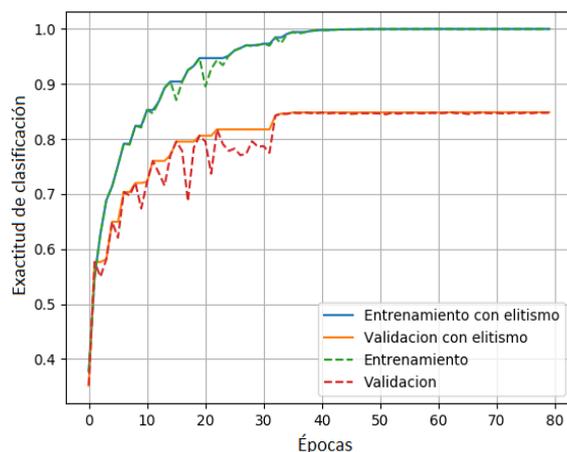


Fig. 11. Resultado de clasificación con 25,000 imágenes y variante de entrenamiento dinámico. Se grafican también las curvas de desempeño con elitismo.

Finalmente, para comprobar que el Entrenamiento Dinámico complementa otras técnicas y mantiene su relevancia en los resultados de clasificación, se realizó un experimento (con 20 réplicas) donde se aplica conjuntamente el entrenamiento dinámico y la técnica de aumento de datos. Esto se compara contra utilizar sólo el aumento de datos en un entrenamiento típico. Los resultados se reportan en la Tabla 5.

Como puede observarse, mediante el entrenamiento dinámico se logra ahorrar 46% de tiempo de cómputo y sólo se sacrifica 1.62% el desempeño de la red, comparado contra el entrenamiento tradicional. La Fig. 13 presenta el promedio de la exactitud alcanzada por época, como promedio sobre las 20 réplicas realizadas, así como su desviación estándar.

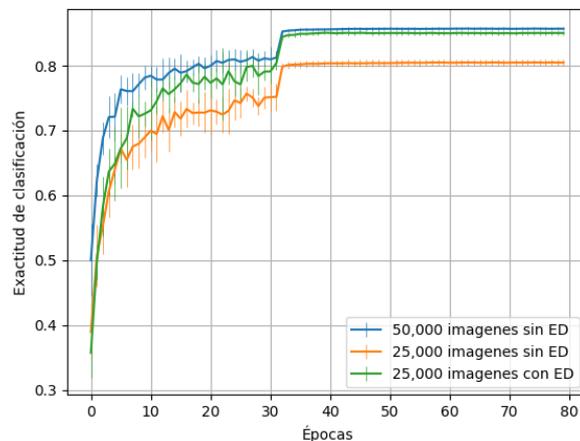


Fig. 12. Resultados de clasificación (validación). Se compara el entrenamiento típico vs Entrenamiento Dinámico (ED).

Tabla 5. Resultados comparativos, entrenamiento dinámico y aumento de datos.

Imágenes de entrenamiento	Aumento de datos	Entrenamiento dinámico	Exactitud de clasificación	Tiempo consumido
100%	Sí	No	90.65 %	25m:36s
50%	Sí	No	86.95 %	13m:40s
50%	Sí	Sí	89.03 %	13m:50s

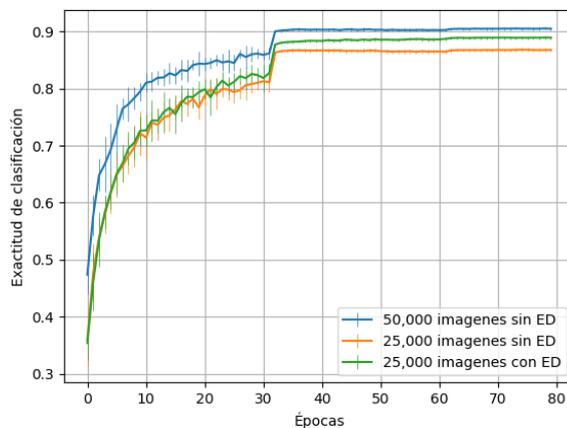


Fig. 13. Resultados de clasificación (validación) con aumento de datos. Se compara el entrenamiento típico vs Entrenamiento Dinámico (ED).

6. Conclusiones y trabajo a futuro

Se observó que mediante nuestra propuesta de entrenamiento dinámico, la cual consiste en un muestreo selectivo-adaptativo de los datos de entrenamiento, se puede disminuir en más de la mitad el tiempo de cómputo necesario para entrenar una CNN

aplicada a la tarea de clasificación de imágenes. Los resultados hasta el momento demuestran que este ahorro en el tiempo de entrenamiento se logra sin afectar negativamente el desempeño de la red. Además, el Entrenamiento Dinámico puede usarse junto con otras técnicas para mejora de desempeño, como Aumento de Datos, con ganancias de casi la mitad de tiempo de entrenamiento ahorrado.

Aunque aún se requiere más evidencia y argumentación, consideramos que la presente propuesta se podría extender y aplicarse en otros problemas de clasificación con diferentes conjuntos de datos, e incluso en otras tareas como detección de objetos o segmentación semántica. Es por ello que como trabajo futuro se contempla aplicar la metodología desarrollada en la tarea de detección de objetos utilizando el modelo SSD, con el objetivo de acelerar el entrenamiento de la red y que sea factible (en términos de costo computacional) emplear una técnica evolutiva para optimizar la arquitectura y los hiper-parámetros de la misma.

Agradecimientos. Este trabajo se llevó a cabo gracias al apoyo del Consejo Nacional de Ciencia y Tecnología (CONACYT) de México, a través de los apoyos 703570 (O. Frausto) y CÁTEDRAS-2598 (A. Rojas). Parte de la experimentación se llevó a cabo en equipos adquiridos mediante el proyecto: "Evolución de Diseños Combinatorios para la Optimización de Problemas de Satisfacción de Restricciones", aprobado por TecNM en 2019, oficio M00.1/0379/2019.

Referencias

1. Deng, L., Dong, Y.: Deep Learning : methods and applications. pp. 197–387 (2014)
2. Arel, I., Rose, D.C., Karnowski, T.P.: Deep machine learning - a new frontier in artificial intelligence research. *IEEE Computational Intelligence Magazine* 5(4), 13–18 (2010)
3. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, Las Vegas (2016)
4. Liu, W. (et al.): SSD: Single Shot MultiBoxDetector. In: *European conference on computer vision*, Amsterdam (2016)
5. LeCun, Y.B.L., Bengio, Y., Haffner, P.: Gradient-Based Learning Applied to document recognition. In : *Proceedings of the IEEE* 86 (11), 2278–2324 (1998)
6. Garcia, A., Lara-Cabrera, R.: EvoDeep: a new Evolutionary approach for automatic Deep Neural Networks. *Journal of Parallel and Distributed Computing* 117, pp. 180–191 (2018)
7. Sun, Y., Xue, B., Zhang, M.: Automatically Designing CNN Architectures Using Genetic Algorithm for Image Classification. *arXiv:1808.03818*, 12 (2018)
8. Real, E. (et al.): Large-scale evolution of image classifiers. In: *Proceedings of the 34th International Conference on Machine Learning*, Sydney, Australia (2017)
9. Berger, L., Hyde, E., Cardoso, M.J., Ourselin, S.: An adaptive sampling scheme to efficiently train fully convolutional networks for semantic segmentation. In: *Annual Conference on Medical Image Understanding and Analysis*, Southampton (2018)
10. van Grinsven, M.J., van Ginneken, B., Hoyng, C.B.: Fast convolutional neural network training using. In: *IEEE transactions on medical imaging* 35 (5), 1273–1284 (2016)
11. Paszke, A., Gross, S., Chintala, S.: Automatic differentiation in PyTorch (2017)

Oscar Frausto-Pérez, Alfonso Rojas-Domínguez, Manuel Ornelas-Rodríguez, Héctor Puga, et al.

12. Truong, T.-D., Nguyen, V.-T., Tran, M.-T.: Lightweight Deep Convolutional Network for Tiny Object Recognition. In: International Conference on Pattern Recognition Applications and Methods, Madeira (2018)
13. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)

Comunicación política en Twitter y su análisis automático mediante el uso de datos ordenados y visualización de información

Rocío Abascal Mena

Universidad Autónoma Metropolitana, Unidad Cuajimalpa,
Ciudad de México, México
mabascal@correo.cua.uam.mx

Resumen. Actualmente, las redes sociales cobran vital importancia debido a que plantean una nueva forma de comunicación a través de medios digitales, donde su alcance es local y global, y su tiempo de difusión es inmediato. Cada vez más, en los procesos electorales no es sorprendente una mayor presencia de los políticos para hacer campañas y ganar contiendas debido a su escasa regulación y a la alta potencialidad para evadir las restricciones que se ponen a las campañas anticipadas. Independientemente de las regulaciones, la presencia de movimientos ciudadanos en diversos ámbitos de la vida diaria muestra claramente que las redes sociodigitales crean una ola polifónica y, definitivamente, un gran gesto de democratización. Este artículo muestra los primeros resultados de un estudio de caso, a partir del uso de datos ordenados y la visualización de información, sobre los perfiles y tuits de los candidatos a Jefe de Gobierno de la CDMX en 2018. Se explican algunas limitaciones de las técnicas actuales para el análisis de tuits y finalmente se presentan unas breves conclusiones con el fin de presentar las posibilidades de un análisis automático que pueda realizado de manera rápida.

Palabras clave: datos ordenados, análisis de redes sociales, visualización de información, comunicación política, Twitter.

Political Communication on Twitter and its Automatic Analysis through the Use of Ordered Data and Information Visualization

Abstract. Currently, social networks are vitally important because they pose a new form of communication through digital media, where its reach is local and global, and its time of dissemination is immediate. Increasingly, in the electoral processes it is not surprising to see a greater presence of politicians to make their campaigns and win contests due to the lack of regulation and the high potential to evade the restrictions placed on the advance campaigns. Regardless of the regulations, the presence of citizen movements in various areas of our daily life clearly shows that socio-digital networks create a polyphonic wave and, definitely, a great gesture of democratization. This article shows the first results

of a case study, from the use of sorted data and the visualization of information, about the profiles and tweets of the candidates for Head of Government of the CDMX in 2018. Some limitations of the techniques for the analysis of tweets are explained and finally some brief conclusions are presented in order to show the possibilities of an automatic analysis that can be done in quickly way.

Keywords: ordered data, social network analysis, information visualization, political communication, Twitter.

1. Introducción

Actualmente, las redes sociales cobran vital importancia debido a que plantean una nueva forma de comunicación a través de medios digitales, donde su alcance es local y global, y su tiempo de difusión es inmediato. Una red social está definida como: “un conjunto organizado de personas formado por dos tipos de elementos: seres humanos y conexiones, entre ellos, las cuales evolucionan orgánicamente a partir de la tendencia natural de toda persona a establecer relaciones y hacer pocos o muchos amigos” [1].

En los últimos años, el uso de las redes sociodigitales ha desempeñado un papel cada vez más visible en la política. Por ejemplo, un número creciente de pakistaníes se dirigió a la gente a través de mensajes de texto, como medios de comunicación alternativos durante la Emergencia de Pakistán del 2007 al 2008. Esto comenzó después de que el presidente del país, el General Pervez Musharraf, suspendió al Jefe de Justicia de la Suprema Corte. En consecuencia, el gobierno canceló redes de teléfonos celulares y bloqueó los canales de noticias [2]. En el año 2010, en algunos países árabes, como Irán, se llevó a cabo la llamada “Primavera Árabe” a través de protestas a gran escala que fueron coordinadas a partir de Twitter.

En México, se recuerdan movimientos políticos generados inicialmente en las redes sociales como #Yosoy132 en el 2012, #Ayotzinapa, #TodosSomosIPN, #nosfaltan43 en el 2014 o #Gasolinazo2017 en 2017. Todos ellos tienen características semejantes como es el uso de las redes sociales para manifestar el repudio de hechos suscitados permitiendo llegar a cientos o miles de seguidores. Esto se debe en gran medida a una regla incontestable del juego: ni emisores ni receptores registran su verdadero nombre. De igual manera, en los procesos electorales no es de sorprenderse una mayor presencia de los políticos para hacer campañas y ganar contiendas debido a su escasa regulación y a la alta potencialidad para evadir las restricciones que se ponen a las campañas anticipadas.

Independientemente de las regulaciones, la presencia de movimientos ciudadanos, en diversos ámbitos de la vida diaria, muestra claramente que las redes sociodigitales crean una ola polifónica y, definitivamente, un gran gesto de democratización.

En las siguientes secciones se presenta: 1) el trabajo relacionado con el análisis de Twitter y la política desde la Computación, 2) se muestran los primeros resultados de un estudio de caso sobre los perfiles y tuits de los candidatos a Jefe de Gobierno de la CDMX en 2018 a partir del uso de datos ordenados, 3) se explican algunas limitaciones de las técnicas actuales para el análisis de tuits y 4) finalmente se presentan unas breves conclusiones.

2. El estudio de Twitter a través de la computación

El estudio de las redes sociales aunado a la política es un campo de investigación que resulta interesante e importante al aprovechar la información generada, por la sociedad, en la que las formas de interacción social están mediadas por el uso de las nuevas tecnologías. En Twitter, la continua evolución de las etiquetas, precedidas por el símbolo # y nombradas como hashtags, permite seguir el acontecimiento diario a partir de los ojos de miles de ciudadanos que son organizadores, y protagonistas creando símbolos que permiten la colaboración en red. Podemos, así, observar una red social que se ha convertido en un espacio para el humor, el ingenio y el discurso directo.

Twitter es una red social bastante efectiva al aprovechar en la mínima expresión de sus mensajes la posibilidad de llegar al otro. Esta característica, que nos remite a tiempos del uso del telegrama, resulta paradójica al contrastarla con el flujo incesante de información que tenemos hoy en día. La restricción en el uso de caracteres parece ser una ventaja que incita a una rápida publicación, recepción y visualización de los mensajes. Así, la principal ventaja es la facilidad e invitación a la participación masiva, como la creación de recursos encaminada a la optimización de la significación y el espacio.

En el área de la Computación existen numerosas investigaciones que recuperan información proveniente de Twitter ya sea a través de un determinado hashtag (#) o de un actor (@). A partir de los datos recuperados es posible analizar a los individuos y/o colectivos que componen los mensajes, determinar la posición de los actores en la red, conocer la intención del mensaje (si es positivo o negativo) o analizar la evolución de la red a través del tiempo, por citar algunos ejemplos de aplicación.

En relación a los métodos de análisis de datos obtenidos de Twitter, en [3] se describen los tres más recurrentes los cuales son: 1) análisis de contenido: estudio de tuits mediante técnicas de minería de texto; 2) análisis de redes: detección de patrones de relaciones entre usuarios y 3) análisis de sentimiento: clasificación de tuits de acuerdo con la actitud expresada en ellos. Sin embargo, una gran mayoría de las investigaciones que utilizan Twitter como su fuente principal de datos realizan lo que se llama análisis de sentimiento y minería de opiniones. Para ello se requieren técnicas provenientes del área de Procesamiento del Lenguaje Natural, la Lingüística computacional y la Minería de textos. Este último, también se refiere a la minería de opiniones y al análisis de sentimientos. Ambos están relacionados con la obtención de información a través del análisis de las emociones que experimenta el usuario o su punto de vista acerca de algo en particular (reseñas, *posts*, etc.). El cálculo de la inclinación promedio de la opinión hacia cualquier entidad no solo ayuda a las organizaciones comerciales a saber más sobre un posible cliente, en el caso de Twitter es un excelente medio para conocer el punto de vista político del público en general. A su vez, la minería de opiniones ayuda a las organizaciones políticas a tener un mejor entendimiento de sus contrincantes [4].

Algunos estudios sugieren que en el ámbito político y de movimientos sociales basados en redes sociodigitales, los mensajes que aumentan su ira son los más virales y predisponen a los manifestantes a ir más decididos a votar [5], [6]. En el mismo sentido, recientemente, está el trabajo de [4] que afirma que una considerable proporción del discurso político de las elecciones nacionales se lleva a cabo en las redes sociodigitales. A partir del estudio de los temas de mayor prevalencia durante un

determinado tema, se investigó la variación temporal de los sentimientos positivos y negativos a la vez que se examinó la proximidad semántica entre los distintos temas. En este caso, el estudio se realizó en el contexto del referéndum griego del 2015 y las subsecuentes elecciones legislativas en el que como resultado se obtuvo que existía un fuerte sentimiento de anti-austeridad, acompañado con una visión crítica sobre las acciones políticas europeas y griegas [4].

En general, en el contexto mexicano, la gran mayoría de los análisis de Twitter se han centrado en relaciones político-ciudadano, procesos decisorios y movimientos sociales [7,8,9]. Además de ello, es de interés el estudio de los *bots* como un elemento distractor en protestas y contiendas electores. Tal es el caso de investigación que se presenta en [10] y en la que se analizan a los *bots* durante el movimiento mexicano #YaMeCansé.

En la siguiente sección se presenta el uso de datos ordenados a partir de perfiles de Twitter con el fin de analizar tuits recuperados durante un periodo de tiempo.

3. Uso de datos ordenados para el análisis de tuits a partir de perfiles de usuario

Uno de los principales intereses en el estudio de los tuits tiene que ver con el seguimiento de las cuentas o perfiles de los usuarios (precedidos por el símbolo @) debido a que la sola recuperación de información puede darnos indicios acerca de la influencia o liderazgo de estos.

El año 2018, para México, se caracterizó por su intensa actividad comicial dado que se llevaron a cabo elecciones federales como locales en todos los estados del país, incluyendo la Ciudad de México (CDMX). Para la CDMX, en la contienda, participaron 7 candidatos, los seis primeros forman parte del ejercicio que se presenta en este artículo:

1. Alejandra Barrales Magdaleno, de la coalición “Por la CDMX al Frente”, @Ale_BarralesM.
2. Claudia Sheinbaum Pardo, de la coalición “Juntos Haremos Historia”, @Claudiashein.
3. Mikel Arriola, del Partido Institucional Revolucionario (PRI), @MikelArriolaP.
4. Mariana Boy Tamborrell, candidata del Partido Verde Ecologista de México, @mariana_boy.
5. Purificación Carpinteyro Calderón, del Partido Nueva Alianza, @PuriCarpinteyro.
6. Marco Antonio Ignacio Rascón Córdova, del Partido Humanista de la Ciudad de México, @MarcoRascon.
7. Lorena Osornio Elizondo, aspirante sin partido, @LoreOsornio_.

El objetivo principal del ejercicio es mostrar algunos grafos y tablas que son posibles generar con solo algunos datos recuperados de cada perfil. En este caso se presentan 4 diferentes ejemplos en los que se comparan los perfiles de los candidatos con el fin de analizarlos y poder sacar algunas conclusiones acerca de la información que comparten

en sus redes: 1) frecuencia de las palabras utilizadas; 2) frecuencia de las palabras y su proximidad con respecto a otro perfil; 3) palabras distintas más frecuentes entre dos perfiles y 4) palabras con mayor frecuencia en los retuits hechos por los seguidores para un perfil en particular.

Antes de comenzar, es necesario considerar aspectos técnicos relacionados con la extracción de estos a partir de Twitter, así como conocer los principales métodos de procesamiento. A su vez, es importante tener en mente las limitaciones inherentes a este objeto de estudio antes de iniciar una investigación basada en datos de trazo digital (los datos que dejan los usuarios a su paso por las redes sociodigitales). Una restricción para la extracción tiene relación con los API (*Application Programming Interface*) de Twitter, el cual cuenta con dos de ellas para obtener los datos: *streaming* y *REST*. La primera API permite descargar una muestra aleatoria del flujo total de tuits en tiempo real o filtrar por características específicas, y únicamente se tiene acceso al 1% del flujo total de tuits generados. Por su parte, la API REST accede a las bases de datos de Twitter y ofrece información almacenada en ellas. La solicitud a la API REST está limitada a periodos de quince minutos y a cierta cantidad de ítems según el dato solicitado. También, es posible obtener los datos a partir de diferentes herramientas tales como TAGS (<https://tags.hawksey.info/>), The Archivist (<https://es.tweetarchivist.com/>), Gnip (<https://gnip.com/>), NodeXL (<http://nodexl.codeplex.com>), o la librería *twitteR* (<https://cran.r-project.org/web/packages/twitteR/twitteR.pdf>) de R, las cuales combinan extracción, procesamiento y visualización, aunque están sometidas a las restricciones de las API.

Para el presente caso de uso se utilizó la librería *twitteR* de R, lenguaje de programación en el que se pueden aplicar técnicas estadísticas, para realizar la conexión con Twitter y especificar la cuenta de los seis usuarios con el fin de recuperar sus respectivos tuits. Así, el primer paso del proceso consistió en realizar la conexión con Twitter para luego recuperar los tuits a partir de cada perfil. Para el presente ejemplo se utilizan únicamente los tuits generados durante el debate del 18 de abril de 2018.

Para poder trabajar con los datos recuperados de cada perfil es importante que estos estén ordenados (en inglés se nombra a los datos ordenados como *tidy data*), para ello se utilizó la librería de R que se llama *tidytext* [11] y cuyo principal propósito es limpiar los datos antes de proceder a utilizarlos. Usando la librería es posible contar con conjuntos de datos limpios que son fáciles de manipular, modelar y visualizar, y tienen una estructura específica: cada variable es una columna, cada observación es una fila y cada tipo de unidad de observación es una tabla. Los conjuntos de datos ordenados proveen una forma estandarizada para vincular la estructura de un conjunto de datos (diseño físico) con su semántica (su significado) [12].

Los datos ordenados son una forma estándar de mapear el significado de un conjunto de datos a su estructura. Un conjunto de datos puede ser desordenado u ordenado dependiendo de cómo las filas, columnas y tablas coinciden con las observaciones, variables y tipos [12]. En este caso los datos ordenados, que se han extraído de Twitter, responden a:

1. Cada variable forma una columna. Por ejemplo, como variable se tiene el tuit, la fecha de emisión, la localización, el idioma, etc.
2. Cada observación forma una fila. En este sentido, una observación corresponde a cada tuit realizado, así como todos sus atributos o variables.

3. Cada tipo de unidad de observación forma una tabla. En este caso corresponde a cada uno de los perfiles de los candidatos.

El conjunto de datos es una colección de valores, usualmente pueden ser tanto números o cadenas de caracteres. Cada valor pertenece a una variable y a una observación. Por su parte, una variable, contiene todos los valores que miden al mismo atributo subyacente en unidades (número de seguidores, número de respuestas, etc.). Mientras tanto, una observación contiene todos los valores medidos en la misma unidad (como el nombre del perfil, a quién está dirigido el mensaje, o la fecha) a través de los atributos.

En la Fig. 1, se muestra una tabla para el perfil, donde la unidad de observación corresponde a @Ale_BarralesM. En este caso se han extraído 19 parámetros los cuales corresponden a las variables: *text*, *favorited*, *favoriteCount*, *replyToSN*, *created*, *truncated*, *replyToSID*, *id*, *replyToUID*, *statusSource*, *screenName*, *retweetCount*, *isRetweet*, *retweeted*, *longitude*, *latitude*, *location*, *language*, y *profileImageUrl*. Se extrae una tabla, unidad de observación, para cada uno de los perfiles.

text	favorited	favoriteCount	replyToSN	created
Es inaceptable que hoy, @Claudiashein nos hable com...	FALSE	217	Ale_BarralesM	2018-04-19 01:47:11
Mientras la ciudad necesitaba agua, @Claudiashein la ...	FALSE	113	Ale_BarralesM	2018-04-19 01:40:32
.@Claudiashein es la <U+00FA>nica Secretaria de Me...	FALSE	147	Ale_BarralesM	2018-04-19 01:39:01
Estoy obligada a se<U+00F1>alar que @Claudiashein...	FALSE	330	NA	2018-04-19 01:31:28

Fig. 1. Ejemplo de tabla, unidad de observación, para @Ale_BarralesM antes de aplicar *tidytext* para ordenar los datos.

Posteriormente, se aplica la librería *tidytext*, ver Fig. 2, para trabajar con datos ordenados y solo con aquellos que se necesitarán para el análisis posterior. En este caso, los atributos seleccionados y que forman parte de los datos ordenados son: *screenName*, *created*, *retweetCount* y *favoriteCount*. En la Fig. 2 se muestra un ejemplo con el perfil de @Ale_BarralesM aunque la tabla creada contiene a todos los candidatos.

text	screenName	created	retweetCount	favoriteCount
#PorLaCDMXalFrente seguiremos luchando por la equ...	Ale_BarralesM	2018-01-10 21:19:39	14	30
#PorLaCDMXalFrente seguiremos trabajando para que...	Ale_BarralesM	2018-01-22 01:16:46	20	56
#PorLaCDMXalFrente vamos a crear las condiciones p...	Ale_BarralesM	2017-12-29 01:30:07	24	68
#PorLaCDMXalFrente vamos a fortalecer los programa...	Ale_BarralesM	2018-02-03 02:28:52	45	106

Fig. 2. Ejemplo de tabla con *tidytext* donde todos los perfiles están ordenados con las variables.

Para el análisis es importante precisar que no es de interés analizar los tuits si están repetidos, es decir que si son retuits enviados por los seguidores con una vez que se analicen es suficiente. Por lo tanto, uno de los primeros pasos es eliminar los retuits a su vez que se quitan las palabras vacías. Sin embargo, es importante conservar los *hashtags* ya que permitirán realizar un análisis posterior.

Uno de los clásicos análisis que se realizan con texto corresponde a calcular la frecuencia de palabras, es decir cuántas veces cada uno de los candidatos uso cada palabra que aparece en sus tuits. En este caso se presenta, en la Fig. 3, el número total de palabras diferentes utilizadas por cada uno de los actores.

screenName	word	n	total	freq
Ale_BarralesM	frente	154	41167	0.003740860
Ale_BarralesM	buen	152	41167	0.003692278
Ale_BarralesM	mujeres	152	41167	0.003692278
MarcoRascon	como	150	30809	0.004868707
Claudiashein	colonia	149	29545	0.005043155
Ale_BarralesM	buenos	148	41167	0.003595113
MarcoRascon	hoy	147	30809	0.004771333
MarcoRascon	independiente	146	30809	0.004738875

Fig. 3. Ejemplo de la tabla que muestra el total y el porcentaje de frecuencia para cada una de las palabras utilizadas por cada candidato.

Una de las razones de utilizar R está en su gran potencial para generar gráficos que permiten visualizar los resultados obtenidos. Por ejemplo, en la Fig. 4, se hace una comparación del número de distintas palabras empleadas entre los candidatos Claudia Sheinbaum (@Claudiashein) y Mikel Arriola (@MikelArriolaP).

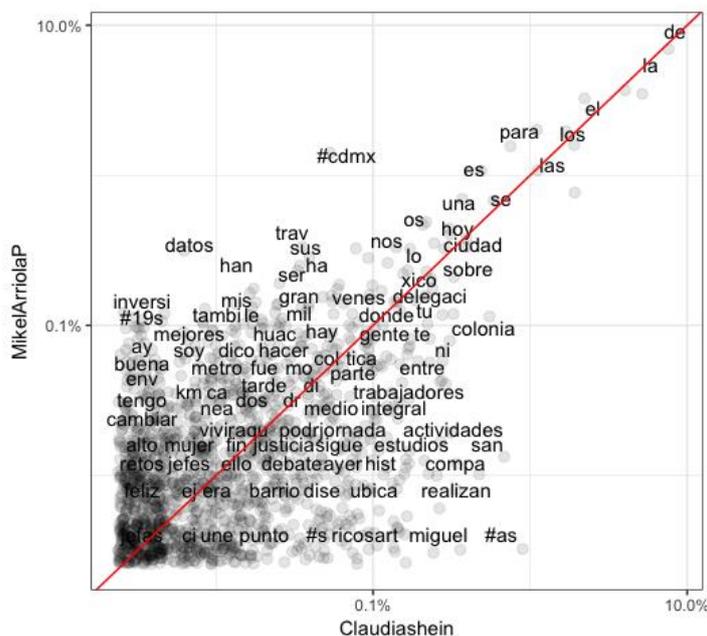


Fig. 4. Comparación en el uso de palabras entre dos de los candidatos.

Las palabras que están cercanas a la línea, en la Fig. 4, tienen frecuencias similares en ambos corpus de los candidatos. Por ejemplo, “fin”, “aquí”, “gente”, “hoy”, “ciudad”. En cambio, las palabras que están lejos de la línea se encuentran más en un

perfil que en el otro. Por ejemplo, Mikel Arriola aborda más el caso de “#19s” o “#cdmx” que Claudia Sheinbaum.

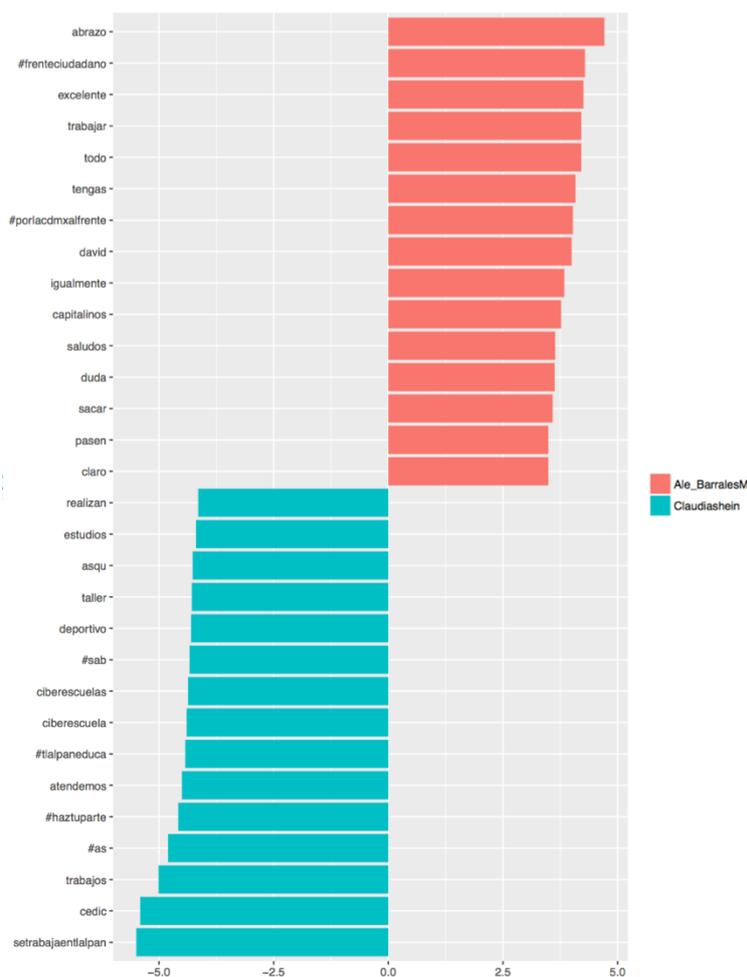


Fig. 5. Comparación en el uso de palabras entre las candidatas Alejandra Barrales y Claudia Sheinbaum.

Para poder obtener mayor información acerca del contenido que cada candidato publica, es posible saber cuáles son las 15 palabras más distintivas para cada cuenta. En las Fig. 5 y 6 se muestran dos diferentes ejercicios de comparación. En el primero, se utilizaron las cuentas de @Ale_BarralesM y @Claudiashein en el que se aprecia por una parte las palabras de “#frenteciudadano” y “#porlacdmxalfrente” característicos de la candidata Alejandra Barrales. En el caso de la candidata Claudia Sheinbaum destaca su trabajo en la Delegación Tlalpan con las palabras “#tlalpaneduca”, “setrabajaentlalpan” o “cedic” que se refiere a uno de los centros comunitarios.

Con el análisis de la Fig. 6, es posible vislumbrar diferentes problemas abordados, respectivamente, por cada uno de los perfiles. En el caso del candidato Mikel Arriola, al igual que Claudia Sheinbaum, están presentes los temas de la delegación, así como el agua, el sismo, y Xochimilco. Por su parte, las palabras características del discurso de Marco Rascón están representadas con: “#humanicemoslaciudad”, “constituyente”, “independiente”, “asamblea”, “cnte”, “cultura” y hasta “Trump”.

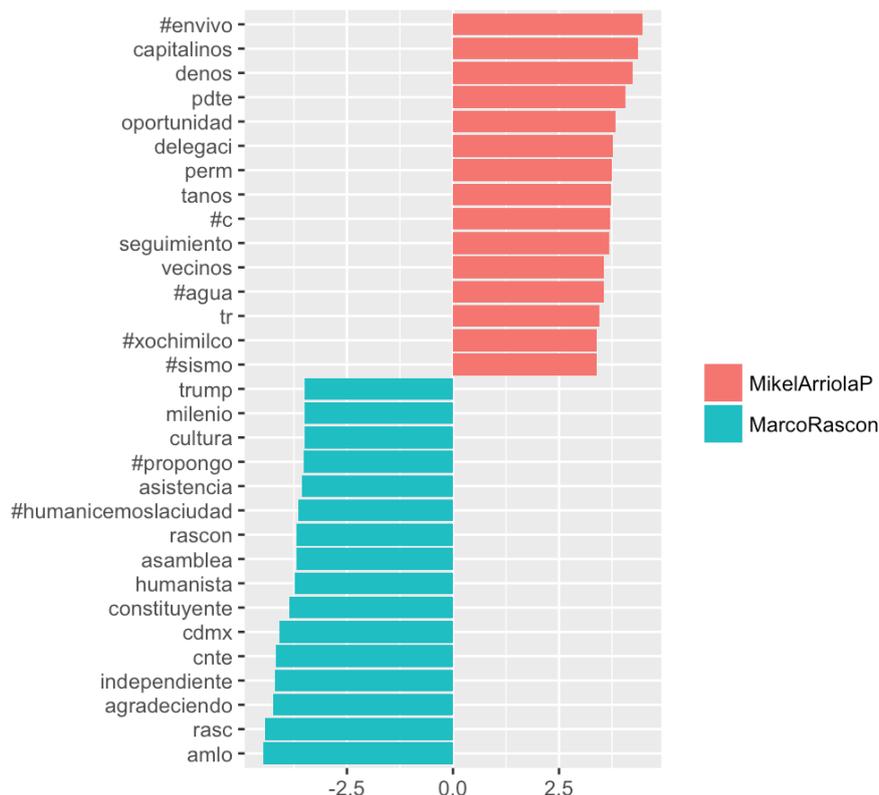


Fig. 6. Comparación en el uso de palabras entre los candidatos Mikel Arriola y Marco Rascón.

Por lo regular, el usuario que sigue, lee y apoya a los candidatos es proclive a compartir, en forma de retuit, información que considera importante e interesante. Sin embargo, hay algunos tuits que son de mayor interés para los usuarios y que pueden ser detectados a partir de las palabras que usan. En el siguiente ejemplo, se muestran las palabras que están contenidas en una mayor cantidad en los tuits compartidos, retuiteados. Para ello, para cada perfil se extrajo el número total de tuits que fueron a su vez retuiteados. Por ejemplo, en la Fig. 7, las palabras que más se encuentran en los retuits de la candidata Purificación Carpinteyro (@PuriCarpinteyro) son: “encuentro”, “neta”, “candidatos”, “trabajo”, “apoyo”, “momento”, “nuestra”, “campa”, “vamos” y “propuestas”. La gráfica muestra la mediana de retuits efectuados a partir de la cual se puede conocer, también, la campaña hecha por los propios ciudadanos en apoyo a su respectivo candidato.

A partir de los ejemplos previos es importante destacar que no se muestra la comparativa de todos los perfiles pero que, en algunos casos, como en la Fig. 7, se puede concluir acerca de quiénes tienen un mayor número de seguidores y apoyo en las redes. Desde luego, para una conclusión así es deseable un análisis más profundo sobre quiénes son estos seguidores y en qué casos los *bots* hacen de las suyas.

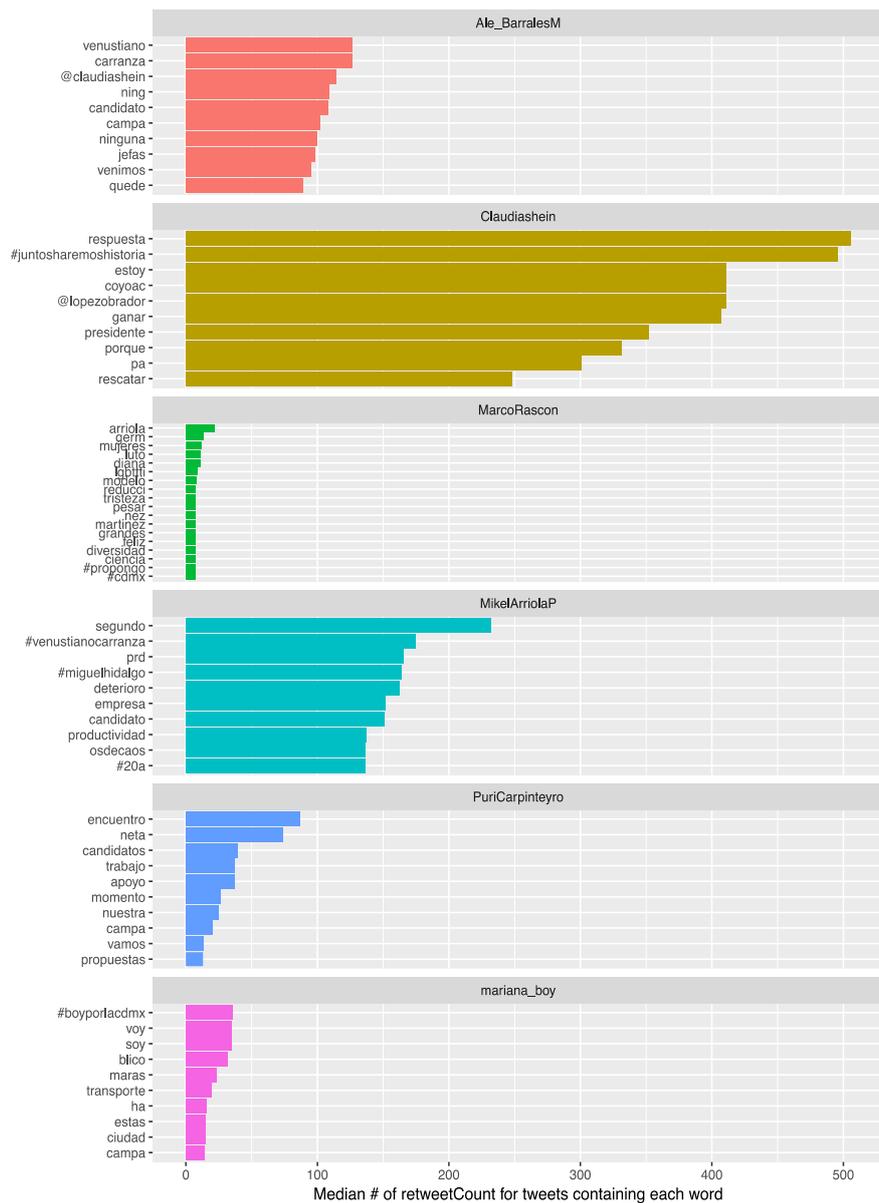


Fig. 7. Palabras que aparecen con mayor frecuencia en los retuits que hacen los ciudadanos de cada uno de los candidatos.

4. Discusión

Al trabajar con datos que provienen de redes sociodigitales es importante considerar que estos se generan a un ritmo considerable y que su manipulación requiere de técnicas que permitan pasar de un formato semiestructurado a uno estructurado.

El uso de principios de ordenamiento de datos puede hacer que muchas tareas de minería de textos sean más fáciles, más efectivas y consistentes con las herramientas que ya están en uso. El ordenamiento de los datos permite manipular, resumir y visualizar las características del texto fácilmente e integrar el procesamiento del lenguaje natural en flujos de trabajo efectivos.

Los datos ordenados permiten que su manipulación sea más fácil al trabajar sabiendo que toda la información con la se cuenta tiene la misma estructura y cuenta con las mismas características. Así es posible realizar diferentes acciones tales como: 1) filtrar el contenido para trabajar solo con ciertas variables; 2) unir o colapsar múltiples valores; y, por ejemplo, 3) ordenar para permitir una visualización de la información pertinente.

A partir de los ejercicios anteriores es posible apreciar algunos de los posibles análisis que pueden realizarse a partir de la recuperación de tuits de determinados perfiles. En estos ejercicios se ha realizado lo siguiente:

1. Para cada perfil se contabilizó la frecuencia de las palabras utilizadas (ver Fig. 3).
2. Para dos perfiles seleccionados se comparó la frecuencia de las palabras y su proximidad (ver Fig. 4).
3. Para dos perfiles distintos, se compararon las palabras frecuentes más distintas entre ellos (ver Fig. 5 y 6).
4. Para cada perfil, se encontraron las palabras que aparecen con mayor frecuencia en los retuits hechos por sus seguidores (ver Fig. 7).

Se ha utilizado únicamente el aspecto sintáctico de las palabras que aparecen en los tuits debido a que se considera que la gran mayoría de estos provienen del mismo contexto por lo que no es necesario explorar su semántica. Sin embargo, algunas de las limitaciones a las que los investigadores se enfrentan al trabajar con tuits tienen que ver principalmente con 1) la inmediatez de los tuits generados; 2) la gran cantidad de tuits, que al menos en contextos políticos, se generan; 3) la ironía de los tuits, la cual sigue siendo un gran reto a la hora de identificar el sentido del tuit y 4) el contexto en el que se generan los tuits que pueden proveer mayor información sobre qué está diciendo cada candidato.

5. Conclusiones

En este artículo se presenta una breve introducción al tipo de análisis que puede realizarse con datos provenientes de Twitter. Debido a la gran velocidad con la que circula la información en las redes sociodigitales, es importante contar con técnicas que permitan que el trabajo de recuperación, extracción, análisis y visualización sea más rápido y sencillo. Para ello, se utiliza R como lenguaje de programación al ofrecer librerías que permiten agilizar la minería de textos, tal como lo hace *tidytext*.

La interacción entre los políticos y los ciudadanos adquiere muchas formas, dependiendo del contexto y espacio donde tenga lugar. En el ambiente digital, ocurre principalmente en medios sociodigitales. Actualmente, Twitter permite recuperar los tuits para poderlos analizar y poder observar a detalle distintas configuraciones comunicativas para estudiarlas, considerando su principal activo en los usuarios y sus redes.

Una comprensión más profunda de la actividad en Twitter se lograría a través del análisis del contenido o del sentimiento expresados en los tuits. Incluir esta dimensión semántica ayudaría en la depuración de los datos extraídos. El análisis puede ser complementado con la sistematización de acciones, como la búsqueda de patrones estructurales a través de distintas redes y la detección de micro-eventos –sucesos relevantes dentro de cada evento– mediante la comparación de fechas de publicación. Asimismo, el análisis del contenido difundido mediante Twitter si bien se hace mediante algoritmos computacionales debería ser complementado con un equipo interdisciplinario de manera a obtener un panorama más completo sobre la implicación, influencia y determinación que tiene la comunicación política a través de redes sociodigitales.

Referencias

1. Christakis, N.A. (et. al.): Conectados: el sorprendente poder de las redes sociales y cómo nos afectan. Pensamiento (Taurus (Firm)). Aguilar, Altea, Taurus, Alfaguara, Santillana Ediciones Generales, S.A. de C.V., 368 p. (2010)
2. Yusuf, H.: Old and New Media: Converging During the Pakistan Emergency (Marzo 2007–Febrero 2008), 61 p., Cambridge, Mass.: Massachusetts Institute of Technology, Center for Future Civic Media. <http://civic.mit.edu/blog/humayusuf/old-and-new-media-converging-during-the-pakistan-emergency-march-2007-february-2008>. Último acceso: 07/05/2019
3. Zimmer, M., Proferes, N.: A topology of Twitter: disciplines, methods, and ethics. *Aslib Journal of Information Management* 66(3), 250–261. DOI: 10.1108/AJIM-09-2013-0083 (2014)
4. Vyas, V., Uma, V.: Open Issues in Opinion Mining. *Extracting Knowledge from Opinion Mining*, p. 283. (2018)
5. Fan, R. (et. al.): Anger is more influential than joy: sentiment correlation in Weibo. *PLoS One* 9(10) e110184. DOI: 10.1371/journal.pone.0110184 (2013)
6. Castells, M.: *Redes de Indignación y Esperanza- Los movimientos Sociales en la Era de Internet*. (1ra. Ed) Madrid: Alianza Editorial, S. A (2012)
7. Cruz, A.: Esfera pública digital: viejas prácticas en 140 caracteres. *Twitter en las elecciones México*. *VIRTUalis* 5(9), 76–89 (2015)
8. Cano, C.: Las redes, las calles y los medios. Análisis visual de las protestas del #1Dmx 2014 en Twitter-México. *VIRTUalis* 6(11), 49–72 (2015)
9. Reyes, J.M.: Twitter, tuits y tuiteros: Análisis de una sentencia en materia electoral. *Juez. Cuadernos de Investigación* (5) (2016)
10. Suárez-Serrato, P. (et al.): On the influence of social bots in online protests. In: *International Conference on Social Informatics*. Springer, Cham, 2016, pp. 269–278 (2016)
11. Silge, J., Robinson, D.: tidytext: Text Mining and Analysis Using Tidy Data Principles in R. *JOSS* 1 (3). The Open Journal. <https://doi.org/10.21105/joss.00037> (2016)
12. Wickham, H., (et al.): Tidy data. *Journal of Statistical Software* 59(10), 1–23 (2014)

Extracción semiautomática de metadatos en documentos no estructurados utilizando procesamiento de lenguaje natural y propiedades tipográficas

Alberto Iturbe Herrera¹, Azucena Montes Rendón²,
Juan-Manuel Torres-Moreno^{3,4}, Gerardo Sierra Martínez⁵,
Noé Alejandro Castro Sánchez¹, Juan Gabriel González Serna¹

¹ Tecnológico Nacional de México / Cenidet, Cuernavaca, Morelos, México

² Tecnológico Nacional de México / Tecnológico de Tlalpan, San Miguel Topilejo,
Ciudad de México

³ LIA / Université d'Avignon, Avignon, France

⁴ Polytechnique de Montréal, Montréal (Québec), Canada

⁵ Universidad Nacional Autónoma de México, Instituto de Ingeniería,
Ciudad de México, México

{iturbe, ncastro, gabriel, amr}@cenidet.edu.mx,
juan-manuel.torres@univ-avignon.fr, gsierram@ingen.unam.mx

Resumen. Hoy en día existen distintos sistemas capaces de extraer metadatos de documento digitales. Sin embargo, la ausencia de una estructura definida en la distribución de los metadatos en documentos de una biblioteca digital de arte presenta un gran problema, esto se debe generalmente al estilo que cada autor o editorial decide utilizar tanto en la portada como en la portadilla del documento. A pesar de que existen herramientas de software que realizan la tarea de extracción de metadatos, éstas se enfocan únicamente en documentos estructurados como publicaciones de revistas, artículos científicos, etc. Los metadatos no son más que datos estructurados de información, es decir, información de información o datos de datos. Este trabajo introduce el uso de técnicas de lenguaje natural y la información tipográfica del texto en el documento para la extracción de metadatos, tales como: título, autores, editorial y fecha de publicación. Los resultados obtenidos en la evaluación con documentos digitales no estructurados indican el potencial del enfoque propuesto, que es capaz de producir buenos resultados en la extracción de metadatos.

Palabras clave: procesamiento de lenguaje natural, extracción de metadatos, información tipográfica del texto, extracción de información, documentos no estructurados.

Semiautomatic Metadata Extraction from Unstructured Documents Using Natural Language Processing and Typographic Text Properties

Abstract. Today there are different systems capable of extracting digital document metadata. However, the absence of a structure defined in the distribution of metadata in documents from a digital art library presents a major problem, this is generally due to the style that each author or publisher decides to use both on the cover and in the cover of the document. Although there are software tools that perform the task of extracting metadata, they focus only on structured documents such as journals, scientific articles, etc. Metadata is not more than structured information data, that is, information information or data data. This paper introduces the use of natural language techniques and typographic information of the text in the document for the extraction of metadata, such as: title, authors, publisher and date of publication. The results obtained in the evaluation with unstructured digital documents indicate the potential of the proposed approach, which is capable of producing good results in the extraction of metadata.

Keywords: natural language processing, metadata extraction, typographic information, information extraction, unstructured documents.

1. Introducción

Los metadatos no son más que un dato estructurado sobre información, es decir, información sobre información o datos sobre datos. Los metadatos en la web se consideran datos que pueden ser guardados, intercambiados y procesados por un sistema informático, generando estructuras que proporcionarán una mejor identificación, clasificación, ubicación y descripción del contenido en la web desde documentos o sitios de Internet [8].

Hoy en día, los medios electrónicos y digitales son indispensables en la vida cotidiana, esto se debe a la gran facilidad de manipulación y movilidad que ofrecen estos medios, a diferencia de los objetos físicos como libros o revistas. Actualmente, es posible obtener una vasta cantidad de información digital de libros que se encuentran disponibles físicamente, es aquí donde intervienen las diferentes formas de obtención de información como repositorios institucionales o bibliotecas digitales.

A su vez, estos eliminan los diferentes estragos que pueden ocurrir en los libros con el tiempo. Es por eso que son considerados un medio indispensable para la preservación de la memoria histórica. Sin embargo, la gran variedad de fuentes de información en los medios electrónicos no se encuentra clasificada, motivo por el cual las búsquedas suelen complicarse, por tal razón es necesario establecer mecanismos capaces de ordenar y extraer información apoyándose en el concepto metadato.

Este trabajo se enfoca en realizar la extracción de metadatos en documentos digitales de arte en español, con el objetivo de semiautomatizar la tarea de extracción de información en documentos de esta naturaleza.

Sin embargo, los documentos no estructurados, denominados así por las siguientes características: la ausencia de una estructura en la distribución de

los metadatos, los errores generados al utilizar técnicas de OCR o la omisión de las mismas, el tipo y calidad de impresión, y el deterioro de estos al haber sido digitalizados dificulta la extracción de información, motivo por el cual, los sistemas actuales de uso comercial no logran extraer la información de forma apropiada.

Este documento está estructurado de la siguiente manera: la Sección 2 presenta información de antecedentes sobre la extracción de información y el procesamiento del lenguaje natural. La sección 3 presenta trabajos relacionados con la extracción de metadatos. En la sección 4 se describen los detalles sobre el sistema de extracción de metadatos propuesto. Los experimentos y resultados se muestran en la sección 5. Finalmente, la sección 6 describe las conclusiones y el trabajo futuro.

2. Background

2.1. Metadato

Por definición, un metadato es un dato estructurado con el objetivo de proporcionar información, describir el contenido, la condición, la calidad y una variedad de características de los datos según sea necesario. El propósito de los metadatos es describir varios atributos de los objetos de información, es decir, otorga significado, contexto y organización, además de mejorar la navegación, búsqueda y administración de archivos en la web.

Etimológicamente los metadatos surgen del griego, $\mu\epsilon\tau\alpha$, meta, que significa "más allá de, después de", en conjunto con el latín *datum*, que significa "lo que se da". Conjunta y literalmente se traduce como "información sobre información." "datos sobre datos" [2].

2.2. Procesamiento de Lenguaje Natural

El procesamiento del lenguaje natural (PLN) se considera un campo interdisciplinario cuyas tareas principales son: la traducción automática, el resumen automático, las búsquedas semánticas, la extracción y recuperación de información, entre otras. El procesamiento del lenguaje natural tiene como objetivo crear sistemas y mecanismos capaces de emular actividades del lenguaje natural, principalmente considerando aspectos lingüísticos.

2.3. Extracción de Información

Esta tarea consiste en extraer las entidades, los eventos y la relación que tienen con respecto a los elementos de un texto o un conjunto de textos. La extracción de información consiste en localizar las partes del texto que contienen información relevante, de acuerdo con las necesidades de los usuarios. Además, esta información se obtiene para ser procesada de forma manual o automática [1].

2.4. Entidades Nombradas

Una entidad nombrada (EN) se define como una unidad léxica que consiste en una secuencia de palabras contiguas que se refieren a una entidad específica, es decir, este elemento referenciado puede ser una persona, un lugar, una organización, cantidades, porcentajes o una fecha [4].

2.5. OCR

OCR, por sus siglas en inglés *Optical Character Recognition*, es un proceso que permite al usuario convertir documentos escaneados, imágenes capturadas previamente con una cámara digital o documentos en formato PDF en documentos con información editable y con opción de búsqueda en ellos.

3. Trabajos relacionados

Una de las tareas de Procesamiento del lenguaje natural es la extracción de información (EI), que implica la extracción automática o semiautomática de información de documentos digitales estructurados, semiestructurados o no estructurados.

Por ejemplo, en [11], Rendón propone una metodología para extraer metadatos en objetos de conocimiento con contenido no estructurado. Este trabajo realiza la extracción de metainformación utilizando las librerías LA-PDFText y GROBID, obteniendo como salida un documento XML con la información identificada a través de etiquetas.

Tkaczyk et. al [12], presentan un sistema integral para la extracción de metadatos en artículos escolares basados en el análisis de la estructura del documento, de encabezado a pie de página. Este trabajo implementó las siguientes bibliotecas: biblioteca iText y LibSVM; y los algoritmos: Docstrum, algoritmos basados en heurísticas de abajo a arriba, agrupación KMeans y campos aleatorios condicionales.

Por otra parte, Guo et. al [6] presentan el sistema SemreX para la extracción de metadatos en artículos científicos. Esta metodología se centra en los documentos IEEE, ACM y LNCS, implementa una serie de heurísticas definidas por los autores, así como el uso de información del texto en el documento como: posición, contenido (texto extraído), tipo de fuente, tamaño de fuente y número de página en el que se encuentra el texto.

Del mismo modo, Huynh et. al [7] presenta la herramienta GATE utilizada para realizar la extracción de metadatos con el propósito de enriquecer una ontología. Este trabajo se centra solo en artículos científicos y utiliza un enfoque basado en las propiedades del texto que implementa Apache PDFBox, JAPE Grammar y el complemento ANNIE.

Morales-Solares et. al presentan en [9] una metodología para extraer los metadatos de las cubiertas de los documentos. Para esto se utilizaron dos métodos: regiones extremas máximas estables (MSER) para texto e imágenes con

fondos complejos, y campos aleatorios condicionales (CRF) para elementos de etiquetado lógico en el documento.

Finalmente, y el enfoque más cercano a este trabajo es presentado por Gao et. al [5] quienes desarrollaron una metodología que utiliza propiedades espaciales del texto para determinar cada metadata. Sin embargo, los documentos que procesaron estaban en idioma chino estructurado.

Analizando los trabajos descritos anteriormente, la mayoría de estos realizar la extracción de metadatos en documentos estructurados, tales como: artículos científicos o de investigación. Basado en esta observación, el presente trabajo introduce el uso de técnicas de procesamiento del lenguaje natural y las propiedades tipográficas del texto en documentos no estructurados en español. La metodología propuesta se evaluó utilizando un corpus de 300 documentos en español en formato PDF, de los cuales se obtuvieron los siguientes metadatos: título, autores, editorial y fecha de publicación.

4. Sistema propuesto

Esta sección presenta las herramientas utilizadas en este trabajo y describelos métodos desarrollados en esta metodología y los corpus utilizados en los experimentos.

4.1. Implementación de la herramienta Freeling

Freeling es una biblioteca desarrollada en el lenguaje de programación C++, esta herramienta nos permite realizar tareas de procesamiento del lenguaje natural, tales como: etiquetado gramatical (PoS, *Part of Speech*), reconocimiento y clasificación de entidades nombradas, desambiguación semántica, análisis morfológico, Detección automática de lenguaje, entre otras.

Además, Freeling puede realizar todas las tareas ya mencionadas en los idiomas: español, inglés, portugués, italiano, francés, alemán, ruso, catalán, galés, croata, esloveno, entre otros. Más información y la documentación de esta herramienta se describen en [10].

4.2. Implementación de la herramienta PDFMiner

PDFMiner es una herramienta para extraer información de documentos en formato PDF, proporcionando una serie de propiedades como: obtener la ubicación exacta del texto en una página y el tipo de fuente correspondiente. Además, esta herramienta permite la conversión de archivos a formato HTML y XML.

Esta herramienta utiliza un enfoque basado en el diseño en el que se utilizan los siguientes parámetros: **M**: *char_margin*, esta variable corresponde a la separación entre caracteres (letras, números o signos), **L**: representa la variable de separación por líneas (*line_margin*), por lo tanto, si dos fragmentos de texto específicos están en líneas diferentes, pero la distancia entre ellos se aproxima al valor de L, los elementos se agruparán como un cuadro de texto. Finalmente, **W**:

es la variable responsable de determinar la distancia mínima entre dos palabras (word_margin) para asignar posteriormente los respectivos espacios en blanco.

Las variables mencionadas anteriormente tienen los valores predeterminados de $M = 1.0$, $L = 0.3$ y $W = 0.2$. De acuerdo con las necesidades del usuario, estos valores pueden modificarse [3]. La figura 1 ejemplifica lo mencionado anteriormente.

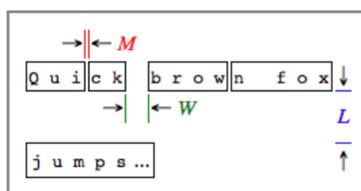


Fig. 1. Parámetros basados en el diseño.

Se requirió una serie de pruebas para determinar que la información extraída de los documentos PDF retenía la mayor cantidad de información posible. Para esto, se utilizó la herramienta en cuestión para extraer el texto usando una línea de comandos. El comando para realizar la extracción es el siguiente:

```
pdf2txt.py -o destination_directory/myfile.html -t html -p 1,3 myfile.pdf
```

En el que podemos utilizar diferentes opciones como:

- o esta variable indica el nombre del archivo de salida.
- p representa las páginas separadas por comas a partir de las cuales se realiza la extracción.
- t corresponde a la variable de formato de salida de la información. Este puede ser texto plano, html o xml.
- O indica el directorio que almacena las imágenes extraídas.
- P Esta variable se usa para especificar la contraseña del documento si es necesario.

La Figura 2 muestra un ejemplo de la información extraída en formato HTML de un documento PDF, en la que se puede apreciar la sintaxis tradicional de las páginas escritas en este formato. Al mismo tiempo, este documento contiene la información correspondiente a la página y la ubicación del texto en ella. Dicha información se encuentra en la línea 11 para este caso; asimismo, las líneas 12 y 15 contienen la información correspondiente al tamaño y el tipo de fuente para los fragmentos de texto indicados. Más información y documentación de esta herramienta se describe en [3].

4.3. Módulo de pre-procesamiento de documentos PDF

Usando la herramienta mencionada anteriormente, se procesaron los archivos y se almacenaron en las carpetas correspondientes, y seguido de esto, la informa-

```

1 <html>
2 <head>
3   <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
4 </head>
5 <body>
6   <span style="position:absolute; border: gray 1px solid; left:0px; top:50px; width:595px; height:842px;">
7 </span>
8   <div style="position:absolute; top:50px;">
9     <a name="1">Page 1</a>
10 </div>
11 <div style="position:absolute; border: textbox 1px solid; writing-mode:lr-tb; left:286px; top:356px; width:158px;
12 height:52px;">
13   <span style="font-family: KEAHPJ+BodoniBT-Roman; font-size:26px">Conceptos claves
14 <br>
15 </span>
16   <span style="font-family: KEAHPJ+BodoniBT-Roman; font-size:26px">de museologia
17 <br>
18 </span>
19 </div>

```

Fig. 2. Información tipográfica extraída en formato HTML.

ción obtenida se almacenó de acuerdo con el formato del archivo. Es necesario mencionar que se desarrollaron algoritmos para el procesamiento de documentos como: la eliminación de saltos de línea en los títulos de los documentos, la eliminación de páginas con información irrelevante y la detección y eliminación de páginas sin contenido.

4.4. Módulo de etiquetado XML

En este módulo, los archivos XML se generaron con el objetivo principal de identificar entidades nombradas tales como: personas, organizaciones, ubicaciones, fechas, entre otras. Sin embargo, al someter las diferentes copias al proceso de etiquetado de PoS utilizando Freeling, fue posible ver que, la mayoría de las palabras que comienzan con letras mayúsculas se identificaron erróneamente como nombres propios, ya sea como persona u organización.

La Figura 3 (a) muestra los ejemplos relacionados con el problema mencionado anteriormente, donde el token con id: **t1.1** contiene la palabra *conceptos* con una clasificación errónea, indicando esto como una persona. De manera similar, el token con id: **t1.5**, que contiene la misma palabra que el token **t1.1**, recibió una clasificación diferente. Otro ejemplo es el token **t1.9** con la palabra *bajo* y la asignación a la clase de organizaciones de manera incorrecta.

Por lo tanto, se desarrolló un conjunto de métodos de mejora de etiquetado de PoS utilizando repositorios construidos. Estos métodos son responsables de modificar los errores mencionados anteriormente para las clases de persona y fecha.

En la gran mayoría de los libros, la fecha de publicación contiene solo el año sin especificar el día y el mes, debido a esto, la herramienta asignó la etiqueta numérica a todos los números encontrados, generando errores con respecto a los números de página y las posibles fechas de publicación. Por esta razón, se desarrolló un método para identificar números no menores de 1800 y no mayores que 2017. Cabe mencionar que estos valores se pueden adaptar a las necesidades.

Una vez que cada método de corrección identifica errores en los textos, extraen información indispensable del fragmento de texto como: id, form y lemma. Posteriormente, si el atributo de form, que representa la palabra en su forma original no presenta incidentes en el repositorio de nombres de personas,

```

1 <sentencia id="t1">
2 <token id="t1.1" form="Conceptos" lemma="conceptos" tag="NP00SP0" ctag="NP" pos="noun" type="proper" neclase="person" nec="PER" >
3 </token>
4 <token id="t1.2" form="claves" lemma="clave" tag="NCFP000" ctag="NC" pos="noun" type="common" gen="feminine" num="plural" >
5 </token>
6 <token id="t1.3" form="de" lemma="de" tag="SP" ctag="SP" pos="adposition" type="preposition" >
7 </token>
8 <token id="t1.4" form="museología" lemma="museología" tag="NCF5000" ctag="NC" pos="noun" type="common" gen="feminine" num="singular" >
9 </token>
10 <token id="t1.5" form="Conceptos" lemma="conceptos" tag="NP00V00" ctag="NP" pos="noun" type="proper" neclase="other" nec="MISC" >
11 </token>
12 <token id="t1.6" form="claves" lemma="clave" tag="NCFP000" ctag="NC" pos="noun" type="common" gen="feminine" num="plural" >
13 </token>
14 <token id="t1.7" form="de" lemma="de" tag="SP" ctag="SP" pos="adposition" type="preposition" >
15 </token>
16 <token id="t1.8" form="museología" lemma="museología" tag="NCF5000" ctag="NC" pos="noun" type="common" gen="feminine" num="singular" >
17 </token>
18 <token id="t1.9" form="Bajo" lemma="bajo" tag="NP00000" ctag="NP" pos="noun" type="proper" neclase="organization" nec="ORG" >
19 </token>
20 <token id="t1.10" form="la" lemma="el" tag="DABF50" ctag="DA" pos="determiner" type="article" gen="feminine" num="singular" >
21 </token>

```

```

1 <sentencia id="t1">
2 <token id="t1.1" form="conceptos" lemma="concepto" tag="NCP000" ctag="NC" pos="noun" type="common" gen="masculine" num="plural" >
3 </token>
4 <token id="t1.2" form="claves" lemma="clave" tag="NCFP000" ctag="NC" pos="noun" type="common" gen="feminine" num="plural" >
5 </token>
6 <token id="t1.3" form="de" lemma="de" tag="SP" ctag="SP" pos="adposition" type="preposition" >
7 </token>
8 <token id="t1.4" form="museología" lemma="museología" tag="NCF5000" ctag="NC" pos="noun" type="common" gen="feminine" num="singular" >
9 </token>
10 <token id="t1.5" form="Conceptos" lemma="conceptos" tag="NP00V00" ctag="NP" pos="noun" type="proper" neclase="other" nec="MISC" >
11 </token>
12 <token id="t1.6" form="claves" lemma="clave" tag="NCFP000" ctag="NC" pos="noun" type="common" gen="feminine" num="plural" >
13 </token>
14 <token id="t1.7" form="de" lemma="de" tag="SP" ctag="SP" pos="adposition" type="preposition" >
15 </token>
16 <token id="t1.8" form="museología" lemma="museología" tag="NCF5000" ctag="NC" pos="noun" type="common" gen="feminine" num="singular" >
17 </token>
18 <token id="t1.9" form="bajo" lemma="bajo" tag="SP" ctag="SP" pos="adposition" type="preposition" >
19 </token>
20 <token id="t1.10" form="la" lemma="el" tag="DABF50" ctag="DA" pos="determiner" type="article" gen="feminine" num="singular" >
21 </token>

```

Fig. 3. Errores iniciales y correcciones con el algoritmo de optimización NER / NEC de las herramientas de Freeling.

palabras clave de organización, organización, número de cuatro dígitos entre los rangos ya mencionados o no es ciudad o país, se modifica a letras minúsculas y, posteriormente, se vuelve a enviar al proceso de etiquetado de PoS y la información anterior se reemplaza con los datos obtenidos, al tiempo que se retiene solo la identificación del token para respetar la posición original. Toda esta información se almacena en archivos temporales separados para cada archivo original respectivamente. La Figura 3 (b) muestra la corrección de los errores en los tokens: **t1.1**, **t1.5**, **t1.9**.

4.5. Módulo de extracción y almacenamiento de metadatos

El siguiente módulo describe el enfoque de este trabajo y las heurísticas para determinar cada uno de los cuatro metadatos.

Detección de autores. Combinando el algoritmo de optimización integrado de Freeling y los repositorios de nombres propios, se agregó un algoritmo capaz de generar tres formas diferentes de los registros en el repositorio de nombres propios, es decir, nombre propio capitalizado (Nombre Propio), minúscula (nombre propio) y mayúscula (NOMBRE PROPIO). A su vez, si este contiene algún signo de acentuación como: acento agudo (´), acento circunflejo (ˆ), acento grave (˘), virgulilla de la eñe (˜), virgulilla de la cedilla (Ç / ç), diéresis (¨), entre otros. La palabra se procesa para generar nuevos registros con la ausencia de estos caracteres (Gutiérrez - Gutierrez), en las tres variantes mencionadas anteriormente.

Sin embargo, a pesar de haber reducido en gran medida el rango de error en relación con la identificación de las entidades nombradas, algunos casos continuaron apareciendo en los que no se detectaron correctamente en los casos en que otros fragmentos del texto tenían el tamaño de fuente más grande como se muestra en la Figura 4, por lo que se sugirió optimizar este módulo descartando todos los elementos de una cadena de texto que tuvieran incidentes en el diccionario de la RAE, realizando el proceso de tokenización de las cadenas de texto y luego realizando una búsqueda en el repositorio generado de la RAE. Por otro lado,

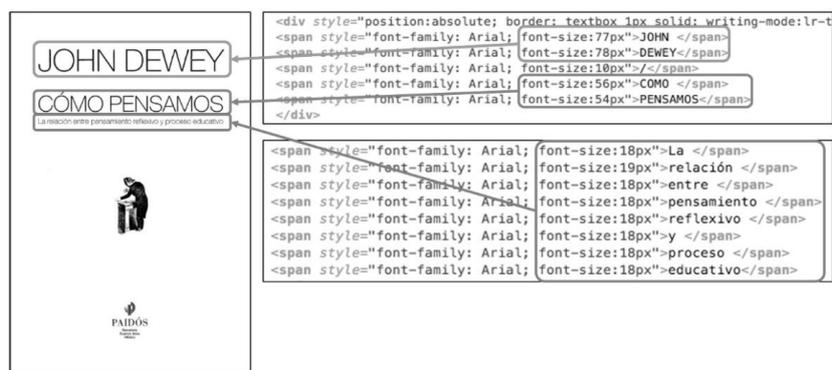


Fig. 4. Ejemplo de distribución de los metadatos.

debido a que existen varios nombres propios con incidencias en el diccionario de la RAE como Abad, Abarca, Bravo, entre otros, se diseñó un algoritmo capaz de evaluar si una palabra determinada como las ya mencionadas, puede o no pertenecer a una entidad nombrada, que recibe como entrada el archivo XML que contiene el etiquetado PoS y la identificación de las entidades nombradas, seguido de un análisis contextual de la palabra, es decir, identifica si el token está precedido o seguido por un nombre propio, ya sea primer nombre o apellido.

Más adelante, se evalúa si la palabra puede o no pertenecer a una entidad nombrada y esta información se actualiza en el archivo XML, agregando: *tag = "NP00SP0"*, *ctag = "NP"*, *pos = "noun"*, *type = "proper"*, *neclass = "Person"*, *nec = "PER"* a la etiqueta PoS en caso de que no la contenga y la elimine en caso de que haya una entidad nombrada de tipo de nombre mal identificada.

Una vez que se realiza la actualización del archivo XML, se obtiene la información de todas las etiquetas PoS que contienen solo la etiqueta "NP00SP0", descartando todas aquellas con un tamaño de fuente inferior a 15 píxeles, en caso de que la heurística no pueda extraer la información suficiente, El valor de tamaño de fuente mínimo se modifica automáticamente para continuar la extracción.

La información obtenida se ordena de dos maneras: por tamaño de fuente y por la ubicación espacial del texto en el documento, para identificar a los autores

principales en primer lugar, que deberían tener un tipo de letra más grande que el resto y, posteriormente, a todos los autores. De menor impacto para concluir con la inserción de estos en la base de datos.

Detección de editorial. Al igual que en la detección de autores, se implementó un algoritmo para la detección de editores, sin embargo, a diferencia del anterior, este utiliza una serie de palabras clave como: editorial, impresión, editores, ediciones, S.L., entre otras.

Todas las palabras clave están sujetas a un proceso automático que genera tres formas diferentes de la palabra clave original, es decir, la palabra clave capitalizada (Palabra Clave), minúscula (palabra clave) y finalmente se genera el uso de mayúsculas (PALABRA CLAVE). Además, la palabra clave se procesa en caso de que contenga signos de acentuación como se mencionó anteriormente.

También fue necesario analizar el contexto de la palabra clave que se busca en el documento, es decir, a modo de ejemplo; Si el sistema detecta la palabra editorial y está precedido por signos de puntuación como: ". (Punto) / , (coma) implica en la mayoría de los casos que el nombre del editor está antes del signo de puntuación y antes de la palabra clave identificada, en el caso de identificar "": (dos puntos) implica que el nombre del editor va seguido del signo de puntuación con la opción de continuar en la misma línea o en la siguiente. Finalmente, el último caso ocurre en ausencia de signos de puntuación, lo que implica que en la mayoría de los casos el nombre del editor está en la misma línea.

Posteriormente, en caso de encontrar varias palabras clave, la información obtenida se ordena de dos maneras: por el tamaño de la fuente y por la ubicación espacial del texto en el documento y de esta manera determina qué información corresponde al editor.

Detección de fecha de publicación. Para detectar la fecha de publicación, surge el siguiente problema porque un archivo puede contener una o más fechas a lo largo de las páginas legales del documento y estas pueden ser o no fechas de publicación, es decir, se pueden presentar otras fechas, tales como: Fechas de ediciones anteriores, fechas de traducción, fechas de revisión, entre otras.

Por lo tanto, se aplicó el mismo enfoque para determinar la fecha de publicación, descartando todos los números menores a 1700 y todas las fechas con fuente inferior a 15 píxeles, en caso de que no haya suficiente información, se modifica automáticamente el tamaño mínimo para continuar con la extracción.

En caso de que el sistema identifique más de una fecha, ordenará todos los resultados obtenidos de mayor a menor, priorizando esa fecha que tenga el tamaño de fuente más grande. Si todas las fechas identificadas tienen un tamaño de fuente común, el sistema evaluará el número de veces que se repite una cantidad, eligiendo la que tenga más repeticiones como fecha de publicación.

Finalmente, si no se cumple ninguno de los casos mencionados, el sistema elegirá la fecha más grande identificada y esta se asignará como la fecha de publicación.

Detección de título. Esta tarea es primordial en la extracción y almacenamiento de los metadatos, ya que esta se encargará de realizar las relaciones respectivas entre los autores y los editores con los libros correspondientes.

Al igual que con los metadatos anteriores, se propuso resolver este problema mediante la obtención de las propiedades tipográficas y espaciales del texto en la página, basándose en el análisis de diferentes colecciones digitales, que, en su mayoría, muestran una tipografía diferente del resto, además de poseer la tipografía más grande. Sin embargo, esto no descarta la posibilidad de que algunos autores manifiesten estas características.

Para mitigar estos casos, se realizó un análisis contextual del fragmento de texto, se analizó qué información se encontraba en las líneas anteriores y en las siguientes, es decir, si el nombre propio se puede conectar al resto del texto mediante preposiciones o artículos formando parte del título como: Biografía de Frida Kahlo, Siendo, se es la tesis de Parménides, De Aristóteles a Newton, entre otros. De esta manera, se evita que el sistema tenga en cuenta a esa entidad nombrada como un posible autor.

Posteriormente, la información extraída anteriormente se filtra de tal manera que solo se conservan las líneas del documento con un tamaño de fuente superior a 15 y, en caso de que no haya suficiente información antes de realizar dicha depuración, el sistema corregirá automáticamente el tamaño mínimo. Fuente para continuar la extracción.

Una vez hecho esto, la información se organiza con la tipografía de mayor a menor, y luego se crea la alineación correspondiente a las propiedades espaciales del texto, generando así una cadena de texto legible correspondiente al título del documento.

Finalmente, al igual que con los metadatos anteriores, se realizaron consultas SQL básicas para los motores de base de datos InnoDB y, en el caso de las bases de datos que usan el motor MyISAM, se utiliza la búsqueda FULLTEXT para obtener la puntuación de relevancia y hacer comparaciones basadas en los valores obtenidos.

Este proceso se usa para evitar registros duplicados para cada metadato debido a que algunos editores usan logotipos o acrónimos para referirse a sus nombres, por lo tanto, esto implica que la detección de estos metadatos no se realizará en estos casos debido a la ausencia de palabras clave.

Es necesario mencionar que, si el archivo no tiene permisos de lectura y escritura, no se ha hecho OCR en el documento o está protegido por contraseña, el sistema no podrá generar los archivos necesarios, si es el caso, el sistema realizará la inserción de la ubicación de vista previa del archivo y el tamaño del archivo solamente.

5. Resultados experimentales

Esta sección presenta los experimentos realizados para evaluar la metodología propuesta en este trabajo. Para realizar la evaluación, se seleccionaron 300 documentos no estructurados en español en formato PDF, es necesario mencionar

que este corpus fue construido manualmente obtenidos de internet, estos fueron analizados manualmente para identificar el número total de autores, editores y fechas de publicación de cada libro.

En total, se obtuvieron 300 títulos, 549 autores, 265 editoriales y 282 fechas de publicación. Estos valores se compararon con los datos generados por el sistema para determinar la precisión y cobertura correspondientes utilizando las ecuaciones 1 y 2:

$$\text{Precisión} = \frac{tp}{(tp + fp)}, \tag{1}$$

$$\text{Cobertura} = \frac{tp}{(tp + fn)}, \tag{2}$$

donde *tp*: representa los valores llamados verdaderos positivos (*true positive*), es decir, los valores que fueron identificados por el sistema como los metadatos correspondientes y, de hecho, lo son. *fp*: representa los valores denominados falsos positivos (*false positive*), es decir, los valores identificados por el sistema como metadatos específicos, pero no son los correctos. Finalmente, *fn*: representa los valores denominados falsos negativos (*false negative*), es decir, aquellos que el sistema **no** identificó como los metadatos correspondientes, pero deben considerarse correctos. La Tabla 1 describe los verdaderos positivos, falsos positivos, falsos negativos, precisión y cobertura para cada metadato.

Tabla 1. Resultados de Precisión y Cobertura para cada metadato.

	Título	Autor	Editorial	Fecha de publicación
True positive	224	406	239	239
False positive	76	81	61	61
False Negative	76	62	61	61
ve				
Precisión	74.66 %	83.36 %	79.66 %	82.33 %
Cobertura	74.66 %	86.76 %	79.66 %	82.33 %

A partir de los resultados obtenidos, se determinó que la razón principal de algunos títulos que no se identificaron correctamente se debe a la problemática en la distribución de la información y la variación en los textos, lo que genera textos incompletos. Además, en algunos casos, la información tipográfica de los documentos era la misma, independientemente del nivel visual que fuera diferente, por lo tanto, las heurísticas basadas en estas propiedades no podían realizar la extracción correctamente. Por otro lado, el sistema no pudo identificar a todos los autores debido a los diferentes orígenes de los que pueden provenir los nombres propios. Por lo tanto, ya que estos no se encuentran en los repositorios

y el freeling no ha podido identificarlos, produce la ausencia del etiquetado de la entidad nombrada.

Con respecto a los metadatos editoriales, los errores se produjeron en algunos casos por la ausencia de palabras clave utilizadas para determinar el editor. Además, algunos libros utilizan el logotipo del editor, por lo que están ausentes las palabras clave.

Finalmente, al igual que con los metadatos anteriores, si la información tipográfica en algunos documentos era la misma sin importar si eran visualmente diferentes y había más de una incidencia con las palabras clave, había errores en la extracción de la información.

Es necesario mencionar que los resultados de la extracción de fechas de publicación se vieron afectados principalmente cuando hay dos o más fechas con el mismo número de incidentes entre ellos y cuando el documento tiene demasiadas fechas, que a veces representan en número de reimpresiones o Ediciones de cualquier tema.

Adicionalmente, se realizó la comparación de los valores de precisión y cobertura obtenidos para este trabajo y aquellos mencionados anteriormente, estos datos se muestran en la Tabla 2. Es necesario mencionar que estos trabajos se enfocan en distintos metadatos en distintos idiomas y en distintos tipos de documentos. Si bien los resultados obtenidos en este trabajo no obtuvieron los mejores resultados, estos destacan por el hecho de enfocarse en documentos no estructurados y en idioma español, los cuales no podría ser extraídos con los sistemas enfocados a documentos estructurados.

Tabla 2. Tabla comparativa.

	Título		Autor		Editorial		Fecha de publicación	
	PREC	REC	PREC	REC	PREC	REC	PREC	REC
Gao et. al [5]	98.1 %	36.4 %	93.9 %	34.2 %	-	-	-	-
Guo et. al [6]	89 %	-	84.5 %	-	-	-	-	-
Huynh et. al [7]	100 %	100 %	92.72 %	89.47 %	95.83 %	92 %	-	-
Tkaczyk et. al [12]	84.07 %	-	87.68 %	-	97.5 %	-	93.67 %	-
Nuestro trabajo	74.66 %	74.66 %	83.36 %	86.76 %	79.66 %	79.66 %	82.33 %	82.33 %

El objetivo de semiautomatizar el proceso de extracción de metadatos en documentos no estructurados se resolvió utilizando este enfoque. Sin embargo, los errores en la extracción y clasificación de la información continúan presentándose, por lo que, al ser un proceso semiautomático, el usuario final puede verificar la información y corregirla si es necesario mediante una interfaz web.

6. Conclusiones y trabajos futuros

Este trabajo presentó un enfoque para la extracción de metadatos utilizando técnicas de procesamiento de lenguaje natural e información tipográfica del texto en documentos no estructurados en español.

Es necesario mencionar que a lo largo de esta investigación no fue posible determinar un patrón en la organización de los metadatos en este tipo de documentos. Por lo tanto, esta metodología se diseñó bajo el enfoque del análisis de las propiedades tipográficas del texto en el documento para determinar los metadatos que utilizan este enfoque.

Además, se construyeron una serie de repositorios de nombres propios, ciudades, países y palabras clave editoriales, que se pueden usar en otras aplicaciones de procesamiento de lenguaje natural, más específicamente en el Reconocimiento y Clasificación de Entidades Nombradas.

Finalmente, la tarea de extraer metadatos se vuelve bastante compleja porque carece de un orden en la distribución de estos, por lo que esta investigación se centró solo en la extracción de los metadatos: título, autores, editorial y fecha de publicación. Sin embargo, en algunos casos, el OCR aplicado a cada documento generalmente genera errores en la interpretación de los caracteres, por lo tanto, obtiene cadenas de texto incoherentes que causan errores de extracción de información.

Referencias

1. Information Extraction - Universitat Pompeu Fabra. <https://www.upf.edu/hipertextnet/numero-5/pln.html>, accessed: 2019-03-4
2. Metadata - Universidad Nacional de Colombia Sede Amazonia. <http://www.unal.edu.co/siamac/sig/metadatos1.html>, accessed: 2019-03-4
3. PDFMiner. <https://media.readthedocs.org/pdf/pdfminerdocs/latest/pdfminerdocs.pdf>, accessed: 2019-03-4
4. Carreras, X., Màrquez, L., Padró, L.: Named entity recognition for Catalan using Spanish resources. In: Proceedings of the Tenth Conference on European Chapter of the Association for Computational Linguistics - Volume 1. pp. 43–50. EACL '03, Association for Computational Linguistics, Stroudsburg, PA, USA (2003), <https://doi.org/10.3115/1067807.1067815>
5. Gao, L., Zhong, Y., Tang, Y., Tang, Z., Lin, X., Hu, X.: Metadata extraction system for Chinese books. In: 2011 International Conference on Document Analysis and Recognition. pp. 749–753 (Sep 2011)
6. Guo, Z., Jin, H.: A rule-based framework of metadata extraction from scientific papers. In: 2011 10th International Symposium on Distributed Computing and Applications to Business, Engineering and Science. pp. 400–404 (Oct 2011)
7. Huynh, T., Hoang, K.: Gate framework based metadata extraction from scientific papers. In: 2010 International Conference on Education and Management Technology. pp. 188–191 (Nov 2010)
8. Lapuente, M.J.L.: Hipertextos. <http://www.hipertexto.info/documentos/metadatos.htm>, accessed: 2019-03-4
9. Morales-Solares, C., Sierra, G., Escalante-Ramirez, B.: An unsupervised approach for automatic discovery of metadata in document images. In: 2016 Fifteenth Mexican International Conference on Artificial Intelligence (MICAI). pp. 1–7 (Oct 2016)
10. Padró, L., Stanilovsky, E.: Freeling 3.0: Towards wider multilinguality. In: Proceedings of the Language Resources and Evaluation Conference (LREC 2012). ELRA, Istanbul, Turkey (May 2012)

11. Rendón, J.C.: Clasificación automática de objetos de conocimiento con contenido no estructurado para el poblado semiautomático de ontologías multidimensionales. Master's thesis, Centro Nacional de Investigación y Desarrollo Tecnológico - Sin publicar (2014)
12. Tkaczyk, D., Bolikowski, L., Czeczko, A., Rusek, K.: A modular metadata extraction system for born-digital articles. In: 2012 10th IAPR International Workshop on Document Analysis Systems. pp. 11–16 (March 2012)

Detección de comandos de voz con modelos compactos de aprendizaje profundo

Edoardo Bucheli-Susarrey¹, Miguel González-Mendoza¹,
Oscar Herrera-Alcántara²

¹ Tecnológico de Monterrey, Escuela de Ingeniería y Ciencias, Monterrey, N.L.,
México

² Universidad Autónoma Metropolitana Azcapotzalco, Ciudad de México, México
edoardo.bucheli@tec.mx, mgonza@tec.mx, oha@azc.uam.mx

Resumen. El problema de Detección de Comandos consiste en localizar palabras de un vocabulario corto en un segmento de audio. La Detección de Comandos corre constantemente en el fondo de dispositivos móviles por lo que resulta necesario crear modelos económicos en cuanto a memoria y cómputo. Utilizando el conjunto de datos *Simple Speech Commands Detection* presentamos un estudio comparativo con dos tipos de capas. Las capas diseñadas consisten en métodos de extracción de representaciones de audio basadas en la Transformada de Fourier y los Bancos de Filtros Mel. Las capas diseñadas pertenecen al área de Aprendizaje Profundo e incluyen capas de perceptrones, capas recurrentes y capas convolucionales. Mediante el Canal de Aprendizaje profundo, organizamos estas capas para resolver el problema.

Palabras clave: detección de comandos, reconocimiento de voz, aprendizaje profundo, simple commands speech detection, preprocesamiento de audio.

Voice Command Detection with Compact Deep Learning Models

Abstract. The Keyword Detection problem consists in localizing a small vocabulary of words embedded in some stream of audio. Keyword Detection constantly runs in the background of many mobile devices and thus it becomes a requirement to create models with a small memory footprint and low computational power. Using the Simple Speech Commands Detection data set, we present a comparative study using two types of layers. Hand-Engineered layers are created from audio feature extraction models based on the Fourier Transform and Mel Filterbanks. Learned layers belong to the Deep Learning literature and include dense layers, recurrent layers and convolutional layers. Using the Deep Learning Pipeline, we organize these layers to solve the problem.

Keywords: keyword spotting, speech recognition, deep learning, simple commands speech detection, audio preprocessing.

1. Introducción

Con la omnipresencia de dispositivos móviles y pantallas en la vida diaria, surgen también riesgos relacionados con visión como pérdida de vista, saturación visual y distracción. Las interfaces de audio son una buena manera de reducir el uso de pantallas. Dichas interfaces contienen una variedad de módulos importantes incluyendo uno de Reconocimiento de Voz y un procedimiento de activación llevado a cabo por un módulo de Detección de Comandos.

El problema de Reconocimiento de Voz Automático (RVA) consiste en mapear una señal acústica que represente habla con la secuencia de palabras correspondiente [4]. La Detección de Comandos consiste en localizar palabras de un vocabulario corto en un segmento de audio [26].

Un módulo de RVA podría resolver el problema de Detección de Comandos sin problema. Sin embargo, como se menciona en [29,34], la Detección de Comandos se ejecuta usualmente en dispositivos móviles y surge la necesidad de crear modelos compactos en cuanto a memoria y procesamiento.

TensorFlow ha publicado el *Simple Speech Commands Dataset* [34] para auxiliar en el desarrollo de sistemas de Detección de Comandos. Además del conjunto de datos, la competencia *TensorFlow Speech Recognition Challenge* (2018) [17] fue publicada en Kaggle para alentar a gente de todo el mundo a proponer modelos de Aprendizaje Profundo que puedan resolver este problema.

La estructura de este documento es la siguiente. En la Sección 2 presentamos formalmente el problema de Detección de Comandos y explicamos nuestro acercamiento al desarrollo de arquitecturas. La Sección 3 presenta trabajo relacionado en los campos de Reconocimiento de Voz y Detección de Comandos. La Sección 4 describe nuestro acercamiento a la solución del problema. En la Sección 5, se hace una descripción del conjunto de datos *Simple Speech Commands Detection*. La Sección 6 detalla las arquitecturas seleccionadas para resolver el problema. La Sección 7 presenta nuestra metodología con los resultados presentados en la Sección 8 y, finalmente, discutimos conclusiones y trabajo posterior en la Sección 9.

2. El problema de detección de comandos

En Detección de Comandos, no se busca encontrar la secuencia más probable como sería el caso en RVA, sino que se desea localizar la ocurrencia de algún comando de voz perteneciente a un vocabulario pequeño en una señal de audio [26]. Sea $\mathbf{X} = (x^{(1)}, x^{(2)}, \dots, x^{(T)})$ la secuencia de entrada, $\mathbf{V} = (v_1, v_2, \dots, v_N)$ el vocabulario de comandos a detectar y $\mathbf{y} = (y_1, y_2, \dots, y_N)$ una secuencia de salida que indica cada vez que un comando de \mathbf{V} ha sido pronunciado. En Detección de Comandos, la meta es crear una función f_{DC} que para cada instante de tiempo en la secuencia \mathbf{y} haga una predicción sobre la probabilidad que cada palabra de \mathbf{V} haya sido pronunciada.

Como método de evaluación utilizaremos la métrica de Error *Top-One*, descrita para este problema en [34] que consiste en generar una predicción para un

segmento corto de audio de la probabilidad del habla de cada comando de \mathbf{V} , otros comandos y silencio. Ésta fue la métrica utilizada en [17] y consecuentemente en la literatura [31,2,11,25].

3. Trabajo relacionado

A partir de los años ochenta y hasta los años 2010-2012 el estado del arte en RVA consistía principalmente en Modelos Ocultos de Markov (MOM) combinados con Modelos de Mezcla Gausiana (MMG). [4]. El campo eventualmente transitó hacia Redes Neuronales a partir de [21,22] donde se utilizó una Máquina de Boltzmann Restringida (MBR) como sustituto al MMG. Otros avances durante estos años [14] incluyeron redes con activaciones ReLU [9] y Dropout [32][35,7], Redes Neuronales Convolucionales (CNN) [28] y Redes Neuronales Recurrentes (RNN) [12].

El problema de Detección de Comandos fue descrito por primera vez en [26]. Al igual que en RVA, los modelos clásicos utilizaron MOM pero el campo transitó hacia el aprendizaje profundo manteniendo énfasis en los modelos compactos. En [29] se proponen varios modelos de CNN. Otros enfoques incluyen [3] que utiliza una Red Neuronal Recurrente Convolutional (CRNN) y [2] que propone un CRNN con unidades Bidireccionales-LSTM y un módulo de atención.

Como resultado del *TensorFlow Speech Recognition Challenge* se publicaron varias propuestas; entre ellas [31,11,25] que utilizan modelos CNN basados en redes propuestas para visión por computadora como VGG [30] y Resnet [13] o las redes compactas de [29]. El primer lugar de la competencia obtuvo un puntaje en exactitud de 91.06 % sin embargo no existen anotaciones disponibles de clase para el conjunto de prueba utilizado en Kaggle por lo que los resultados obtenidos en esta investigación no son directamente comparables.

4. Hacia un sistema *End-to-End* de detección de comandos

Dado el análisis del estado del arte, algunas de las preguntas que nos hemos planteado para esta investigación son: (1) ¿Podemos crear un verdadero modelo *End-to-End* utilizando solo capas entrenadas? (2) Para cada tipo de representación, ¿cuál es el *trade-off* entre resolución, exactitud y eficiencia? (3) ¿Qué capas diseñadas son más adecuadas para esta tarea considerando la exactitud y la eficiencia? (4) ¿Qué combinación de capas da el mejor rendimiento en nuestras métricas de evaluación? (5) ¿Cómo manejan los modelos la precisión y la exhaustividad en sus diferentes clases? Para contestar estas preguntas, en esta sección detallamos el modelo propuesto.

4.1. El canal de aprendizaje profundo

Sistemas clasificadores para datos de alta dimensión como el audio, pueden modelarse como un canal muy general (Figura 1) que incluye una etapa de

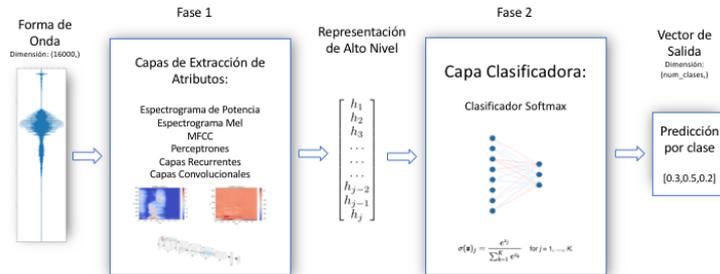


Fig. 1. Nuestro canal de aprendizaje profundo para la clasificación de audio.

extracción de representación (Fase 1) seguida de una etapa de clasificación (Fase 2) que dada la representación obtenida hace una predicción de clase. La Fase 1 obtiene como entrada una forma de onda de audio y sus capas transforman los datos en algo que es más fácil de separar linealmente. Un modelo para la Fase 1 puede ser compuesto de una combinación de capas entrenadas o diseñadas, aunque las capas diseñadas aparecerán invariablemente antes de las capas entrenadas.

Esta sección presenta una visión general de tales capas. Las capas diseñadas incluyen la Transformada de Fourier de Tiempo Reducido (STFT) [23], un Banco de Filtros Mel [8] y la Transformada Discreta de Cosenos (DCT) [8,24]. Las capas entrenadas incluyen las Capas de Perceptrones [27], las Capas Recurrentes [16,15,5] y las Capas Convolucionales [10].

4.2. Las capas diseñadas: Extracción de atributos de audio

Una forma de onda es un arreglo unidimensional de valores de punto flotante dentro del rango $[-1, 1]$. Este arreglo representa una señal en el tiempo $x[n]$ donde n es una marca de tiempo y $x[n]$ es el nivel de amplitud de la onda en el tiempo n . A cada uno de los valores del arreglo se le conoce como muestra. La frecuencia de muestreo f_s representa cuántas muestras deben tomarse en un segundo.

Las formas de onda proporcionan toda la información necesaria para reproducir una señal. Sin embargo, la cantidad de información que podemos extraer fácilmente de ellas es limitada. Esta es la razón por la que es muy común aplicar preprocesamiento a las señales de audio en sistemas de reconocimiento.

Los atributos más comúnmente extraídos de las formas de onda son las representaciones frecuencia-tiempo. Estos incluyen el Espectrograma de Potencia (EP), el Espectrograma Mel (EM) y los Coeficientes Cepstrales de Frecuencia

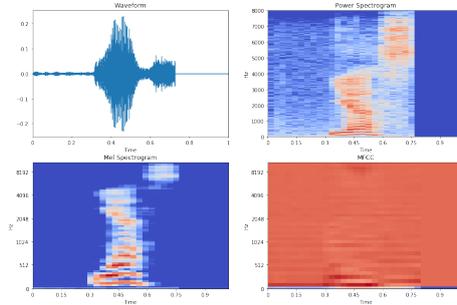


Fig. 2. Cuatro Representaciones de Audio; Forma de Onda (superior izquierda), Espectrograma de Potencia (superior derecha), Espectrograma Mel (inferior izquierda), MFCC (inferior derecha)

Mel (MFCC). En la Figura 2, se presenta una comparación de todas las representaciones mencionadas hasta ahora.

Cuando aplicamos la capa de STFT se obtiene una matriz de valores complejos donde cada columna representa una ventana en el tiempo con valores para cada componente de frecuencia. Tomando la magnitud de cada valor complejo y aplicando una escala logarítmica se obtiene el EP. Al aplicar un Banco de Filtros Mel se obtiene la potencia del EP a través de bandas de frecuencia que están linealmente espaciadas y tienen el mismo ancho de banda en la escala Mel [33], también conocido como el EM. Finalmente, podemos aplicar una DCT al EM para de-correlacionar los vectores obteniendo así el MFCC. En [14] se argumenta que los MFCC eran necesarios cuando se utilizaban MMG, pero para el Aprendizaje Profundo esto ya no es una necesidad.

4.3. Las capas entrenadas: Redes neuronales artificiales

Una red neuronal típica se puede pensar como un extractor de atributos con un clasificador lineal adjunto al final. Para clasificar correctamente, la red debe primero aprender cómo crear alguna representación que sea linealmente separable. Las redes crean representaciones ocultas o codificaciones $\mathbf{h}^{(l)}$, donde l es la profundidad de la capa que generó $\mathbf{h}^{(l)}$. A medida que crezca l , $\mathbf{h}^{(l)}$ debería ser más sencilla de separar.

Las capas de Redes Neuronales Artificiales a utilizar son: (1) Capas de Perceptrones [27] (2) Capas Recurrentes [16] que incluyen LSTM [15] y GRU [5] y (3) Capas Convolucionales [10].

5. El conjunto de datos: Simple Speech Commands Detection

El conjunto de datos *Simple Speech Commands Detection* [34] fue publicado por TensorFlow y está compuesto de varios miles de ejemplos por miles de

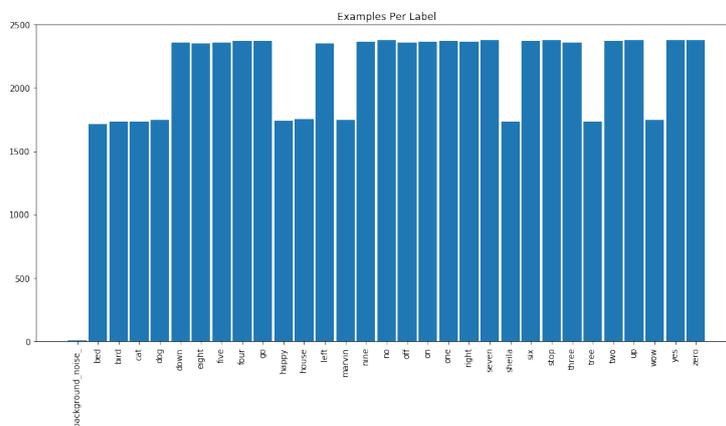


Fig. 3. Distribución del conjunto de datos Simple Commands Speech Detection v0.01. El eje horizontal representa todos los comandos presentes y el eje vertical representa el número de ejemplos para el comando correspondiente.

diferentes oradores de más de 30 palabras y algunas señales de ruido de fondo. El objetivo es crear un modelo que pueda predecir un subconjunto de esta lista de palabras clasificando cualquier otro comando como desconocido o la falta de un comando como silencio.

Existen dos versiones de este conjunto de datos. La primera (v0.01) contiene 64,727 archivos de audio que representan 30 comandos y 6 archivos de ruido de fondo, ésta fue la versión que se utilizó para el *TensorFlow Speech Recognition Challenge*. El conjunto de datos fue eventualmente actualizado a la versión 0.02 que contiene 105,829 archivos de audio que representan 35 comandos y 6 archivos de ruido de fondo. La figura 3 muestra la distribución de archivos para la versión 0.01 del conjunto de datos.

5.1. Subproblemas

Hemos identificado dos versiones del problema. La primera se utilizará para medir el rendimiento general y la segunda para analizar de manera más cercana las métricas de Precisión y Exhaustividad así como la capacidad de los modelos para identificar un comando como *desconocido*.

- Clasificación de 10 palabras (12 clases): 'yes', 'no', 'up', 'down', 'left', 'right', 'on', 'off', 'stop' y 'go'. Cualquier otra palabra deberá ser clasificada como 'desconocido' y la ausencia de palabras deberá ser clasificada como 'silencio'. Estos modelos serán entrenados con la versión 0.01 del conjunto de datos.
- Clasificación "Left/Right" (3 clases): 'left', 'right' y 'unknown'. Estos modelos serán entrenados con la versión 0.02 del conjunto de datos.

5.2. Datos de entrada

Los archivos de entrada se presentan en formato WAV con una frecuencia de muestreo de 16,000 Hz y una profundidad de palabra de 16 bits. Los clips duran aproximadamente un segundo pero pueden variar ligeramente, en estos casos se agregan ceros al final de la cadena o se corta para que cada clip tenga una longitud exacta de 16,000 muestras.

5.3. Preprocesamiento de datos

Como parte de la Fase 1 del canal de aprendizaje, queremos comparar cuatro tipos de representaciones de audio. Formas de onda, EP, EM y MFCC obtenidos mediante el uso de las capas diseñadas correspondientes.

Para los EP, EM y MFCC podemos obtener diferentes resoluciones ya sea en la frecuencia o en el dominio del tiempo. Para nuestros experimentos hemos fijado la resolución de tiempo a 32 pasos de tiempo, lo que significa que cada paso de tiempo representa exactamente 31.25 milisegundos de audio.

6. Algoritmos de aprendizaje

Basados en la literatura y los resultados de la competencia de Kaggle, hemos hecho una selección de arquitecturas en cuatro categorías; (1) Perceptrones Multi-Capa, (2) CNN para una dimensión, (3) CNN para dos dimensiones y (4) CRNN.

Dada la necesidad de utilizar modelos compactos, como pauta utilizaremos la restricción colocada en Kaggle que un modelo no debe sobrepasar los 5 MB. Trataremos esta restricción como una restricción suave sujeta al rendimiento de los modelos en las métricas de evaluación. No obstante, no se incluirán las arquitecturas que superen en gran medida esta limitación como es el caso de VGG y Resnet.

Todas las arquitecturas utilizan activaciones ReLU y *Dropout* para evitar sobre-entrenamiento a menos que se especifique de otra manera. En la salida todas las arquitecturas tienen una capa *Softmax* de 12 o 3 unidades (una por cada clase) dependiendo de la versión del problema.

6.1. Perceptrón Multi-Capa

De manera similar a [29], como modelo base entrenaremos un Perceptrón Multi-Capa (PCM) con 3 Capas Ocultas, 128 Unidades Ocultas. Esta red podría aplicarse a formas de onda, pero el tamaño de la entrada hace muy difícil obtener un resultado que sea si quiera cercano a aceptable.

6.2. Redes convolucionales de una dimensión

Hemos elegido el modelo detallado en [19] para formas de onda que se presentó para anotación general de audio al cual nos referimos como *Conv1D*. Ese modelo se compone de cuatro bloques cada uno con dos capas convolucionales (filtros de tamaño [9,3,3,3] respectivamente), *Dropout* y *MaxPooling* (filtros de tamaño [16,4,4,Global]). Seguido de dos capas de Perceptrones con 64 y 128 unidades respectivamente.

6.3. Redes convolucionales de dos dimensiones

Una gran parte de la literatura está centrada alrededor de modelos CNN de dos dimensiones utilizando las representaciones de EP, EM ó MFCC. Por eso la mayoría de nuestros modelos encajan en esta categoría.

Para los modelos CNN de dos dimensiones, nos inspiramos en [29] que explora el uso de CNN compactas. De las arquitecturas presentadas en este trabajo, utilizaremos *cnn-one-fstride4* y *cnn-trad-fpool3*.

Por último, existe una red propuesta en el *Tensor Flow Speech Recognition Challenge* por Thomas O'Malley y que es detallada en [25] y nos referimos a ella como *O'Malley*. La idea principal de esta red es aplicar capas de convolución que afectan a la dimensión de frecuencia pero no a la dimensión temporal. Cuenta con cuatro capas convolucionales y una capa oculta de perceptrones.

6.4. Redes convolucionales recurrentes

Utilizamos tres redes CRNN para EP, EM y MFCC y dos para formas de onda. Todas ellas utilizan capas convolucionales para transformar la entrada antes de ser alimentada a las unidades recurrentes. Las dos primeras se basan en [3] y nos referimos a ellas como CRNN1 y CRNN2. Ambas tienen dos capas convolucionales a la entrada, la primera con 32 filtros de tamaño 5×5 y la segunda con un filtro de tamaño 1×1 seguida de dos capas recurrentes de 32 unidades, la diferencia entre CRNN1 y CRNN2 radica en que CRNN1 utiliza unidades GRU y CRNN2 utiliza unidades LSTM bidireccionales. Utilizamos las inferencias de todas las unidades en el tiempo y alimentamos la salida a una capa de perceptrones con 64 unidades. Una variación de CRNN1 fue creada para manejar las formas de onda utilizando convoluciones de una dimensión en lugar de las convoluciones de dos dimensiones al principio.

El último modelo a utilizar fue tomado de [2] y nos referimos a él como *attRNN*. En él se propone un módulo de atención que indica a la red dónde se producen los datos más importantes. La arquitectura se detalla en el documento mencionado anteriormente. Al igual que con CRNN1, se ha propuesto una versión para forma de onda de esta arquitectura utilizando convoluciones de una dimensión.

7. Metodología

Tenemos un total de 10 arquitecturas de capas entrenadas y 9 representaciones de entrada obtenidas con las capas diseñadas. Sin embargo, no todas las arquitecturas y representaciones coinciden. De las diez arquitecturas, tres sólo funcionan con formas de onda y siete sólo trabajan con EP, EM o MFCC. Esta sección proporciona información sobre las combinaciones aplicadas, el método de evaluación y los pasos seguidos durante la implementación.

7.1. Representaciones de entrada

Usando 4 representaciones y sus variaciones obtenemos 9 representaciones de entrada, todas con una resolución en el tiempo de 32 a excepción de la forma de onda:

- Forma de onda muestreada a 16,000Hz.
- Espectrograma de potencia con resoluciones de frecuencia de 257 y 129.
- Espectrograma Mel con resoluciones de frecuencia de 120, 80 y 40.
- MFCCs con resoluciones de frecuencia de 120, 80 y 40.

7.2. Arquitecturas de redes neuronales

La Sección 6 ha detallado las arquitecturas a utilizar. La siguiente lista las presenta de manera reducida.

- Conv1D: descrita en la sección 6.2 sólo compatible con formas de onda.
- CRNN 1D: descrita en la sección 6.4 sólo compatible con formas de onda.
- attRNN 1D: descrita en la sección 6.4 sólo compatible con formas de onda.
- PMC: descrita en la sección 6.1 compatible con EP, EM y MFCC.
- trad_fpool3: descrita en la sección 6.3 compatible con EP, EM y MFCC.
- one_fstride4: descrita en la sección 6.3 compatible con EP, EM y MFCC.
- CRNN1 2D: descrita en la sección 6.4 compatible con EP, EM y MFCC.
- CRNN2 2D: descrita en la sección 6.4 compatible con EP, EM y MFCC.
- attRNN 2D: descrita en la sección 6.4 compatible con EP, EM y MFCC.
- O'Malley: descrita en la sección 6.3 compatible con EP, EM y MFCC.

7.3. Implementación

Tomando las combinaciones descritas anteriormente, tenemos un total de 59 redes. Cada red fue programada usando la API de Keras [6] junto con TensorFlow [1]. Las capas diseñadas que incluyen; la STFT, el Banco de Filtros Mel y la DCT fueron creadas usando el paquete Librosa [20].

Cada modelo fue entrenado por 30 épocas con una tasa de aprendizaje de $1e - 3$ y el algoritmo de optimización *Adam* [18]. La única excepción fueron los Perceptrones Multicapa que se entrenaron por 100 épocas cada uno. Las redes

fueron entrenadas en un sistema de escritorio usando una GPU GTX 970 con 3.5 GB de Memoria.

Para evaluar los modelos se utilizó validación cruzada de 10 pliegues. Dado que hay demasiados ejemplos para la clase "desconocido", muestreamos aleatoriamente el 20 % de los ejemplos y los mantuvimos consistentes a través de las 10 particiones. Para todas las separaciones, nos aseguramos de que ningún hablante apareciera a través de las particiones, asegurando así que los modelos sean independientes del hablante.

8. Resultados

Experimento 1: 10 Palabras Las Tablas 1 y 2 muestran los resultados de exactitud media, desviación estándar y tamaño del modelo para el problema "10 palabras". En la Tabla 1 se presentan los resultados de los modelos para formas de onda, hemos destacado en negrita el modelo con la máxima exactitud media. En la Tabla 2 se presentan los resultados de las representaciones del EP, el EM y el MFCC. Para cada arquitectura hemos puesto en negritas la representación con la máxima exactitud media.

Tabla 1. Resultados de modelos entrenados con formas de onda.

	Conv1D	CRNN1D	attRNN 1D
Exactitud WF-16k	93.08 %	91.61 %	92.83 %
Des. Est. WF-16k	1.59 %	1.51 %	1.64 %
Memoria WF-16k	3.2 MB	1.41MB	2.4 MB

En respuesta a la pregunta de investigación (1) dados los resultados de la Tabla 1 podemos ver claramente que se pueden lograr resultados aceptables sin ninguna capa diseñada, Conv1D logró una exactitud promedio de 93.08 % y un tamaño de memoria muy por debajo de la restricción de 5 MB. CRNN 1D y attRNN logran resultados similares y por lo tanto concluimos que los verdaderos sistemas de aprendizaje profundo *End-to-End* por lo menos para la Detección de Comandos son más que plausibles.

En cuanto a las preguntas (2) y (3) en la Tabla 2, podemos ver que el PMC, trad_fpool3 y on_fstride4 funcionan mejor con representaciones más simples y sin necesidad de los vectores de-correlacionados del MFCC. Es por eso que la representación EM-40 domina claramente en estos ejemplos. Sin embargo, la diferencia se vuelve menos clara con las redes CRNN1 2D, CRNN2 2D, attRNN 2D y O'Malley donde las representaciones más grandes obtienen una ligera ventaja sobre EM-40 y los resultados obtenidos con MFCCs son muy similares a los obtenidos con EM.

Teniendo en cuenta estos resultados, nuestra conclusión es que para sistemas compactos, es probablemente mejor utilizar EM-40, pero pequeñas mejoras podrían lograrse en ciertas redes utilizando representaciones más grandes como

Tabla 2. Resultados de modelos entrenados con EM, MFCC y EP.

	PMC	trad_fpool3	one_stride4	CRNN1 2D	CRNN2 2D	attRNN 2D	O'Malley
Exactitud EM-40	77.75 %	87.71 %	84.18 %	89.62 %	90.13 %	93.36 %	94.94 %
Des. Est. EM-40	3.44 %	3.72 %	3.60 %	2.45 %	2.55 %	2.48 %	2.07 %
Memoria EM-40	2.4 MB	3.0 MB	1.5 MB	556.1 kB	1.2 MB	2.3 MB	17.0 MB
Exactitud EM-80	74.93 %	85.94 %	82.44 %	89.39 %	90.15 %	92.55 %	95.19 %
Des. Est. EM-80	5.04 %	3.54 %	4.73 %	2.43 %	2.79 %	2.56 %	2.15 %
Memoria EM-80	4.4 MB	2.8 MB	2.4 MB	579.1 kB	1.3 MB	2.5 MB	18.9 MB
Exactitud EM-120	73.01 %	85.89 %	81.11 %	88.72 %	89.70 %	91.48 %	94.77 %
Des. Est. EM-120	5.32 %	3.54 %	5.58 %	3.15 %	2.94 %	2.57 %	2.17 %
Memoria EM-120	6.4 MB	3.1 MB	2.7 MB	602.2 kB	1.4 MB	2.8 MB	18.9 MB
Exactitud MFCC-40	73.58 %	81.90 %	76.32 %	71.75 %	68.74 %	90.00 %	93.82 %
Des. Est. MFCC-40	4.11 %	1.58 %	2.29 %	2.51 %	3.95 %	1.37 %	1.41 %
Memoria MFCC-40	2.4 MB	3.0 MB	1.5 MB	556.1 kB	1.2 MB	2.3 MB	17.0 MB
Exactitud MFCC-80	69.65 %	79.48 %	74.70 %	70.25 %	83.95 %	92.85 %	95.36 %
Des. Est. MFCC-80	6.84 %	1.53 %	3.33 %	2.88 %	3.58 %	2.50 %	1.85 %
Memoria MFCC-80	4.4 MB	2.8 MB	2.4 MB	579.1 kB	1.3 MB	2.5 MB	18.9 MB
Exactitud MFCC-120	64.51 %	83.46 %	77.66 %	85.41 %	84.72 %	92.47 %	94.84 %
Des. Est. MFCC-120	6.70 %	5.46 %	8.29 %	2.74 %	4.53 %	2.39 %	2.05 %
Memoria MFCC-120	6.4 MB	3.1 MB	2.7 MB	602.2 kB	1.4 MB	2.8 MB	18.9 MB
Exactitud EP-129	57.60 %	81.60 %	78.12 %	89.33 %	90.33 %	93.37 %	95.31 %
Des. Est. EP-129	3.41 %	3.40 %	5.12 %	1.61 %	2.15 %	2.14 %	2.05 %
Memoria EP-129	6.8 MB	3.2 MB	2.7 MB	607.9 kB	1.4 MB	2.8 MB	18.9 MB
Exactitud EP-257	50.05 %	82.43 %	77.19 %	89.56 %	90.31 %	92.83 %	95.34 %
Des. Est. EP-257	3.84 %	3.84 %	5.40 %	2.13 %	3.03 %	2.26 %	1.95 %
Memoria EP-257	13.1 MB	4.7 MB	3.5 MB	681.6 kB	1.6 MB	3.6 MB	18.9 MB

EM-80. EM-120 no parece dar ninguna ventaja adicional más allá de EM-80. Lo mismo parece ser válido para el MFCC. En cuanto a la comparación entre EM-120, MFCC-120 y EP-129 la respuesta parece depender del modelo, pero ninguna representación tiene una victoria clara en ese sentido.

En cuanto a la pregunta (4) el mejor resultado de exactitud promedio en general se obtuvo usando MFCC-80 y O'Malley, pero otras arquitecturas producen resultados muy similares como O'Malley y EM-80, O'Malley y EP-129 y O'Malley y EP-257 por lo que podrían ser intercambiables. Si aplicamos la restricción de 5 MB, el modelo con la mayor precisión media es attRNN 2D y EP-129 pero al igual que con O'Malley, este resultado es prácticamente idéntico al obtenido con attRNN y EM-40.

Experimento 2: Left/Right Las Tablas 3 y 4 presentan la precisión y exhaustividad medias y el tamaño en memoria para cada una de las tres clases en este problema. Debido a su incompatibilidad una vez más hemos separado ciertas arquitecturas y representaciones. También hemos destacado en ambas tablas la mejor precisión y exhaustividad para cada clase y representación.

En la Tabla 4 vemos en general resultados que parecen ser coherentes con los resultados del experimento anterior. O'Malley y attRNN 2D aún poseen las puntuaciones más altas en las métricas seleccionadas.

Si contamos el número de victorias y empates para cada representación en cada uno de los seis parámetros de la Tabla 4 (Precisión y Exhaustividad para las tres clases) EP-127 con 12 victorias y 1 empate y EM-40 con 11 victorias

Tabla 3. Precisión y Exhaustividad para cada clase del problema *Left/Right* con formas de onda.

	Conv1D	CRNN 1D	AttRNN 1D
Prec/Exh 'Left'	95.3 %/93.1 %	95.1 %/89.1 %	94.5 %/91.6 %
Prec/Exh 'Right'	94.7 %/ 93.5 %	95.1 %/90.1 %	95.5 %/92.6 %
Prec/Exh 'Desconocido'	97.8 %/98.5 %	96.7 %/98.7 %	97.4 %/ 98.6 %

Tabla 4. Precisión y Exhaustividad para cada clase del problema *Left/Right* con EM, MFCC y EP.

	PMC	trad_fpool3	one_fstride4	CRNN1 2D	CRNN2 2D	AttRNN 2D	O'Malley
	Prec/Exh	Prec/Exh	Prec/Exh	Prec/Exh	Prec/Exh	Prec/Exh	Prec/Exh
MS-40	'Left'	86.7 %/ 79.7 %	88.1 %/ 88.5 %	86.5 %/ 83.3 %	92.0 %/89.5 %	91.2 %/ 91.1 %	94.8 %/94.8 %
	'Right'	86.1 %/ 84.1 %	91.3 %/87.8 %	89.1 %/83.5 %	94.3 %/ 92.7 %	94.4 %/91.5 %	95.7 %/95.3 %
	'Desc'	94.4 %/96.3 %	96.2 %/96.8 %	94.8 %/96.6 %	97.1 %/97.9 %	97.2 %/97.8 %	97.5 %/ 98.6 %
MS-80	'Left'	86.5 %/77.0 %	89.4 %/85.0 %	84.8 %/81.3 %	91.0 %/88.9 %	92.3 %/89.3 %	92.3 %/91.4 %
	'Right'	84.9 %/80.3 %	89.8 %/87.5 %	87.0 %/82.4 %	94.3 %/90.3 %	94.0 %/91.1 %	95.8 %/91.8 %
	'Desc'	93.2 %/96.1 %	95.6 %/96.9 %	94.2 %/95.8 %	96.6 %/97.8 %	96.8 %/98.0 %	97.3 %/98.2 %
MS-120	'Left'	86.8 %/74.6 %	87.0 %/85.5 %	85.2 %/78.8 %	91.4 %/89.3 %	90.3 %/88.7 %	90.9 %/91.5 %
	'Right'	84.0 %/78.9 %	90.1 %/85.5 %	86.9 %/81.9 %	93.1 %/91.0 %	94.0 %/90.1 %	94.5 %/91.6 %
	'Desc'	92.6 %/96.2 %	95.4 %/96.7 %	93.8 %/96.1 %	96.8 %/97.6 %	96.6 %/97.7 %	97.4 %/97.8 %
MFCC-40	'Left'	71.3 %/77.2 %	89.3 %/86.4 %	85.3 %/78.6 %	90.8 %/84.4 %	88.2 %/85.6 %	94.3 %/93.3 %
	'Right'	74.2 %/77.7 %	89.6 %/ 90.0 %	86.5 %/82.3 %	93.8 %/86.8 %	92.3 %/89.3 %	95.7 %/94.3 %
	'Desc'	94.2 %/91.8 %	96.4 %/96.9 %	94.1 %/96.3 %	95.5 %/98.1 %	95.9 %/97.1 %	97.9 %/98.5 %
MFCC-80	'Left'	65.5 %/70.5 %	88.8 %/85.2 %	85.9 %/75.4 %	89.7 %/86.0 %	88.5 %/87.3 %	91.5 %/92.3 %
	'Right'	65.1 %/74.1 %	91.6 %/87.5 %	84.5 %/79.7 %	93.6 %/88.5 %	92.8 %/88.5 %	94.8 %/92.6 %
	'Desc'	93.2 %/89.4 %	95.6 %/97.1 %	93.0 %/96.1 %	95.9 %/97.6 %	96.2 %/97.3 %	97.5 %/97.7 %
MFCC-120	'Left'	63.2 %/67.0 %	89.1 %/81.9 %	86.7 %/72.4 %	90.8 %/85.5 %	90.1 %/85.6 %	93.3 %/90.5 %
	'Right'	61.3 %/64.3 %	89.1 %/87.2 %	85.2 %/79.4 %	93.3 %/88.8 %	91.4 %/89.2 %	94.3 %/92.7 %
	'Desc'	92.1 %/90.1 %	95.2 %/97.0 %	92.4 %/96.5 %	95.8 %/97.7 %	96.1 %/97.4 %	97.3 %/98.1 %
PS-127	'Left'	83.0 %/64.5 %	89.6 %/84.9 %	87.6 %/74.3 %	94.0 %/89.9 %	93.3 %/90.7 %	94.4 %/ 93.4 %
	'Right'	80.3 %/71.6 %	90.6 %/85.9 %	86.5 %/79.6 %	94.3 %/91.9 %	93.7 %/ 93.3 %	95.7 %/ 94.4 %
	'Desc'	90.5 %/96.2 %	95.5 %/ 97.4 %	92.6 %/96.7 %	97.0 %/98.2 %	97.4 %/98.0 %	98.0 %/98.4 %
PS-257	'Left'	70.9 %/58.8 %	88.4 %/83.3 %	83.9 %/78.2 %	93.6 %/ 90.4 %	92.8 %/89.7 %	94.4 %/92.6 %
	'Right'	76.7 %/60.8 %	88.7 %/84.2 %	89.1 %/75.7 %	95.0 %/91.7 %	94.9 %/90.0 %	96.5 %/93.4 %
	'Desc'	89.3 %/ 95.5 %	94.9 %/96.8 %	92.8 %/96.6 %	97.0 %/ 98.3 %	96.8 %/ 98.2 %	97.6 %/ 98.6 %

y 3 empates tienen los mejores resultados. PS-257 con 8 victorias y 3 empates vendría después.

En respuesta a la pregunta de investigación (5), podemos ver que en la mayoría de los casos, la precisión es mejor que la exhaustividad para los comandos *Left/Right*, pero para los comandos "*desconocido*" la exhaustividad es generalmente mejor que precisión. Esto sucede porque cada vez que los algoritmos predicen un comando *Left/Right* como positivo suelen ser correctos (de ahí la alta precisión), pero hay algunos comandos que deben ser clasificados como positivos que se clasifican como "*desconocido*" (esta es la razón por la que la precisión no es tan buena para "*desconocido*" y la exhaustividad no es tan buena para los comandos).

9. Conclusiones

Hemos presentado un análisis de varias arquitecturas y representaciones posibles para el Problema de Detección de Comandos con un énfasis en sistemas *End-to-End* compactos. También hemos presentado un Canal de Aprendizaje

Profundo enfocado en extraer mejores representaciones para el problema usando una mezcla de capas diseñadas y entrenadas.

Se ha observado también que ciertas CNN compactas pueden ser eficaces con las representaciones más pequeñas pero sufren al aumentar el tamaño de los datos. Por otro lado, los CRNN en general utilizan menos recursos de memoria y presentan mejoras al aumentar la resolución de frecuencia.

Asegurarse de que los hablantes no aparezcan en más de una partición del conjunto de datos y tener muchos equipos de grabación diferentes nos ayudó a hacer un modelo más robusto. Sin embargo, estos resultados se pueden mejorar si se terminan de entrenar con ejemplos específicos del usuario.

Nos hemos centrado en la Detección de Comandos para dispositivos móviles en este documento, pero muchos de estos principios pueden aplicarse a la anotación de audio en general y trabajo futuros podría hacer una exploración en esta línea.

Referencias

1. Abadi, M., et al.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015), <https://www.tensorflow.org/>, software available from tensorflow.org
2. de Andrade, D.C., Leo, S., Viana, M.L.D.S., Bernkopf, C.: A neural attention model for speech command recognition. In: undefined (2018), <http://arxiv.org/abs/1808.08929>
3. Arik, S.O., et al.: Convolutional recurrent neural networks for small-footprint keyword spotting. In: Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH. vol. 2017-Augus, pp. 1606–1610 (mar 2017)
4. Bengio, Y., Goodfellow, I., Courville, A.: Applications: Speech Recognition, pp. 446–448 (2016), <https://mitpress.mit.edu/books/deep-learning>
5. Cho, K., et al.: Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) pp. 1724–1734 (2014)
6. Chollet, F., et al.: Keras. <https://keras.io> (2015)
7. Dahl, G.E., Sainath, T.N., Hinton, G.E.: Improving deep neural networks for LVCSR using rectified linear units and dropout. In: ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings. pp. 8609–8613. IEEE (may 2013)
8. Davis, S.B., Mermelstein, P.: Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences (aug 1980)
9. Fred Agarap, A.M.: Deep Learning using Rectified Linear Units (ReLU). Tech. rep., <https://github.com/AFAgarap/relu-classifier>.
10. Fukushima, K.: Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics* 36(4), 193–202 (apr 1980)
11. Gouda, S.K., Kanetkar, S., Harrison, D., Warmuth, M.K.: Speech Recognition: Keyword Spotting Through Image Recognition (mar 2018), <http://arxiv.org/abs/1803.03759>
12. Graves, A., Mohamed, A.R., Hinton, G.: Speech recognition with deep recurrent neural networks. In: ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings. pp. 6645–6649. IEEE (may 2013)

13. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition pp. 770–778 (2015), <http://arxiv.org/abs/1512.03385>
14. Hinton, G., et al.: Deep Neural Networks for Acoustic Modeling in Speech Recognition. *IEEE Signal Processing Magazine* 29(November), 82–97 (nov 2012), <https://www.microsoft.com/en-us/research/publication/deep-neural-networks-for-acoustic-modeling-in-speech-recognition/>
15. Hochreiter, S., Schmidhuber, J.: Long Short-Term Memory. *Neural Computation* 9(8), 1735–1780 (nov 1997)
16. Hopfield, J.J.: Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America* 79(8), 2554–8 (apr 1982)
17. Kaggle: Tensorflow speech recognition challenge (2018), kaggle Challenge, <https://www.kaggle.com/c/tensorflow-speech-recognition-challenge/>
18. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization (dec 2014), <http://arxiv.org/abs/1412.6980>
19. Mahmood, Z.: Beginner's guide to audio data (2018), python notebook posted in Freesound General-Purpose Audio Tagging Challenge (2018) <https://www.kaggle.com/fizzbuzz/beginner-s-guide-to-audio-data>
20. McFee, B., Raffel, C., Liang, D., Ellis, D., McVicar, M., Battenberg, E., Nieto, O.: librosa: Audio and Music Signal Analysis in Python. In: *Proceedings of the 14th Python in Science Conference*. pp. 18–24 (2018)
21. Mohamed, A.R., Dahl, G., Hinton, G.: Deep belief networks for phone recognition. *Scholarpedia* 4(5), 5947 (2009)
22. Mohamed, A.R., Dahl, G.E., Hinton, G.: Acoustic modeling using deep belief networks. *IEEE Transactions on Audio, Speech and Language Processing* 20(1), 14–22 (jan 2012)
23. Müller, M.: *Fourier Analysis of Signals: Short-Time Fourier Transform*, pp. 93–105. Springer International Publishing, Cham (2015), <http://link.springer.com/10.1007/978-3-319-21945-5>
24. Narasimha, M.J., Peterson, A.M.: On the Computation of the Discrete Cosine Transform. *IEEE Transactions on Communications* 26(6), 934–936 (jun 1978)
25. O'Malley, T.: Second place solution (2018), discussion posted in Tensorflow Speech Recognition Challenge (2018) <https://www.kaggle.com/c/tensorflow-speech-recognition-challenge/discussion/47715>
26. Rohlicek, J., et al.: Continuous hidden Markov modeling for speaker-independent word spotting. In: *International Conference on Acoustics, Speech, and Signal Processing*. pp. 627–630. IEEE (1989)
27. Rumelhart, David E. Hinton, G.E., Williams, R.J.: Learning Internal Representations by Error Propagation. In: *Parallel Distributed Processing Explorations in the Microstructure of Cognition Volume 1 Foundations*, vol. 1, p. 567. MIT Press (1986), <https://ieeexplore.ieee.org/document/6302929>
28. Sainath, T.N., Mohamed, A.R., Kingsbury, B., Ramabhadran, B.: Deep convolutional neural networks for LVCSR. In: *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*. pp. 8614–8618. IEEE (may 2013)
29. Sainath, T.N., Parada, C.: Convolutional Neural Networks for Small-footprint Keyword Spotting. In: *Proceedings INTERSPEECH*. pp. 1478–1482 (2015), <https://storage.googleapis.com/pub-tools-public-publication-data/pdf/43969.pdf>
30. Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition (sep 2014), <http://arxiv.org/abs/1409.1556>

31. Solovyev, R.A., Vakhrushev, M., Radionov, A., Aliev, V., Shvets, A.A.: Deep Learning Approaches for Understanding Simple Speech Commands. Tech. rep. (oct 2018), <http://arxiv.org/abs/1810.02364>
32. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15, 1929–1958 (2014), <http://jmlr.org/papers/v15/srivastava14a.html>
33. Stevens, S.S., Volkman, J.: The Relation of Pitch to Frequency: A Revised Scale. *The American Journal of Psychology* 53(3), 329 (jul 1940)
34. Warden, P.: Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition (apr 2018), <http://arxiv.org/abs/1804.03209>
35. Zeiler, M.D., et al.: On rectified linear units for speech processing. In: ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings. pp. 3517–3521. IEEE (may 2013)

Detección del árbol de dzidzilché usando los descriptores de Halarick y redes neuronales artificiales

Juan Jose Negron-Granados

Universidad Autónoma de Yucatán, Facultad de Matemáticas, Mérida, Yucatán,
México
A08001373@alumnos.uady.mx

Resumen. La apicultura es una actividad pecuaria de gran importancia para la economía de México, siendo Yucatán uno de los principales estados productores, a pesar de ello, esta actividad ha sufrido grandes deterioros que han disminuido la productividad de miel en la entidad, lo cual es posible de detener si se aprovechan correctamente los recursos tecnológicos para el fortalecimiento de la producción de miel [6]. Una manera de mejorar el cultivo de miel es con la caracterización de los recursos poliníferos que se encuentran alrededor de una colonia de abejas, con dicha información se puede establecer centros de producción estratégica, disminuir la destrucción selvática o prevenir plagas y enfermedades. La caracterización de la cobertura terrestre suele realizarse con el análisis de información espectral en función de la luz reflejada por la superficie. En este trabajo se hace la propuesta de utilizar el dispositivo Sequoia Parrot como sensor espectral para la recolección de imágenes aéreas y el uso de los descriptores de Halarick para el entrenamiento de una red neuronal que permita detectar el árbol de dzidzilché, un árbol endémico de la Península de Yucatán de gran relevancia para la producción de productos poliníferos.

Palabras clave: redes neuronales, imágenes multiespectrales, descriptores de Halarick.

Detection of the Dzidzilché Tree Using the Halarick Descriptors and Artificial Neural Networks

Abstract. Beekeeping is a livestock activity of great importance for the economy of Mexico, being Yucatan one of the main states that produces more honey, in spite of this, this activity has suffered great deteriorations that have diminished the productivity of honey in the state, which is possible to stop if the technological resources for the strengthening of the production of honey are harnessed correctly [6]. One way to improve the cultivation of honey is with the characterization of polliniferous resources that are around a colony of bees, with this information can establish strategic production centers, reduction of jungle destruction or

prevention of pests and diseases. The characterization of the terrestrial cover is usually carried out with the analysis of spectral information as a function of the light reflected by the surface. In this paper, the proposal is made to use the Sequoia Parrot device as a spectral sensor for the collection of image areas and the use of Halarick descriptors for the training of a neural network that allows the detection of the dzidzilché tree, an endemic tree of the Yucatan Peninsula of great relevance for the production of polliferous products.

Keywords: neural networks, multispectral images, Halarick descriptors.

1. Introducción

La apicultura es una de las actividades agropecuarias de gran importancia económica y ambiental para México, pues los principales productores y exportadores de miel a nivel internacional son mexicanos [12]. La Península de Yucatán es una importante región productora de miel para México, ya que el 95 % de su producción se destina al mercado internacional [9]; sin embargo, la apicultura ha sufrido un grave deterioro en los últimos años ha presentado una disminución en la producción de miel que se ve reflejado en los porcentajes de exportaciones, los cuales se redujeron en un 40 % a nivel nacional y un 35 % en Yucatán. Uno de los principales factores de esta reciente declinación ha sido la disminución de los recursos néctar-polinívoros, lo que se ve reflejado en una baja producción melífera. Por lo anterior, existe la necesidad de mejorar los sistemas de comercialización y diversificación de la actividad así como actualizar las técnicas de producción y administración del proceso productivo por parte de los apicultores para obtener la cantidad de miel requerida por el mercado [9].

La caracterización del ambiente es la principal labor para comprender los procesos ecológicos y conservar la fauna [14], algunas de las aplicaciones de esta labor se han aplicado en la conservación de los humedales del norte de Argentina [18] y Chile [11], de igual forma es posible encontrar trabajos que apliquen la caracterización de plantas que sean de utilidad para los actividades pecuarias, como la ganadería [16] o para mejorar la producción de miel, por ejemplo, en Tabasco se utilizó este procedimiento para producción de mieles monoflorales y competir en el mercado internacional, pues dicho tipo de mieles son altamente apreciadas en Europa, a pesar de que Tabasco no destaca en producción de miel [4]. Por lo tanto, una manera de mejorar la producción y calidad de la miel es caracterizando la flora que se encuentra alrededor de una colmena, pues se espera que en 5km a la redonda de una colonia de abejas se encuentre abundante vegetación polinífera; la dieta de las abejas es muy específica y se espera que tengan al alcance algunas de las plantas que se encuentran en la Tabla 1.

Usualmente, la caracterización de coberturas terrestres se realiza utilizando imágenes satelitales, pues los sensores que llevan a bordo son una herramienta adecuada para la caracterización del suelo [2]; no obstante, el tamaño del píxel en

Tabla 1. Especies de plantas más importantes para la producción de miel en Yucatán.

Nombre común	Nombre científico
Ts'its'ilche'	Gymnopodium floribundum
Taj	Viguiera dentata
Jaabin	Piscidia piscipula
Box katsim	Acacia gaumeri
Sac katsim	Mimosa bahamensis
Tsalam	Lysiloma latisiliquum
X-tabentún	Turbina corymbosa
Kitimché	Caesalpinia gaumeri
Enredaderas	Varias especies de la familia Convolvulaceae

imágenes satelitales puede llegar medir varios metros en cada lado, por ejemplo, el Landsat TM tiene una resolución de 30x30 metros por píxel en sus imágenes [3]. Sin embargo, es posible mejorar la resolución del espacio físico por píxel usando una cámara colocada en una aeronave no tripulada para la captura de imágenes aéreas, adicionalmente, si se utiliza una cámara espectral es posible obtener información de la corteza terrestre en función de la reflectancia de las ondas electromagnéticas. Para realizar el análisis de texturas suelen usarse métodos estadísticos como la matriz de co-ocurrencia o los descriptores de Halarick, por ejemplo, usando estas técnicas en imágenes espectrales es posible analizar la biodiversidad en las superficies de : volcanes [7], bosques [1], zonas urbanas [21] e incluso ríos [17].

Este trabajo tiene el objetivo de identificar el árbol de dzidzilché, una especie propia de la península yucateca y de alto valor para las actividades apícola. Para lograr el objetivo se utiliza el sensor multiespectral Sequoia Parrot, el cual cuenta con lentes que permite capturar la información espectral de la superficie terrestre con una resolución espacial de 3,7 y 18,6 centímetros por píxel para vuelos de 30 y 150 metros de altura, respectivamente. Así como también el uso de los descriptores de Halarick para el análisis de la textura, siendo estos parámetros el vector de entrada para el modelo de red neuronal que se propone. En el siguiente capítulo se describe el área de estudio y los elementos utilizados para cumplir el objetivo propuesto, como la cámara Parrot, los descriptores de Halarick y la red neuronal utilizada, de igual manera, se menciona la metodología usada. En el capítulo 3 se realiza un análisis del modelo a partir de los resultados obtenidos y finalmente, en el capítulo 4 se realizan las conclusiones.

2. Materiales y métodos

2.1. Área de muestreo

Debido a la actividad apícola de en Yucatán es posible encontrar vegetación polinífera en gran parte de la superficie del estado, una de estas zonas es el Campus de Ciencias Biológicas y Agropecuarias (CCBA) de la Universidad

Autónoma de Yucatán (UADY), que cuenta con el entorno adecuado para realizar sus investigaciones apícolas lo que hace del CCBA el lugar ideal para realizar la toma de fotos que se utilizará en este trabajo (Figura 1).



Fig. 1. Ubicación del CCBA se puede notar que alrededor se conserva la flora nativa originaria del estado de Yucatán. (Fuente: maps.google.com)

2.2. Sensor multiespectral

Para la recolección de imágenes se ha utilizado el sensor espectral Sequoia Parrot (Figura 2), este dispositivo se define como espectral debido a que las imágenes están en función de diferentes frecuencias de luz, la información que se pueden obtener son las bandas: verde (530-570 nm), rojo (640-680 nm), borde rojo (730-740 nm) e infrarrojo cercano (770-810 nm); además de tomar fotos a color (imágenes RGB). El sensor se compone por dos módulos: el cuerpo (*body*) el cual tiene integrada los 4 lentes espectrales; y el sensor de luz, (*sunshine sensor*) que sirve para tener información más precisa de la reflectancia solar. Este sensor tiene mejor resolución espacial que las imágenes satelitales, además se evita influencias de gases en la atmósferas y nubes.

2.3. Corrección y alineación de las imágenes

Los sensores espectrales traen una distorsión de ojo de pescado y para corregirlo se debe obtener el punto proyectado del plano del mundo real en el plano de la imagen (ecuación 1), donde P y P' son puntos compuestos por (X, Y, Z) y (X', Y', Z') , respectivamente; R es la matriz de rotación con una dimensión de 3×3 y T el vector de traslación con dimensión 3, ambos están dados por el fabricante. A continuación se debe calcular el valor de un parámetro θ el cual se define en la ecuación 2, este valor se utiliza para calcular el coeficiente de distorsión ρ usando la ecuación 3; finalmente estos valores junto con la matriz afín (M) y el centro de la imagen es posible calcular la proyección del plano



Fig. 2. Cuerpo del sensor (Izquierda). Sensor de luz (Derecha).

del mundo real en el plano de la imagen (x_d, y_d) sin la distorsión de ojo de pescado (ecuación 4) [19]. Los componentes de la matriz M afín están definidos por C, D, E, F , valores que son dados por el fabricante al igual que el vector de desplazamiento (c_x, c_y) :

$$P' = R^T(P - T), \quad (1)$$

$$\theta = \frac{2}{\pi} \arctan\left(\frac{\sqrt{X'^2 + Y'^2}}{Z'}\right); \quad \theta \in [0, 1], \quad (2)$$

$$\rho = \theta + p_2\theta^2 + p_3\theta^3 + p_4\theta^4, \quad (3)$$

$$\begin{aligned} \begin{pmatrix} x_d \\ y_d \end{pmatrix} &= M \begin{pmatrix} \frac{\rho X'}{\sqrt{X'^2 + Y'^2}} \\ \frac{\rho Y'}{\sqrt{X'^2 + Y'^2}} \end{pmatrix} + \begin{pmatrix} c_x \\ c_y \end{pmatrix} \\ &= \begin{pmatrix} C & D \\ E & F \end{pmatrix} \begin{pmatrix} \frac{\rho X'}{\sqrt{X'^2 + Y'^2}} \\ \frac{\rho Y'}{\sqrt{X'^2 + Y'^2}} \end{pmatrix} + \begin{pmatrix} c_x \\ c_y \end{pmatrix}. \end{aligned} \quad (4)$$

2.4. Descriptores de Halarick

La matriz de co-ocurrencia del nivel de gris [22] una de las mejores herramientas para el análisis de las texturas, este método permite cuantificar las características presentes en una imagen y es ampliamente utilizado para la caracterización de diferentes tipos de terrenos en imágenes satelitales [20]. Esta matriz cuenta las parejas de píxeles vecinos en determinados ángulos ($0, 45, 90$ y 135°) y separados en una determinada distancia, usualmente 1, la misma que se usa para este trabajo (Figura 3).

Dada una matriz de co-ocurrencia es posible calcular los descriptores de Halarick, las cuales son variables que describen la textura en la imagen basados en la estadística y frecuencia de distribución [10]. Estos valores definen en un valor numérico la uniformidad, contraste, correlación entre los tonos de la imagen,

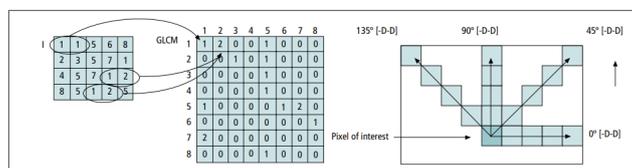


Fig. 3. Matriz de co-ocurrencia para una matriz con 8 escalas [20].

entropía, homogeneidad, valor medio del nivel de gris, desviación estándar del nivel de gris y el coeficiente de variación entre la desviación estándar y la media, para calcularlos se usan las ecuaciones 5-11 respectivamente, donde P es la matriz de co-ocurrencia de tamaño N y x cada píxel de una imagen de tamaño n .

$$\sum_{i,j=0}^{N-1} P_{i,j}^2, \tag{5}$$

$$\sum_{i,j=0}^{N-1} P_{i,j}(i-j)^2, \tag{6}$$

$$\sum_{i,j=0}^{N-1} P_{i,j} \frac{(i-\mu_i)(j-\mu_j)}{\sqrt{\sigma_i\sigma_j}}, \tag{7}$$

$$\sum_{i,j=0}^{N-1} P_{i,j}(-LnP_{i,j}), \tag{8}$$

$$\sum_{i,j=0}^{N-1} \frac{P_{i,j}}{1+(i+j)^2}, \tag{9}$$

$$\frac{1}{n} \sum_{i=1}^n x_i n_i, \tag{10}$$

$$\frac{\sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2 n_i^{-2}}}{\frac{1}{n} \sum_{i=1}^n x_i n_i}. \tag{11}$$

2.5. Red neuronal artificial

Las redes neuronales artificiales (RN) son algoritmos matemáticos basados en el funcionamiento del cerebro. Es un método de aprendizaje supervisado, es decir que tiene una fase de entrenamiento. Existe un gran número de arquitecturas de RN; la propuesta para este trabajo es una perceptrón multicapa con dos capas ocultas completamente conectadas, debido a que la más empleada en

percepción remota [8] por ser más eficiente que otros métodos para la clasificación de coberturas terrestres [15].

Lo primero a realizar es la alineación de las imágenes espectrales de la cámara, siguiendo la documentación del fabricante se encontró que la alineación entre las imágenes tienen un error entre 2 y 5 píxeles dependiendo de la banda, por lo que se ha recurrido la técnica de homografías para reducir dicho error. Se realizaron 100 fotos a un tablero ajedrez dando un total de 400 imágenes (100 por cada cámara espectral), luego se hallaron las homografías de las parejas verde-rojo, verde-borde rojo y verde-infrarrojo, tomando la banda verde como la imagen maestra, tal como se señala en la documentación del sensor, finalmente se calcularon las homografías promedio, de cada pareja, y su respectiva desviación estándar (Tabla 2). Realizada la alineación se procedió a entrenar la red neuronal en un periodo de mil épocas, tomando 500 muestras para entrenamiento, 250 para la validación y 250 para la etapa de prueba.

Tabla 2. Homografías promedio para la alineación entre bandas y su desviación estándar tomando como referencia la banda verde.

Banda.	Homografía promedio.	Desviación estándar.
Roja	$\begin{pmatrix} 1,0 & 0,0 & 53,5 \\ 0,0 & 1,0 & -9,3 \\ 0,0 & 0,0 & 1,0 \end{pmatrix}$	$\begin{pmatrix} 0,0 & 0,0 & 0,27 \\ 0,0 & 0,0 & 0,74 \\ 0,0 & 0,0 & 0,0 \end{pmatrix}$
Borde Rojo	$\begin{pmatrix} 1,0 & 0,0 & -9,9 \\ 0,0 & 1,0 & 27,2 \\ 0,0 & 0,0 & 1,0 \end{pmatrix}$	$\begin{pmatrix} 0,0 & 0,0 & 0,89 \\ 0,0 & 0,0 & 0,33 \\ 0,0 & 0,0 & 0,0 \end{pmatrix}$
Infrarrojo	$\begin{pmatrix} 1,0 & 0,0 & 41,8 \\ 0,0 & 1,0 & 15,7 \\ 0,0 & 0,0 & 1,0 \end{pmatrix}$	$\begin{pmatrix} 0,0 & 0,0 & 0,70 \\ 0,0 & 0,0 & 0,56 \\ 0,0 & 0,0 & 0,0 \end{pmatrix}$

El uso de redes neuronales es muy efectivo para estimar modelos de aprendizaje supervisado de datos complejos no lineales [23]. El perceptrón multicapa completamente conectado es una de las arquitecturas más utilizadas, se compone por una capa de entrada, capas ocultas de neuronas ordenadas conectadas a una capa anterior y posterior, capas de normalización de los datos y una capa de salida donde se encuentra el valor de la clasificación [5]. Un perceptrón multicapa completamente conectado con una capa oculta es conocido como aproximador universal, sin embargo es posible mejorar los resultados agregando más capas ocultas [13]. Se le denomina completamente conectada debido a que cada neurona de la capa de entrada y de las ocultas tienen sus salidas conectadas a cada neurona de la siguiente capa. La red que se compone esta compuesta por una capa de entrada de dimensión 52, que representan los descriptores de Haralick calculados, posteriormente se encuentra una capa oculta de 500 neuronas, seguido de una capa de normalización, seguido de 5 capas ocultas de dimensiones: 250, 140, 70, 35 y 7; posteriormente una segunda capa de

normalización, finalmente una capa de salida de una sola neurona la cual indica si la imagen corresponde al árbol de dzidzilché o no (Figura 4).

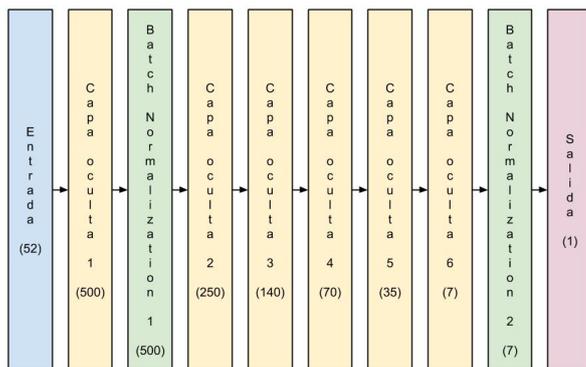


Fig. 4. Estructura de la neurona artificial completamente conectada propuesta. En la parte inferior de cada capa se encuentra el número de neuronas que se componen.

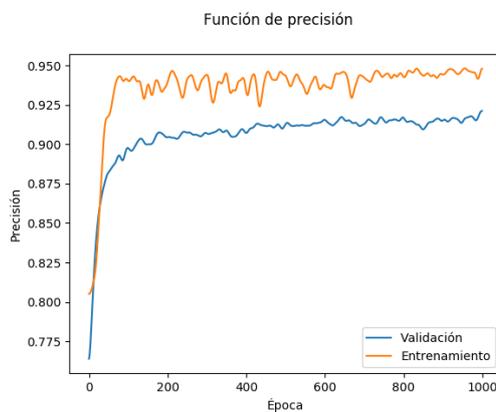


Fig. 5. Función de exactitud por época.

3. Resultados

La red se entrenó durante 1000 épocas con una tasa de aprendizaje de 0,01, el cuál logró alcanzar, al final de la etapa de entrenamiento, una precisión promedio del 94 % y 91 % para los conjuntos de entrenamiento y validación,

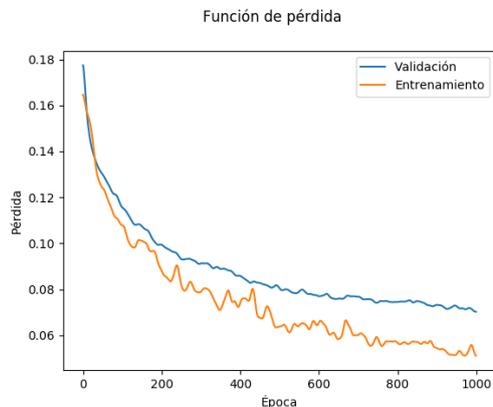


Fig. 6. Función de pérdida por época.

respectivamente (Figura 5); adicionalmente es posible observar que la función de pérdida respalda la robustez del modelo propuesto (Figura 6), pues se alcanza errores menores a 0,1.

Finalmente, en la etapa de prueba se obtuvo una precisión del 93 % con un valor de pérdida de 0,05, adicionalmente, se calcularon las tablas de contingencia con los valores absolutos (Tabla 3), y con valores relativos (Tabla 4); además se calculó la curva característica operativa del receptor (roc) así como el área bajo dicha curva (auc) (Figura 7) como parámetros adicionales para evaluar el modelo, cuyo resultado fue bastante bueno, pues en el valor del área auc (0.98) está muy próximo a 1, por lo cual se puede considerar un modelo excelente que cumpla el objetivo de detectar el árbol de dzidzilché.

Tabla 3. Tabla de contingencia con los valores absolutos.

		Valor real		Total
		Positivo	Negativo	
Predicción	Positivo	111	11	122
	Negativo	5	123	128
Total		116	134	250

4. Conclusiones

Con la realización de este trabajo hemos demostrado que utilizar los descriptores de Halarick en imágenes áreas capturadas con el sensor Sequoia Parrot proporcionan la información adecuada para que una red neuronal pueda discriminar una planta polinífera en específico, el árbol de dzidzilché. Sin embargo, es

Tabla 4. Tabla de contingencia con los valores relativos.

		Valor real		Total
		Positivo	Negativo	
Predicción	Positivo	44,4 %	4,4 %	48,8 %
	Negativo	2 %	49,2 %	51,2 %
Total		46,4 %	53,6 %	100 %

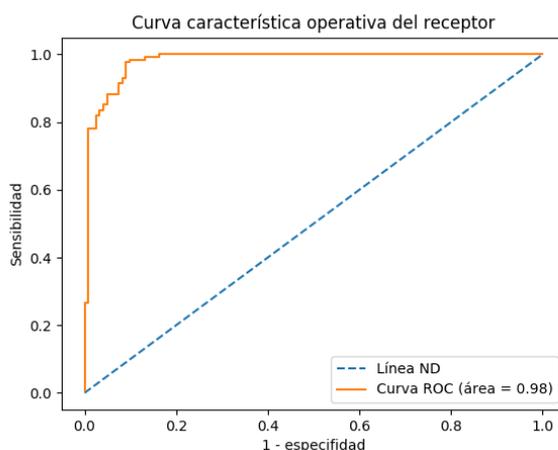


Fig. 7. Función de exactitud por época.

posible utilizar la información espectral colectada para evaluar y comparar con otros modelos, por ejemplo una red neuronal convolucional o un autoencoder y de tal manera aumentar la complejidad de la caracterización de las diferentes texturas que puedan encontrarse en las regiones políníferas.

Finalmente, el modelo planteado cumple el objetivo de clasificación propuesto, lo cual permitiría cuantificar el espacio de crecimiento del árbol de dzidzilché, valor que es de utilidad para mejorar la producción de miel del estado de Yucatán.

Referencias

1. Anaya, A., Jesus, A., Duque, L., Rosa, A., Valencia, H., German, M.: Texture analysis of satellite imagery in the field of biodiversity and forest structure of the colombian andes; analisis de textura en imagenes de satelite en el ambito de la biodiversidad y la estructura en un bosque de los andes colombianos (2008)
2. Atkinson, P.M., Lewis, P.: Geostatistical classification for remote sensing: an introduction. *Computers & Geosciences* 26(4), 361–371 (2000)
3. Centro Nacional de Información y Comunicación Educativa: Resolución de las imágenes satélite (2006), http://concurso.cnice.mec.es/cnice2006/material121/unidad1/i_resolucion.htm, último acceso: 2019/05/11

4. Córdova-Córdova, C.I., Ramírez-Arriaga, E., Martínez-Hernández, E., Zaldívar-Cruz, J.M.: Caracterización botánica de miel de abeja (*apis mellifera* l.) de cuatro regiones del estado de tabasco, méxico, mediante técnicas melisopalinológicas. *Universidad y ciencia* 29(2), 163–178 (2013)
5. Daumé III, H.: *A course in machine learning*. Publisher, ciml. info 5, 69 (2012)
6. Echazarreta González, C.: Apicultura y producción de miel, <https://www.cicy.mx/documentos/CICY/Sitios/Biodiversidad/pdfs/Cap2/18%20Apicultura%20y%20produccion.pdf>, último acceso: 2019/05/11
7. Fernández, L., Álvarez, G., Salinas, R.: Methods for analyzing surface texture effects of volcanoes with plinian and subplinian eruptions types: Cases of study lascar (23° 22's) and chaiten (42° 50's), chile (2018)
8. Foody, G.M.: Thematic mapping from remotely sensed data with neural networks: Mlp, rbf and pnn based approaches. *Journal of Geographical Systems* 3(3), 217–232 (2001)
9. Güemes Ricalde, F.J., Echazarreta González, C., Villanueva, R., Pat Fernández, J.M., Gómez Alvarez, R.: La apicultura en la península de yucatán. actividad de subsistencia en un entorno globalizado. *Revista Mexicana del Caribe* 8(16) (2003)
10. Haralick, R.M., Shanmugam, K., et al.: Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics* (6), 610–621 (1973)
11. Hauenstein, E., González, M., Peña-Cortés, F., Muñoz-Pedreras, A.: Clasificación y caracterización de la flora y vegetación de los humedales de la costa de toltén (ix región, chile). *Gayana. Botánica* 59(2), 87–100 (2002)
12. Instituto Nacional de Economía Social: Historia e importancia de la apicultura (2018), <https://www.gob.mx/inaes/articulos/historia-e-importancia-de-la-apicultura?idiom=es>, último acceso: 2019/05/11
13. Lab, L.: *Deep Learning Tutorial*. University of Montreal, Canada (2015)
14. Leopold, A.: *Game management* charles scribner's sons. New York 481 (1933)
15. Lizarazo, I.: Clasificación de la cobertura y del uso del suelo urbano usando imágenes de satélite y algoritmos supervisados de inteligencia artificial. *UD y la geomática* (2), 4–18 (2008)
16. Machado, R., Roche, R., Toral, O., García, E.G.: Metodología para la colecta, conservación y caracterización de especies herbáceas, arbóreas y arbustivas útiles para la ganadería. *Pastos y Forrajes* 22(3), 181–204 (1999)
17. Maldonado, F., Quignard, I., Zamboni, L., Rodriguez, E., Sione, W.: Uso de imágenes landsat tm para la delimitacion de unidades de muestreo ambiental en el delta del río paraná
18. Perotti, M.G., Diéguez, M.C., Jara, F.G.: Estado del conocimiento de humedales del norte patagónico (argentina): aspectos relevantes e importancia para la conservación de la biodiversidad regional. *Revista chilena de historia natural* 78(4), 723–737 (2005)
19. Pix4D: <https://support.pix4d.com/hc/en-us/articles/202559089-How-are-the-Internal-and-External-Camera-Parameters-defined>, último acceso: 2019/05/11
20. Ríos-Díaz, J., Martínez-Payá, J.J., del Baño Aledo, M.E.: El análisis textural mediante las matrices de co-ocurrencia (glcm) sobre imagen ecográfica del tendón rotuliano es de utilidad para la detección cambios histológicos tras un entrenamiento con plataforma de vibración.(textural analysis by means of a grey level co-occurrence matrix method on patellar tendon ultrasonography is useful for the detection of histological changes after whole-body vibration training). *CCD. Cultura.Ciencia.Deporte*. doi: 10.12800/ccd 4(11), 91–102 (2009)

Juan Jose Negron-Granados

21. Rodríguez-Cruz, M., Moctezuma-Flores, M.: Análisis bayesiano y fusión de datos para la clasificación de escenas urbanas del distrito federal. *Ingeniería, investigación y tecnología* 7(1), 17–28 (2006)
22. Tahir, M.A., Bouridane, A., Kurugollu, F.: An fpga based coprocessor for glcm and haralick texture features and their application in prostate cancer classification. *Analog Integrated Circuits and Signal Processing* 43(2), 205–215 (2005)
23. Wali, A.: *Clojure for Machine Learning*. Packt Publishing (2014)

Métodos computacionales para la creación artística

Sandra Rodríguez-Mondragón, Luis Jorge Soto-Walls,
Manuel Martín Clavé-Almeida, Oscar Herrera-Alcántara

Universidad Autónoma Metropolitana, Unidad Azcapotzalco,
Ciudad de México, México

{sandra.rgz.mondragon, luissotowalls, mclavealmeida}@gmail.com,
oha@correo.azc.uam.mx

Resumen. Se presentan algunas formas de aplicar principios algorítmicos al desarrollo de obra artística con carácter digital. Como parte inicial se abordan las definiciones de algunos conceptos tales como obra de arte, arte digital, algoritmo y tiempo real; se describen cuatro métodos algorítmicos donde es posible incluir el concepto de tiempo real. Se abordan algunas formas de expresión artística desarrolladas por medio de algoritmos como los fractales, el arte sonoro y el data mining. Finalmente se describe cómo se pueden implementar las metodologías del diseño de algoritmos descritas al desarrollo de arte digital, donde interviene el tiempo real como variable independiente para la creación artística y se presentan algunos desarrollos de producción artística con IA.

Palabras clave: arte, arte digital, tiempo real, algoritmo, fractal, data mining.

Computational Methods for Artistic Creation

Abstract. Some ways of applying algorithmic principles to the development of artistic work with digital character are presented. As an initial part, the definitions of some concepts such as work of art, digital art, algorithm and real time are addressed; Four algorithmic methods are described where it is possible to include the concept of real time. Some forms of artistic expression developed through algorithms such as fractals, sound art and data mining are also tackled. Finally, we describe how the methodologies of the described algorithm design can be implemented, to the development of digital art where real time intervenes as an independent variable for artistic creation and are presented some developments of artistic production with AI.

Keywords: art, digital art, real time, algorithm, fractal, data mining.

1. Definiciones

1.1. Obra de arte

Con objeto de lograr definir el concepto de obra de arte, revisaremos algunas afirmaciones: Julio Domínguez define la obra de arte como “la manifestación concreta

de genio creativo del hombre, cuando modifica o reorganiza su entorno para impactar en los sentimientos de sus semejantes” [1].

Por su parte Ricardo Padilla, nos dice que “la obra plástica es de distintas clases y texturas: papel, tela, madera, mármol, bronce o cualquier otro material susceptible de transformarse como la arcilla o el acrílico, y son tanto bidimensionales (dibujos, cuadros, grabados, litografías, etc.) como tridimensionales (esculturas, estatuas, obras arquitectónicas, monumentos, edificios, etc.), independientemente de su género (figurativo o abstracto) y de su finalidad” [2].

Así, Laura Fernández afirma que “en la obra plástica van a incidir dos partes, la formal... y la conceptual... existen elementos que le son comunes a todas las obras del arte que son configuradores...: líneas, áreas, volumen, hay elementos diferenciadores: color, calor tonal, textura y otras leyes que componen las obras de arte son las estéticas, en las cuales se encuentran: proporción, equilibrio, ritmo o énfasis” [3]. Y finalmente Terry Smith afirma que “antes que cualquier otra cosa, las obras de arte son testimonio de cada una de estas contemporaneidades: el mero hecho de llegar a ser en y por sí mismas, su existencia en un mundo repleto de otros y su persistencia en distintos mundos determinados por la repetición y la diferencia” [4].

Así la obra de arte nos permite expresar un concepto y éste se puede materializar bidimensional o tridimensionalmente, dando pie al objeto artístico que es perceptible por medio de los sentidos de la vista (visión) y del tacto; de la misma manera hay manifestaciones artísticas que se pueden percibir por medio de los otros sentidos el oído (audición), el olfato, el gusto.

1.2. Generación de la obra artística

Los factores que intervienen en el momento de la producción de obra artística pueden ser múltiples, sin embargo, existen aspectos básicos que se deben abordar. A lo largo de la historia ha existido la obra artística y los artistas, cada uno en su tiempo, se han valido de la tecnología para la generación de su obra, algunos han innovado y otros han explotado el uso de herramientas o materiales; así, por ejemplo, las primeras esculturas realizadas en mármol se labraron con cincel que era la tecnología disponible en ese momento histórico. Actualmente existen múltiples herramientas digitales que permiten al artista lograr la generación de obra de vanguardia.

En su momento el diseño gráfico dio un salto al diseño asistido por computadora y en la actualidad, las computadoras son herramientas fundamentales de esta disciplina. Los elementos para la generación de una obra artística escultórica (por ejemplo) son básicamente tres: el diseño (concepto y forma), el material y la tecnología a emplear en el desarrollo de la obra, en ello el escultor al realizar un diseño lo visualiza de forma ideal, después decide en qué material se producirá y finalmente por medio de herramientas logrará que el material adquiera la forma proyectada. Así, el diseño regirá sobre el material y la tecnología, tomando como principio que la manifestación artística es preponderante sobre las tecnologías que permiten su producción (ver Fig. 1).

1.3. Arte digital

Estamos inmersos en un medio digital –tecnológico- en nuestra vida cotidiana, donde las nuevas tecnologías de comunicación e información (TIC's) se manifiestan en el arte, por ello surgen nuevos modos de expresión basados en las TIC's que potencializan los modos de hacer arte. En ello surge el concepto de **arte digital** como uno dentro de las nuevas artes.

De esta manera, el vínculo entre el arte y la tecnología se remonta a los años cincuenta, cuando un reducido grupo de artistas, de hecho, los pioneros del arte digital, empezó a utilizar la computadora para crear lo que se llamó las primeras pinturas electrónicas; es decir, arte hecho a través de programas informáticos comerciales para la edición de imagen o creación de dibujos. De acuerdo con Laura Regil, este género es producto de la creatividad artística y la mezcla de tecnologías digitales, ciberespacio y vida artificial, constituyéndose en un universo original de estética, espacio y tiempo. A su vez, las actuales manifestaciones artísticas experimentan y amplían tanto las posibilidades expresivas y de divulgación como las de la participación y apreciación de los espectadores [5].

Dentro de las “nuevas artes” (ver Fig. 2) podemos incluir a la producción artística que se genera a partir de las nuevas tecnologías y por medio de conceptos matemáticos que sugieren el arte a partir de esquemas susceptibles de reproducción controlada. Las nuevas artes provienen de un proceso creativo que se beneficia de la tecnología para lograr formas y efectos que, sin ella, simplemente no podrían realizarse, quedando en arte conceptual. Surgen así formas innovadoras de alta calidad y estructuras resueltas matemáticamente de forma muy precisa, que, si bien pueden generarse por medio de otros procesos, con la tecnología son resueltas eficientemente. Este tipo de obra es producto de conjugar diversos medios de expresión, vía la tecnología, y obtener un concepto más rico para todos los sentidos del ser humano, de esta manera se integra la música con la imagen, el arte que se difunde en Internet, el dibujo y el modelado computarizado, entre otros medios que combinados logran un resultado expresivo muy diverso.

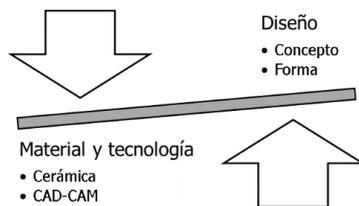


Fig. 1. Producción de obra artística.



Fig. 2. Las nuevas artes, basado en las definiciones de Irma Escobar [6].

A modo de ejemplo de las manifestaciones artísticas que emplean principios matemáticos tenemos la clasificación que hace Ricardo Zalaya, de la escultura

matemática, ello nos permite visualizar el potencial de este tipo de formas de expresión (ver Fig. 3) [7].

1.4. Algoritmo

Como dato histórico, tenemos que la palabra algoritmo trae su origen en el nombre del matemático persa "Mohamed ibn Musa al Khwarizmi" (825 d.C.). Su apellido fue traducido al latín como *algorismus* y posteriormente pasó al español como Algoritmo. Khwarizmi fue bibliotecario en la corte del califa al- Mamun y astrónomo en el observatorio de Bagdad. Sus trabajos de álgebra, aritmética y tablas astronómicas adelantaron enormemente el pensamiento matemático y fue el primero en utilizar la expresión *al-yabr* (de la que procede la palabra álgebra). Su trabajo con los algoritmos introdujo el método de cálculo utilizando la numeración arábiga y la notación decimal [8].

Por otro lado, Francisco Pinares *et. al.*, afirman que “para implementar la solución de un problema mediante el uso de una computadora, es necesario establecer una serie de pasos que permitan resolver el problema, a este conjunto de pasos se le denomina algoritmo, el cual debe tener como característica final la posibilidad de transcribirlo fácilmente a un lenguaje de programación, para esto se utilizan herramientas de programación, las cuales son métodos que permiten la elaboración de algoritmos escritos en un lenguaje entendible” [9]. Gustavo López *et. al.*, lo definen como “un conjunto de pasos que, ejecutados de la manera correcta, permiten obtener un resultado (en un tiempo acotado)” [10].

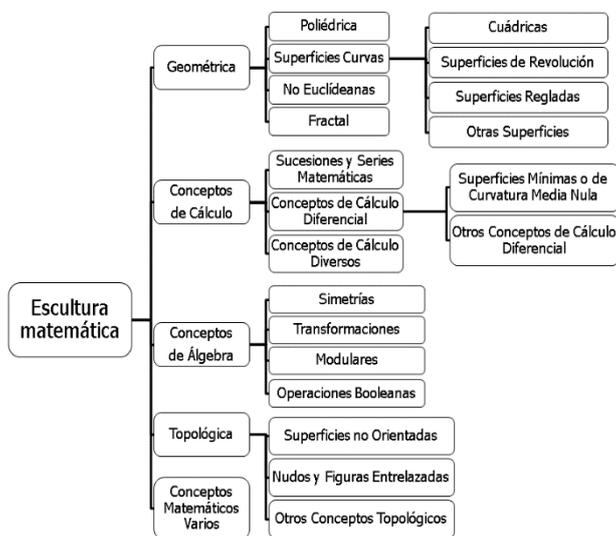


Fig. 3. Clasificación de escultura matemática. Basado en la propuesta de Zalaya.

Pueden existir varios algoritmos para resolver un mismo problema. Cuando se estudian los algoritmos es importante analizar tanto su diseño como su eficiencia. Se

puede complementar la definición con las tres características que indica Pinales [9], además de la facilidad para transcribirlo: preciso (debe indicar el orden en el cual debe realizarse cada uno de los pasos que conducen a la solución del problema), definido (el resultado nunca debe cambiar bajo las mismas condiciones del problema) y finito (no se debe caer en repeticiones de procesos de manera innecesaria; deberá terminar en algún momento).

Así un algoritmo es un método para resolver un problema. En lo que a esta investigación refiere, citaremos algunos métodos computarizados con base en algoritmos para producir obra artística, aplicando el concepto de tiempo real. Donde la metodología que aplica algoritmos es una metodología que resuelve problemas por medio de una secuencia de pasos precisa, definida y finita.

Por otro lado, otra cualidad de los algoritmos es el lenguaje en que se pueden representar y ello lo retomamos de la descripción que hace Matías García [11], donde indica básicamente tres:

- Lenguaje natural, es el lenguaje común (coloquial).
- Lenguaje estructurado, con reglas sintaxis y semántica definidas, donde se puede usar pseudocódigo o lenguaje universal de programación; o de código que implica el uso de un programa en específico.
- Lenguaje simbólico que se puede desarrollar principalmente por medio de diagramas de flujo.

Así en lo que a algoritmo se refiere, de acuerdo con Ramírez, se deben considerar sus tres partes [12]: (1) Entrada: Información dada al algoritmo. (2) Proceso: Operaciones o cálculos necesarios para encontrar la solución del problema. (3) Salida: Respuestas dadas por el algoritmo o resultados finales de los procesos realizados.

De acuerdo con Carlos Balderrama [13], existen dos tipos de algoritmos: cualitativos que son aquellos en los que se describen los pasos utilizando palabras; y cuantitativos que son aquellos en los que se utilizan cálculos numéricos para definir los pasos del proceso.

1.5. Tiempo real

En el ámbito de la tecnología, el tiempo real se aplica a sistemas digitales en los que se produce una interacción en que se corresponde el tiempo interno del sistema con el tiempo externo o del ambiente. Por lo tanto, en un sistema en tiempo real se produce una interacción con el mundo real, entendido como un proceso físico, en el que se emiten respuestas correctas siguiendo las restricciones temporales. Algunos ejemplos de áreas concretas en que se puede utilizar este término son determinadas telecomunicaciones como las emisiones televisivas en directo o en meteorología, cuando un dispositivo indica la temperatura existente en el momento actual.

Los sistemas en tiempo real se utilizan ampliamente el día de hoy, tanto en la industria aérea, automotriz, ferroviaria, en el sector militar, en la medicina, en los electrodomésticos, prácticamente en cualquier aplicación que utilice software.

De la misma forma Alejandro Veigía [13], define a continuación un sistema de tiempo real como “un sistema que debe satisfacer restricciones explícitas en el tiempo

de respuesta o arriesgarse a severas consecuencias, incluida la falla. Por tanto, un sistema de tiempo real es un sistema que responde a un estímulo externo dentro de un tiempo especificado. Su eficiencia no solo depende de la exactitud de los resultados de cómputo, sino también del momento en que los entrega. La predictibilidad es su característica principal. A diferencia de los sistemas tradicionales, que tienden a distribuir en forma equitativa los recursos disponibles entre las diferentes tareas a ejecutar, los sistemas de tiempo real deben asegurar la distribución de recursos de tal forma que se cumplan los requerimientos de tiempo”. De la anterior afirmación cabe mencionar que la definición de tiempo real depende del tipo de aplicación que se dé al concepto.

2. Métodos algorítmicos

Si queremos producir una obra artística basada en *software*, primero tendremos que atravesar un proceso de diseño, conceptualización, escritura, compilación, depuración de errores, etc. Este proceso se encuentra mucho más relacionado con la ingeniería que con las artes plásticas.

De acuerdo con Balderrama [14], algunas de las herramientas utilizadas comúnmente para diseñar algoritmos son: seudocódigo, diagrama de flujo, diagramas Nassi/Shneiderman-S (Chapin), método Jackson, método Bertini, método Tabourier, método Warnier, entre otros (ver fig. 4).

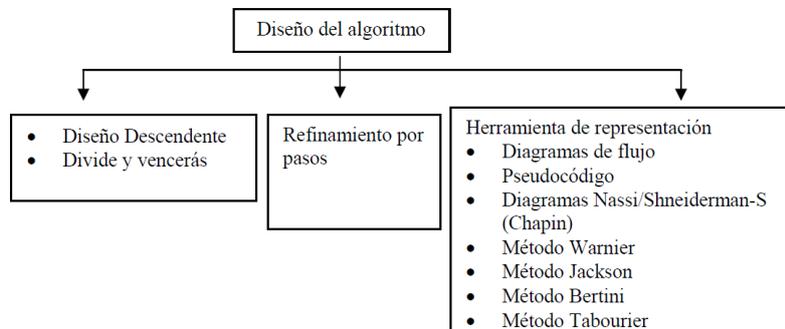


Fig. 4. Diseño de algoritmo.

Un computador no tiene capacidad para solucionar problemas más que cuando se le proporciona los sucesivos pasos a realizar, estos pasos indican las instrucciones a ejecutar por la máquina y se denomina algoritmo. Normalmente los pasos diseñados en un primer esbozo del algoritmo son incompletos e indican sólo unos pocos pasos, tras esta primera descripción estos se amplían en una descripción más detallada con pasos específicos; dicho proceso se denomina refinamiento del algoritmo. Las características de un buen algoritmo son:

- Debe tener un punto particular de inicio.
- Debe ser definido, no debe permitir dobles interpretaciones.

- Debe ser general, es decir, soportar la mayoría de las variantes que se puedan presentar en la definición del problema.
- Debe ser finito en tamaño y tiempo de ejecución [14].

En lo que refiere a la generación artística, es posible aplicar el principio algorítmico que se desee, ello dependerá del concepto de la obra, el material o forma en que se representará y por supuesto los recursos con que se cuente para su producción. A continuación, menciono algunos de los métodos que ya se han usado para tal fin, sin que ello se contemple como una compilación exhaustiva, pero si con carácter didáctico; cabe mencionar que en los casos que se aplica el concepto de *tiempo real* generalmente nos referimos a eventos en los que el medio de expresión es el sonido, aunque también podemos hablar de visualización (video, foto, imagen digital, etc.).

2.1. Composición algorítmica

El concepto de composición algorítmica se refiere al método de creación musical por medio de procesos formales o algoritmos. Es un método para la creación de música no lineal, es decir cuando los eventos sonoros no están predeterminados hasta el momento en que ocurren [14].

La composición algorítmica se puede desarrollar por medio de algoritmos genéticos, gramáticas formales, autómatas celulares y cadenas de Markov; en este ensayo sólo se describen las antes mencionadas, pese a que se pueden aplicar otras.

2.2. Algoritmos genéticos

Retomando la definición de Fogel, son métodos adaptativos, generalmente usados en problemas de búsqueda y optimización de parámetros, basados en la reproducción sexual y en el principio de supervivencia del más apto [15]. Así, Goldberg afirma que “son algoritmos de búsqueda basados en la mecánica de selección natural y de la genética natural. Combinan la supervivencia del más apto entre estructuras de secuencias con un intercambio de información estructurado, aunque aleatorizado, para constituir así un algoritmo de búsqueda que tenga algo de las genialidades de las búsquedas humanas” [16].

Tabla 1. Operatividad de gramática formal [18].

Gramática	Lenguaje	Reglas de Producción	Si $\mu \rightarrow \varphi$, relación entre $ \mu $ y $ \varphi $	Solución
Tipo-0	Recursivas	Sin restricciones		Máquinas de Turing
Tipo-1	Dependiente de contexto	$\alpha A \beta \rightarrow \alpha \gamma \beta$	$ \mu \leq \varphi $	Autómatas lineales acotados
Tipo-2	Independiente de contexto	$A \rightarrow \gamma$	$ \mu = 1$	Autómatas de pila
Tipo-3	Regular	$A \rightarrow aB$ $A \rightarrow a$		Autómatas finitos, regulares

α β y γ son cadenas de terminales y no terminales. La cadena γ no puede ser la cadena vacía.

Para alcanzar la solución a un problema se parte de un conjunto inicial de individuos, llamado población, generado de manera aleatoria. Cada uno de estos individuos

representa una posible solución al problema. Estos individuos evolucionarán tomando como base los esquemas propuestos por Darwin sobre la selección natural, y se adaptarán en mayor medida tras el paso de cada generación a la solución requerida [17].

2.3. Gramáticas formales

Una gramática formal consta de un conjunto finito de símbolos terminales (las palabras en un lenguaje formal), un conjunto finito de símbolos no terminales, un conjunto de reglas de producción con un lado izquierdo y otro derecho, y un símbolo inicial. Las reglas se aplican sustituyendo la parte de la izquierda por la parte de la derecha. Una derivación es una secuencia de aplicaciones de reglas. Cada gramática define el lenguaje formal de todas las sentencias que están formadas exclusivamente por los símbolos terminales a los que se puede llegar mediante derivación a partir del símbolo inicial.

Así, Gonzalo, describe cuatro tipos de gramáticas formales: tipo 0 (sin restricciones, recursivas), tipo 1 (dependientes de contexto), tipo 2 (independientes de contexto, libre de contexto) y tipo 3 (gramáticas regulares). Y también afirma que: todo lenguaje de tipo 3 es de tipo 2, todo lenguaje de tipo 2 es de tipo 1, y todo lenguaje de tipo 1 es de tipo 0 [18]. Lo anterior también, en lenguaje de programación, se puede representar como en el cuadro que se muestra a continuación en la tabla 1.

2.4. Autómatas celulares

Un autómata celular es un modelo matemático para un sistema dinámico, compuesto por un conjunto de celdas o células que adquieren distintos estados o valores. Estos estados son alterados de un instante a otro en unidades de tiempo discreto, es decir, que se puede cuantificar con valores enteros a intervalos regulares. De esta manera, este conjunto de células logra una evolución según una determinada expresión matemática, que es sensible a los estados de las células vecinas, la cual se le conoce como regla de transición local. El aspecto que más los caracteriza es su capacidad de lograr una serie de propiedades que surgen de la propia dinámica local a través del paso del tiempo y no desde un inicio, aplicándose a todo el sistema en general. Por lo que no es fácil analizar sus propiedades globales desde su comienzo, complejo por naturaleza, a no ser por vía de la simulación, partiendo de un estado o configuración inicial de células y cambiando en cada instante los estados de todas ellas de forma síncrona [19].



Fig. 5. Formas arquitectónicas básicas que surgen de un autómata celular.

Algunas de las aplicaciones de autómatas celulares se dan en la arquitectura, bioinformática y la criptografía.

Arquitectura. La conexión que se hace con ésta es la capacidad de generar patrones o modelos y, de una forma organizada, estos modelos nos pueden sugerir formas arquitectónicas (ver Fig. 5).

La Bioinformática consiste en analizar, comprender y predecir procesos biológicos con la ayuda de herramientas computacionales. Puede ser vista como la disciplina que une dos ciencias: Biología y Computación (ver Fig. 6).

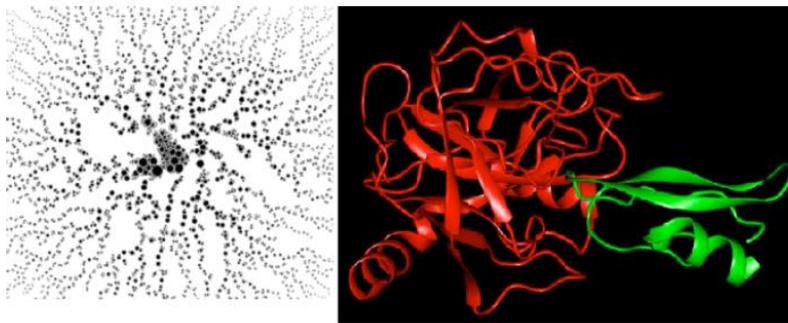


Fig. 6. Interacciones entre proteínas, tema común de estudio de la Bioinformática.

Por necesidades del ser humano, transmitir información de forma segura, surge la criptología (del griego cripto-oculto- y logos-tratado, ciencia), ésta se divide en dos ramas: criptografía, cuya tarea es cifrar la información a enviar, y criptoanálisis que se encarga de analizar técnicas y métodos para obtener la información cifrada (ver Fig. 7).

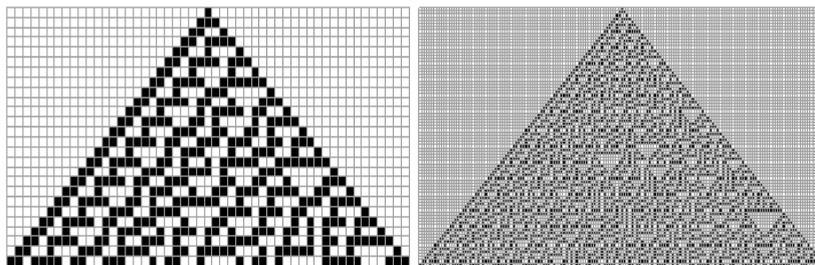


Fig. 7. Diagrama de evolución del autómata celular, regla 30 en 25 y 100 generaciones.

3. Arte algorítmico

A pesar de su aparente especificidad, la creación de imágenes de síntesis no difiere en lo fundamental del proceso creativo de las artes plásticas. Por lo que en ambos procesos de creación de imágenes (abstracciones visibles), a partir de modelos surgidos

de la imaginación o la observación de sus creadores; a diferencia de cualquier otra técnica anterior, las técnicas digitales permiten además reproducir (y modificar) imágenes de cualquier origen de un modo sencillo, rápido y barato. Un orden visual renovado del que surge una nueva situación iconográfica.

3.1. Fractal

Todas las imágenes creadas y reproducidas por computadora son el resultado de una sucesión de algoritmos matemáticos. Un caso particular de imagen sintética son las imágenes fractales, figuras geométricas formadas por un número infinito de elementos de pequeñez infinita, contenidos en una superficie infinita que sólo se pueden ver, siguiendo determinados algoritmos, con la ayuda de computadoras (ver. Fig. 8).



Fig. 8. Ejemplo de fractal desarrollado por computadora.

3.2. Algoritmos en arte sonoro

Con la ayuda de medios digitales y analógicos actuales e inspirándose en la tradición de pioneros del vídeo experimental como Nam June Paik y Woody y Steina Vasulka, Koolwijk y Prins compartieron sus conocimientos técnicos y con su proyecto Synchronator se lanzaron a resolver las dificultades involucradas en el registro de señales de vídeo y audio complejas, híbridas y distorsionadas. En directo, su performance audiovisual cuenta con dos proyecciones simultáneas y es consecuencia directa de su interés mutuo en convertir la imagen en sonido y viceversa, haciendo uso de los métodos que han desarrollado para conseguir que equipos de vídeo convencionales acepten señales analógicas, y a veces de audio y vídeo perturbados, como si fueran señales legítimas.

Regularmente cuando se trabaja con arte sonoro se hace uso de Procesadores Digitales de Señal (DSP), lo que permite captar el sonido y convertirlo a datos que se pueden procesar en un ordenador. Los procesadores DSP pueden realizar tareas en tiempo real, sin embargo, la definición de tiempo real depende del tipo de aplicación [20].

3.3. Data mining o minería de datos (DM)

Sonia Pighin la describe como la extracción de información en grandes bases de datos. Su nombre deriva de la analogía existente entre buscar dicha información valiosa y minar una montaña para encontrar un yacimiento de metales preciosos, ya que ambos procesos requieren examinar una inmensa cantidad de material o investigar inteligentemente hasta concretar la búsqueda.

Es un conjunto de tecnologías que ayuda a las empresas a enfocar sus objetivos sobre la información más importante de sus fuentes de datos. Las herramientas de DM pueden responder preguntas que generalmente demandan demasiado tiempo, encontrando información que ni un profesional experto podría hallar porque se encuentra fuera de sus expectativas. DM es en realidad, un proceso iterativo de descubrimiento de patrones y tendencias dentro de los datos, a través de métodos automáticos, manuales o más generalmente semiautomáticos, y que no serían necesariamente revelados por otros métodos tradicionales de análisis. Las herramientas de DM exploran las bases de datos en busca de patrones ocultos, permitiendo a partir de éstos, predecir las futuras tendencias y comportamientos de información nueva [21].

Una de las ventajas del DM, consiste en que el manejo de la información se puede realizar por medio de texto o de forma visual, algunas de las formas de visualizar los datos son: agrupación, estructuras jerárquicas, redes y panorámicas.

Con base en el DM, algunos artistas visuales como Ulises Carrión, Daniel Lara, Víctor Manuel Dávalos, Felipe Ehremberg, entre otros, han implementado el concepto de tiempo real o transmisión automática debido a que por medio de un DSP se captan las frecuencias en un concierto en vivo ya sea de música tradicional o electrónica.

4. Conclusión

Es complicado citar un método general para la resolución de problemas, en este caso para producir obra artística por medio de algoritmos aplicando la variable de tiempo real, la resolución de un problema es un proceso creativo donde el conocimiento, la habilidad y la experiencia tienen un papel importante, el proceder de manera sistemática ayuda a resolver cualquier problema y el diseño de algoritmos es una forma estructurada de sistematizar la forma de resolver un problema.

Al comenzar a abordar un problema hay que tener en cuenta que para la mayoría de ellos hay muchas maneras de resolverlos y pueden existir muchas soluciones, se deben explorar los criterios conceptuales de la obra y decidir la estrategia para su creación.

En lo que refiere a tiempo real, dicho término aplica a sistemas ideales, que, en sentido estricto, es inexistente; por lo que ningún modelo es una representación exacta de la realidad, su validez depende de lo razonables que sean las estimaciones o suposiciones tomadas y de su grado de aproximación a la realidad. Si se pretende generar obra evolutiva a partir de la variable de tiempo real como dato fijo, el resultado sólo puede ser un sistema que contabiliza tiempo con precisión.

El desarrollo tecnológico ha impulsado (e impulsa) la aparición de nuevas formas de expresión, las herramientas por sí mismas no representan un estímulo para la

creatividad personal, pues la capacidad artística depende de factores más profundos que la mera disposición de medios tecnológicos.

Para producir obra artística por medio de tecnologías e implementar el uso de algoritmos, es necesario hacer el desarrollo conceptual de la obra y proponer una hipótesis que permita visualizar de forma esquemática los momentos o etapas que se deben resolver por medio de uno o varios algoritmos. Un ejemplo de desarrollo escultórico, aplicando modelado basado en algoritmos y sucesión Fibonacci se muestra a continuación (ver Fig. 9 y 10) [22].

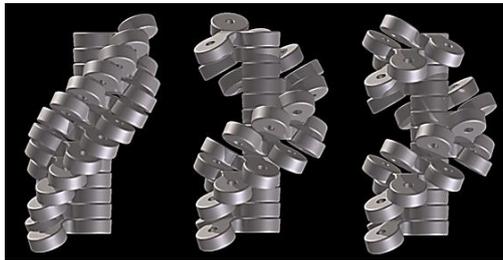


Fig. 9. Escultura modular con un desarrollo basado en un acomodo de orden radial con un número de módulos correspondiente a la sucesión de Fibonacci y un ángulo de separación variable.

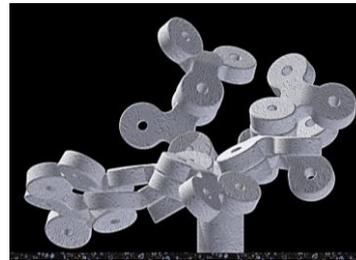


Fig. 10. Escultura modular con un desarrollo basado en un acomodo de orden programado en sucesión de Fibonacci y número de módulos con base en un algoritmo de equilibrio.

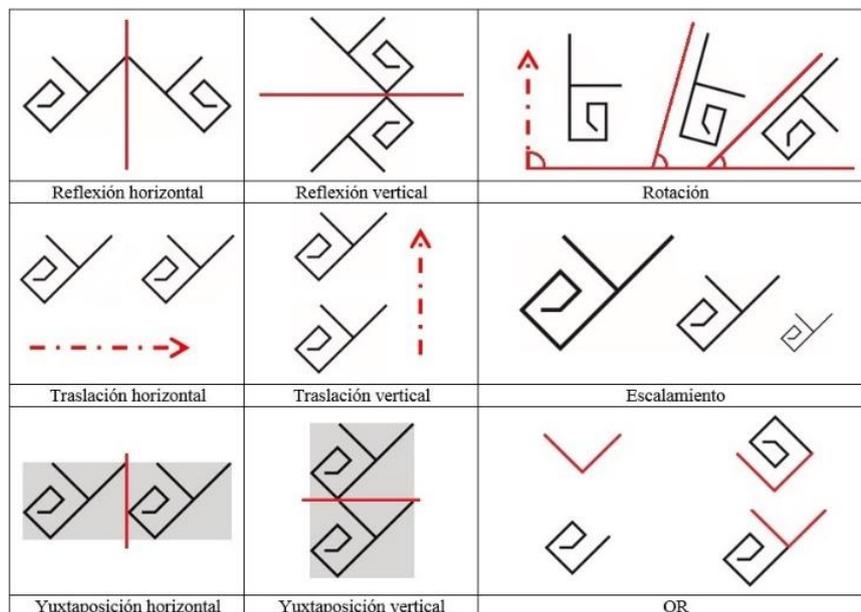


Fig. 11. Representación visual de operadores.

Otro ejemplo de desarrollo modular basado en un programa de reconocimiento de patrones se muestra a continuación, donde dicho reconocimiento de patrones opera por medio de un autómata de pila que procesa íconos para obtener imágenes de indumentaria indígena por medio de un algoritmo de evaluación de cadenas en notación postfija. Estos desarrollos se basan en modelos formales de iconografía de textil indígena de la región de los Altos de Chiapas, Chiapas, México. Así, la aplicación de operadores a los íconos ha permitido reconstruir una imagen original (ver Fig. 11 y 12) [23].

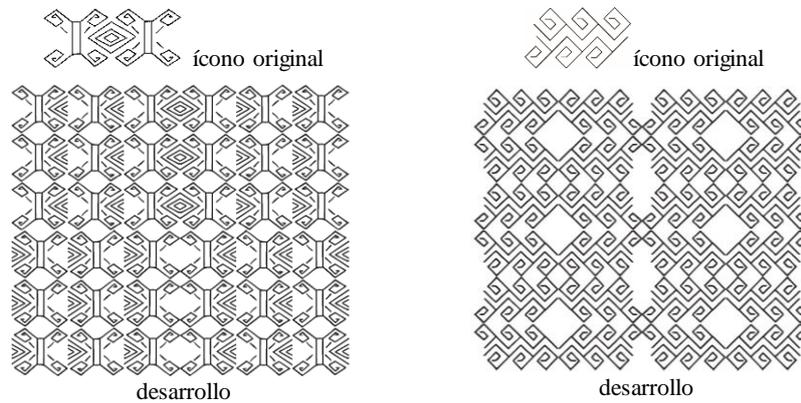


Fig. 12. Desarrollos visuales obtenidos por medio de aplicación de operadores en un autómata de pila que opera por medio de un algoritmo de evaluación de cadenas en notación postfija.



Fig. 13. Comparativo de obra artística producida por artistas e IA [23].

Finalmente, y como un ejemplo reciente, pero con base de desarrollo similar a los casos antes mencionados, Francisco Ordóñez [24], en su conferencia “¿Sueñan los androides con obras de Banksy? [...]”, nos hace una reseña de algunas aplicaciones de Inteligencia Artificial (AI) para la generación de expresiones artísticas en diversos campos, entre ellos la de obra pictórica producto de algoritmos y redes neuronales con bancos de datos visuales tales como fotografías de cuadros de artistas plásticos, lo que

ha permitido generar patrones de cualidades formales de la obra analizada con el fin de producir obra original aplicando estas cualidades.

En esta conferencia, para demostrar su premisa, Ordóñez somete a evaluación de la audiencia un grupo de imágenes para identificar cual es producto de IA, sin embargo, se precisa de un análisis que permita demostrar el proceso para identificar la obra de IA y comprobar la hipótesis (ver Fig. 13).

Cabe resaltar que la IA por sí sola tiene la capacidad de producir propuestas visuales con mayor velocidad que el ser humano, pero ello no garantiza la creación de conceptos o estilos para la producción artística, aunque sí es una poderosa herramienta que potencializa capacidad productiva de un artista y hasta el momento depende del desarrollador del concepto, que a su vez debe identificar qué procesos de la producción artística se pueden mejorar por medio IA.

Referencias

1. Domínguez Bedoya, J.: El concepto de obra de arte en la legislación mexicana. Alegatos, noviembre 2010, núm. 11 pp. 1–7, www.azc.uam.mx/publicaciones/alegatos/pdfs/11/11-03. Último acceso: 26/03/19 (2010)
2. Padilla Padilla, R.A.: La obra plástica y su protección en la legislación panameña. http://www.legalinfo-panama.com/articulos/articulos_29a.htm, Último acceso: 20/02/19 (2003)
3. Fernández Martínez, L.M.: Los retos sociales y comunicativos de la pintura mural santiaguera, en la formación del individuo. Revista caribeña de ciencias sociales (2013)
4. Smith, T.: ¿Qué es el arte contemporáneo? Siglo XXI, México, p.18 (2012)
5. Regil Vargas, L.: Hipermedia: medio, lenguaje herramienta del arte digital. Revista Digital Universitaria 6(10), 2–18 (2005)
6. Escobar Rodríguez, I.L.: Las bellas artes, <http://plasticas.dgenp.unam.mx/inicio/introduccion/bellas-artes>, Último acceso: 26/03/19 (2009)
7. Zalaya Báez, R.: Tesis doctoral: Escultura matemática “Antecedentes en la historia del arte, desarrollo, perspectivas de evolución y clasificación por conceptos matemáticos”. Universidad Politécnica de Valencia, España, pp. 152–163 (2005)
8. López García, J.C.: Algoritmos y programación. Fundación Gabriel Piedrahita Uribe, Colombia, pp. 96 (2009)
9. Pinales Delgado, F.J., Velázquez Amador, C.E.: Problemario de algoritmos resueltos con diagramas de flujo y pseudocódigo. Universidad Autónoma de Aguascalientes, México, pp. 172 (2014)
10. López, G., Jeder, I., Vega, A.: Análisis y Diseño de Algoritmos. Implementación en C y Pascal. Alfaomega, Argentina, pp. 336 (2009)
11. García, M.E.: Introducción a la informática. http://www.profmatisgarcia.com.ar/uploads/tutoriales/Ej_resueltos_algoritmos.pdf, último acceso 05/03/19 (2013)
12. Ramírez, C.: Metodología de programación, programación en C, aplicaciones electrónicas. https://www.academia.edu/14334848/algoritmos_y_programas, Último acceso: 05/03/19 (2011)
13. Veiga, L.A.: Tesis: Sistemas jerárquicos de tiempo real para adquisición de datos y control., http://sedici.unlp.edu.ar/bitstream/handle/10915/1350/Documento_completo.pdf?sequence=29, Último acceso: 13/04/19 (1999)

14. Balderrama Vásquez, C.: Algoritmos. http://moodle2.unid.edu.mx/dts_cursos_mdlic/IC/EA/AM/06/Algoritmos.pdf, Último acceso: 03/04/19 (2009)
15. Fogel, D.B.: What is evolutionary computation? *Spectrum, IEEE*, 37(2), 26, 28–32 (2000) & Fogel, D. B.: *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. IEEE Press, Piscataway, NJ. Third Edition (2006) <http://www.galeon.com/dantethedestroyer/algoritmos.pdf>, último acceso 15/04/19 (2010)
16. Goldberg, D.E: *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (1989)
17. Darwin, Ch: *Descent of Man*: Nuvision Publications (2007)
18. Gonzalo García, M.: Apuntes de: Teoría de Autómatas y Lenguajes Formales, ciencias.uis.edu.co/lenguajes/doc/chomsky.doc, último acceso 10/04/19 (2002)
19. Reyes Gómez, D.A.: Descripción y aplicaciones de los autómatas celulares. Universidad Autónoma de Puebla, Puebla, México, http://delta.cs.cinvestav.mx/~mcintosh/cellularautomata/Summer_Research_files/Arti_Ver_Inv_2011_DARG.pdf. Último acceso: 14/04/19 (2011)
20. Prins, G.-J., van Koolwijk, B.: Synchronator, <https://lullcec.org/2010/concerts/synchronator-gert-jan-prins-bas-van-koolwijk/>. Último acceso: 14/04/19 (2010)
21. Pighin, S.: *Informática Aplicada a la Ingeniería de Procesos I (Orientación I)*. Universidad Tecnológica Nacional, Facultad Regional, Rosario, Argentina, https://www.fro.utn.edu.ar/repositorio/catedras/quimica/5_anio/orientadora1/monograis/pighin-datamining.pdf. Último acceso: 14/04/19 (2001)
22. Rodríguez Mondragón, S.: Tesis: Sistema modular para conformación de escultura cerámica monumental. UAM, México (2014)
23. Rodríguez, S., Soto, L.J., Herrera, O.: Modelo de proceso para identificación visual a partir de íconos. EAE (2018)
24. Ordóñez Morales, F.J.: ¿Sueñan los androides con obras de Banksy? *Creatividad, arte e IA*. En: T3chFest 2019, Universidad Carlos III de Madrid, España, <https://www.youtube.com/watch?v=gNkjD1JSbw&feature=share>, Último acceso: 7/05/19, (2019)

Red neuronal convolucional con extracción de características multi-columna para clasificación de imágenes

Fidel López Saca, Andrés Ferreyra Ramírez, Carlos Avilés Cruz,
Juan Villegas Cortéz

Universidad Autónoma Metropolitana, Unidad Azcapotzalco, Ciudad de México,
México

fidelosmcc@gmail.com, {fra, caviles, juanvc}@correo.azc.uam.mx

Resumen. En los últimos años, las redes neuronales convolucionales han traído una evolución significativa en la clasificación de imágenes, estos métodos poderosos de aprendizaje automático extraen características de nivel inferior y las envían a las siguientes capas para identificar características de nivel superior que mejoran el rendimiento. En el presente artículo se propone una arquitectura de una red convolucional con un enfoque que logra incrementar el rendimiento de clasificación, es una estructura jerárquica que es fácil de construir, es adaptable y fácil de entrenar con un buen rendimiento en tareas de clasificación de imágenes. En la arquitectura propuesta, la imagen ingresa a la red por tres secciones de extracción de características que utilizan diferentes filtros de convolución, las características extraídas son combinadas y enviadas a capas totalmente conectadas para realizar la clasificación. Los resultados experimentales muestran que la arquitectura propuesta tiene un rendimiento superior a diferentes redes convolucionales tradicionales tales como AlexNet, GoogleNet y ResNet 152.

Palabras clave: redes neuronales convolucionales, clasificación, extracción de características.

Convolutional Neural Network with Extraction of Multi-Column Features for Image Classification

Abstract. Due to the immense amount of information available on the Internet, it causes users to feel overwhelmed with so much information, making it difficult to search for products and/or services that suit the tastes and needs of each user. For this reason the development of intelligent tools has become indispensable as are the Recommendation Systems, where its main objective is to help users find information of products and/or services in a better way filtering all the available information thus achieving a better use of it. In the present research work a Hybrid Recommendation Algorithm is designed and developed to create a list of recommended items (restaurants) for users (consumers), merging the

algorithms: Collaborative and Content Based Filter, using a Bayesian Classifier with techniques of Natural Language Processing. Besides, the user experience is improved by applying your GPS location as a filter to the recommendations. To measure the performance of the proposed system, we experimented with a set of data extracted from the Foursquare and TripAdvisor Websites.

Keywords: convolutional neural network, classification, feature extraction.

1. Introducción

En las últimas décadas, el crecimiento constante de imágenes digitales -como fuente principal de representación de la información para aplicaciones científicas- ha hecho de la clasificación de imágenes una tarea desafiante. Con el fin de alcanzar rendimientos de clasificación altos, se han propuesto diferentes técnicas de reconocimiento de patrones, entre las que se encuentran los métodos de aprendizaje profundo que hoy en día son un foco de estudio en el procesamiento de imágenes y visión por computadora. En este enfoque, la arquitectura más popular para la tarea de clasificación de imágenes son las redes neuronales convolucionales (CNNs, por sus siglas en inglés); una red construida de múltiples capas y en donde cada capa modela un campo receptivo de la corteza visual lo que la hace mucho más efectiva en tareas de visión artificial [11].

Las CNN combinan las características de bajo nivel dentro de características abstractas de alto nivel con transformaciones no lineales, lo que le permite tener la capacidad de aprender la representación semántica de las imágenes. Estas redes extraen características generalmente útiles de los datos con o sin etiquetas, detectan y eliminan redundancias de entrada y preservan solo aspectos esenciales de los datos en representaciones sólidas y discriminativas [2]; pueden capturar las características más obvias de los datos [19], por lo que podrían lograr mejores resultados en varias aplicaciones. A diferencia de las características creadas a mano, como SIFT [12] y HOG [3]; las características extraídas por la CNN se generan de extremo a extremo, lo que elimina la intervención humana. Las CNN tienen menos conexiones y parámetros lo que favorece que la extracción de características sea más eficiente.

La mayoría de arquitecturas de redes convolucionales tales como AlexNet [10], GoogleNet [16], VGG [15], ResNet 152 [8], y muchas otras [7], [18], [13], [6], [5], utilizan el mismo concepto para producir mapas de características en las capas de convolución, seguidas de capas de ‘pooling’ para reducir la dimensión de los mapas y conforme profundizan en la arquitectura, duplican el número de filtros para compensar la reducción a la mitad del tamaño de los mapas de características posteriores. La profundidad de la red favorece el rendimiento de clasificación y evita la desaparición de los gradientes mediante el uso de la inferencia de clases en capas de convolución consecutivas y la capa de pooling máximo, o el uso de capas softmax que realza el desvanecimiento de los gradientes [16], [15],[19]. Algunas de estas arquitecturas, utilizan nuevas funciones

de activación, métodos de regularización de actualización de pesos, inferencias de clase, entrenamiento previo por capas en enfoques supervisados; mostrado resultados muy prometedores [18], [5].

Aumentar el número de capas de una CNN significa aumentar la profundidad y el número de parámetros de la red; lo que complica el entrenamiento y reduce considerablemente el rendimiento, sobretodo con bases de datos pequeñas. Por otra parte, debido a la falta de características únicas, la fusión de características es cada vez mas importante para tareas como la clasificación y la recuperación de imágenes. Estas son técnicas que simplemente concatenan un par de características diferentes o utilizan métodos basados en el análisis de correlación canónica para la reducción de la dimensionalidad conjunta en el espacio de características.

En este artículo, se propone una red neuronal convolucional con extracción de características multi-columna para clasificación de imágenes. La red integra la capacidad de abstracción de las redes neuronales profundas y la capacidad de concatenar diferentes características. La red crece tanto en profundidad como en amplitud, la imagen a clasificar ingresa a través de tres secciones diferentes de extracción de características con diferentes filtros en las operaciones de convolución, las características extraídas son entonces concatenadas e ingresadas a capas totalmente conectadas que realizan la etapa de clasificación.

Los resultados muestran que la red tiene alto rendimiento en conjunto de imágenes como Oliva & Torralba [14], Stanford Dogs [9] y Caltech 256 [1]. Se realiza una comparación de la red propuesta con las redes existentes: AlexNet [10], GoogleNet [16] y ResNet [8], cuyas características están descritas en [4].

El documento está estructurado de la siguiente manera: En la sección 2 se describe el diseño de la red. La sección 3 analiza los conjuntos de datos empleados para los experimentos. En la sección 4 se describen los parámetros de entrenamiento. La sección 5 afirma la validez de la arquitectura propuesta mediante los experimentos realizados y los resultados experimentales. Finalmente en la sección 6, se discuten las conclusiones y el trabajo futuro.

2. Arquitectura propuesta

Aunque una sola CNN puede trabajar bien en problemas de clasificación de imágenes, puede presentar problemas para alcanzar precisiones altas en la fase de predicción. En la etapa de entrenamiento la red puede presentar un problema estadístico ya que el algoritmo de aprendizaje está buscando un espacio de parámetros (pesos y bias) que puede ser demasiado grande con respecto a la cantidad de datos de entrenamiento. En estos casos, puede haber muchos conjuntos de parámetros diferentes que produzcan la misma precisión, por lo que el algoritmo elige una de estas opciones; sin embargo, se corre el riesgo de que los parámetros elegidos no tengan buenas predicciones con las datos de prueba. En muchas aplicaciones, el algoritmo de aprendizaje de la CNN, no puede garantizar encontrar el mejor conjunto de pesos y bias; por lo que se produce un problema computacional, ya que se tiene que jugar con el ajuste de las variables de entrenamiento de la red, tales como: número de épocas, tasa

inicial de aprendizaje, tamaño del lote, etc. Por otra parte, cuando el espacio de soluciones no contiene un conjunto de parámetros que sea una aproximación correcta a la función objetivo verdadera, la red puede tener problemas en la representación de las características de las imágenes lo que dificulta aun más la clasificación y predicción. Los problemas mencionados anteriormente se pueden superar resolviendo el mismo problema de clasificación con varias CNN con arquitectura diferente y combinando los resultados. Esta expectativa nos llevó a proponer una arquitectura que en lugar de estar compuesta por diferentes CNN, esta diseñada con diferentes secciones de extracción de características; las cuales se combinan para realizar la clasificación. La arquitectura propuesta se muestra en la figura 1, está compuesta de 18 capas, las primeras quince son para la extracción de características y las últimas tres son para la clasificación. A continuación se hace una descripción de las capas de la red:

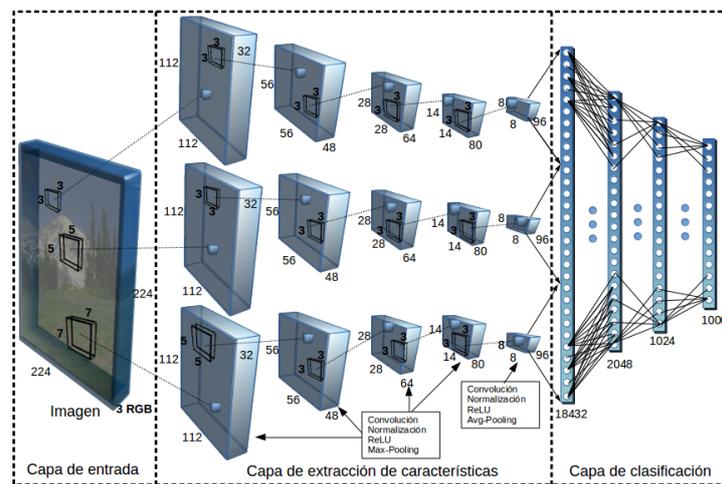


Fig. 1. Arquitectura general de la red neuronal propuesta.

- **Capa de entrada de la imagen:** esta capa establece una etapa de pre-procesamiento de las imágenes de entrada a la red. En esta capa las imágenes se pueden redimensionar, rotar e incluso se pueden tomar pequeñas muestras al azar. Esta capa esta diseñada para que la red acepte imágenes con dimensiones 224×224 con una profundidad de tres, ya que la imagen ingresa con el formato RGB (ver figura 1, parte izquierda).
- **Capas de extracción de características:** la imagen ingresa por tres diferentes secciones de extracción de características, cada una extrae diferentes características utilizando filtros de diferentes tamaños. La primera sección utiliza un filtro de 3×3 en sus cinco capas. La segunda sección utiliza en la primera capa un filtro de 5×5 y en las siguientes cuatro, filtros de 3×3 .

En la tercera sección, la primera capa utiliza un filtro de 7×7 en la segunda uno de 5×5 y en las últimas tres, filtros de 3×3 . Las primeras cuatro capas de cada sección de extracción utilizan convolución, normalización, ReLU y max-pooling con un filtro de 3×3 y paso de 2 con relleno. En la última capa de cada sección se utilizan convolución, normalización, ReLU y avg-pooling con un filtro de 7×7 , un paso de uno sin relleno (ver figura 1, parte central).

- **Clasificación:** la salida de cada sección de extracción de características se concatena para generar un vector unidimensional. Así se puede continuar con las capas totalmente conectadas para realizar la clasificación, como se describe a continuación (ver figura 1, parte derecha).
 - **Concatenación de las salidas de las 3 secciones de extracción de características:** la clasificación inicia con la concatenación de la salida de cada sección de extracción, se tendrá un vector con dimensiones 18432×1 (características de la imagen).
 - **Primera capa totalmente conectada:** tiene una profundidad de 2048 neuronas y es seguida de una capa de ReLU y una capa de Dropout.
 - **Segunda capa totalmente conectada:** consta de 1024 neuronas y es seguida de una capa de ReLU y una capa de Dropout.
 - **Tercera capa totalmente conectada:** esta capa es utilizada para ajustar la red convolucional a cada uno de los conjuntos de entrenamiento utilizados ya que es necesario igualar el número de neuronas de salida al número de clases de cada conjunto de entrenamiento. Esta capa es seguida de una capa de Softmax, una función de regresión, que ayuda a clasificar múltiples categorías.
- **Capa de salida:** es la capa final que se encarga de mostrar el porcentaje de éxito de clasificación.

Una red convolucional puede tener la tendencia a memorizar datos de entrenamiento, fenómeno conocido como sobreajuste, presentando porcentajes bajos de generalización. Para evitar esto, la red propuesta utiliza una capa de regularización comúnmente llamada “Dropout” en las primeras dos capas completamente conectadas. Las capas de Dropout hacen que la red sea más robusta a los datos de entrada imprevistos, y solo están activas durante la etapa de entrenamiento de la red, es decir, no están presentes durante la etapa de predicción.

3. Conjuntos de datos utilizados

En este trabajo se utilizaron 3 conjuntos de datos diferentes, los cuales se describen brevemente a continuación:

- **Oliva & Torralba [14]:** este conjunto de datos está compuesto por 2,688 imágenes de escenas a color que pertenecen a la misma categoría semántica. La base de datos tiene un total de 8 categorías y las imágenes fueron obtenidas de diferentes fuentes: bases de datos comerciales, sitios web y cámaras digitales.

- **Stanford Dogs** [9]: este conjunto de datos consta de 20,580 imágenes a color, pertenecientes a 120 clases o razas de perros de todo el mundo. Este conjunto de datos se ha creado utilizando imágenes y anotaciones de ImageNet para la tarea de categorización de imágenes de grano fino.
- **Caltech 256** [1]: este conjunto de datos consta de 30,607 imágenes a color pertenecientes a 256 categorías más una de nombre “clutter” que contiene múltiples escenas. Cada categoría contiene de 80 a 827 imágenes y la mayoría de las categorías tiene alrededor de 100 imágenes.

En la tabla 1 se muestran algunas estadísticas de los conjuntos de datos.

Tabla 1. Conjuntos de datos utilizados y sus estadísticas.

Conjunto de datos	Clases	Dimensiones			No. imágenes		
		Ancho	Alto	Prof	Total	Min x clase	Max x clase
Oliva & Torralba	8	256	256	3	2,688	260	410
Stanford Dogs	120	200 – 500	150 – 500	3	20,580	148	252
Caltech 256	257	300	200	3	30,607	80	827

4. Parámetros de entrenamiento

Tanto la red propuesta como las redes con las que se hace la comparación: AlexNet, GoogleNet y ResNet, fueron entrenadas utilizando el algoritmo de optimización ADAM (adaptive moment estimation), con un tamaño de lote de $\beta = 32$ imágenes y un decaimiento de pesos (factor de regularización) de $\lambda = 0,0005$. Los pesos iniciales en cada una de las capas fueron inicializados con una distribución gaussiana con una media de 0 y una desviación estándar de 0.01. Los umbrales de activación en cada una de las capas fueron inicializados a cero. Se inició con una tasa de aprendizaje de $\mu = 0,001$ la cual se disminuyó en un factor de 10 después de cada 50 épocas, para tener cambios de aprendizaje más específicos en 250 épocas de entrenamiento. Las redes fueron entrenadas en 4 GPUs NVIDIA GTX 1080, con 8 GB de memoria RAM y 2560 núcleos, con sistema operativo Linux Ubuntu 16.04, Linux kernel 4.12, Python 2.7, TensorFlow 1.12, NVIDIA CUDA®8.0, NVIDIA cuDNN v5.1.

La falta de una cantidad suficiente de imágenes de entrenamiento, el desequilibrio de imágenes por clase y el balance desigual de clases dentro de cada conjunto de datos, puede provocar que una red se sobreajuste. Para hacer a las redes más robustas e invariantes a transformaciones en los datos, mejorar la eficiencia del proceso de entrenamiento e incrementar la generalización; utilizamos el método de aumento de datos como una forma de crear imágenes nuevas con diferentes orientaciones [17].

Ya que las bases de datos contienen imágenes de diferentes tamaños, se hace un recorte aleatorio de la imagen de entrenamiento, el recorte es del mismo



Fig. 2. Ejemplo de imágenes modificadas para el ingreso a la CNN.

tamaño que acepta la capa de entrada de la red; es decir, de $224 \times 224 \times 3$. La imagen de entrada es reflejada de manera horizontal, de izquierda a derecha, con un 50 % de probabilidad. El brillo y el contraste de la imagen se ajustan de manera aleatoria en intervalos del 63 % y 0,2 – 1,8, respectivamente. Finalmente la imagen es normalizada para que la red tenga cierta independencia de las propiedades de la imagen. En la figura 2, se muestran algunos ejemplos de imágenes modificadas con el aumento de datos.

5. Experimentos y resultados

Para evaluar el rendimiento, las redes se entrenaron desde cero con cada una de las bases de datos y se utilizó aumento de datos. Los conjuntos de entrenamiento fueron formados con el 70 % de las imágenes de cada base de datos y el 30 % restante se utilizó para formar los conjuntos de prueba; la selección de las imágenes se realizó de manera aleatoria. Para separar los conjunto de entrenamiento y prueba, se utilizó el toolbox desarrollado en [4], para crear archivos de tipo tfrecord para evitar el desbordamiento de la memoria al realizar el entrenamiento. En la tabla 2 se muestran las características de cada conjunto de datos utilizadas para realizar los experimentos en este proyecto.

Tabla 2. Conjuntos de imágenes separadas en entrenamiento y pruebas, utilizando el 70 % de imágenes por clase para entrenamiento y el 30 % para pruebas.

Conjunto de datos	Clases	Cantidad de imágenes		
		Entrenamiento	Prueba	Total
Oliva & Torralba	8	1, 879	809	2, 688
Stanford Dogs	120	14, 358	6, 222	20, 580
Caltech 256	257	21, 314	9, 293	30, 607

Para describir la precisión de las redes en la tarea de clasificación, utilizamos los términos **top-1** y **top-5**. El número **top-1** es la cantidad de veces que la

etiqueta correcta tiene la probabilidad más alta predicha por la red. El número **top-5** es la cantidad de veces que la etiqueta correcta está dentro de las 5 clases principales predichas por la red.

5.1. Evaluación de la red propuesta

En la figura 3 se muestran los resultados de entrenamiento de la red propuesta, la cual a partir de este momento llamaremos ToniNet para diferenciarla de las otras redes. Note que la red llega a una exactitud del 100 % para el conjunto de entrenamiento Oliva & Torralba, mientras que para los conjuntos Stanford Dogs y Caltech 256 alcanza solo el 98 %.

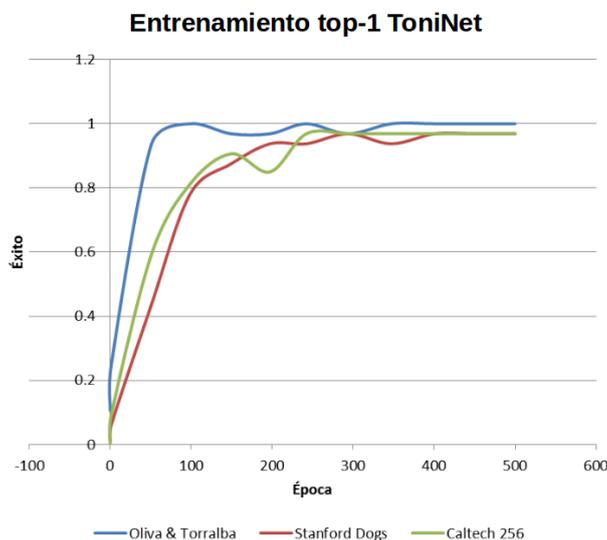


Fig. 3. Resultados de precisión top-1 en el entrenamiento de la red ToniNet.

Para evitar el sobre ajuste de la red ToniNet, se incluyeron capas de regularización (Dropout) en las dos primeras capas completamente conectas y se utilizó el aumento de datos. En la figura 4, se muestra los resultados obtenidos en la evaluación del sobreajuste de la red. Note que el error en la fase de entrenamiento disminuye de manera constante conforme aumenta el número de épocas, lo que refleja el crecimiento constante del rendimiento (Exactitud) de aprendizaje de la red.

5.2. Comparación

En la tabla 3 se muestra un resumen de los resultados para la etapa de prueba de las redes AlexNet, GoogleNet, ResNet 152 y ToniNet. Note que

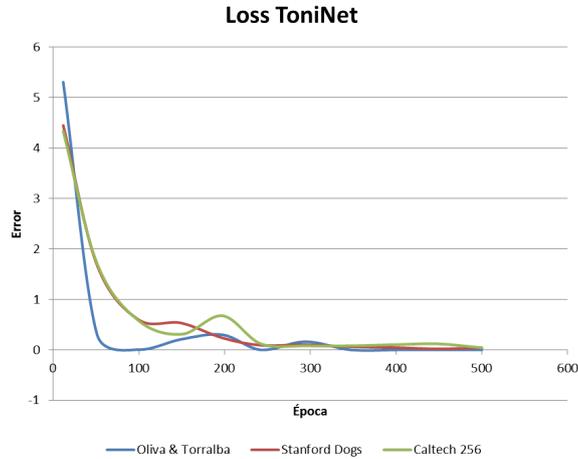


Fig. 4. Error de entrenamiento para la red ToniNet.

ToniNet supera a AlexNet y a GoogleNet tanto en top-1 como en top-5 para los 3 conjuntos de prueba; sin embargo, ToniNet es superada por ResNet 152 solo con el conjunto de prueba Caltech 256. GoogleNet supera en precisión a AlexNet ya que es una red con mayor profundidad; es decir, es una red que cuenta con un número mayor de capas de extracción; por la misma razón, ResNet 152 supera a GoogleNet. Sin embargo, aunque ResNet 152 también tiene una profundidad mayor a ToniNet, está solo supera a ToniNet en precisión con Caltech 256; lo que enfatiza tanto la ventaja como la importancia de tener tres secciones de extracción de características en paralelo idénticas pero con filtros de diferentes tamaños.

Tabla 3. Resultados y comparación de redes

CNN	Oliva & Torralba			Stanford Dogs			Caltech 256		
	Minutos	Top-1	Top-5	Minutos	Top-1	Top-5	Minutos	Top-1	Top-5
AlexNet	19	90,4 %	99,8 %	175	43,2 %	70,8 %	213	57,5 %	74,7 %
GoogleNet	14	91,5 %	99,8 %	164	51,9 %	80,2 %	196	60,6 %	79,0 %
ResNet 152	85	92,8 %	99,8 %	538	53,6 %	82,7 %	739	64,7 %	81,8 %
ToniNet	34	94,6 %	100 %	255	55,2 %	84,7 %	371	62,5 %	80,8 %

En la tabla 3 también se muestran los tiempos de entrenamiento de las redes. GoogleNet es la red con el menor tiempo de entrenamiento seguida de AlexNet, ToniNet y ResNet 152. Tomando como referencia el mayor tiempo de entrenamiento, que corresponde a ResNet 152, para las tres bases de datos podemos comentar que en promedio: ToniNet tiene un tiempo de entrenamiento del 18.03 % mayor a AlexNet y un 21.3 % mayor a GoogleNet, pero un 54.13 % menor a ResNet 152.

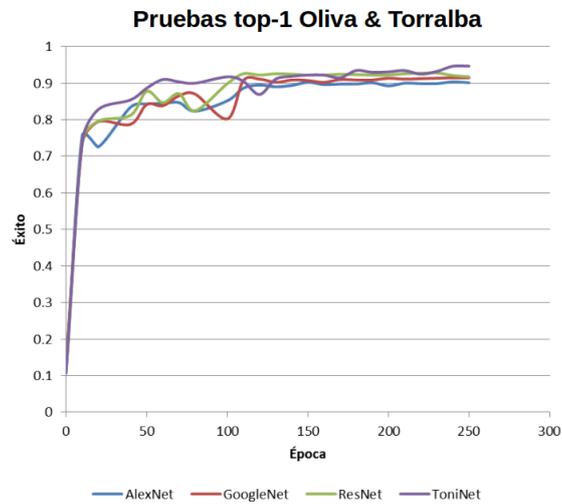


Fig. 5. Resultados de precisión top-1 (Oliva & Torralba).



Fig. 6. Matriz de confusión de la red ToniNet (Oliva & Torralba).

- Resultados con Oliva & Torralba:** Esta es la base de datos relativamente más simple que se utilizó, consta de 1,879 imágenes de entrenamiento y 809 de prueba pertenecientes a 8 clases. Analizando el comportamiento de

las redes en la fase de prueba, en la figura 5, podemos observar que las 4 redes aprenden muy bien y alcanzan una buena precisión top-1 en tan solo 150 épocas. Sin embargo, vale la pena mencionar que ToniNet muestra una mejor generalización durante toda la fase de aprendizaje y obtiene el mejor rendimiento con un 94,6 % de éxito en la época 250. Con esta base de datos ToniNet supera en 4,2% a AlexNet, en 3,1% a GoogleNet y en 1,8% a ResNet 256.

Para visualizar mejor el desempeño de ToniNet, en la figura 6 se muestra la matriz de confusión con C_1, C_2, \dots, C_8 que corresponden a las clases *Opencountry*, *Coast*, *Forest*, *Highway*, *Inside_city*, *Mountain*, *Street* y *Tallbuilding*, respectivamente. En la diagonal principal se observa que la red obtiene un 94,56% de éxito con 765 imágenes correctas de 809.

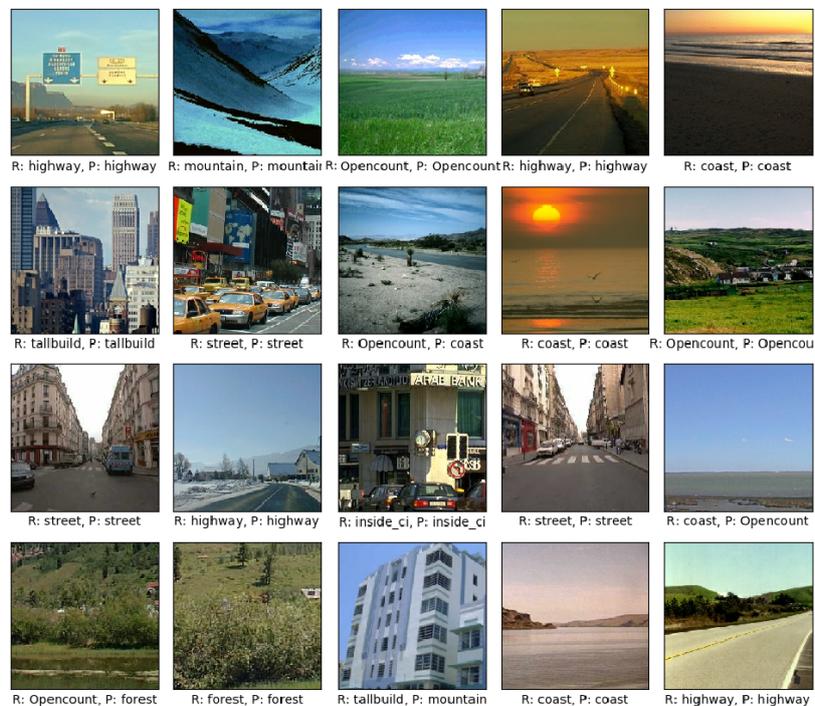


Fig. 7. Prueba con Oliva & Torralba, la letra *R* significa que es la clasificación real, la letra *P* significa que es la clasificación dada por la red.

La red tiene la mejor predicción con la clase *Tallbuilding* con el 96,19% seguida de *Forest* con el 96,04%.

La clase con menor éxito de predicción es *Inside_city* con 91,67%, confundándose con *Street* y *Tallbuilding*. La clase real *Highway* es la más confundida por la red con *Opencountry*, *Coast* y *Street*, también *Opencountry*

es confundida con *Coast*, *Forest* y *Mountain*. *Mountain* es la clase que mejor aprende la red. En la figura 7 se muestran algunos de los resultados en la fase de prueba, en donde se puede observar que la red ToniNet confundió una imagen de la clase *coast* por una de la clase *Opencountry*, entre otras que clasificó de forma incorrecta.

- **Resultados con Stanford Dogs:** Esta base de datos es mas complicada que Oliva & Torralba ya que el tamaño de las imágenes es muy variante, contiene 14,358 imágenes de entrenamiento y 6,222 de prueba pertenecientes a 120 clases.

Analizando el comportamiento de las redes en la fase de prueba, en la figura 8, podemos observar que ToniNet tiene el mejor rendimiento con un 55,2% de éxito alcanzado en 250 épocas y supera en 12% a AlexNet, en 3,3% a GoogleNet y en 1,6% a ResNet 256. En esta prueba, ToniNet se llegó a entrenar con 500 épocas logrando un 56,2% como máximo en rendimiento en prueba; sin embargo, el pequeño incremento en el rendimiento es poco significativo comparado al tiempo de entrenamiento.

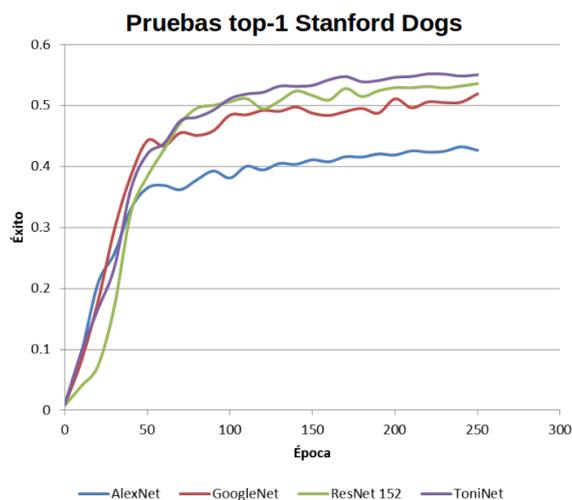


Fig. 8. Resultados de precisión top-1 (Stanford Dogs).

- **Resultados con Caltech 256:** Esta base de datos es más desafiante que Oliva & Torralba y Stanford Dogs, ya que tiene el mayor desequilibrio de imágenes por categoría, contiene 21,314 imágenes de entrenamiento y 9,293 de prueba pertenecientes a 256 clases. Analizando el comportamiento de las redes en la fase de prueba, en la figura 9, podemos observar que ResNet 152 obtuvo el mejor rendimiento con un 64,7% de éxito en 250 épocas, superando a ToniNet en un 2,2%. Con esta base de datos ToniNet supera en 5,0% a AlexNet, en 1,9% a GoogleNet.

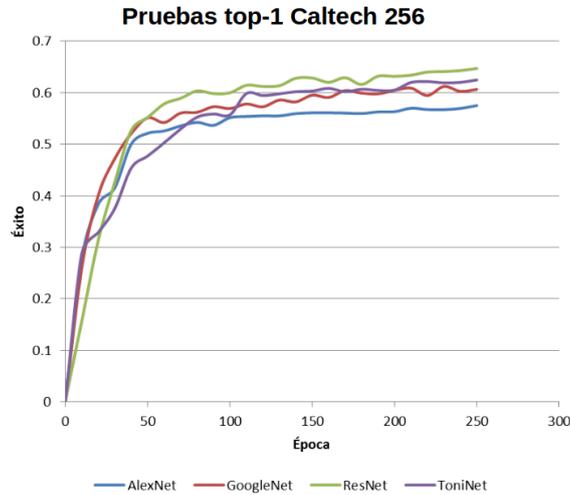


Fig. 9. Resultados de precisión top-1 (Caltech 256).

6. Conclusiones

La red propuesta dio mejores resultados con el conjunto de imágenes de Oliva & Torralba y StanfordDogs comparándola con AlexNet, GoogleNet y ResNet 152. Fue superada por ResNet 152 con el conjunto de imágenes Caltech 256, pero el tiempo de entrenamiento de nuestra red fue menor y con mayor rendimiento que AlexNet y GoogleNet.

La ventaja de la red con respecto a las comparadas; es que tiene diferentes secciones de extracción de características, con esto necesita menos capas, como por ejemplo: ResNet 152 y GoogleNet. La desventaja principal es que necesita más tiempo de entrenamiento que AlexNet y GoogleNet para lograr mejores resultados pero con menor tiempo que ResNet.

Se ponen las bases para la construcción de redes usando diferentes secciones para la extracción de características. Se pueden añadir más secciones dependiendo del hardware con el que se cuenta. Las redes profundas como ResNet con 152 capas extraen las características solo de una sección, en la red propuesta se pueden extraer características de múltiples secciones.

La red se puede mejorar en la etapa de clasificación, principalmente en las capas totalmente conectadas, también ampliando las secciones.

Como trabajo futuro se planea utilizar la red para la detección de objetos y segmentación de imágenes. También se planea experimentar con más secciones; para trabajar con el color, la forma, la textura, entre otras.

Referencias

1. Caltech256: Caltech 256 Dataset. www.vision.caltech.edu/ImageDatasets/Caltech256 (Mayo 2016)
2. Ciresan, D.C., Meier, U., Masci, J., Gambardella, L.M., Schmidhuber, J.: Flexible, high performance convolutional neural networks for image classification. In: Twenty-Second International Joint Conference on Artificial Intelligence (2011)
3. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: international Conference on computer vision & Pattern Recognition (CVPR'05). vol. 1, pp. 886–893. IEEE Computer Society (2005)
4. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE international conference on computer vision. pp. 1026–1034 (2015)
5. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence* 37(9), 1904–1916 (2015)
6. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. In: Proceedings of the 22nd ACM international conference on Multimedia. pp. 675–678. ACM (2014)
7. Kaiming He, Xiangyu Zhang, S.R.J.S.: Deep residual learning for image recognition. *IEEE Xplore* (2015)
8. Khosla, A., Nityananda, Jayadevaprakash, Yao, B., Fei-Fei, L.: Stanford Dogs Dataset. <http://vision.stanford.edu/aditya86/ImageNetDogs/> (Septiembre 2017)
9. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)
10. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. *Nature* 521, 436–44 (05 2015)
11. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *International journal of computer vision* 60(2), 91–110 (2004)
12. López, F., Ferreyra, A., Avilés, C., Villegas, J., Zúñiga, A., Rodríguez, E.: Preprocesamiento de bases de datos de imágenes para mejorar el rendimiento de redes neuronales convolucionales. *Research in Computing Science* 147(7): Robotics and Computer Vision, 35–45 (2018)
13. Mairal, J., Koniusz, P., Harchaoui, Z., Schmid, C.: Convolutional kernel networks. In: Advances in neural information processing systems. pp. 2627–2635 (2014)
14. Oliva, Torralba: . <http://cvcl.mit.edu/database.htm> (Mayo 2016)
15. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014)
16. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1–9 (2015)
17. Taylor, L., Nitschke, G.: Improving deep learning using generic data augmentation. *CoRR abs/1708.06020* (2017), <http://arxiv.org/abs/1708.06020>
18. Wan, L., Zeiler, M., Zhang, S., Le Cun, Y., Fergus, R.: Regularization of neural networks using dropconnect. In: International conference on machine learning. pp. 1058–1066 (2013)
19. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: European conference on computer vision. pp. 818–833. Springer (2014)

Clasificación de eventos académicos a partir de su descripción textual

Ariadna Gutiérrez-Rosales, José A. Reyes-Ortiz, Maricela Bravo

Universidad Autónoma Metropolitana, División de Ciencias Básicas e Ingeniería,
Departamento de Sistemas, Azcapotzalco, Ciudad de México, México
ariadna.gtzt08@gmail.com, {jaro, mbc}@azc.uam.mx

Resumen. El mundo se ha introducido cada vez más en la era inteligente y con ello surgen los espacios inteligentes, donde predecir los eventos que suceden en él no es una labor sencilla para lo cual se necesita brindarle inteligencia al medio para que éste sea capaz de anticipar eventos por suceder y proporcionar los servicios adecuados al usuario de acuerdo con sus necesidades. La clasificación de eventos académicos es deseable ya que permite predecir acontecimientos, por mencionar algunos ejemplos: visitas, reuniones, seminarios, cursos académicos y asesorías. En los espacios académicos, los métodos de clasificación pueden considerar eventos pasados y el tiempo en que ocurrieron, tales modelos dotarían al espacio académico de cierto grado de inteligencia para actuar sobre algunas situaciones o decisiones a futuro. Los eventos a clasificar en este artículo están relacionados con docencia, investigación y difusión de la cultura, pertenecientes a cuatro clases: evento de difusión, evento ambiental, evento de cursos académicos y evento de asesoría. Esta clasificación tiene como objetivo determinar el tipo de evento que sucede dentro del espacio académico, para ello se evalúan cuatro de los principales modelos de clasificación más utilizados en la literatura (Naïve Bayes (NB), K-Nearest-Neighbors (KNN), C4.5 y Support Vector Machine (SVM)) y se expone cuál es el más adecuado para un espacio académico.

Palabras clave: eventos académicos, clasificación, aprendizaje automático, procesamiento de lenguaje natural.

Academic Event Classification from Textual Descriptions

Abstract. The world has been introduced more and more into the intelligent age and with it intelligent spaces arise, where to predict the events that take place in it is not a simple task for which it is necessary to provide intelligence to the environment so that it is capable of anticipating events to happen and provide the appropriate services to the user according to their needs. The classification of academic events is desirable since it allows predicting events, to mention a few examples: visits, meetings, seminars, academic courses and consultancies. In academic spaces, classification methods can consider past events and the time they occurred, such models would give the academic space a certain degree of intelligence to act on some situations or decisions in the future. The events to be

classified in this article are related to teaching, research and dissemination of culture, belonging to four classes: dissemination events, environmental events, academic courses and advice. This classification aims to determine the type of event that happens within the academic space, for which four of the main classification models most used in the literature are evaluated (Naïve Bayes (NB), K-Nearest-Neighbors (KNN), C4.5 and Support Vector Machine (SVM)) and which is best suited for an academic space.

Keywords: academic events, classification, machine learning, natural language processing.

1. Introducción

El mundo se ha introducido cada vez más en la era inteligente y con ello surgen los espacios inteligentes, donde identificar los eventos que suceden en él no es una labor sencilla, motivo por el cual se necesitan enfoques computacionales para que éste sea capaz de anticipar eventos por suceder y proporcionar los servicios adecuados al usuario de acuerdo con sus necesidades. A pesar de que el término ambiente inteligente es utilizado principalmente en “casas inteligentes” es posible extender su aplicación de estudio a los espacios académicos.

La clasificación, es una de las principales tareas del aprendizaje automático, ofrece información que puede ser utilizada para la toma de decisiones, y eliminación de tareas manuales y repetitivas. Algunos ejemplos en los que se utiliza la clasificación son: la medicina, detección de fraude y seguridad, sistemas de recomendación, identificación de correo no deseado, y ambientes inteligentes. Es este último, donde se enfoca este trabajo.

La clasificación de eventos académicos es deseable ya que permite predecir acontecimientos, por mencionar algunos ejemplos: visitas, reuniones, seminarios, cursos académicos y asesorías. En los espacios académicos, los métodos de clasificación pueden considerar eventos pasados y el tiempo en que ocurrieron, tales modelos dotarían al espacio académico de cierto grado de inteligencia para actuar sobre algunas situaciones o decisiones a futuro. Los eventos a clasificar en este artículo están relacionados con docencia, investigación y difusión de la cultura, pertenecientes a cuatro clases: evento de difusión, evento ambiental, evento de cursos académicos, y evento de asesoría.

Esta clasificación tiene como objetivo determinar el tipo de evento que sucede dentro del espacio académico, para ello se evalúan cuatro de los principales modelos de clasificación más utilizados en la literatura (Naïve Bayes (NB), K-Nearest-Neighbors (KNN), C4.5 y Support Vector Machine (SVM)) y se expone cuál es el más adecuado para un espacio académico.

El resto del trabajo se organiza como sigue. En la Sección 2 se presentan los trabajos relacionados con la clasificación de eventos basado en sus descripciones textuales. La Sección 3 expone el enfoque utilizado para la clasificación de eventos que incluye los algoritmos de aprendizaje automático. La Sección 4, presenta los resultados obtenidos con los diversos algoritmos y combinando las características textuales extraídas. Finalmente, las conclusiones y el trabajo a futuro son presentado en la Sección 5.

2. Trabajos relacionados

En esta sección se describe el trabajo realizado en el área de clasificación de eventos basada en información no estructurada como sus descripciones textuales. Además, se explora el uso de algoritmos de aprendizaje automático para dicha tarea, así como el uso de recursos externos como las ontologías.

Con respecto a la clasificación de eventos utilizando textos, existen trabajos que han utilizado las redes sociales como su fuente de información. En [1] se presenta un enfoque tradicional basado en “Bolsa de palabras” para la clasificación de mensajes de Twitter en diversas categorías entre las que destacan los eventos y noticias. [2] expone un enfoque para clasificar mensajes de la red social en dos categorías mensajes sobre eventos del mundo real y mensajes que no son eventos; los autores utilizan la técnica de clasificación en línea y agrupamiento basado en tópicos junto con características textuales.

Un sistema para la extracción y clasificación de eventos en un dominio abierto a partir de Twitter es presentado en [3]. Los autores proponen un enfoque basado en modelos de variables latentes que descubren un conjunto apropiado de tipos de eventos que coinciden con los datos. Los eventos descubiertos automáticamente se inspeccionan posteriormente para filtrar los que son incoherentes y el resto se anota con etiquetas informativas, algunas como: finanzas, educación, religión, deportes y política. El conjunto resultante de clases de eventos se aplica luego para categorizar cientos de millones de eventos reales extraídos de manera automática.

En [4] se propone un enfoque no supervisado para explorar eventos a partir de Twitter, el cual consiste en un proceso de filtrado, extracción y categorización de eventos. En la etapa de filtrado el ruido de los tweets es eliminado, mientras que para la extracción se utiliza un lexicón para separar los tweets de aquellos que no son relevantes. Finalmente, para la categorización, los tweets son representados en vectores de características textuales y un modelo Bayesiano es utilizado para clasificar los eventos sin el uso de datos etiquetados.

En el dominio de la bioinformática, la clasificación de eventos a partir de textos médicos ha sido de gran ayuda para la identificación y extracción automática de eventos adversos, como en [5], donde se utilizan un método de aprendizaje automático para la detección efectiva de eventos en biomedicina; en [6] que extraen las relaciones entre medicamentos y efectos adversos como eventos a partir de literatura médica. En [7] se presenta un sistema que extrae seis tipos de eventos (pruebas, problema, tipo de diagnóstico, tratamiento, evidencias y ocurrencias) a partir de notas médicas, utilizando características semánticas como nombres de medicamentos, tratamientos, enfermedades, síntomas y regiones anatómicas extraídas del conjunto de datos utilizado como entrenamiento.

Finalmente, el uso de ontologías para apoyar la minería de textos en biomedicina, se presenta en [8], donde exponen un enfoque basado en reglas de decisión para la extracción y clasificación de eventos y hechos. Las ontologías ayudan a la identificación de características semánticas como el reconocimiento de entidades nombradas.

Con la revisión de los trabajos relacionados, se puede notar que la mayoría de los esfuerzos se centran en dominios como la medicina y utilizando textos en inglés extraídos de redes sociales y literatura científica. Con ello, es evidente la necesidad de

un enfoque para la clasificación de eventos académicos utilizando textos en español, como lo presenta este trabajo de investigación.

3. Clasificación de eventos

El proceso de clasificación de eventos, involucra una serie de etapas que a continuación se enumeran en la Figura 1.

1. Recopilar datos y formación del conjunto de datos.
2. Limpieza y transformación de los eventos (Selección de datos).
3. Minería de datos (Seleccionar el método de minería): Clasificación.
4. Evaluación e interpretación del método.

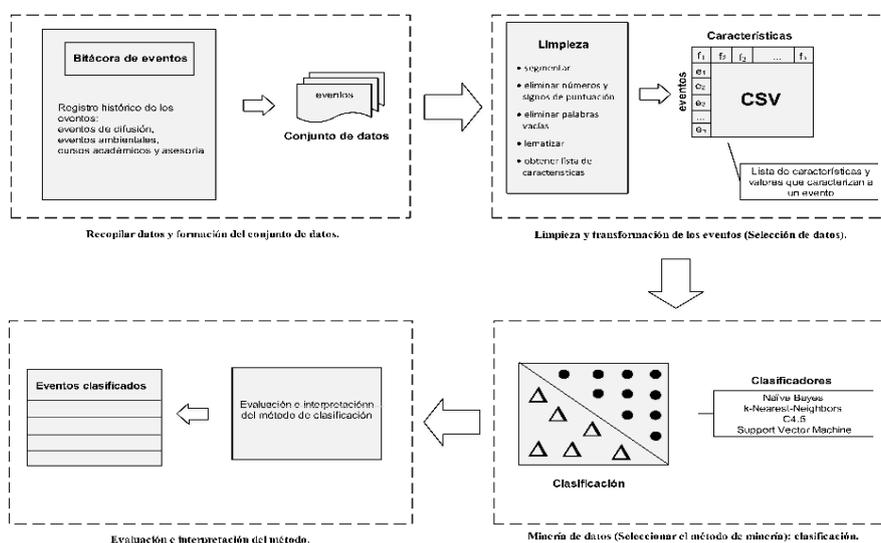


Fig. 1. Proceso de clasificación de eventos.

3.1. Recopilar datos y formación del conjunto de datos

La etapa de recopilación de datos parte de una serie de bitácoras las cuales tienen como objetivo mantener el registro histórico de los eventos ocurridos en el espacio académico: eventos de difusión, cursos académicos, asesorías y eventos ambientales. Estas bitácoras son obtenidas a partir de la implementación de un programa que, de manera automática realiza la simulación de eventos aleatorios ocurridos en un espacio académico. Los datos que se almacenan en las bitácoras de eventos son: el tipo participante, el horario y el lugar en el que se registra el evento, así como el nombre y descripción del evento.

Se describe la implementación de un método para la extracción automática y el análisis de estos eventos, que se almacenan en archivos de texto plano y forman el conjunto de datos de entrada, en una etapa siguiente, serán representados como un conjunto de características morfológicas de cada evento.

3.2. Limpieza y transformación de los eventos (selección de datos)

La limpieza y transformación depende de la recopilación de datos pues se realiza la traducción (transformación) de los eventos almacenados en las bitácoras de eventos a la lista de características y valores que caracterizan a un evento para ello es necesario un proceso de limpieza de los datos. Los eventos se representan por pares del tipo EVENT: FEATURE (evento: característica). Las características para cada EVENT: FEATURE se representan utilizando el modelo espacio vectorial como en [9], donde las características se representan numéricamente.

En este modelo es muy común representar a los elementos en una tabla, las filas representan los eventos y las columnas f_i representan las características de cada evento.

Las características f_i de cada evento se representan por el conjunto de todas las características:

$$F = \{f_1, f_2, f_3 \dots f_n\}. \quad (1)$$

Los eventos son el conjunto de todos los eventos:

$$E = \{e_1, e_2, e_3 \dots e_n\}. \quad (2)$$

3.3. Minería de datos (seleccionar el método de minería): Clasificación

Las técnicas de minería de datos se clasifican en dos categorías: supervisadas o predictivas y no supervisadas o descriptivas [10]. En esta fase es donde se decide cuál es la tarea (clasificación) a realizar y las técnicas descriptivas o predictivas a utilizar (seleccionar el método de clasificación). A continuación, se describen las utilizadas en este trabajo.

3.3.1. Clasificación bayesiana

Los clasificadores Bayesianos son clasificadores estadísticos, que pueden predecir tanto las probabilidades del número de miembros de clase, como la probabilidad de que una muestra dada pertenezca a una clase particular. La clasificación Bayesiana se basa en el teorema de Bayes [10].

Clasificadores como Naïve Bayes [11] permiten simplificar el coste computacional del modelo probabilístico, sin pérdida de expresividad por parte del mismo demostrando una alta exactitud y velocidad cuando se han aplicado a grandes bases de datos.

La teoría de la probabilidad y los métodos bayesianos son uno de los principales enfoques utilizados en el aprendizaje automático y la minería de datos; las razones por las que estos métodos resultan importantes son:

- Los métodos bayesianos permiten hacer inferencias a partir de los datos, formular hipótesis sobre nuevos valores, y además permiten calcular explícitamente la probabilidad asociada a cada una de las hipótesis posibles.
- Facilitan el trabajo para el análisis de numerosas técnicas de aprendizaje y minería de datos que no trabajan explícitamente con probabilidades.

Naïve Bayes [11] es el modelo más simple de clasificación en redes bayesianas. Su principal característica es que supone que todos los atributos son independientes esto

da lugar a un modelo gráfico probabilístico en el que existe un único nodo raíz (clase), y en la que todos los atributos son nodos hoja en donde el único nodo padre es la clase.

El clasificador Naïve (ingenuo) Bayes [11], es utilizado cuando se quiere clasificar un ejemplo descrito por un conjunto de atributos (x_i 's) en un conjunto finito de clases (c) de acuerdo con el valor más probable dados los valores de sus atributos [10] por lo tanto el objetivo de este clasificador es encontrar la clase óptima para un determinado evento, calculando la clase que da la probabilidad posterior máxima.

3.3.2. Aprendizaje basado en ejemplos

La clasificación basada en ejemplos se realiza por medio de una función que mide la proximidad o parecido con los ejemplos existentes usando una métrica de distancia y los ejemplos más próximos son utilizados para asignar la clase a la nueva instancia [12].

El clasificador k vecinos más próximos [13], es un método de aprendizaje perezoso basado en ejemplares, se basa en el modelo de espacio vectorial, el cual representa un conjunto de vectores de la forma $(a_1(x), a_2(x), \dots, a_n(x))$ en la que $a_r(x)$ es el valor de la instancia para el atributo a_r . El algoritmo KNN [13] procura por todos los ejemplos de entrenamiento comparar la similitud entre sus vectores de características, para encontrar los k ejemplos de entrenamiento más próximos y el ejemplar desconocido es designado a los k vecinos más próximos con mayor valor de clasificación.

La principal ventaja de este algoritmo es su facilidad de implementación, pero su costo computacional es alto cuando el tamaño de las instancias usadas en el entrenamiento crece. El mejor valor de k depende del conjunto de datos y del dominio de la aplicación.

3.3.3. Árboles de decisión

El árbol de decisión es una estructura en árbol, donde cada nodo representa un atributo a ser probado; las ramas representan la salida de la prueba y los nodos finales (hojas) representan la clasificación.

El algoritmo de árboles de decisión posee dos fases principales: en la primera llamada fase de crecimiento del árbol, el algoritmo inicia con todo el conjunto de datos como nodos raíz. Los datos son divididos en subconjuntos utilizando algún criterio de división. En la segunda fase, etapa de poda del árbol, el árbol total formado se poda para prevenir el exceso de ajuste (*over-fitting*) del árbol a los datos de entrenamiento.

Existen diversos algoritmos para construir árboles de decisión entre ellos ID3 [14], C4.5 [15], SPRINT [16], SLIQ [17] y PUBLIC [18]. El utilizado en este trabajo es el algoritmo C4.5 [15] que incluye diversos métodos para trabajar con atributos numéricos, valores ausentes, datos con ruidos y para generar reglas a partir de árboles de decisión.

3.3.4 Máquinas de soporte vectorial (SVM)

Máquinas de Soporte Vectorial [19] es un método de aprendizaje supervisado con un alto grado de clasificación, su funcionamiento está basado en la clasificación lineal separando los datos en dos clases. El algoritmo pretende encontrar el hiperplano que maximiza el margen entre los vectores de soporte que define la posición del hiperplano

ideal. La ventaja de utilizar este método es su buen desempeño cuando se cuenta con un gran número de características y también cuando se tienen pocos elementos de entrenamiento en tareas de múltiples clases [19].

3.4. Evaluación e interpretación del método

En esta etapa se evalúa y se validan las conclusiones obtenidas comparando los modelos y se determina cuál ofrece mejores resultados de clasificación, para lograr esto se realizó la representación de eventos basada en el modelo vectorial y el esquema de pesado TF-IDF (Frecuencia del Término - Frecuencia Inversa del Término). TF - IDF es la unión del esquema de pesado TF (Frecuencia del Término) [20] con IDF (Frecuencia Inversa del Término) [21]. En TF-IDF [22] cada vector está conformado por los pesos que representan la relevancia que tiene una característica en un evento. De acuerdo con [22] aquellas características que ocurren con menor frecuencia se consideran más importantes que aquellas que ocurren con mayor frecuencia. Su fórmula se muestra en la ecuación 3:

$$tf - idf_{ij} = f_{ij} * \log\left(\frac{N}{df}\right), \quad (3)$$

donde f_{ij} es la frecuencia de la característica i en el evento j , N es el número de descripciones y df es el número de descripciones en donde aparece el término i .

La precisión, el recuerdo y la exactitud son las métricas de evaluación más comunes en la evaluación de los algoritmos de clasificación, en este trabajo se utiliza a la *precisión*. La *precisión* indica qué tan exacta fue la clasificación de los eventos, mientras que el recuerdo da a conocer si los eventos que pertenecen a una clase i , se clasificaron dentro de esa clase; la *exactitud* representa el porcentaje de las predicciones que son correctas. La fórmula de la *precisión* se muestra en la ecuación 4:

$$Precisión = \frac{\text{Número de eventos clasificados correctamente}}{\text{Total de eventos}}. \quad (4)$$

4. Experimentación y resultados

En esta sección se presentan la experimentación y los resultados obtenidos con los algoritmos de aprendizaje supervisado presentados anteriormente. Además, del conjunto de datos utilizado para esta experimentación y su transformación para lograr la clasificación de eventos.

4.1. Conjunto de datos

En la etapa de recopilación de datos y formación del conjunto de datos, a partir de una serie de bitácoras se obtuvieron 363 eventos académicos de cuatro clases de eventos: eventos de asesoría, cursos académicos, eventos de difusión y eventos ambientales. De estos eventos son de interés los participantes del evento, el lugar y el horario en el cual se llevó a cabo cada evento y su variación en caso de que se trate de eventos ambientales, así como el nombre del evento y su descripción. En la Tabla 1 se muestran los cuatro tipos de eventos considerados, algunos de los cuales incluyen subtipos de eventos y sus descripciones correspondientes.

Tabla 1. Descripción de eventos.

Tipo	Descripción de evento	Tipos de eventos
Asesoría	Consulta que brinda un profesor a un estudiante para resolver cuestiones sobre temas que domina	
Cursos académicos	Su objetivo es la formación académica y profesional de estudiantes y profesores	Licenciatura, posgrado y actualización
Evento de difusión	Evento cuyo objetivo es difundir temas relacionados con la investigación y la cultura	Congreso, panel de discusión, taller, seminario y presentación
Ambiental	Evento en el cual se encuentran involucradas las variables del ambiente	Presencia, luminosidad, temperatura y humedad

4.2. Extracción de características

En esta etapa se realizó la representación de los eventos y características mediante el modelo espacio vectorial propuesto por Salton [23], para cada par se extraen trece características que se dividen de acuerdo con la información morfológica del evento: la(s) persona(s) participante(s) en el evento, tiempo en el que ocurre el evento, espacio en el cual ocurre dicho evento, y en el caso de los eventos ambientales, la variación, así como la clase de evento, además con el modelo “Bolsa de palabras”, se obtiene el conjunto de características lexicográficas que componen a un evento: verbos, adjetivos y sustantivos. Estas características se describen en la tabla 2.

La obtención de las características se realizó mediante un pre-procesado de las descripciones de los eventos. Este pre-procesado fue realizado con la herramienta NLTK de Scikit-learn [24] y el módulo Pattern desarrollado por el Centro de investigación CLiPs (Computational Linguistics & Psycholinguistics) [25] en el lenguaje de programación denominado Python.

Tabla 2. Características morfológicas de un evento.

Característica	Descripción
Características de agente	Definen al participante que inicia o participa en un evento
Características de tiempo	Describen el tiempo en el que ocurren los eventos
Características de espacio	Describen el espacio físico en el cual ocurren los eventos
Características de magnitud	Describen variables de ambiente en un evento ambiental
Características de clase	Describen el tipo de evento
Características lexicográficas	Definen las palabras más representativas de un evento

–**Pre-procesado.** En este paso se seleccionan los datos que serán utilizados en la clasificación. El pre-procesado en este trabajo consiste en segmentar, limpiar, eliminar palabras vacías, lematizar y obtener la lista de características (bolsa de palabras) de un evento.

–**Segmentar.** Tarea que consiste en obtener las cadenas delimitadas por un espacio en blanco.

–**Limpieza.** Proceso que descarta aquellos datos que no aportan información relevante al proceso de clasificación, estos datos son: números, signos de puntuación, caracteres

especiales y aquellas palabras que carecen de un significado por sí solas, denominadas palabras vacías (stop words), algunos ejemplos son: artículos, preposiciones y conjunciones. Este trabajo utiliza el módulo de stop words de *NLTK* para español.

–**Lematización.** Proceso mediante el cual se eliminan partes no esenciales de una palabra para obtener su forma base, este proceso implica un análisis morfológico de cada palabra en el que se identifica a través de un etiquetado automático (POS tagging) su categoría gramatical. Existen muchas herramientas que permiten realizar esta labor. Sin embargo, las pruebas realizadas a los eventos académicos mostraron ambigüedades en la asociación de una palabra con su categoría gramatical razón por la que se optó por la implementación de un módulo de lematización automático para español. Este módulo se desarrolló en *Python 2.7* como lenguaje de programación y la identificación de la categoría gramatical se realiza con la ayuda de *Pattern.es* [25] en su versión para el español. A la variación que sufre una palabra dependiendo de su género, número o tamaño se le conoce como flexión, en español forman flexión nominal los adjetivos, sustantivos y pronombres con los morfemas flexivos de género y número (masculino, femenino y singular o plural respectivamente), los verbos lo hacen con la conjugación. La asignación de categorías gramaticales de un verbo se realiza identificando sus diferentes formas verbales que dependiendo de esta se trasladan al infinitivo. La identificación de adjetivos se realiza a través de sus morfemas flexivos de género y número (-o, -a, -os, -as, -as o -es), según sea el género se obtiene su forma base y se singularizan. En el caso de los sustantivos se identifican sus morfemas de número (-s, -es), en plural para posteriormente singularizar, en su mayoría, los sustantivos son invariables (no cambian de género) son masculinos o femeninos, de estos, se descartan los sustantivos derivados de verbos (expresan acciones, eventos o procesos).

–**Bolsa de palabras.** El modelo "bolsa de palabras" (del inglés, Bag of Words) está compuesto por el conjunto de características lexicográficas obtenidas durante el proceso de lematización, cabe mencionar que en este modelo no se admiten términos repetidos. En la tabla 3 se muestra el listado de características utilizadas para la clasificación de eventos junto con su descripción.

Tabla 3. Conjunto de características.

Característica	Descripción	Posibles valores
Número de estudiantes	Total de alumnos participantes en un evento	Valor numérico que indica la cantidad de estudiantes en un evento
Número de profesores	Total de profesores participantes en un evento	Valor numérico que indica la cantidad de profesores en un evento
Número de visitantes	Total de participantes externos a un espacio académico	Valor numérico que indica la cantidad de visitantes en un evento
Total de participantes	Total de participantes en un evento	Valor numérico que indica la cantidad de estudiantes, profesores y visitantes en un evento
Horario inicial del evento	Rango de tiempo en el que sucede un evento	Valor nominal = { turno matutino = 1, turno vespertino = 2, turno intermedio = 3, turno nocturno = 4 }
Horario final del evento	Rango de tiempo en el que sucede un evento	Valor nominal = { turno matutino = 1, turno vespertino = 2, turno intermedio = 3, turno nocturno = 4 }

Tiempo del evento	Tiempo que tarda en desarrollarse un evento	Valor numérico que indica la duración en minutos de un evento
Tipo de espacio	Lugar en el que sucede un evento	Valor booleano que indica si es un espacio interior o al aire libre = {si = 0, no =1}
Tipo de lugar	Espacio físico en el que se desarrolla un evento	Valor nominal = {salón = 1, oficina de un profesor = 2, laboratorio = 3, auditorio = 4, plaza = 5, jardín = 6 }
Variación ambiental	Cambio en luminosidad, temperatura o humedad	Valor numérico que indica la variación en eventos de luminosidad, temperatura y humedad
Clase del evento	Clase a la que pertenece un evento	Valor nominal = {Evento de difusión = 0, Cursos académicos = 1, Asesoría =2, Evento ambiental = 3}
Nombre del evento	Denominación verbal en español que se le asigna a un evento	Valor nominal = {cadena}
Descripción del evento	Narración de corta extensión, en español que se hace sobre un evento	Valor nominal = {cadena}
Verbo	306	Valor numérico que indica el total de verbos en los eventos
Adjetivo	294	Valor numérico que indica el total de adjetivos en los eventos
Sustantivo	1233	Valor numérico que indica el total de sustantivos en los eventos

4.3. Clasificación de eventos

En esta sección se describen las pruebas realizadas con cuatro de los métodos de clasificación más usados en la literatura: Naïve Bayes, KNN, Árboles de decisión (C4.5) y Support vector Machine (SVM). Posteriormente se muestra el score obtenido para cada uno.

Para la realización de las diferentes pruebas se utilizó el esquema de pesado TF – IDF sobre cada conjunto de datos y se dividió en dos pequeños subconjuntos de eventos seleccionados aleatoriamente, el primer grupo corresponde al de entrenamiento con el 70% de los eventos y el segundo con el 30% restante para su evaluación.

4.4. Resultados

Las pruebas se realizaron de manera individual para las características lexicográficas (verbos, adjetivos y sustantivos) y haciendo una combinación de estas, además se hizo la combinación de las características lexicográficas con sus características nominales, por último, la combinación de todos, es decir, características lexicográficas: verbos, adjetivos, sustantivos y las características nominales de cada evento.

En el conjunto de pruebas llevadas a cabo se puede observar que Naïve Bayes y C4.5 muestran mejores resultados para el conjunto de verbos con una precisión del 67% mientras que C4.5 y SVM obtienen mayor precisión para los adjetivos con un 56% y un 63% respectivamente, Naïve Bayes y C4.5 nuevamente ofrecen mejores resultados para los sustantivos.

En el caso de las pruebas con combinaciones de características, se tiene que combinando verbos, adjetivos, sustantivos y características nominales se ha logrado un 94 % de precisión con el algoritmo C4.5. En el total de pruebas realizadas, se ha concluido que el algoritmo C4.5 obtiene los mejores resultados en la clasificación de eventos académicos como se observa en la Tabla 4 que expone los resultados de precisión obtenidos de la evaluación de cada uno de los algoritmos aplicados con los diferentes conjuntos de características.

Tabla 4. Resultados de precisión.

Conjunto de datos	NB	KNN	C4.5	SVM
Verbo	0.67	0.51	0.67	0.60
Adjetivo	0.39	0.44	0.56	0.63
Sustantivo	0.75	0.44	0.90	0.44
Verbo + Adjetivo + Sustantivo	0.77	0.44	0.88	0.44
Nominal + Verbo	0.78	0.91	0.87	0.57
Nominal + Adjetivo	0.78	0.91	0.91	0.52
Nominal + Sustantivo	0.80	0.85	0.90	0.44
Nominal + Verbo + Adjetivo + Sustantivo	0.79	0.85	0.94	0.67

5. Conclusiones

En este artículo se ha presentado un enfoque para la clasificación de eventos académicos utilizando algoritmos de aprendizaje automático basado en sus descripciones textuales. El enfoque que presenta consiste de una etapa de entrenamiento de los modelos de clasificación, donde se utilizan características textuales como la frecuencia de palabras y se hace uso de información morfosintáctica, como la categoría de las palabras (verbos, sustantivos, adjetivos). Los cuatro algoritmos utilizados son Naïve Bayes (NB), k vecinos más próximos (KNN), C4.5 y máquinas de soporte vectorial (SVM).

Las principales aportaciones de este trabajo son a) el conjunto de datos sobre eventos académicos etiquetados en cuatro categorías; b) el enfoque para la clasificación automática de eventos académicos basada en sus descripciones textuales en español; c) la comparación de diversos clasificadores combinándolos con diversos tipos de características.

Con la experimentación y resultados, se hace notar que la mejor configuración de experimentos es utilizando el algoritmo de árboles de decisión (C4.5) y haciendo uso de todas las características: nominales, verbos, adjetivos y sustantivos. Esta configuración ha logrado un 94 % de precisión en la tarea de clasificación de eventos académicos.

Los resultados de este trabajo son de gran utilidad para los analistas de eventos académicos, debido a que ellos realizan un análisis y categorización de este tipo de eventos de manera manual. El enfoque propuesto en este artículo apoyaría en disminuir los tiempos de análisis de eventos desde que propone un razonamiento automático a partir de sus descripciones textuales.

Como trabajo a futuro se propone la experimentación con eventos de otros dominios como la medicina, la política y seguridad. Además, se propone el modelo de n-gramas y n-gramas sintácticos, por su simplicidad e independencia de idioma. Un sistema de clasificación automática de eventos con comunicación directa con los usuarios, sería de gran utilidad para la comunidad que se dedica al análisis de eventos.

Referencias

1. Sriram, B., Fuhry, D., Demir, E., Ferhatosmanoglu, H., Demirbas, M.: Short text classification in twitter to improve information filtering. In: 33rd international Proceedings ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 841–842. ACM, New York (2010)
2. Becker, H., Naaman, M., Gravano, L.: Beyond trending topics: Real-world event identification on twitter. In: Fifth AAAI International Conference on Weblogs and Social Media (2011)
3. Ritter, A., Etzioni, O., Clark, S.: Open Domain Event Extraction from Twitter. In: 18th ACM SIGKDD International conference on Knowledge discovery and data mining, pp. 1104–1112. ACM, New York, NY, USA (2012)
4. Zhou, D., Chen, L., He, Y.: An unsupervised framework of exploring events on twitter: Filtering, extraction and categorization. In: Twenty-Ninth AAAI Conference on Artificial Intelligence (2015)
5. Miwa, M., Sætre, R., Kim, J. D., Tsujii, J. I.: Event Extraction with Complex Event Classification Using Rich Features. *Journal of Bioinformatics and Computational Biology*. 8(1), pp. 131–146 (2010)
6. Kang, N., Singh, B., Bui, C., Afzal, Z., van Mulligen, E. M., Kors, J. A.: Knowledge-Based Extraction of Adverse Drug Events from Biomedical Text. *BMC bioinformatics*. 15(1), 64 (2014)
7. Sohn, S., Waghlikar, K. B., Li, D., Jonnalagadda, S. R., Tao, C., Komandur Elayavilli, R., Liu, H.: Comprehensive Temporal Information Detection from Clinical Text: Medical Events, Time, and TLINK Identification. *Journal of the American Medical Informatics Association*. 20(5), pp. 836–842 (2013)
8. Spasic, I., Ananiadou, S., McNaught, J., Kumar, A.: Text Mining and Ontologies in Biomedicine: Making Sense of Raw Text. *Briefings in Bioinformatics*. 6(3), pp. 239–251 (2005)
9. Reyes, J. A., Montes, A., González, J. G., Pinto, D. E.: Clasificación de Roles Semánticos Usando Características Sintácticas, Semánticas y Contextuales. *J. Comp. y Sist.* 17(2), pp. 263–272 (2013)
10. Molina, J., García, J.: Técnicas de Minería de Datos basadas en Aprendizaje Automático. *Técnicas de Análisis de Datos*, pp. 96–66 (2008)
11. Friedman, N., Geiger, D., Goldszmidt M.: Bayesian Network Classifiers. *Mach. Learn.* 29, pp. 131–163 (1997)
12. Viera, A. F. G.: Técnicas de aprendizaje de Máquina Utilizadas para la Minería de Texto. *Investigación bibliotecológica*. 31(71), pp. 103–126 (2017)
13. Cover, T. M., Hart, P. E.: Nearest Neighbor Pattern Classification. *IEEE Trans. Inf. Theor.* 13, pp. 21–27 (1967)
14. Quinlan J. R.: Induction of Decision Trees, *J. Mach. Learn.* 1 (1), pp. 81–106 (1986)
15. Ross, J.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Francisco (1993)
16. Shafer, J.C., Agrawal, R., Mehta, M.: SPRINT: A Scalable Parallel Classifier for Data Mining. In: 22th International Conference on Very Large Data Bases, pp. 544–555. Morgan Kaufmann, San Francisco (1996)

17. Mehta, M., Agrawal, R., Rissanen, J.: SLIQ: A Fast Scalable Classifier for Data Mining. In: 5th International Conference on Extending Database Technology: Advances in Database Technology, Springer-Verlag, London, UK, pp. 18–32 (1996)
18. Rastogi, R., Shim, K.: PUBLIC: A Decision Tree Classifier that Integrates Building and Pruning. In: 24th International Conference on Very Large Data Bases, pp. 24–27. Morgan Kaufmann, San Francisco (1998)
19. Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J., Scholkopf, B.: Support Vector Machines. *IEEE Intelligent Systems*. 13(4), pp. 18–28 (1998)
20. Salton, G., McGill, M. J.: *Introduction to Modern Information Retrieval*. McGraw-Hill, New York (1986)
21. Robertson, S.: Understanding Inverse Document Frequency: on Theoretical Arguments for IDF. *Journal of documentation*. 60(5), pp. 503–520 (2004)
22. Salton, G., Yang, C. S.: On the Specification of Term Values in Automatic Indexing. *Journal of documentation*. 29(4), pp. 351–372 (1973)
23. Salton, G.: *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley Longman Publishing Co., Boston (1989)
24. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Vanderplas, J.: Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* 12, pp. 2825–2830 (2011)
25. Smedt, T. D., Daelemans, W.: Pattern for Python. *J. Mach. Learn. Res.*, 13, pp. 2063–2067 (2012)

Detección de enfermedades en el sector agrícola utilizando Inteligencia Artificial

Balzhojt Roldán Ortega, Rajesh Roshan Biswal, Eddy Sánchez DelaCruz

Instituto Tecnológico Superior de Misantla, Departamento de Posgrado,
Misantla, Veracruz, México

balzhojt@gmail.com, rjshbswl@gmail.com, eddsacx@gmail.com

Resumen. El presente artículo, expone varios trabajos de investigación sobre la posibilidad de detectar enfermedades en los cultivos agrícolas, utilizando técnicas de inteligencia artificial. Los autores de dichos trabajos, hacen propuestas de técnicas de obtención de características de las hojas o frutos de las plantas, así como también el uso de algoritmos clasificadores o de agrupación, todo esto con el fin de determinar si una hoja, presenta signos de alguna enfermedad. Al haber diversos tipos de enfermedades y diversas variedades de plantas, los autores hacen propuestas para utilizar el algoritmo que ellos consideran el que obtendrá mejores resultados. Al final concluimos que, sí es posible detectar enfermedades trayendo consigo un beneficio directo para el agricultor que la implemente, ya que un diagnóstico oportuno, daría una respuesta a la enfermedad y por lo tanto reducción del riesgo en pérdidas económicas.

Palabras clave: inteligencia artificial, reconocimiento de patrones, enfermedades, plantas.

Detection of Diseases in the Agricultural Sector using Artificial Intelligence

Abstract. The present article, exposes several works of investigation on the possibility of detecting diseases in the agricultural crops, using artificial intelligence techniques. The authors of these works, proposed different techniques for obtaining leaf or fruits characteristics of plants, using classification or clustering algorithms, in order to determine whether or not, a leaf showed signs of disease. Depending on the disease and the plant type, adequate algorithms with the optimum results were proposed. It can be concluded that, it is possible to detect leaf diseases, thereby benefitting the farmer directly. Moreover, a timely diagnosis would result in an effective treatment of the disease, thus avoiding the risk of any economic loss as result.

Keywords: artificial intelligence, pattern recognition, diseases, plants.

1. Introducción

Es bien sabido que la prevención y el diagnóstico oportuno de cualquier enfermedad nos traerá la ventaja estratégica sobre dicho padecimiento y en la agricultura no es la excepción, pues sabiendo lo que aqueja a un cultivo o planta, se incrementa las posibilidades de éxito en el tratamiento.

En el mundo en desarrollo, más del 80% de la producción agrícola es generada por pequeños agricultores [1], y los informes de pérdidas de rendimiento de más del 50% debido a plagas y enfermedades son comunes [2]. Además, la mayor proporción de personas con problemas de pobreza y hambruna (50%) vive en estas zonas productivas [3], lo que hace que los pequeños agricultores sean un grupo particularmente vulnerable a las interrupciones en el suministro de alimentos derivadas de patógenos.

Existen métodos para determinar las enfermedades de cualquier planta, como llevar muestras de tejido vegetativo a un laboratorio especializado o llevar a un ingeniero agrónomo experto al sitio del cultivo, en cualquiera de los dos métodos, la desventaja radica en el tiempo necesario para obtener los resultados.

Es por eso, que se han considerado el uso de técnicas de visión artificial y reconocimiento de patrones, así como algunos algoritmos de clasificación que de forma automática determinen la posible enfermedad, facilitando la tarea de los especialistas para desarrollar su trabajo y que puedan dar con un diagnóstico oportuno para su tratamiento. Y como dice Barbedo [4], las herramientas para el reconocimiento automático de enfermedades de las plantas tienen el potencial de convertirse en una valiosa fuente de información para ayudar a la toma de decisiones en la agricultura.

En la siguiente sección, se muestran las técnicas utilizadas, así como los trabajos de investigación más significativos, que atienden el desafío de la detección de una enfermedad, a partir del análisis de las características presentes en las hojas de los cultivos.

2. Técnicas y algoritmos empleados

Para llegar a la clasificación de las enfermedades, los autores se basan en metodologías ya probadas, como se muestra en la figura 1. Adquisición de los datos, preprocesamiento, extracción de características y reconocimiento, son los pasos o procedimientos que se siguen para obtener resultados asertivos.

Para la adquisición de las imágenes, se hace uso de cámaras digitales para capturar las hojas o las partes donde es visible el daño causado por la enfermedad, así como también conjuntos de imágenes disponibles en la web y que se han puesto a disposición de todo el público sirviendo como base para el entrenamiento del modelo. Para las imágenes adquiridas por cámara, las tomas fueron en ambientes controlados con resoluciones aceptables, aunque es importante mencionar que Bhange [6] utilizó un teléfono móvil, para adquirir las imágenes y hacer un experimento alterno para ver qué tanto afectaba la calidad de las mismas.

Una vez adquiridas las imágenes, pasan al preprocesamiento, en donde se les darán un tratamiento como el escalado, eliminación de ruido, transformación del espacio de color, ecualización de histograma y todo lo que se pueda hacer para maximizar las características. Para cuando pasa al siguiente proceso, la imagen va más limpia, por lo

tanto, al aplicarle las técnicas de segmentación, éstas separarán los puntos de interés con mejor precisión, obteniendo datos valiosos que serán más descriptivos de la enfermedad. Una vez teniendo las características, lo siguiente es hacer la clasificación, hacer uso de los algoritmos, obtener los resultados y hacer una descripción de los mismos. Es en este paso, cuando se muestra la eficacia del procedimiento, si se ha clasificado de acuerdo a lo esperado.

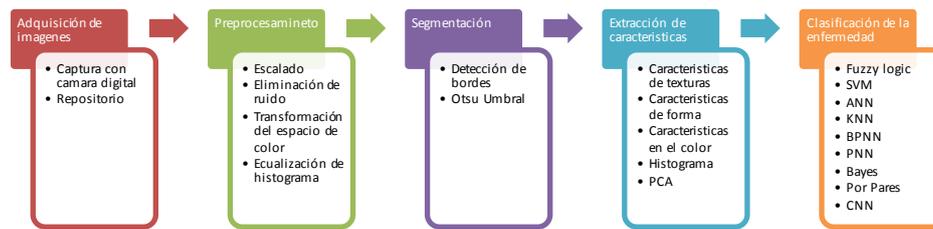


Fig. 1. Metodología generalizada usada en estos artículos.

A continuación, los algoritmos con sus particularidades para ser elegidos como clasificadores en estos estudios.

Tabla 1. Algoritmos y sus características.

Algoritmo	Propiedades
Fuzzy logic	Se basa en reglas heurísticas, se usa para procesos altamente no lineales [5]. Fácil implementación.
SVM	Busca un hiperplano que funciona como separador. En el entrenamiento y problemas típicos es muy eficiente [6].
Bayes	Es muy eficiente donde se utiliza este tipo de entorno de aprendizaje supervisado. No se requiere de gran cantidad de datos para su entrenamiento [7].
KNN	Busca de las observaciones más cercanas a la que está tratando de predecir y clasifica el punto de interés basado en la mayoría de los datos de entrenamiento.[10]
ANN	La clasificación es muy eficiente, a costa de un entrenamiento costoso computacionalmente.[11]
CNN	El desempeño de las redes neuronales convolucionales en el reconocimiento de objetos y la clasificación de imágenes ha hecho un gran progreso en los últimos años. Tienden a ser más precisas a costa del alto costo computacional. En el entrenamiento puede requerir un número considerable de imágenes para producir resultados fiables [15]

En la siguiente sección, se muestra los trabajos significativos, relacionados con la detección de las enfermedades en varios tipos de plantas, utilizando diversos algoritmos de aprendizaje automático, pero siempre siguiendo la misma metodología.

3. Revisión de la literatura

Derwin Suhartono et al. (2013) utilizaron un sistema de lógica difusa y árboles de decisión y con la ayuda de un experto humano, pudieron hacer un reconocimiento de las enfermedades del café, donde obtuvieron las características de los síntomas que se presentan en la planta y así poder hacer su árbol de decisión. Los resultados que obtuvieron están dentro del 85% de exactitud. Aunque ésta investigación no trata de técnicas de visión artificial, se puede notar que se ha dado un paso muy importante para la detección de enfermedades, utilizando sistemas expertos aplicando arboles de decisión con lógica difusa [5].

Bhange y Hingoliwala (2015) proponen una herramienta web, donde los agricultores que cultivan la granada, subirán una imagen del fruto para ser analizada y con el modelo entrenado se puede comprobar si está o no infectado el fruto. La técnica utilizada se basa en extraer las características de las imágenes de la granada, como el color, morfología y el vector de coherencia de color para posteriormente realizar la clasificación mediante el algoritmo SVM dándoles una precisión del 85% con una cámara de 10 megapíxeles. Los autores sabiendo que los agricultores no siempre tendrán los medios óptimos de captura para subir las imágenes al sistema, realizaron pruebas significativas con las cámaras de los dispositivos móviles con resoluciones de 5 y 3 megapíxeles, y obtuvieron resultados de 82% y 79% respectivamente [6].

Mengistu et al. (2016) utilizaron una ANN, KNN, Naive Bayes, un híbrido de mapas auto organizables (SOM) y función de base radial (RBF), todo esto para poder determinar el mejor algoritmo que clasificara las enfermedades de roya, marchitamiento del café (CWD) y la CFD que afecta al fruto del café. En su trabajo exponen que consiguieron para el KNN 58.16%, para Naive Bayes obtuvieron 53.47%, para ANN 79.04%, y para la combinación de RBF y SOM obtienen 90.07% de exactitud, lo cual muestra una gran mejora con respecto a los anteriores algoritmos, aunque dejan nota que éste último, les lleva más tiempo en el entrenamiento [7].

Singh y Misra (2017), realizaron pruebas para obtener reconocimiento de enfermedades o quemaduras presentadas en las hojas, Los cultivos de estudio fueron de plátano, frijol, limón y rosas. Después de procesar las imágenes, Proponen hacer la segmentación mediante algoritmos genéticos, y hacer el agrupamiento. Para la extracción de características utilizaron el método de concurrencia de color, ya que consideran mejor utilizar la imagen a color que la tradicional escala de grises. Para hacer la clasificación, utilizaron MDC con K-Mean obteniendo 86.54 %, MDC con un algoritmo propuesto por ellos, obteniendo una mejora de 93.63% y SVM con algoritmo propuesto obteniendo una mejora significativa de 95.71%. Todos estos porcentajes describiendo un promedio general de los cuatro cultivos de estudio [8]

Barbedo et al. (2016) proponen algoritmos para identificar múltiples enfermedades de las plantas, basándose en el análisis del color y usando un algoritmo de clasificación por pares. Según ellos con su metodología, les permite operar en condiciones no controladas y así poder abarcar un gran número de enfermedades. Este método fue probado con un gran conjunto, sin restricciones de las imágenes de las hojas que contienen los síntomas que pertenecen a 74 enfermedades, 4 plagas y 4 trastornos abióticos, que afecta a 12 especies de plantas diferentes. Los resultados obtenidos fueron entre 40% y 80% de precisión [9]

Qin et al. (2016), nos dicen que, encontraron una solución factible para el diagnóstico e identificación de cuatro enfermedades de la alfalfa. Extrajeron 129 características de textura, color y forma de las 1651 imágenes utilizando los métodos ReliefF, 1R y CFS. Para clasificar las enfermedades, utilizaron SVM, KNN y Random Forest. obteniendo que el mejor clasificador fue SVM y el método RelifF para la obtención de característica, ya que consiguieron 97.64% en precisión para el conjunto de entrenamiento y 94.74% para el conjunto de pruebas [10].

Pujari et al. (2016) compararon los algoritmos SVM y ANN para clasificar enfermedades en diversos cultivos que han sido atacados por hongos, bacterias, nematodos y deficiencia de nutrientes. Mencionan que los síntomas de las enfermedades de las plantas exhiben diferentes propiedades como el color, forma y textura y en base a esto, se obtiene las características. Consideran el color como una dimensión importante, pero aplicando reducción de dimensionalidad descubren por experimentación que, de 24 solo 8 características son significativas para la clasificación de las enfermedades. Al final ponen a prueba los dos algoritmos y encuentran que con SVM obtienen 92.17 % de precisión y que con ANN solo 87.48% por lo cual demuestran que, para este caso, el SVM es mejor clasificador [11].

Kiani y Mamedov (2017) tratan mediante su trabajo, de probar que no es necesario utilizar una red neuronal o algoritmo complejo para verificar si existe o no una enfermedad característica en las hojas de las fresas. Para los autores les es mejor usar algoritmos de lógica difusa para la clasificación que cualquier otro método de que consuma más recurso computacional como las redes neuronales. Los resultados que obtienen son alentadores, ya que alcanzaron una optimización del 97% en la segmentación y clasificación de las enfermedades con un tiempo de procesamiento de 1.2 segundos. Con este tipo de técnicas comprueban que es factible de ser implantadas en un circuito integrado, adaptado a un robot con visión artificial y que pudiera hacer una inspección automatizada en un centro botánico moderno [12].

Mohanty et al. (2016) realizó 60 experimentos donde usa redes neuronales convolucionales profunda para identificar 14 especies de cultivos y 26 enfermedades, utilizando modelos entrenados como AlexNet [17] y GoogleNet [18]. Utilizan el enfoque de Krizhevsky et al.[20] que demuestra por primera vez que la capacitación supervisada de extremo a extremo con una arquitectura CNN es una posible opción para un gran número de clases, superando el enfoque tradicional de utilizar características diseñadas a mano. Dentro del conjunto de datos de PlantVillage de 54,306 imágenes que contienen 38 clases de 14 especies de cultivos y 26 enfermedades (o su ausencia), este objetivo se ha logrado, como lo demuestra la precisión máxima del 99.35%. Por lo tanto, sin ningún tipo de ingeniería de características, el modelo clasifica correctamente los cultivos y las enfermedades de 38 clases posibles en 993 de cada 1000 imágenes. Los autores notaron que para el entrenamiento se requiere mucho trabajo computacional pero para la clasificación es menos de un segundo, creen que se podría implementar en un teléfono celular [13].

Ashqary y Abu-Naser (2019) realizaron un estudio que consistió en analizar 9,000 imágenes de hojas de tomate, para producir un modelo que se pudiera utilizar en teléfonos inteligentes, con el propósito de identificar 5 tipos de enfermedades. Su modelo estaría basado en una red convolucional profunda, pero que estaría conformada por dos partes, La primera parte del modelo (las funciones de extracción), que era el mismo para enfoque a todo color y enfoque de escala de grises, que consta de 4 capas

convolucionales con función de activación Relu, cada una seguida por capa Max Pooling y la segunda parte contendría dos capas densas para contener los dos enfoques, color y escala de grises. Al final, demostraron que trabajar con las características a color, les da mejores resultados 99.84% con respecto a escala de grises con un 95.54% [14].

Barbedo (2019) ha estado trabajando con redes neuronales convolucionales de varios cultivos, buscando niveles de enfermedad en las plantas, pudiendo clasificar a cultivos sanos con un 89% de exactitud, ligeramente enfermos con un 31%, moderadamente enfermos 87% y gravemente enfermos con un 94%. Aunque los resultados son satisfactorios, concluye que la clasificación de las enfermedades de las plantas a partir de imágenes digitales, es muy difícil. Por otra parte, las limitaciones del conjunto de datos en términos de cantidad y variedad de muestras, siguen impidiendo la aparición de sistemas verdaderamente integrales para hacer funciones de clasificación de enfermedades [15].

Khan et al (2019) propone el uso de VGGNet [19] y AlexNetOWTBn [20] como arquitecturas de una red convolucional profunda, para automatizar el proceso de clasificación de las enfermedades que se presentan en las hojas de tomate. Las enfermedades a clasificar fueron el tizón temprano, oidio y mildiu. Las imágenes recogidas son pre-procesadas utilizando técnicas de procesamiento de imágenes como la reducción de ruido, regresión y mejora de procesamiento de imágenes que ayuda en la reducción de costes y tiempo de cálculo. Posteriormente extrajo las características del conjunto de datos utilizando mapas de convolución donde se aplicaban a los datos de entrada, imágenes de hojas sana e infectadas. Aunque la arquitectura ha presentado resultados muy precisos, para este estudio solo obtuvieron 32.23% usando la arquitectura AlexNetOWTBn y 33.27% para VGG [16].

En la tabla 2 se muestra el resumen de los resultados obtenidos por las investigaciones expuestas.

Tabla 2. Resumen de los algoritmos y sus resultados.

Referencia.	Cultivo	Clasificador	Precisión en los resultados
[5]	Café	Fuzzy logic	85.00%
[6]	Granada	SVM	82.00%
[7]	Café	KNN	58.16%
		ANN	79.04%
		N. BAYES	53.47%
		RBF y SOM	90.07%
[8]	Plátano, Frijol, Limón, Rosa	MDC + K-Means	86.54%
		MDC + propuesta	93.63%
		SVM + propuesta	95.71%
[9]	Frijol	Propuesta Clasificación por pares	50.00%
	Mandioca		46.00%
	Agrios		56.00%
	Árbol de coco		71.00%
	Café		53.00%
	Maíz		40.00%
	Algodón		76.00%
Uva	58.00%		

Referencia.	Cultivo	Clasificador	Precisión en los resultados
	Maracuyá		56.00%
	Soja		58.00%
	Caña de Azúcar		59.00%
	Trigo		70.00%
[10]	Alfalfa	SVM	94.74%
[11]	Trigo, girasol, uva, maíz, pepino, algodón, col, tomate	ANN SVM	87.48% 92.17%
[12]	Fresa	Fuzzy logic	97.00%
[13]	Varios	CNN	99.35%
[14]	Tomate	CNN	99.84%
	Frijol		95.00%
	Mandioca		83.00%
	Naranja		62.00%
	Coco		97.00%
	Maíz		66.00%
	Café		77.00%
	Algodón		100%
[15]	Anacardo		83.00%
	Uva		81.00%
	Col rizada		100%
	Maracuyá		90.00%
	Haba de soja		76.00%
	Caña de azúcar		100%
	Trigo		61.00%
	Total		82.00%
[16]	Tomate	CNN	33.27%

4. Discusión y resultados

Una de las ventajas que podemos observar cuando Derwin Suhartono et al. (2013) [5] utilizó el algoritmo clasificador de lógica difusa, es lo fácil y rápido que pudo implementarlo, pues no requiere de una gran colección de datos para su entrenamiento, y le presentó resultados más o menos favorables al momento de querer clasificar. Ahora que, si se disminuyen la clase objetivo, pues tendría mejores resultados, como lo demuestra el trabajo de Kiani y Mamedov (2017) [12], que consiguió mejor precisión por solo el hecho de predecir solo una enfermedad.

La propuesta que hace Mengistu et al. (2016) [7], con utilizar RBF y SOM parece bastante buena, pues son las que les dio mejores resultados, comparados con KNN, ANN, Bayes, y esto se debe a que su enfoque de obtener las características por su textura y color, hizo que tuviera mejores resultados, pues la elección de esta última es la que mejor se desenvuelve para estos casos.

De los algoritmos SVM se puede concluir que es otra buena alternativa para la resolución de estos casos, aunque en un principio, no fueron pensados para resolver este tipo de problemas, con el paso del tiempo se han ido adaptando y presentando resultados muy buenos, como los de Pujari et al. (2016) [11], Singh y Misra (2017) [8] y Bhangé y Hingoliwala (2015) [6].

Cuando la clasificación es para pocas enfermedades, se podría decir que es relativamente fácil, pero cuando realmente se quiere abarcar más enfermedades o más cultivos, es cuando aumenta la complejidad, y una propuesta de solución es la que presenta Barbedo et al. (2016) [9], con su clasificación por pares donde se basa en la teoría de que las plantas presentan síntomas parecidos cuando son atacados por la misma enfermedad y que su algoritmo puede ser reentrenado para nuevas enfermedades.

Otra propuesta para las mismas plantas y las mismas enfermedades, fue presentada por el mismo Barbedo [15] pero en el año 2019 al utilizar redes neuronales convolucionales, declara que es la mejor forma para hacer este tipo de estudios, y que la única limitante hoy en día, es la disponibilidad de grandes cantidades de datos para hacer un buen entrenamiento de la red.

5. Conclusión

Como se pudo observar en esta revisión de artículos, los trabajos que obtuvieron mejores resultados son redes neuronales convolucionales, su uso es cada vez más tendencioso para este tipo de problemáticas, pues se obtienen diagnósticos más cercanos a lo que un experto humano determinaría. El único problema sería que el entrenamiento es demasiado costoso computacionalmente y se requiere una gran cantidad de datos para hacerlo.

Si los datos de estudio no fuesen suficientes, la recomendación sería árboles de decisión con lógica difusa, ya que es el algoritmo con mejores resultados después de las redes convolucionales. Aunque no es muy preciso, podemos confiar que a medida que avancen las tecnologías, se podrá llegar a valores más óptimos y con un costo menor computacionalmente.

Otro punto importante que podemos hacer notar es que, con el uso de nuevas tecnologías como el aprendizaje automático y el reconocimiento de patrones, se pueden detectar enfermedades en los cultivos y hacer un diagnóstico oportuno, disminuyendo el riesgo de pérdidas agrícolas y económicas, lo que traería un beneficio directo a los agricultores que la implementase.

Referencias

1. Tai, A.P., Martin, M.V., Heald, C.L.: Threat to future global food security from climate change and ozone air pollution. *Nature Climate Change* 4(9), 817 (2014)
2. Harvey, C.A., Rakotobe, Z.L., Rao, N.S., Dave, R., Razafimahatratra, H., Rabarijohn, R.H., MacKinnon, J.L.: Extreme vulnerability of smallholder farmers to agricultural risks and climate change in Madagascar. *Philosophical Transactions of the Royal Society B: Biological Sciences* 369(1639), 20130089 (2014)
3. Sanchez, P.A., Swaminathan, M.S.: Cutting world hunger in half. *Science* 307(5708), 357–359 (2005)
4. Barbedo, J.G.A.: Digital image processing techniques for detecting, quantifying and classifying plant diseases. *SpringerPlus* 2(1), 660 (2013)
5. Suhartono, D., Aditya, W., Lestari, M., Yasin, M.: Expert system in detecting coffee plant diseases. *Int. J. Electr. Energy* 1(3), 156–162 (2013)

6. Bhange, M., Hingoliwala, H.A.: Smart farming: Pomegranate disease detection using image processing. *Procedia Computer Science* 58, 280–288 (2015)
7. Mengistu, A.D., Alemayehu, D.M., Mengistu, S.G.: Ethiopian coffee plant diseases recognition based on imaging and machine learning techniques. *International Journal of Database Theory and Application* 9(4), 79–88 (2016)
8. Singh, V., Misra, A.K.: Detection of plant leaf diseases using image segmentation and soft computing techniques. *Information processing in Agriculture* 4(1), 41–49 (2017)
9. Barbedo, J.G.A., Koenigkan, L.V., Santos, T.T.: Identifying multiple plant diseases using digital image processing. *Biosystems engineering*, vol. 147, pp. 104–116 (2016)
10. Qin, F., Liu, D., Sun, B., Ruan, L., Ma, Z., Wang, H.: Identification of alfalfa leaf diseases using image recognition technology. *PloS one* 11(12), e0168274 (2016)
11. Pujari, D., Yakkundimath, R., Byadgi, A.S.: SVM and ANN based classification of plant diseases using feature reduction technique. *IJIMAI* 3(7), 6–14 (2016)
12. Kiani, E., Mamedov, T.: Identification of plant disease infection using soft-computing: Application to modern botany. *Procedia computer science*, vol. 120, pp. 893–900 (2017)
13. Mohanty, S.P., Hughes, D.P., Salathé, M.: Using deep learning for image-based plant disease detection. *Frontiers in plant science*, 7, 1419 (2016)
14. Ashqar, B.A., Abu-Naser, S.S.: Image-Based Tomato Leaves Diseases Detection Using Deep Learning (2019)
15. Barbedo, J.G.A.: Plant disease identification from individual lesions and spots using deep learning. *Biosystems Engineering*, 180, pp. 96–107 (2019)
16. Khan, S., Shaikh, A.A., Ansari, H., Ansari, N.: Disorder Detection in Tomato Plant Using Deep Learning. In: Ayesha, A., Ansari, H., Ansari, N. (eds.), *Disorder Detection in Tomato Plant Using Deep Learning* (February 24, 2019) (2019)
17. Krizhevskv. A., Sutskever. I., Hinton. G.F.: Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, pp. 1097–1105 (2012)
18. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Rabinovich, A.: Going deeper with convolutions. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9 (2015)
19. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014)
20. Krizhevsky, A.: One weird trick for parallelizing convolutional neural networks. *arXiv preprint arXiv:1404.5997* (2014)

Estado del arte de clasificación de nódulos pulmonares utilizando Aprendizaje Automático

Cecilia Irene Loeza Mejía, Rajesh Roshan Biswal, Eddy Sánchez-DelaCruz

Instituto Tecnológico Superior de Misantla, Departamento de Posgrado,
Misantla, Veracruz, México

cecilialoeza@yahoo.com, rroshanb@itsm.edu.mx

Resumen. Los nódulos pulmonares son manchas o lesiones que se diagnostican en los pulmones y pueden tener tanto causas benignas como malignas, principalmente relacionadas con cáncer de pulmón. En México, de acuerdo con estadísticas, el cáncer de pulmón se ubica en el séptimo lugar y es el más letal. Diversas instituciones médicas internacionales trabajan para mejorar el diagnóstico del cáncer de pulmón, debido a que la principal causa de muerte es el diagnóstico tardío. Por esta razón, el análisis de nódulos pulmonares es un desafío en el procesamiento de imágenes médicas para determinar el tratamiento adecuado, se proponen nuevas metodologías y técnicas no sólo por expertos en medicina sino también por otros científicos. El objetivo de este estudio es mostrar técnicas de aprendizaje automático aplicadas en la clasificación de nódulos pulmonares, específicamente se revisarán redes neuronales artificiales y máquina de vector soporte. Se concluye que el aprendizaje automático es efectivo para clasificar nódulos pulmonares en malignos (cancerosos) y benignos (no cancerosos).

Palabras clave: cáncer de pulmón, clasificación de nódulos pulmonares, aprendizaje automático

A Review on Classification of Pulmonary Nodules Using Machine Learning

Abstract. Pulmonary nodules are stains or lesions that are diagnosed in the lungs and can have both benignant and malignant causes, mainly related to lung cancer. In Mexico, according to statistics, lung cancer is in seventh place and is the most lethal. Several international medical institutions continue to work to improve the diagnosis of lung cancer, as delayed diagnosis leads to the principal cause of the mortality. For this reason, the processing of medical images and the diagnosis of pulmonary nodules can be life saving, to determine the appropriate treatment techniques as proposed by medical experts and scientists in the field. The aim of this study is to review the application of state-of-the-art machine learning techniques for the classification of pulmonary nodules, specifically

artificial neural networks and support vector machine. It is concluded that machine learning is very much effective in classifying pulmonary nodules as malignant (cancerous) and benign (non-cancerous).

Keywords: lung cancer, classification of pulmonary nodules, machine learning.

1. Introducción

El cáncer de pulmón es un problema mundial de salud, en México cada año fallecen aproximadamente ocho mil personas a causa del mismo, siendo el principal problema el diagnóstico tardío [1]. En el 99 % de los casos se detecta el cáncer en fase III o IV, existen pocas iniciativas para detectarlo antes [2]. La detección temprana puede prolongar la supervivencia y facilitar la recuperación de las personas. Además, el diagnóstico exacto es clave para el tratamiento adecuado, por lo que es necesario implementar medidas de diagnóstico oportuno.

Las imágenes médicas son muy importantes, ya que ayudan a emitir diagnósticos, determinar el tratamiento y evaluar los cambios que ocurren con el tiempo [3], ejemplo de éstas son la radiografía, la tomografía computarizada (TC) y el ultrasonido. TC es un estudio radiológico que utiliza rayos X para generar múltiples imágenes transversales del interior del cuerpo. Para la detección del cáncer de pulmón, un radiólogo interpreta una tomografía computarizada del tórax, a través de su experiencia y conocimiento médico, dicha interpretación la recibe el oncólogo para emitir el tratamiento de cáncer de pulmón, sin embargo, la visualización de múltiples tomografías puede provocar cansancio al radiólogo y hacer que cometa errores.

Mediante el aprendizaje automático se pueden detectar y predecir patrones [4], ha mostrado ser efectivo en aplicaciones del ámbito médico, tales como: detección temprana de enfermedades, planificación quirúrgica, análisis de patologías, oncología y radiología. El aprendizaje automático es una de las disciplinas de la inteligencia artificial, que emplea estadística, probabilidad y técnicas de optimización, para comprender reglas complicadas y características de alta dimensión; en el tema de cáncer de pulmón se ha aplicado para detectarlo [5], clasificar la gravedad del cáncer [6], predecir la supervivencia de pacientes [7], clasificar el cáncer de pulmón [8,9] y clasificar nódulos pulmonares [10–19].

A continuación, se muestra la revisión de la literatura en la sección 2, mientras que en la sección 3 se exponen los datasets y métodos que generalmente se utilizan en la clasificación de nódulos, en la sección 4 se resumen los resultados que han obtenido diversos autores utilizando redes neuronales artificiales y máquina de vector soporte; mientras que en la sección 5 se proporciona una visión general del aprendizaje automático en la clasificación de nódulos pulmonares.

2. Revisión de la literatura

Liu y Kang [10] utilizan la arquitectura de redes neuronales multivista, la cual toma múltiples vistas de cada nódulo ingresado, permitiendo realizar la clasificación binaria de nódulos (benigno y maligno) y una clasificación ternaria (benigno, maligno primario y maligno metastásico). Para validar su arquitectura utilizan el dataset LIDC-IDRI.

Zhu *et al.* [11] proponen DeepLung el cual es un sistema automático de diagnóstico de cáncer. DeepLung consta de dos fases: detección de nódulos y clasificación de nódulos (en benignos o malignos), utiliza redes convolucionales basadas en redes de doble vía 3D y potenciación del gradiente. Las redes de doble vía son más compactas y puede tener mejor rendimiento que redes residuales. Para entrenar y evaluar la detección de nódulos utilizaron el dataset LUNA16, en cambio, para validar la clasificación utilizan el dataset LIDC-IDRI.

Abbas [12] presenta el sistema Nodular-Deep que permite clasificar nódulos pulmonares, contempla tres etapas: (i) extracción de características mediante una red neuronal convolucional (ii) optimización de características utilizando red neuronal recurrente, la cual es una técnica de aprendizaje supervisado (iii) clasificación de nódulos (benignos y malignos) optimizadas a través del clasificador lineal softmax. Debido a la arquitectura presentada, Nodular-Deep no requiere preprocesamiento ni postprocesamiento. Para validar sus resultados utiliza el dataset LIDC-IDRI.

Yan *et al.* [13] proponen utilizar una red neuronal convolucional para clasificar nódulos pulmonares (en benignos y malignos), así mismo, implementan tres redes neuronales: (i) 2D a nivel de división (ii) 2D a nivel de nódulos (iii) 3D a nivel de nódulos. Para comparar las redes neuronales utilizan el dataset LIDC-IDRI, obteniendo mejores resultados en la red 3D a nivel de nódulos.

Shukla *et al.* [14] muestran un sistema de diagnóstico automático, el cual consta de cuatro etapas (i) preprocesamiento, en la cual utilizan un filtro difuso (ii) segmentación utilizando el algoritmo Watershed (iii) extracción de características mediante una matriz de co-ocurrencia de nivel de gris (iv) clasificación utilizando máquina de vector soporte.

Parven y Kavitha [15] proponen un sistema que considera cuatro etapas (i) preprocesamiento donde extraen la región del pulmón de la TC (ii) segmentación para identificar regiones del pulmón con nódulos (iii) extracción de características utilizando una matriz de co-ocurrencia de nivel de gris (iv) clasificación de nódulos en benignos y malignos utilizando máquina de vector soporte con distintos kernels (lineal, cuadrático polinomial y función de base radial). Utilizan un dataset con TC de 11 pacientes de un hospital reconocido. Las pruebas las realizan en tiempo real.

Mabrouk *et al.* [16] exponen el desarrollo de un sistema de diagnóstico asistido por computadora para clasificar nódulos grandes. La metodología que utilizan es (i) extracción de regiones pulmonares mediante umbralización (ii) extracción de las características más relevantes (iii) selección de características, en esta etapa eligen las características más importantes para diferenciar entre tejidos normales y tejidos cancerosos (iv) clasificación de tejidos normales y cancerosos utilizando

máquina de vector soporte con diferentes kernels (lineal, cuadrático y función de base radial). Utilizan un dataset de 12 TC recuperadas de la Universidad de Cornell.

Kuruville y Gunavathi [17] clasifican nódulos pulmonares, la metodología que utilizan es (i) segmentación mediante operaciones morfológicas y el método de OTSU (ii) extraen parámetros estadísticos de las regiones segmentadas de los pulmones (iii) utilizan redes neuronales para la clasificación de nódulos cancerosos y nódulos no cancerosos. Las tomografías que utilizan son las del dataset LIDC-IDRI y de un hospital.

Madero *et al.* [18] proponen una metodología para la clasificación de nódulos compuesta por las etapas (i) extracción de la región de interés (ii) transformada Wavelet (iii) extracción de características (iv) selección de atributos y sub-bandas, en esta etapa buscan el mejor subconjunto de atributos en el vector de características para diferenciar entre nódulos cancerosos y no nódulos (v) clasificación utilizando máquina de vector soporte. Los datasets que utilizan son LIDC-IDRI y ELCAP.

Son *et al.* [19] proponen un método de aprendizaje profundo para mejorar la precisión de clasificación de nódulos pulmonares en la TC. Utilizan una red neuronal convolucional profunda 2D de 15 capas para la extracción y clasificación de nódulos y no nódulos. Además, aplican la función de pérdida focal en el proceso de entrenamiento para aumentar la precisión de clasificación del modelo de aprendizaje automático. Para validar sus resultados utilizan el dataset LUNA16.

3. Materiales y métodos de clasificación utilizando Aprendizaje Automático

El aprendizaje automático permite la resolución de tareas complejas, en la literatura existen diversos enfoques de aprendizaje automático para clasificar nódulos pulmonares, a continuación, se hace un compendio de los materiales y métodos utilizados en la literatura, se muestran los datasets que utilizan y el procesamiento de la TC aplicando aprendizaje automático.

3.1. Dataset

Para entrenar y evaluar el rendimiento de algoritmos se debe obtener un dataset de imágenes de TC del tórax. Dichas imágenes son en formato DICOM, el cual es un estándar que se creó para promover la comunicación de información asociada a imágenes médicas entre dispositivos fabricados por diferentes proveedores [20]. Una imagen DICOM contiene un encabezado con información para identificarla en el dataset, datos personales del paciente, parámetros de adquisición y datos de la imagen. Cada TC de un paciente se agrupa en una carpeta y contiene múltiples imágenes transversales de diferentes planos o cortes del tórax: (i) sagital, es paralelo al plano lateral del paciente (ii) coronal, es paralelo al plano frontal del paciente, (iii) axial, es perpendicular al eje longitudinal del paciente.

En la literatura se han utilizado datasets privados los cuales se generan en hospitales y en universidades, tales como la de Michigan, de Harvard y de Toronto. También se han utilizado datasets públicos como LIDC-IDRI, LUNA16 y ELCAP. Cada uno de los datasets contienen distinto número de carpetas que corresponde

Datasets privados. En [8] se mencionan 3 datasets privados: (i) Universidad de Michigan contiene 96 muestras, de las cuales 86 son adenocarcinoma primario (donde 67 están en etapa I y 19 en etapa III) y 10 de tejidos no neoplásicos. (ii) Brigham and Women's Hospital, Harvard contiene 181 muestras de tejidos malignos, clasificados en mesotelioma pleural y adenocarcinoma. (iii) Universidad de Toronto contiene 39 casos de cáncer de células no pequeñas.

Datasets públicos. LIDC-IDRI (del inglés *Lung Image Database Consortium and Image Database Resource Initiative*) es un repositorio público con tomografías de 1018 casos recuperadas de diversos centros médicos e instituciones de imágenes médicas. Dentro de cada caso se incluyen imágenes de una TC y un archivo XML asociado que registra los resultados de un proceso de anotación de imagen de dos fases realizado por cuatro radiólogos torácicos experimentados [21]. (ii) LUNA16 contiene 888 TC, es un subconjunto de tomografías del dataset LIDC-IDRI que excluye los escaneos con un grosor de corte superior a 2.5 mm [22]. (iii) ELCAP consta de un conjunto imágenes de 50 TC de pulmón completo documentadas en dosis bajas [23].

3.2. Preprocesamiento

El preprocesamiento permite mejorar la calidad de las imágenes y eliminar la redundancia de los datos. Además, puede realzar características para su posterior procesamiento, permitiendo visualizar detalles que no se veían en la imagen original [24]. El preprocesamiento generalmente es el paso que ocupa más tiempo. En esta etapa se convierten las tomografías a escala de grises y se elimina el ruido, el cual se puede observar en las imágenes de TC como un granulado de la misma, es decir, aparecen como si tuvieran manchas [25]. Para realizar el preprocesamiento [14] aplica un filtro difuso, el cual comúnmente se utiliza para disminuir el ruido de imágenes. Por otra parte [15] emplea técnicas de binarización, etiquetado, encogimiento y expansión.

3.3. Segmentación

La segmentación permite obtener objetos o regiones de interés significativas de la imagen, para ello, se utilizan distintos métodos, los cuales trabajan sobre las variaciones de intensidad o textura [24]. La segmentación en tomografías identifica las regiones pulmonares. El nódulo pulmonar es un tejido anormal que puede ser benigno o maligno, máximo mide 3 centímetros de diámetro y su forma es redonda u ovalada [26]. La segmentación de la TC de tórax es un

desafío ya que hay falta de homogeneidad en las estructuras pulmonares con densidades similares, tales como venas, arterias, bronquiolos, además, tienen diferentes protocolos de escaneo y escáneres [14].

Para segmentar la TC, [9] utiliza la umbralización y el algoritmo Watershed, en cambio [14] aplica el algoritmo Watershed, por otra parte [17] utiliza operaciones morfológicas y el método de OTSU, mientras que [16] emplea la transformada de Radón.

3.4. Extracción de características

Es una etapa crucial e importante, obtiene las características más relevantes en las cuales se tienen sospechas de la presencia de nódulos, para que posteriormente se puedan diferenciar entre benignos y malignos. La calidad de la extracción de características es de suma importancia para clasificar correctamente los nódulos. Para extraer características [14–16, 18] emplean una matriz de co-ocurrencia de nivel de gris (GLCM) la cual contiene información sobre los píxeles con valores de nivel de gris similares. GLCM es de los métodos más utilizados en el análisis de textura de imágenes biológicas. Por otra [11] parte utiliza red profunda 3D, mientras que [12] utiliza una red neuronal convolucional.

3.5. Clasificación de nódulos

La clasificación automática es fundamental en el diagnóstico asistido por computadora cuando se emplean imágenes médicas. En esta etapa se clasifican los nódulos en benignos (no cancerosos) o malignos (cancerosos), con base en su tamaño, densidad, contorno y forma. Para clasificar los nódulos se utilizan dos conjuntos de datos uno para entrenamiento y el otro para validación. Las técnicas de aprendizaje automático que se han aplicado para la clasificación de nódulos son redes neuronales artificiales (ANN) y máquina de vector soporte (SVM), las cuales se utilizan ampliamente en el campo médico para la detección de enfermedades.

Redes neuronales artificiales. Las ANN son neuronas entrelazadas que se inspiran en las redes biológicas y se procesan de forma paralela para obtener un resultado. Las ANN tienen la propiedad de aprender a partir de unas pocas representaciones de los datos, su estructura es jerárquica, así mismo, tienen flexibilidad lo que permite reconocer patrones con ruido o incompletos. Además, tienen redundancia intrínseca, lo cual permite un funcionamiento perfecto del sistema aun estando dañado alguno de sus componentes [27]. Para clasificar nódulos pulmonares se han utilizado redes multivista, backpropagation, 2D, 3D y 3D con potenciación del gradiente.

Máquina de vector soporte. Es una técnica de aprendizaje supervisado que permite analizar e identificar categorías o clases utilizando los datos entrenados y son robustas al ruido en los datos, así mismo, se pueden aplicar en problemas

de alta dimensionalidad. Para separar las clases analizadas se utilizan kernels, los cuales permiten transformar un problema no linealmente separable a uno linealmente separable. Además, la SVM tiene una gran capacidad de generalización incluso cuando el conjunto de entrenamiento es pequeño [28]. Los kernels que utilizan en la literatura son: lineal, cuadrático, polinomial y función de base radial.

En la sección de resultados se muestran los algoritmos que se han utilizado en diversos proyectos de clasificación de nódulos pulmonares. Los algoritmos se han programado en diversas herramientas como Python, R y Matlab; empleando distribuciones del sistema operativo Windows y Linux. Algunos autores también han empleado librerías como TensorFlow en Python.

3.6. Validación de resultados

Se verifica que el algoritmo clasifica correctamente los nódulos. En la literatura se han utilizado: matriz de confusión y validación cruzada para evaluar el modelo. En la matriz de confusión en cada columna se representa el número de predicciones de cada clase y cada fila representa los valores reales. Los parámetros que se utilizan son TP (verdadero positivo), TN (verdadero negativo), FP (falso positivo) y FN (falso negativo). La matriz de confusión permite calcular la precisión (*accuracy*), como la proporción entre las predicciones correctas que ha hecho el modelo y el total de predicciones, la ecuación 1 permite obtener la precisión. La ecuación 2 obtiene la sensibilidad, la cual es la fracción de nódulos clasificados correctamente. En la ecuación 3 se muestra la especificidad la cual es la fracción de no nódulos clasificados correctamente. También se puede obtener la curva ROC, la cual que describe la fracción de identificación correcta (sensibilidad) contra la tasa de falsas positivos (1 - especificidad):

$$ACC = \frac{TP + TN}{TP + FP + FN + TN}, \quad (1)$$

$$SE = \frac{TP}{TP + FN}, \quad (2)$$

$$SP = \frac{TN}{TN + FP}. \quad (3)$$

Para validar sus resultados [18] utiliza sensibilidad, especificidad, precisión y curva ROC. En cambio [16, 17, 19] calculan precisión, especificidad y sensibilidad. Por otra parte [10] calcula sensibilidad, especificidad y curva ROC. En contraste [15] utiliza sensibilidad y especificidad. Mientras que [14] calcula la precisión.

Otros autores utilizan la validación cruzada para evaluar la clasificación, en la cual, el dataset original se divide en k subconjuntos o pliegues, para cada pliegue, el algoritmo se entrena en la unión de los otros pliegues y luego el error de su salida se estima utilizando el pliegue. Finalmente, el promedio de todos estos

errores es la estimación del error verdadero [29]. En la ecuación 4 se muestra la fórmula para el cálculo del error de validación cruzada en k iteraciones:

$$E = \frac{1}{K} \sum_{i=1}^K E_i. \quad (4)$$

El número de iteraciones k depende del tamaño del dataset, [11] utiliza validación cruzada de 10 iteraciones en el dataset LUNA16, así mismo [12] aplica validación de 10 iteraciones en el dataset LIDC-IDRI, por otra parte [13] emplea validación de 5 iteraciones en LIDC-IDRI.

3.7. Resumen de metodología de clasificación de nódulos

En la figura 1 se expone de manera general la metodología que utilizan diversos autores para clasificar los nódulos: (i) se preprocesa el dataset de TC de tórax (ii) se identifican regiones con nódulos (iii) se obtienen las características más relevantes (iv) clasifican los nódulos pulmonares en benignos o malignos y (v) validan que los resultados hayan sido los correctos.

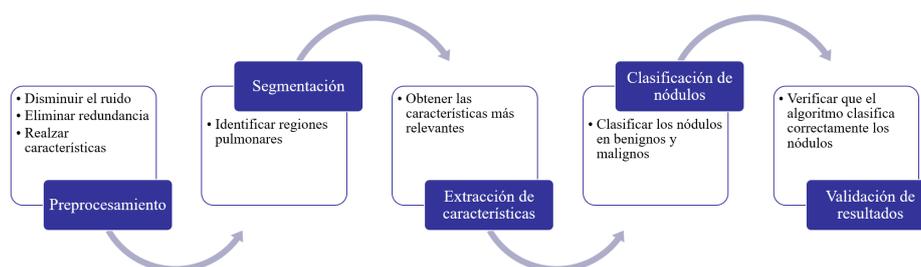


Fig. 1. Metodología para Clasificación de Nódulos Pulmonares utilizando Aprendizaje Automático

4. Resultados y discusión

En el estado del arte generalmente se clasifican los nódulos pulmonares en benignos y malignos. Para entrenar y validar la clasificación mediante redes neuronales los autores han trabajado con los datasets LUNA16 y LIDC-IDRI, los cuales contienen 888 y 1018 TC respectivamente. Mientras que para el entrenamiento y validación utilizando máquina de vector soporte han utilizado datasets privados que contienen desde 11 TC hasta los datasets públicos como ELCAP con 50 TC y LIDC-IDRI con 1018 TC.

Los resultados que obtuvieron distintos autores se muestran en la tabla 1 se mencionan los métodos de aprendizaje automático utilizados (específicamente

redes neuronales artificiales y máquina de vector soporte) aplicados en proyectos de clasificación de nódulos pulmonares, la precisión que tienen y los tipos de nódulos que clasifican.

El porcentaje de acierto varía desde 81.41 % utilizando redes neuronales convolucionales 3D hasta el 97.2 % utilizando red neuronal convolucional 2D y función de pérdida focal. Como se puede observar el porcentaje de precisión puede variar, aunque se utilice la misma técnica de clasificación, esto se debe a que utilizan diversas técnicas para el preprocesamiento, segmentación y extracción de características.

Tabla 1. Métodos de aprendizaje automático para clasificar nódulos pulmonares.

Referencia	Método utilizado	Acierto	Clasificación de nódulos
[10]	Redes neuronales convolucionales multivista	94.59 % 86.09 %	Benignos y malignos Benigno, maligno primario y maligno metastásico
[11]	Red neuronal convolucional 3D y potenciación del gradiente	81.41 %	Benignos y malignos
[12]	Redes neuronales y softmax	95 %	Benignos y malignos
[13]	Red neuronal convolucional: 2D a nivel de división 2D a nivel de nódulos 3D a nivel de nódulos	86.7 % 87.3 % 87.4 %	Malignos
[14]	SVM	92.5 %	Benignos y malignos
[15]	SVM: Kernel Lineal Kernel Polinomial Kernel función de base radial	83.45 % 85.79 % 91.38 %	Normales, benignos y malignos
[16]	SVM: Kernel Lineal Kernel Cuadrático Kernel función de base radial	91.9 % 94.9 % 91.9 %	Normales, benignos y malignos
[17]	Red neuronal backpropagation utilizando pendiente de gradiente con velocidad de aprendizaje variable	91 %	Benignos y malignos
[18]	SVM	82 %	De 2mm a 30mm de diámetro
[19]	Red neuronal convolucional 2D y función de pérdida focal	97.2 %	Nódulos y no nódulos

5. Conclusión

El aprendizaje automático ha mostrado diversas aplicaciones en el tema de cáncer de pulmón. En el estudio realizado se expone la metodología común que siguen diversos autores para clasificar nódulos pulmonares, lo cual da una pauta para la detección y clasificación del cáncer de pulmón. Para la clasificación de nódulos generalmente se realiza lo siguiente: (i) preprocesamiento de las tomografías para disminuir el ruido y la redundancia, donde emplean diversas técnicas como binarizar y aplicar filtro difuso (ii) segmentación para identificar regiones pulmonares, las técnicas que utilizan son: umbralización, algoritmo Watershed, operaciones morfológicas y transformada de Radón (iii) extracción de las características más relevantes, generalmente los autores emplean la matriz de co-ocurrencia de nivel de gris (iv) clasificación de nódulos en benignos y malignos, aplican métodos de aprendizaje automático como redes neuronales artificiales y máquina de vector soporte (v) validación de los resultados mediante matriz de confusión o validación cruzada, haciendo uso de dataset de TC de tórax, el cual proviene de universidades y hospitales. En el estudio se muestra una visión general de los datasets de tomografías del tórax que utilizan, los métodos que aplican y los resultados que obtienen. Se concluye que el aprendizaje automático se puede aplicar en la detección y clasificación de nódulos pulmonares en tomografías computarizadas para ayudar en la detección y determinación del tratamiento.

Referencias

1. Secretaría de salud: Cada año mueren cerca de ocho mil mexicanos por cáncer de pulmón (2018) <https://www.gob.mx/salud/prensa/145-cada-ano-mueren-cerca-de-ocho-mil-mexicanos-por-cancer-de-pulmon?idiom=es>. Último acceso: 2019/03/09
2. The Economist Intelligence Unit Limited: Lung cancer in Latin America: Time to stop looking away (2018) <https://perspectives.eiu.com>. Último acceso: 2019/03/27
3. Restrepo, A.: Procesamiento de imágenes médicas. *Iatreia* 12(1) (1999)
4. Russell, S., Norvig, P.: *Artificial Intelligence: A Modern Approach*. 3rd edn. Prentice Hall, New Jersey (2010)
5. Zhuo, L., Chenhui, Y., Hang Y., Taihua W.: Deep Reinforcement Learning with Its Application for Lung Cancer Detection in Medical. *Internet of Things*. vol. 97, pp. 1–9. <https://doi.org/10.1016/j.future.2019.02.068>
6. Bergquist, S., Brooks, G., Keating, N., Landrum, M., Rose, S.: Classifying Lung Cancer Severity with Ensemble Machine Learning in Health Care Claims Data. In: *2nd Machine Learning for Healthcare Conference*, pp. 25–38 (2017)
7. Pradeep, K., Naveen, N.: Lung Cancer Survivability Prediction based on Performance Using Classification Techniques of Support Vector Machines, C4.5 and Naive Bayes Algorithms for Healthcare Analytics. *Procedia Computer Science*, vol. 132, pp. 412–420 <https://doi.org/10.1016/j.procs.2018.05.162> (2018)
8. Barchuck, A. (et al.): Evaluation of Machine Learning Algorithm Utilization for Lung Cancer Classification Based on Gene Expression Levels. *Asian Pac J Cancer Prev*. 17(2), 835–838. <https://doi.org/10.7314/APJCP.2016.17.2.835> (2016)

9. Alakwaa, W., Nassef, M., Badr, A.: Lung Cancer Detection and Classification with 3D Convolutional Neural Network (3D-CNN) vol. 11 <https://doi.org/10.14569/IJACSA.2017.080853> (2017)
10. Liu, K., Kang, G.: Multiview Convolutional Neural Networks for Lung Nodule Classification. *International Journal of Imaging Systems and Technology* 27(11), 12–22 (2017)
11. Zhuo, W., Liu, C., Fan, W., Xie, X.: DeepLung: Deep 3D Dual Path Nets for Automated Pulmonary Nodule Detection and Classification. <https://doi.org/10.1109/WACV.2018.00079>
12. Abbas, Q.: Nodular-Deep: Classification of Pulmonary Nodules using Deep Neural Network. *International Journal of Medical Research & Health Sciences* 6(8), pp. 111–118 (2017)
13. Yan, X. (et al.): Classification of Lung Nodule Malignancy Risk on Computed Tomography Images Using (2017)Convolutional Neural Network: A Comparison Between 2D and 3D Strategies. https://doi.org/10.1007/978-3-319-54526-4_7
14. Shukla, N., Narayane, A., Nigade, A., Yadav, K., Mhaske, H.: Lung cancer detection and classification using Support Vector Machine. *International Journal of Engineering And Computer Science* 4(11), 14983–14986 (2015)
15. Parveen, S., Kavitha, C.: Classification of Lung Cancer Nodules using SVM Kernels. *International Journal of Computer Applications* 95(25) (2014)
16. Mabrouk, M., Karrar, A., Sharawy, A.: Support Vector Machine Based Computer Aided Diagnosis System for Large Lung Nodules Classification. *Journal of Medical Imaging and Health Informatics*, vol. 3, pp. 214–220, <https://doi.org/10.1166/jmihi.2013.1154> (2013)
17. Kuruvilla, J., Gunavathi, K.: Lung cancer classification using neural networks for CT images. *Computer Methods and Programs in Biomedicine* 113(1), 202–209. <https://doi.org/10.1016/j.cmpb.2013.10.011> (2014)
18. Madero, H., Vergara, O., Cruz, V., Ochoa, H., Nandayapa, M.: Automated system for lung nodules classification based on wavelet feature descriptor and support vector machine. *BioMedical Engineering OnLine* 14. <https://doi.org/10.1186/s12938-015-0003-y> (2015)
19. Son, G., Phuong, T., Thi, V., Mai, C., Burie, J: Improving Accuracy of Lung Nodule Classification Using Deep Learning with Focal Loss. *Journal of Healthcare Engineering*, vol. 2019, pp. 1–9. <https://doi.org/10.1155/2019/5156416> (2019)
20. Pianykh, O.: *Digital Imaging and Communications in Medicine (DICOM): A Practical Introduction and Survival Guide*. Springer, Berlin (2012)
21. Armato, S. et al.: The Lung Image Database Consortium (LIDC) and Image Database Resource Initiative (IDRI): a completed reference database of lung nodules on CT scans. *Med Phys* 38(2), 915–931 <https://doi.org/10.1118/1.3528204> (2011)
22. Grand Challenges (s.f.): Lung nodule analysis 2016 <https://luna16.grand-challenge.org/Data/>. Último acceso: 2019/04/09
23. Vision and Image Analysis Group (s.f.): ELCAP Public Lung Image Database <http://www.via.cornell.edu/databases/lungdb.html>. Último acceso: 2019/04/09
24. Bronzino, J.: *Handbook of Medical Imaging*. Academic Press, USA (2000)
25. Gil, J. et al.: *Técnico especialista en laboratorio*. MAD, España (2006)
26. Clavero, J.: Nódulos pulmonares. *Revista Médica Clínica Las Condes* 26(3), pp. 302–312 (2015)
27. Viñuela, I., Galván, I.: *Redes de Neuronas Artificiales un enfoque práctico*. Pearson educación, Madrid (2004)

Cecilia Irene Loeza Mejía, Rajesh Roshan Biswal, Eddy Sánchez-DelaCruz

28. Maldonado, S., Weber, R.: Modelos de selección de atributos para Support Vector. *Revista Ingeniería de Sistemas*, vol. 26 (2012)
29. Shalev, S., Ben, S.: *Understanding Machine Learning: From Theory to Algorithms*. 3rd edn. Cambridge University Press, New York (2014)

Utilización de un sistema en tiempo real para la predicción de contaminación del aire

Carlos Lino-Ramírez, Rogelio Bautista-Sánchez, Sandra P. Bombela-Jiménez

Tecnológico Nacional de México en León, Guanajuato, México
carloslino@itleon.edu.mx, rogeliobautistasanchez@outlook.com,
sandy.bombela@gmail.com

Resumen. En la actualidad existen varias ciudades afectadas con altos índices de contaminación, debido a varias cosas entre ellas, el crecimiento vehicular, empresas que arrojan contaminantes a la intemperie debido a sus procesos industriales, basura e incendios forestales, entre otras. En este artículo se presenta la propuesta de un sistema en tiempo real que puede estar monitoreando variables ambientales en varios puntos de la ciudad y hacer una predicción del comportamiento de dichas variables. Los datos provienen de datos proporcionados por el Sistema Estatal de Información de Calidad del Aire (SEICA) en el estado de Guanajuato. Se aplican técnicas de rellenado de datos para completar los valores perdidos, se realiza el etiquetado de la calidad del aire de acuerdo al semáforo proporcionado por el SEICA y, se realiza una transformación a la base de datos para el entrenamiento de la red neuronal que se utiliza para la predicción de la calidad del aire. Los datos adquiridos se normalizan, se agrupan, y con ellos se estructuran los componentes de predicción, y se hace también el análisis estadístico y de estructura.

Palabras clave: tiempo real, predicción, contaminación, clasificación.

Use of a Real-Time System to Predict Air Pollution

Abstract. Currently there are several cities affected with high levels of pollution, due to several things among them, vehicle growth, companies that throw pollutants out in the open because of their industrial processes, garbage and forest fires, among others. In this article we present the proposal of a real-time system that can be monitoring environmental variables in several points of the city and make a prediction of the behavior of these variables. The data comes from data provided by the State Air Quality Information System (SEICA) in the state of Guanajuato. Data filling techniques are applied to complete the missing values, the air quality labeling is done according to the semaphore provided by the SEICA and a transformation is made to the database for the training of the neural network that is used for the prediction of air quality. The acquired data are normalized, grouped, and with them the prediction components are structured, and the statistical and structure analysis is also made.

Keywords: real time, prediction, pollution, classification.

1. Introducción

1.1. Contaminación ambiental

Uno de los principales problemas que enfrenta la humanidad actualmente, es la contaminación, debido a que genera gran preocupación para las grandes ciudades, ya que de éste se desprenden diversos problemas para la salud humana [1]. El incremento de la contaminación en las ciudades, se da debido a varios factores, como por ejemplo la industria energética, la movilización urbana, entre otros. El utilizar recursos no renovables, como lo es el petróleo o el carbón, para la producción de energía genera emisiones contaminantes como el dióxido de azufre (SO₂), monóxido de carbono (CO), entre otros. A nivel mundial, se utilizan 38 % derivados del petróleo para generar energía, 28 % utiliza carbón, 21 % gas natural, 6 % energía nuclear y solo el 7 % se centra en generar energía con recursos renovables [2]. Otra fuente de contaminación son los medios de transporte, de los que una gran parte de contaminantes es emitida por los automóviles [2]. Según las Naciones Unidas, hay alrededor de 7 billones de personas actualmente en el mundo [3], esto representa una enorme fuente de emisiones contaminantes, agravando el problema cada vez más, dado que las personas tienden a migrar a grandes ciudades, buscando mejores oportunidades de empleo, vivienda, servicios, entre otros. Esto provoca que las urbes continúen su expansión, y conlleva a que estén generando grandes emisiones de contaminación, lo que deteriora la calidad de aire y con ello acarrea problemas de salud. Según datos del Consejo Nacional de Población (CONAPO), el 72.3% de la población, en México, vive en zonas metropolitanas. Y según la ONU, en los próximos 10 años, las poblaciones rurales comenzarán a disminuir. Todo esto, genera que la salud de las personas de las grandes ciudades, se deteriore cada vez más. Las personas que viven en lugares con altos índices de contaminación, son más propensas a adquirir enfermedades del tipo respiratorias, como el asma o alergias [4].

1.2. Procesamiento Big Data

Se puede considerar que la era del Big Data” nace con el desarrollo de MapReduce y Hadoop como las primeras “tecnologías Big Data”. Estas tecnologías se centran en un enfoque de Batch Processing. Es decir, el objetivo era acumular todos los datos que se pudieran, procesarlos y producir resultados que se “empaquetaban” por lotes. Con este enfoque, Hadoop ha sido la herramienta más empleada. Es una herramienta realmente buena para almacenar enormes cantidades de datos y luego poder escalarlos horizontalmente mientras vamos añadiendo nodos en nuestro clúster de máquinas, ver figura 1.

Como se puede ver en la figura 1, el problema que aparece en este enfoque es que el retraso en tiempo que introduce disponer de un ETL que carga los datos para su procesamiento, no será tan ágil como hacerlo de manera continua con un enfoque de tiempo real. El procesamiento en trabajos batch de Hadoop MapReduce es el que domina en este enfoque. Y lo hace, apoyándose en todo momento de un ETL. Hasta la fecha la gran mayoría de las organizaciones han empleado este paradigma Batch. No era necesaria mayor sofisticación. Sin embargo, como ya comentamos anteriormente, existen exigencias mayores. Los datos, en muchas ocasiones, deben ser procesados en

tiempo real, permitiendo así a la organización tomar decisiones inmediatamente. Esas organizaciones en las que la diferencia entre segundos y minutos sí es crítica. Hadoop, en los últimos tiempos, es consciente de esta economía de tiempo real en la que nos hemos instalado. Por ello, ha mejorado bastante su capacidad de gestión. Sin embargo, todavía es considerado por muchos una solución demasiado rígida para algunas funciones. Por ello, hoy en día, solo es considerado el ideal en casos como cuando no se necesita un cálculo con una periodicidad alta, cuando los cálculos se deban ejecutar solo a final de mes y cuando la generación de informes es con una periodicidad baja.

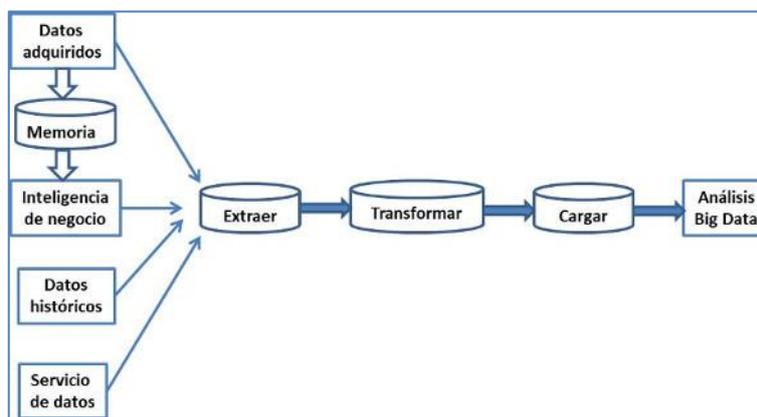


Fig. 1. Procesamiento batch con Big Data.

En los últimos años han surgido una serie de herramientas y tecnologías alrededor de Hadoop para ayudar en esa tarea de analizar grandes cantidades de datos. Para analizar las mismas, lo descomponemos en las cuatro etapas de la cadena de valor de un proyecto de Big Data:

1. Ingesta de datos:

Destacan tecnologías como: Flume: recolectar, agregar y mover grandes cantidades de datos desde diferentes fuentes a un data store centralizado. Comandos HDFS: utilizar los comandos propios de HDFS para trabajar con los datos gestionados en el ecosistema de Hadoop. Sqoop: permitir la transferencia de información entre Hadoop y los grandes almacenes de datos estructurados (MySQL, PostgreSQL, Oracle, SQL Server, DB2 y otras).

2. Procesamiento de datos:

Destacan tecnologías como: MapReduce: El proceso por el cual los datos que se obtienen en tiempo real van siendo capturados temporalmente para un posterior procesamiento. Hive: framework creado originalmente por Facebook para trabajar con el sistemas de ficheros distribuidos de Hadoop (HDFS). El objetivo no era otro que facilitar el trabajo, dado que a través de sus queries SQL (HiveQL) podemos lanzar consultas que luego se traducen a trabajos MapReduce. Pig: herramienta que facilita el análisis de grandes volúmenes de datos a través de un lenguaje de alto nivel. Su estructura permite la paralelización, que hace aún más eficiente el procesamiento de volúmenes de datos, así como la infraestructura necesaria para ello. Cascading: crear y ejecutar flujos

de trabajo de procesamiento de datos en clústeres Hadoop usando cualquier lenguaje basado en JVM (la máquina virtual de Java). De nuevo, el objetivo es quitar la complejidad de trabajar con MapReduce y sus trabajos. Es muy empleado en entornos complejos como la bioinformática, algoritmos de Machine Learning, análisis predictivo, Web Mining y herramientas ETL. Spark: facilita enormemente el desarrollo de programas de uso masivo de datos. Creado en la Universidad de Berkeley, ha sido considerado el primer software de código abierto que hace la programación distribuida accesible y más fácil para “más público” que los muy especializados.

3. Almacenamiento de datos

Destacan tecnologías como: HDFS: sistema de archivos de un cluster Hadoop que funciona de manera más eficiente con un número reducido de archivos de datos de gran volumen, que con una cantidad superior de archivos de datos más pequeños. HBase: permite manejar todos los datos y tenerlos distribuidos a través de lo que denominan regiones, una partición tipo Nodo de Hadoop que se guarda en un servidor. La región aleatoria en la que se guardan los datos de una tabla es decidida, dándole un tamaño fijo a partir del cual la tabla debe distribuirse a través de las regiones. Aporta, así, eficiencia en el trabajo de almacenamiento de datos.

4. Servicio de datos

En esta última etapa, en realidad, no es que destaque una tecnología o herramienta, sino que destacaría el “para qué” se ha hecho todo lo anterior. Es decir, qué podemos ofrecer/servir una vez que los datos han sido procesados y puestos a disposición del proyecto de Big Data [5].

1.3. Sistemas en tiempo real

A la hora de procesar grandes volúmenes de datos existen dos principales enfoques: procesar una gran cantidad de datos por lotes o bien hacerlo, en pequeños fragmentos, y en “tiempo real”. Parece, así, bastante intuitivo pensar cuál es la idea del paradigma en tiempo real que se tratará. Este enfoque de procesamiento y análisis de datos se asienta sobre la idea de implementar un modelo de flujo de datos en el que los datos fluyen constantemente a través de una serie de componentes que integran el sistema de Big Data que se esté implantando. Por ello, se lee como procesamiento streaming o de flujo. Así, en tiempos muy pequeños, procesamos de manera analítica parte de la totalidad de los datos, y, con estas características, se superan muchas de las limitaciones del modelo batch. Por otro lado, una cosa es denominarlo tiempo real y otra es realmente pensar que esto se va a producir en verdadero tiempo real, ver figura 2.

Las limitaciones aparecen por que:

- Se debe disponer de suficiente memoria para almacenar entradas de datos en cola. Fíjense en la diferencia con el paradigma batch, donde los procesos de MapReduce podrían ser algo lentos, dado que escribían en disco entre las diferentes fases.
- La tasa de productividad del sistema debería ser igual o más rápida a la tasa de entrada de datos. Es decir, que la capacidad de procesamiento del sistema sea más ágil y eficiente que la propia ingesta de datos. Esto, de nuevo, limita bastante la capacidad de dotar de instantaneidad al sistema.

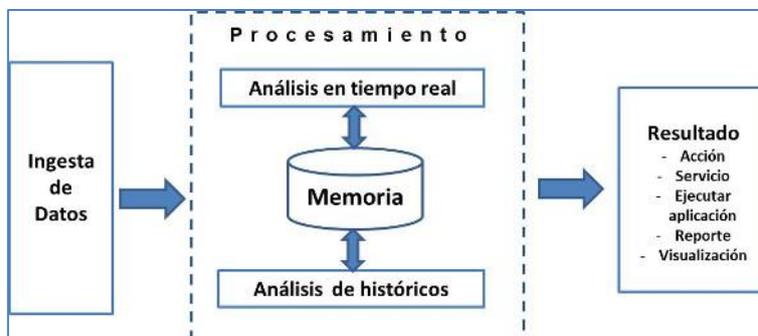


Fig. 2. Procesamiento en tiempo real.

Con la estructura mostrada en la figura 2 podremos procesar flujos de datos en tiempo real con sistemas big data, con algunos de los actuales modelos de procesamiento existentes para esta tecnología.

1.4. Sistema para la predicción de contaminación del aire

En este artículo se hace la propuesta de un sistema para predecir la contaminación del aire haciendo uso de un ambiente big data que permite la recolección de los datos en tiempo real y que nos permita predecir de la mejor manera posible la contaminación del aire, administrando de manera correcta desde la ingesta de datos, el procesamiento, el almacenamiento y el servicio de los datos gestionado con herramientas tecnológicas para Big Data.

2. Trabajos relacionados

A través de determinados algoritmos y empleando las variables y los dispositivos adecuados se pueden detectar desde problemas graves de salud pública hasta tendencias de mercado, pasando por los rendimientos de una cosecha con sistemas Big Data. Lucila Ballarino, responsable global de Transformación Digital de Fundación Telefónica, asegura que también existen ejemplos de gestión de datos climatológicos, de salud o de criminalidad. Pueden predecirse desastres naturales en forma de terremotos y tsunamis; analizarse rutas de propagación de virus en tiempo real para tomar decisiones rápidas y poder controlarlo lo antes posible, evitando pandemias; o incluso anticiparse a la realización de un crimen. Por otra parte, también desde el punto de vista medioambiental el Big data se está empleando para realizar simulaciones de cuál va a ser la evolución climatológica, de las corrientes del mar, de cómo va a afectar eso a la fauna y flora, comenta que permite poner en marcha acciones preventivas para controlarla y minimizar ese impacto. Desde el punto de vista medioambiental el Big data se está empleando para realizar simulaciones de cuál va a ser la evolución climatológica, de las corrientes del mar, de cómo va a afectar eso a la fauna y flora, contaminación ambiental, etc. A nivel local, estas herramientas pueden ser muy útiles para el trabajo de campo de las ONG. Pueden aprovecharse de las redes sociales, por ejemplo, para poder identificar y

localizar a aquellas personas en riesgo de exclusión que necesitan ayuda y que se resisten a pedirla, por ejemplo. Otro ejemplo es el de la segmentación de campañas para hacer llegar a los donantes correctos las causas que de verdad le interesan, favoreciendo así las donaciones. O aplicar esa segmentación para captar a nuevos voluntarios para que participen en campañas enfocadas en sus gustos [6].

En la literatura de proyectos de investigación sobre este tema encontramos actualmente una gran variedad atacando de distintas maneras este tipo de problema, como los autores Donnelly, Misstear y Broderick que en su artículo presentan un modelo para producir pronósticos de calidad del aire en tiempo real con alta precisión y alta eficiencia computacional. Las variaciones temporales en los niveles de dióxido de nitrógeno (NO₂) y las correlaciones históricas entre la meteorología y los niveles de NO₂ se utilizan para estimar la calidad del aire 48 horas antes.

La regresión no paramétrica del núcleo se utiliza para producir factores linealizados que describen las variaciones en las concentraciones con la velocidad y dirección del viento y, además, para producir factores estacionales y diurnos. La base del modelo es una regresión lineal múltiple que utiliza estos factores junto con los parámetros meteorológicos y la persistencia como predictores.

El modelo fue calibrado en tres sitios urbanos y un sitio rural, y el modelo ajustado final logró valores R de entre 0.62 y 0.79 para pronósticos por hora y entre 0.67 y 0.84 para pronósticos máximos diarios. La validación del modelo utilizando cuatro parámetros de evaluación modelo, un índice de concordancia (IA), el coeficiente de correlación (R), la fracción de valores dentro de un factor de 2 (FAC2) y el sesgo fraccional (FB) arrojaron buenos resultados. La IA para pronósticos de 24 horas del NO₂ por hora estuvo entre 0.77 y 0.90 en sitios urbanos y 0.74 en el sitio rural, mientras que para pronósticos máximos diarios estuvo entre 0.89 y 0.94 para sitios urbanos y 0.78 para el sitio rural. Se observaron valores de R de hasta 0.79 y 0.81 y valores de FAC2 de 0.84 y 0.96 para las predicciones máximas por hora y diarias, respectivamente. El modelo solo requiere datos de entrada simples y muy pocos recursos computacionales. Resultó ser un medio preciso y eficiente para producir pronósticos de calidad del aire en tiempo real [7].

En otro artículo realizado por Adams y Kanaroglou abarcan este tema desde el uso de sistemas de mapeo de riesgo ya que estos sistemas proporcionan datos de concentración absoluta o las concentraciones se utilizan para derivar un índice de calidad del aire, que proporciona el riesgo de contaminación del aire para una mezcla de contaminantes del aire con un valor único. Cuando la información de riesgo se presenta como un valor único para una región entera, informa sobre la variación espacial dentro de la región. Sin una comprensión de la variación local, los residentes solo pueden tomar una decisión parcialmente informada al elegir las actividades diarias. El valor único generalmente se proporciona debido a un número limitado de unidades de monitoreo activas en el área. En su trabajo superan ese problema aprovechando las técnicas móviles de control de la contaminación del aire, la información meteorológica y la información sobre el uso de la tierra para cartografiar los riesgos para la salud de la contaminación del aire en tiempo real. Proponen un enfoque que pueda proporcionar información mejorada sobre el riesgo para la salud al público mediante la aplicación de modelos de redes neuronales dentro de un marco inspirado en la regresión del uso de la tierra. Las campañas móviles de monitoreo de la contaminación del aire se modelaron con una serie de variables de predicción que incluían información sobre las características del

uso de la tierra circundante, las condiciones meteorológicas, las concentraciones de contaminación del aire de los monitores de ubicación fija, e información de tráfico durante el tiempo de recolección. Se modelan partículas finas y dióxido de nitrógeno. Durante el proceso de adaptación del modelo, reservan el veinte por ciento de los datos para validar las predicciones. Los rendimientos de los modelos se miden con un coeficiente de determinación de 0.78 y 0.34 para PM_{2.5} y NO₂, respectivamente. Aplican una medida de importancia relativa para identificar la importancia de cada variable en la red neuronal para superar parcialmente los problemas de la caja negra de los modelos de redes neuronales [8].

En otro artículo los autores Nath y Patil hacen uso de la variable MH (mixing height) para representar la profundidad de dispersión de la capa límite atmosférica, es un parámetro de entrada crucial en los modelos de contaminación atmosférica. Sin embargo, existe una enorme incertidumbre en su estimación ya que no es una variable directamente medible. En general, la MH se calcula a partir de las mediciones de radiosonda dos veces al día desde la estación meteorológica más cercana, especialmente en los países en desarrollo. Sin embargo, estos valores extrapolados causan errores severos en la predicción ya que MH depende del sitio y del tiempo. En su trabajo aplican un modelo de crecimiento de altura de mezcla in situ simple (IMG, in situ mixing height growth), que puede estimar los valores de MH en tiempo real in situ a partir de mediciones superficiales de viento y temperatura fácilmente disponibles, a algunos modelos de predicción de contaminación atmosférica comúnmente utilizados. Los modelos de caja (BM, box models) los utilizan para predicciones a gran escala, pero suponen una altura tope constante, aunque su precisión depende en gran medida de su variación. IMG lo aplicaron a un modelo de caja fotoquímica, ya que la formación de ozono depende en gran medida de la insolación y está controlada por valores de MH en tiempo real. Las concentraciones de ozono estimadas por IMG-BM mostraron una mejora del 13% en comparación con las estimadas a partir de los valores de radiosonda extrapolados habituales. La aplicación de IMG a GDM para las industrias mostró que el modelo IMG mejora considerablemente la precisión de predicción y puede utilizarse de manera rentable [9].

El autor Elbir propone el modelo meteorológico CALMET y su modelo de dispersión de sople CALPUFF y se utilizaron para predecir la dispersión de las emisiones de dióxido de azufre de las fuentes de calor industriales y domésticas en Izmir, la tercera provincia más grande de Turquía. El dominio de modelado cubrió un área de 80 × 100 km centrada en el área metropolitana de Izmir con un espaciado de malla de 1000 m. Los análisis estadísticos se llevaron a cabo para evaluar el rendimiento del modelo mediante la comparación de las series temporales pronosticadas y medidas de las concentraciones de dióxido de azufre en cuatro estaciones de monitoreo utilizando dos métodos principales: raíz del error cuadrático medio (RMSE) y un índice de acuerdo (d). El índice de acuerdo varió de 0,51 a 0,77 en cuatro estaciones de monitoreo y el RMSE total varió de 0.36 a 0.66 para el año 2000. El rendimiento general del modelo para cuatro estaciones de monitoreo se encontró bueno con una precisión de aproximadamente 68%. El acuerdo de las predicciones y medidas del modelo fue mejor para dos estaciones de monitoreo urbano Karsiyaka y Bornova, en comparación con las otras estaciones urbanas Alsancak y Konak [10].

Los autores Kim et al., presentan en su artículo una propuesta para controlar la calidad del aire interior (IAQ) en las estaciones de metro, las estrategias de control basadas

en el modelo predictivo que no tiene el efecto de la temperatura debido a variaciones estacionales, se han utilizado actualmente. En este trabajo, se proponen modelos dependientes de la temporada para el monitoreo y la predicción de IAQ, que se ocupan de los cambios estacionales.

Los datos en tiempo real de diversos contaminantes (la concentración de la plataforma PM10 y PM2.5, la temperatura, la humedad y la concentración de nitrógeno) durante marzo de 2008 a febrero de 2009 se obtienen de la estación de metro de Seúl. La prueba MANOVA se ha llevado a cabo para conocer la medida cuantitativa de las diferencias entre diferentes conjuntos de datos de tres estaciones (primavera y otoño, verano e invierno).

Los métodos de regresión de PCA y PLS se aplican en conjuntos de datos de un año (para desarrollar un modelo global) y cuatro temporadas (para desarrollar modelos estacionales) para monitorear y predecir el IAQ. Los resultados de este estudio muestran que los modelos estacionales pueden predecir los datos futuros de PM10 y PM2.5 antes que el modelo global [11].

En otro artículo relacionado los autores Han et al. diseñan un marco para recopilar y almacenar diversos dominios de datos sobre las causas de las enfermedades cardiovasculares, y construyen una base de datos integrada de big data. Una variedad de bases de datos de código abierto se integró y migraron a dispositivos de almacenamiento distribuidos. La base de datos integrada estaba compuesta de datos clínicos sobre enfermedades cardiovasculares, encuestas nacionales de salud y nutrición, información estadística, censos de población y vivienda, datos de administración meteorológica y datos del servicio de evaluación y evaluación del seguro médico.

El marco se compone de datos, velocidad, análisis y capas de servicio, todos almacenados en dispositivos de almacenamiento distribuidos. Finalmente, propusieron un marco para un sistema de predicción de enfermedades cardiovasculares basado en la arquitectura lambda para resolver los problemas asociados con los análisis en tiempo real de big data. Este sistema puede usarse para ayudar a predecir y diagnosticar enfermedades, como enfermedades cardiovasculares [12].

A diferencia de los trabajos que se describen en esta sección, con este proyecto se pretende sentar las bases para establecer estaciones de sensores que puedan estar continuamente monitoreando las principales partículas contaminantes a través de un sistema big data que recolecte toda la información en un sistema central e inmediatamente se procese la información para predecir por algunos periodos de tiempo establecidos el comportamiento de la contaminación en base a los registros actuales e históricos.

3. Componentes de predicción

Existe una distinción y separación de las partes de un conjunto de datos hasta llegar a conocer sus elementos basados en su entorno. Cada uno de los datos tiene características propias que los determinan y distinguen claramente de los demás. De ellos se eligen uno entre otros con el objetivo de predecir la magnitud física que mide la sensación subjetiva de calor o frío. El poder generar un aviso previo de un hecho que va a suceder para poder evitar una proximidad de un daño o peligro ordenado por clases. Antes de entrar en concreto con términos de la predicción es necesario procesar la información en diferentes etapas, para las cuales los datos que describen el comportamiento del set

de datos que fueron generados y que son importantes junto con las características seleccionadas para los procedimientos de descubrimiento de escenarios en el entorno de trabajo. En seguida se da una explicación de los métodos que se están usando:

Bdscan. Para la estimación del número de escenarios para la identificación de multiescenarios. Este tiene un enfoque basado en densidad, modelando los clusters como cúmulos de alta densidad de puntos. Por lo cual, si un punto pertenece o no a un clúster, debe estar cerca de un conjunto de otros puntos de dicho clúster. Haciendo uso de las características seleccionadas, hará el descubrimiento de los diferentes escenarios que se encuentran en un entorno de trabajo. Esta sub etapa tiene la función también de poder detectar outliers que pueden afectar el descubrimiento de estos multiescenarios, los cuales son descartados para repetir el proceso con el conjunto de datos limpio. Como resultado de esta sub etapa se tiene un conjunto de datos representativos de las variables por cada sensor exentos de outliers y un parámetro de estimación del número de clusters identificados.

K-means. Para la partición del entorno por agrupamientos. Este tiene como objetivo la partición de un conjunto de n observaciones (representaciones estadísticas de las variables promedio por día de cada sensor en su ubicación geográfica original) en k grupos en el que cada observación pertenece al grupo cuyo valor medio es el más cercano. El parámetro k es el valor resultante de la estimación del número de clusters identificados.

Chi-Square. Para la selección de características. Es evidente que el uso de todas las características de cada escenario permiten describirlo completamente, analizando el comportamiento se puede entender que no es necesario tener toda la información para poder obtener una buena representación del entorno, además el uso de esfuerzo computacional se puede ahorrar cuando se seleccionan los mejores representantes de un escenario del entorno de trabajo. Es por ello que cada agrupamiento es enviado a esta sub etapa para la selección de las mejores características (sensores con su información) que representar mejor al escenario.

LSTM. Para la predicción de las variables en estudio. Para la tarea de predicción se emplean redes de tipo Long Short Term Memory (LSTMs), siendo éstas un caso especial de redes neuronales tradicionales y de tipo RNN. Este tipo de especial de redes neuronales son ampliamente utilizadas en problemas de predicción en series temporales debido a que su diseño permite recordar la información durante largos períodos y facilita la tarea de hacer estimaciones futuras empleando períodos de registros históricos.

4. Flujo de datos

Para poder procesar el set de datos, es necesario seleccionar el mejor conjunto que tenga un periodo con la mayor cantidad de datos, este periodo seleccionado se vuelve el principal insumo para cada proceso. El siguiente paso es la realización de una reconstrucción de datos usando la técnica MICE (Multivariate Imputation by Chained Equations), con la información completa se realiza una prueba de normalidad para descubrir si el conjunto de datos tiene una distribución normal, posteriormente el conjunto de datos completo ahora puede ser analizado para entender mejor el comportamiento

que tiene obteniendo la información estadística del mismo, dichos resultados estadísticos, proveen la información necesaria para el descubrimiento de los escenarios que se encuentran en el entorno de trabajo.

En términos de los componentes de predicción, el uso de DBSCAN se aplica con 3 características representativas de los diferentes conjuntos de datos generados por cada sensor, para lo cual es indispensable tener la ubicación de los sensores que captan la información y el mejor representante de la información por día, usando DBSCAN con estos parámetros se pueden identificar un número N aproximado de agrupamientos, que a continuación se ratifican usando K-Means.

Hasta este punto el conjunto de datos no ha cambiado, el conjunto de datos completo es necesario para las etapas más importantes, el conjunto de datos estadísticos representativos del conjunto de datos general que se usa en esta etapa y permite generar un mejor desempeño para el procesamiento de la información más adelante.

Como resultado del descubrimiento de los multiescenarios, el conjunto de datos completos se particiona según los multiescenarios descubiertos generando diferentes subsets de datos. A continuación, se hace uso del método de selección por Chi Square, tomando, de cada subset generado, los mejores representantes, reduciendo así la cantidad de información de cada sensor en el periodo de tiempo seleccionado. Cada subset reducido de cada cluster ahora está listo para poder ser procesado por el modelo de predicción, el cual toma la información para entrenar el modelo, una vez entrenado el modelo, ya está listo para poder procesar la información del entorno en términos reales, generando ventanas de predicción de un intervalo de tiempo en el futuro definido por el usuario.

Como resultado de esta predicción se genera el catálogo de documentos de registros de predicción, que es enviado a un predictor de riesgos que evalúa cada documento dentro del catálogo para identificar si pertenece a un conjunto que se considera de riesgo o a un conjunto que se considera dentro de los intervalos establecidos como normales. El resultado de este procesamiento final de la información es la probabilidad existente en el catálogo de documentos de registro de predicción de que en algún momento en el futuro se registre un evento de riesgo.

5. Modelo de predicción

La base de datos fue etiquetada de acuerdo al semáforo de calidad del aire del SEICA que se divide en cinco categorías (Buena, Satisfactoria, No satisfactoria, Mala y Muy mala), para cada una de las instancias se verifica a que categoría pertenece cada uno de los cinco contaminantes obteniendo así cinco posibles categorías, después la categoría asignada a la instancia la peor de estas cinco. Los límites para definir a que categoría pertenece cada contaminante se encuentran especificados en la tabla 1. En los años 2014 al 2017 no existieron registros que pertenecían a la categoría “Mala” y “Muy Mala”.

Tabla 1. Clasificación de las cinco variables contaminantes.

Contaminante	PM ₁₀	O ₃	SO ₂	NO ₂	CO
Unidad de medida	Ug/m ³	Ppb	Ppb	Ppb	Ppm
Bueno	0-54	0-64	0-99	0-198	0-9
Satisfactoria	55-74	65-69	100-109	190-209	9-10

No satisfactoria	75-174	70-130	110-174	210-315	11-15
Mala	175-274	131-184	175-239	316-420	16-22
Muy mala	>275	>185	>240	>420	>22

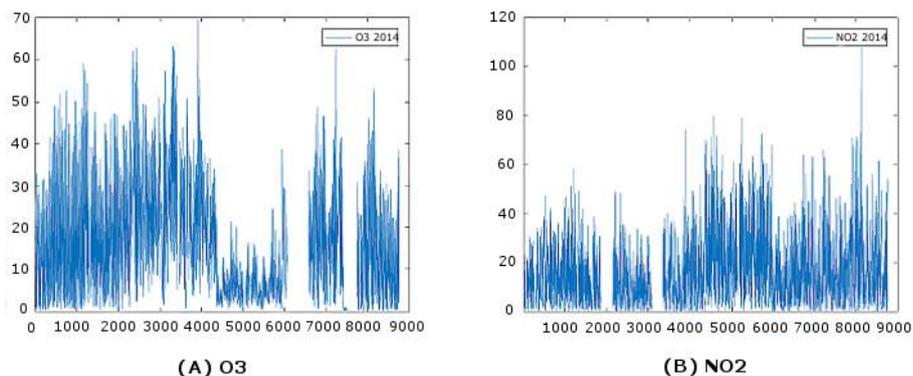


Fig. 3. Comportamiento del O3 y del NO2 en 2014.

Las bases de datos cuentan con un histórico del año 2014, 2015, 2016 y 2017. Cada año tiene registrado el mes, el día y la hora (de la hora 0 hasta la hora 23), es decir que en un año tenemos aproximadamente entre 8757 registros y 8781 (por ser año bisiesto). En cada una de las variables censadas de todos los años, existen valores perdidos como se muestra en la figura 3 (inciso a) O3 e inciso b) NO2), donde se puede mostrar que el eje x representa el periodo (las horas) y en el eje y el valor que toma cada uno de los contaminantes.

Se realizó el mismo método de graficado en todos los años para ver el comportamiento de cada una de las variables. También se realizó un estudio más minucioso para ver la distribución de los datos.

6. Resultados

Después de haber extrapolado los datos faltantes, se aplica una operación de retraso para transformar la base de datos [13]. Es decir, para predecir la calidad del aire de la próxima hora se tienen que incluir en el procesamiento no solo los datos actuales sino también datos pasados, los últimos dos o tres registros pasados como se muestra en la tabla 2 donde la transformación de la base de datos para predecir la etiqueta (calidad del aire) de la instancia 4, se utilizan los atributos de las instancias 1, 2 y 3 como se muestra en la tabla 3.

Tabla 2. Base de datos antes de ser transformada.

Instancia	Atr1	Atr2	Atr3	Atributo
1	A1	B1	C1	E1
2	A2	B2	C2	E2
3	A3	B3	C3	E3
4	A4	B4	C4	E4

Tabla 3. Transformación de la base de datos.

Atr1	Atr2	Atr3	Atr4	Atr5	Atr6	Atr7	Atr8	Atr9	Etiqueta
A1	B1	C1	A2	B2	C2	A3	B3	C3	E4

La base de datos fue dividida en cuatro bases de datos (una por cada año) y se entrenaron cuatro modelos de MLP, obteniendo así resultados por cada uno de los años.

La tabla 4 muestra los resultados de precisión por la base de datos de cada año, utilizando las 3 técnicas utilizadas para el rellenado de datos y una de las técnicas con alguna modificación (el reetiquetado), dando como mejores resultados de porcentaje de clasificación la técnica de múltiple imputación con reetiquetado.

Tabla 4. Resultados de la clasificación con diferentes técnicas.

Año	Ins-tancia	Clasificación con extrapo-lación	Clasificación con técnica de pro-medio móvil	Clasificación con impu-tación	Clasificación con imputación reetiquetado
2014	8757	93.11%	93.20%	93.38%	97.56%
2015	8757	95.98%	93.47%	92.88%	96.25%
2016	8781	94.20%	91.49%	89.65%	96.70%
2017	8757	91.46%	94.17%	90.67%	95.78%

7. Conclusiones

Los resultados obtenidos permiten tener confianza en las predicciones y así poder mantener informada a las personas sobre los cambios en los niveles de la calidad del aire en un futuro cercano, se realizaron ejemplos de cómo obtener información y procesarla en tiempo real para calcular su predicción, esto debido a que se contaba con una estación de monitoreo, pero no con todos los sensores necesarios para calcular la calidad del aire, ya que algunos no se pudieron conseguir, pero con los que se tenía los resultados fueron aceptables como se pudo observar.

Referencias

1. Yang, X., Du, J., Liu, S., Li, R., Liu, H.: Air pollution source estimation profiling via mobile sensor networks. In: 2016 International Conference on Computer, Information and Telecommunication Systems (CITS). <https://doi.org/10.1109/CITS.2016.7546456> M4 Citavi (2016)
2. Bose, B.: Global Warming: Energy, Environmental Pollution, and the Impact of Power Electronics. IEEE Industrial Electronics Magazine 4(1), 6–17. <https://doi.org/10.1109/MIE.2010.935860> (2010)
3. Guo, D., Zhang, Y., He, L., Zhai, K., Tan, H.: Chebyshev-polynomial neuronet, WASD algorithm and world population prediction from past 10000-year rough data. In: Proceedings of the 2015 27th Chinese Control and Decision Conference, CCDC 2015, pp. 1702–1707 <https://doi.org/10.1109/CCDC.2015.7162194> (2015)
4. Garrido-Lestache, J.S., Rodríguez García, V.: Las enfermedades alérgicas. Libro de las enfermedades alérgicas de la fundación BBVA (2012)

5. Rayón, A.: Tecnologías de ingesta de datos en proyectos “big data” en tiempo real, <https://blogs.deusto.es/bigdata/tecnologias-de-ingesta-de-datos-en-proyectos-big-data/>. Último acceso: 2019/01/01
6. Albendea, G.L.: Big data: una herramienta de predicción útil para el sector social, https://www.compromisoempresarial.com/innovacion_social/2017/-11/big-data-una-herramienta-de-prediccion-util-para-el-sector-social/ (2017)
7. Donnelly, A., Misstear, B., Broderick, B.: Real time air quality forecasting using integrated parametric and non-parametric regression techniques. *Atmospheric Environment* 103(2), 53–65. <https://doi.org/10.1016/j.atmosenv.2014.12.011> (2015)
8. Adams, M.D., Kanaroglou, P.S.: Mapping real-time air pollution health risk for environmental management: Combining mobile and stationary air pollution monitoring with neural network models. *Journal of Environmental Management*, 168, pp. 133–141 <https://doi.org/10.1016/j.jenvman.2015.12.012> (2016)
9. Nath, S., Patil, R.S.: Prediction of air pollution concentration using an in situ real time mixing height model. *Atmospheric Environment*, 40(20), 3816–3822 <https://doi.org/10.1016/j.atmosenv.2006.02.034> (2006)
10. Elbir, T.: Comparison of model predictions with the data of an urban air quality monitoring network in Izmir, Turkey. *Atmospheric Environment* 37(15), 2149–2157. [https://doi.org/10.1016/S1352-2310\(03\)00087-6](https://doi.org/10.1016/S1352-2310(03)00087-6) (2003)
11. Kim, M., Sankararao, B., Kang, O., Kim, J., Yoo, C.: Monitoring and prediction of indoor air quality (IAQ) in subway or metro systems using season dependent models. *Energy and Buildings* 46, 48–55. <https://doi.org/10.1016/j.enbuild.2011.10.047> (2012)
12. Han, S.H., Kim, K.O., Cha, E.J., Kim, K.A., Shon, H.S.: System framework for cardiovascular disease prediction based on big data technology. *Symmetry* 9(12), 1–11 <https://doi.org/10.3390/sym9120293> (2017)
13. Soares, F.M., Souza, A.M.F.: *Neural network programming with java*. Second edition, Editor: Packt publishing, 270 pp. (2017)

Algoritmos de aprendizaje automático de vanguardia para el diagnóstico de enfermedades

José Carlos de Jesús Montero Rodríguez, Rajesh Roshan Biswal,
Eddy Sánchez de la Cruz

Instituto Tecnológico Superior de Misantla
Misantla, Veracruz, México
182T0885@itsm.edu.mx, rroshanb@itsm.edu.mx, esanchezd@itsm.edu.mx

Resumen. En el presente trabajo se analizaron artículos sobre el uso de sistemas de aprendizaje automático utilizados para el diagnóstico de enfermedades como: cáncer de mama, cáncer de próstata, cardiovasculares, hipertensión, Parkinson, infartos, Artritis reumatoide, Triage, etc. También para la predicción de mortalidad en los hospitales y supervivencia posterior a eventos cardiovasculares, De los cuales se seleccionaron los más relevantes ya que en ellos los autores comparan el uso de diferentes algoritmos y herramientas de desarrollo dando su punto de vista de los más óptimos y con mayor índice de éxito para realizar diagnósticos basados en base de datos, antecedentes e imágenes de acuerdo sus necesidades y tipos de enfermedades.

Palabras clave: aprendizaje automático, algoritmos, médico, diagnóstico.

Advanced Machine Learning Algorithms for Disease Diagnosis

Abstract. In the present work, articles regarding the use of machine learning systems used for the diagnosis of diseases such as breast cancer, prostate cancer, cardiovascular diseases, hypertension, Parkinson's, heart attacks, rheumatoid arthritis, triage, etc. were analyzed. Also for the prediction of the mortality and survival rate after cardiovascular events, the most relevant cases were selected where the authors compare the use of different algorithms and development tools. giving their point of view of the most optimal and with a higher rate of satisfaction to make diagnoses based on data base, background and images according to their needs and types of diseases.

Keywords: machine learning, algorithms, medic, diagnostic.

1. Introducción

El diagnóstico médico es una de las tareas más complejas de realizar ya que se requiere del análisis de muchos factores como: Anamnesis (es la información

proporcionada por el paciente durante la entrevista clínica útil para analizar su situación clínica), síntomas, signos, exploración física[1]. El análisis de dichos factores en conjunto con la experiencia del médico da como resultado un diagnóstico el cual puede ser presuntivo o definitivo y en ciertos casos tiene que ser confirmados por exámenes médicos para llegar a un diagnóstico definitivo y poder tratar dicha afección. Por lo tanto, el uso de la tecnología para facilitar el diagnóstico de enfermedades es una necesidad en el siglo XXI ya que la creciente población y demanda de los servicios médicos sobrepasa la capacidad de atención de los hospitales dando como resultado tiempos de espera más largos y menos tiempo para la atención del paciente. El uso los Algoritmos de aprendizaje automático para el diagnóstico de enfermedades pueden reducir los tiempos de consulta e incluso en afecciones muy comunes evitar la consulta con el especialista ya que por medio de los sistemas se podría dar un diagnóstico certero y su tratamiento óptimo, incluso en las zonas geográficas más marginadas que no cuenta con especialistas un sistema manejado por un técnico de la salud podría diagnosticar y tratar dichas enfermedades sin la necesidad de que el paciente deje su zona.

En la última década, la tendencia de usar Algoritmos de aprendizaje automático se ha popularizado brindando muchas posibilidades para la toma de decisiones autónomas. En las áreas de la salud teniendo una notable evolución a pesar de que son áreas muy especiales debido a lo complejo que es tratar con la integridad de un ser humano, recalcando que cualquier error puede ser fatal para la salud del individuo pudiendo hasta costarle la vida.[2]

Dejando eso a un lado la evolución de los algoritmos en los últimos años han contribuido a mejorar el diagnóstico de enfermedades muy difíciles de diagnosticar como los desórdenes mentales, alzheimer e incluso poder predecir enfermedades cardiovasculares que de la manera tradicional no se podrían predecir con la misma efectividad[1]. La finalidad del siguiente trabajo es contrastar los diferentes algoritmos y herramientas utilizados por los autores en sus casos de estudios particulares donde podemos observar el índice de efectividad de los diferentes algoritmos tales como redes neuronales, redes bayesianas, K-means, razonamiento basado en casos y el uso de herramientas como weka.

2. Metodología

Se realizó una búsqueda en bibliotecas y revistas virtuales: Rev Ciencias Médicas, Google scholar, revista espacios, Pistas Educativas, European Journal of Internal Medicine. De los cuales se seleccionaron artículos literarios que no fueran mayores a 10 años de antigüedad y donde los autores abordaran el uso de algoritmos de aprendizaje automático para el diagnóstico clínico y la predicción de enfermedades a través de información brindada por los pacientes, estudios clínicos, historias clínicas y casos médicos en los cuales se pudiera cuantificar los resultados para evaluar el desempeño de los diferentes algoritmos utilizados en cada estudio a su vez se descartaron los artículos que no presentaban resultados sólidos y solo hacían propuesta de la importancia del uso de algoritmos de

aprendizaje automático para el diagnóstico médico. Para el control de la calidad de los artículos y de su información se tomó en cuenta si estaban publicados en alguna revista científica o sitio en línea con buena reputación. Se seleccionaron artículos muy variados en cuanto a las enfermedades que diagnosticaban ya que con ello se puede cuantificar cual de los algoritmos tienen mejor desempeño en diversas áreas de la salud.

3. Desarrollo

3.1. Tipos de algoritmos de aprendizaje automático para el diagnóstico de enfermedades

EL diagnóstico de enfermedades es un proceso cognitivo complejo que implica capacitación, experiencia, reconocimiento de patrones y cálculo de probabilidad condicional, entre otros componentes menos comprendidos. En las últimas décadas se han realizado varios esfuerzos por aplicar el análisis predictivo en los sistemas de salud, así como lanzar sistemas de aprendizaje automático para facilitar el diagnóstico[1]. Los algoritmos de aprendizaje automático son un área del conocimiento donde convergen diferentes técnicas, en particular la redes neuronales artificiales (RNA), los árboles de decisión (AD) Y las redes bayesianas, teniendo un gran desarrollo e impacto en diversas áreas del conocimiento, incluida la medicina. Los métodos estudiados se han utilizado en medicina con varios propósitos: clasificación de diagnóstico de enfermedades y predicción de la mortalidad de diversas patologías[1]:

1. Las redes neuronales artificiales,
2. El razonamiento basado en casos,
3. Las redes bayesianas,
4. Árboles de decisión.

Las redes neuronales artificiales han demostrado que producen buenos resultados en bases de datos con muchas instancias. Las redes neuronales no consideran la incertidumbre, actúan como una caja negra, en el sentido de que no es posible saber cómo se ha llegado a los resultados obtenidos y los nodos intermedios no pueden ser interpretados[1].

El razonamiento basado en casos se ha utilizado con éxito en el diagnóstico médico asistido, esta técnica se basa en el paradigma de resolución de problemas en el que un problema nuevo se soluciona atendiendo casos similares del pasado. El ejemplo más conocido, es el vecino más cercano, que consiste en comparar cada caso problema con una serie de casos almacenados; el caso problema nuevo y los casos almacenados consisten en un número de atributos o variables y éstos se comparan para encontrar el caso almacenado que más se parece al caso problema, de esta manera, se puede seleccionar la enfermedad conocida que más se parezca a la descripción del paciente. Una vez resuelto, este nuevo caso pasa a formar parte de la biblioteca de casos disponibles para comparar, con lo que el sistema se vuelve cada vez más sólido[1].

Las redes bayesianas calculan la probabilidad de que un caso con cierto atributo pertenezca a una categoría. En medicina se han usado para predecir recaída de enfermedad o riesgo de cáncer. El filtro no sabe esto por adelantado y tiene que aprender de la mano del usuario. El programa calcula en lo sucesivo la probabilidad de que cada atributo corresponda a una categoría o a otra. Basado en las características que se describen sobre las técnicas de IA que con frecuencia se utilizan para el diagnóstico de enfermedades, se persigue como objetivo comparar dichas técnicas para seleccionar la que mejor se ajusta al diagnóstico de enfermedades, cuando se tienen datos almacenados sobre el comportamiento de las enfermedades que con frecuencia afectan a una población en cuestión[1].

El árbol de decisión es un diagrama que representa en forma secuencial condiciones y acciones, normalmente construidos a partir de la descripción de la narrativa de un problema. Ellos proveen una visión gráfica de la toma de decisión necesaria, especifican las variables que son evaluadas, qué acciones deben ser tomadas y el orden en el cual la toma de decisiones será efectuada, y permite que quien las toma descomponga un problema complejo y grande, en varios más pequeños, proporcionando un método efectivo para la toma de decisiones debido a que: plantea el problema para que todas las opciones sean analizadas, permite considerar totalmente las posibles consecuencias de tomar una decisión y ayuda a realizar las mejores medidas sobre la base de la información existente y de las mejores suposiciones. Al tener en cuenta todos estos elementos, se decidió utilizar la técnica de árbol de decisiones como vía para garantizar un diagnóstico eficiente y ágil, lo que permitirá trabajar con metas más elevadas y necesarias como el diagnóstico en etapas tempranas de la enfermedades [27].

3.2. Algoritmos aproximados empleados en problemas de clasificación

Perceptrón multicapa (Multi-Layer Perceptron, MLP). Un MLP es una red feed-forward con aprendizaje supervisado. El tipo de aprendizaje más utilizado por un perceptrón es el método backpropagation (propagación del error hacia atrás) basado en la generalización de la regla Delta. A pesar de sus limitaciones, este método ha contribuido a ampliar de forma considerable el rango de aplicaciones de los MLPs (diagnóstico de enfermedades, clasificación de señales, simulación de la dinámica del brazo de un robot, puntuación para la solicitud de créditos, etc.). En este tipo de aprendizaje, primero se aplica un patrón de entrada como estímulo a la primera capa de las neuronas de la red, el cual se propaga a través de todas las capas ocultas hasta generar una salida (paso forward). Dicha propagación se lleva a cabo al aplicar una función de activación a la suma ponderada, que surge de la combinación lineal de las entradas y las conexiones (o pesos) sinápticos correspondientes. Luego se evalúa el error cometido al comparar el resultado obtenido por las neuronas de salida con el patrón de salida, por medio de una función de costo.

A continuación, estos errores se transmiten hacia atrás (paso backward), partiendo de la capa de salida hacia todas las neuronas de la capa intermedia que contribuyan directamente a la salida. Este proceso se repite, capa por capa,

hasta que todas las neuronas de la red hayan recibido un error que describa su aporte relativo al error total. En base a esto, se reajustan los pesos de conexión de cada neurona, de manera que la próxima vez que se presente el mismo patrón, la salida esté más cercana a la deseada. Para acelerar el proceso de aprendizaje se puede añadir un término denominado momento o momentum, el cual acelera la convergencia del algoritmo. Otra forma de mejorar el proceso de aprendizaje es adaptar la tasa de aprendizaje, η , durante dicho proceso, conocido como η adaptativo, y consiste en modificar el valor de η durante el entrenamiento con el fin de evitar que el error se estanque debido a la presencia de óptimos locales [24].

Algoritmos Genéticos (Genetic Algorithms, GAs). Los GAs son un tipo popular de Metaheurísticas, en especial de algoritmos evolutivos, que simula la evolución de individuos por medio del proceso de selección natural Darwiniano. Computacionalmente hablando, esto se refiere a que realizan una búsqueda multidireccional manteniendo una población de potenciales soluciones (individuo) y fomentando, por un lado, la formación de información (por medio de la mutación), y por otro lado, el intercambio de información (por medio del crossover) entre dichas soluciones. La selección de los individuos a reproducirse (crossover y mutación), se realiza a través de operadores de selección probabilísticos (ruleta, ranking, torneo binario, entre otros). Generación tras generación, los individuos de la población actual son sustituidos por los recién creados. Este reemplazo usualmente intenta mantener los mejores individuos (elitismo) eliminando los peores. Dependiendo de si se tiene en cuenta la antigua población se pueden obtener dos tipos de estrategias de reemplazo: $(\mu; \lambda)$ si el reemplazo se realiza utilizando únicamente a la población de hijos, y $(\mu + \lambda)$ si el reemplazo μ se realiza seleccionando individuos de la unión de la población antigua y la de hijos. Los dos GAs propuestos, GA1X y GA2X, utilizan números reales en la representación de cada individuo, la selección de los padres se realiza por medio de un torneo binario, método de selección que es utilizado también en el proceso de reemplazo $(\mu + \lambda)$ con elitismo[25]. Además aplican el método swap de un gen (elemento de la solución) elegido al azar para llevar a cabo la mutación de un individuo bajo una determinada probabilidad (P_m)[25]. La diferencia entre GA1X y GA2X radica en el método de crossover empleado. Por un lado, GA1X utiliza crossover de un punto, con el fin de testear un GA estándar. En tanto que, GA2X aplica el crossover de dos puntos, cuyo objetivo es introducir mayor variación en cada descendencia generada. Ambos operadores se aplican bajo una cierta probabilidad (P_c).

El origen de los genes de cada individuo; donde cada uno se compone de los pesos sinápticos de un MLP. Los individuos se estructuran en forma de vector, a diferencia de la estructura matricial utilizada en los MLPs. En otras palabras, cada peso de conexión representa un gen del individuo, siendo creados con valores aleatorios en el rango $[-0.5, 0.5]$ siguiendo una distribución uniforme[25].

El fitness mide la capacidad del individuo para resolver los problemas planteados, es decir, el promedio de error cometido al intentar clasificar patrones de cáncer de mama o diabetes. Mientras menor sea dicho error, el fitness del indi-

viduo será mejor. En otras palabras, la evaluación del fitness de cada individuo se corresponde con el cálculo del error medio obtenido al ejecutar sólo el paso hacia adelante del MLP durante una época [25].

4. Estado del arte

Para el diagnóstico de patologías presentes en la columna vertebral se utilizó el software weka en conjunto con: algoritmos de clasificación como LMT, Redes Bayesianas, Naive Bayes, J48 y Naive Bayes Simple para decidir cuál es el mejor de ellos para el diagnóstico de dicha enfermedad. Los resultados obtenidos durante la clasificación de los datos por medio de los algoritmos utilizados muestran que el algoritmo de decisión LMT clasificó 265 instancias correctamente y solo 45 instancias incorrectamente de un total de 310 instancias con un porcentaje de éxito de 85.483 %, y un error absoluto de 0.2168. El algoritmo Redes Bayesianas clasificó correctamente 237 de las 310 instancias con un porcentaje de éxito de 80 % y un error absoluto de 0.249; el algoritmo Naive Bayes clasificó correctamente 248 instancias con un error absoluto de 0.2, mientras que J48 clasificó 253 instancias, finalmente el algoritmo Naive Bayes Simple clasificó correctamente 241 instancias de las 310, llegando a la conclusión de que el mejor algoritmo de decisión es el LMT[3]. Un perceptrón fue utilizado con el objetivo principal de diseñar un método no invasivo para el diagnóstico de pacientes con cirrosis hepática utilizando únicamente datos de laboratorios. Como resultado del estudio tras comparar las RNA en término del rendimiento, el modelo fue capaz de reconocer 3 tipos de valores: cirrosis, no cirrosis y los no identificables con un tasa de éxito de casi el 90 %[4].

En otras áreas del campo médico se han propuesto las RNA para pronosticar la estancia hospitalaria prolongada de pacientes mayores en el servicio de urgencias, donde el mejor desempeño fue logrado por el MLP con una sensibilidad del 62.5 % y especificidad del 96.6 %[5]; asimismo, pueden predecir la estancia prolongada en la unidad de cuidados intensivos y la mortalidad con la ayuda de algoritmos como, el vecino más cercano (K-NN), SVM, árboles de clasificación (CART), entre otros. Los datos mostraron que el modelo SVM ofreció los mejores resultados con un valor del área bajo la curva de 0.77 para predecir la estancia y del 0.82 para la predicción de la mortalidad [6]. Por otra parte, en la investigación del cáncer han sido aplicadas por su habilidad para reconocer patrones complejos como, en la predicción de cáncer avanzado de vejiga en pacientes sometidos a cistectomía radical[7], cáncer de mama, el modelo fue evaluado utilizando la matriz de confusión y las curvas ROC. Con los datos de prueba, el modelo RNA implementado dió como resultado una tasa de predicción de la enfermedad del 90.5%. Con estos valores predictivos se concluye que el modelo RNA es rápido, fiable y de gran ayuda en esta especialidad médica[8], además puede predecir la supervivencia después de la recesión hepática por cáncer colorrectal. La investigación comparó el uso de las RNA con el modelo de regresión C-index encontrando que a través del análisis de 241 pacientes y de la selección de seis variables de entrada de 28. El modelo RNA predice la supervivencia individual

con mayor exactitud que el modelo de regresión de Cox, con de 0.72 frente a 0.66 [9]. Igualmente han sido utilizadas con gran éxito en predicción de mortalidad. Por ejemplo, la mortalidad hospitalaria de los pacientes con reparaciones de aneurisma aórtico abdominal roto, en el cual la RNA mostró mejor desempeño cuando esta fue comparada con los resultados obtenidos por la metodología de regresión lineal[10]. En el pronóstico de la mortalidad de cáncer de vejiga en este trabajo se utilizaron siete métodos de aprendizaje para predecir la mortalidad a los cinco años después cistectomía radical, incluyendo a la Red neuronal MLP, redes de función de base radial, la máquina de aprendizaje extremo EML, regularizados ELM (RELM), SVM, las RB naive y el clasificador del vecino más cercano (K-NN). Los resultados en Experimentales indican que RELM logra la más alta precisión en la predicción con un 0.8. Los resultados de la investigación demuestran el potencial de la aplicación de esta técnica de aprendizaje para apoyar la toma de decisiones clínicas[11]; en el cáncer de páncreas el estudio utilizó las RNA para predecir la supervivencia individual a largo plazo de los pacientes sometidos a cirugía radical en el tratamiento de este tipo de cáncer. Los resultados fueron comparados con el modelo de regresión C-index mostrando que las RNA tiene mejor desempeño con un 0.79, mientras que el otro modelo mostró un 0.67 de precisión[12]. En pacientes con accidente cerebrovasculares (isquemia y hemorragia) para el pronóstico de la mortalidad a los 10 días después del evento. La investigación aplicó redes MLP y los métodos estadísticos multivariable (Análisis discriminante multivariado [MDA] y análisis de regresión logística [LRA]. El MLP tiene una la clasificación del 100 %, para ambos grupos isquémicos y hemorrágicos en la fase de entrenamiento. Los métodos LRA y MDA mostraron resultados exitosos, a pesar de que estos resultados son más bajos el MLP tanto para el grupos hemorrágico y como para el grupo isquémico en la fase de entrenamiento. Sin embargo, cuando se compararon los resultados de los métodos en la fase de prueba, la mejor clasificación, especificidad, sensibilidad y precisión del 87.8 % fue alcanzado por el modelo MDA para el grupo hemorrágico. La mejor clasificación, especificidad 75.9 %, sensibilidad 85.9 % y precisión 80.9 % para el grupo isquémico fue alcanzado por la técnica LRA[13].

Por otro lado el estudio de colak y cols (2015)[14] mostró buenos resultados en la predicción de accidentes cerebrovasculares aplicando el MLP, las máquinas de vectores y las RBF, los cuáles mostraron una precisión del 80 %. Las RPF también ha mostrado un buen desempeño en el diagnóstico de enfermedades del corazón[15], en el diagnóstico y análisis de la fibrilación articular[16]. Debido a su excelente desempeño el tratamiento de imágenes las redes SOM han aplicado en la segmentación de imágenes médicas (segmentación de tumores de mamas[17] y cerebrales[18]), además para la identificación de desórdenes como la demencia, esquizofrenia o el alzheimer[19]. Los AD que tienen entre otras características su aporte visual para la toma de decisiones, han sido utilizados para identificar los factores de riesgos asociados con la hipertensión arterial. Los modelos fueron evaluados con 9078 sujetos divididos en dos grupos (hipertensos y no hipertenso). Los resultados para el primer modelo fueron 73 % de sensibilidad, 63 % especificidad y 77 % de precisión. Los valores correspondientes para el segundo

modelo fueron 70 %, 61 %, 74 % respectivamente. Según los autores el estudio proporcionó reglas de clasificación fáciles y comprensibles para identificar los factores de riesgo asociado con la hipertensión y esto puede ser útil para el desarrollo de programas para el manejo de hipertensión[20]. Cómo ayuda en la toma de decisiones en paciente con soporte ventricular izquierdo, se estudiaron 183 pacientes con fallas ventriculares derechas, de los cuales 27 requirieron soporte ventricular izquierdo, los restantes continuaron con manejo médico hasta el trasplante o fallecimiento. El modelo identificó 8 variables preoperatorias óptimas. El árbol de decisiones resultante que constaba de 28 ramas y 15 hojas, identificó a los pacientes que necesitaban soporte ventricular con una sensibilidad del 85 % y un área bajo la curva de 0.87[21]. En el pronóstico del cáncer pleural[22], además, en la predicción de los síntomas de Parkinson[23].

5. Estudios y contribuciones

Araújo, A. et al. Comparar diferentes algoritmos de aprendizaje automático para la clasificación de quistes mamarios a través de imágenes termográficas usando redes neuronales artificiales, árboles de decisiones, clasificadores bayesianos y atributos de Haralick y Zernike. La base de datos de imágenes está compuesta por termografía, las imágenes fueron adquiridas en el Hospital Universitario de la Universidad Federal de Pernambuco. Estas imágenes son clínicamente clasificadas en las clases quistes malignos y benignas. Momentos de Zernike y Haralick fueron utilizados como atributos. Las máquinas de aprendizaje extremo (ELM) y las redes de Perceptrón multicapa (MLP) demostraron ser bastante eficientes clasificadores para la clasificación de lesiones mamarias en imágenes termográficas. Utilizando el 75 % de la base de datos para la formación, el valor máximo obtenido por precisión fue de 73.38 %, con un índice Kappa de 0.6007. Este resultado indica a un Sensibilidad del 78 % y especificidad del 88 %. La eficiencia global del sistema fue del 83 %. Conclusión: ELM mostró ser un clasificador prometedor para ser usado en la diferenciación de lesiones de mama en imágenes termográficas, debido a su bajo coste computacional y robustez[26].

González, F., Análisis de diferentes artículos sobre Machine learning models in rheumatology y cómo evolucionaron desde los años 1989 hasta el 2015 sintetizando los algoritmos más con mejores resultados para el diagnósticos de enfermedades en esta rama de la medicina con el uso aprendizaje supervisado como las redes neuronales y las redes bayesianas, y de aprendizaje no supervisado como el método de agrupamiento k-medias. Adicionalmente, para la identificación de endofenotipos los autores utilizaron alineamiento de secuencias de aminoácidos. Los resultados presentados en el artículo son bastante prometedores: los modelos predictivos logran un alto desempeño en la tarea de discriminar entre pacientes con artritis reumatoide y controles, con sensibilidad y especificidad de hasta el 92,3 % y el 93,3 %, respectivamente. Por otro lado, el análisis de conglomerados basado en k-medias permitió establecer 2 grupos de pacientes con diferentes niveles de severidad. Finalmente, el análisis de secuencias de aminoácidos reveló secuencias comunes en pacientes con desenlaces iguales[28].

Morales, L., Comparación de modelos de inteligencia computacional para la clasificación e identificación de endofenotipos (relación entre fenotipo y marcadores genéticos) en pacientes con artritis reumatoide y controles sanos, a partir de información genética, principalmente el HLA DRB1 (antígeno leucocitario humano) y la teoría del epítoto compartido. Esto hace referencia a la asociación entre la artritis reumatoide y el HLA DRB1, principalmente los alelos que contienen un motivo común de aminoácidos de las secuencias QKRAA, QRRAA o RRRRAA en las posiciones 70 a 74 de la cadena del DRB1, los cuales confieren una susceptibilidad particular de la enfermedad a los individuos con técnicas de inteligencia computacional como son las redes neuronales, redes bayesianas y métodos como k-means. Como datos de entrada se utilizaron variables como: factor reumatoide, anticuerpos contra péptido citrulinado, proteína C reactiva, número de articulaciones inflamadas y dolorosas, rigidez matinal, edad, género, antecedentes de comorbilidades y la información del alelo HLA DRB1. Se utilizaron técnicas de bioinformática para la búsqueda de secuencias de aminoácidos relacionadas con la severidad de la enfermedad.

Se obtuvieron resultados importantes para el diagnóstico de la enfermedad, así como también para su categorización y como potencial aplicación en la medicina personalizada de los individuos afectados por esta enfermedad. Se diseñaron modelos computacionales para clasificación, dentro de los cuales la red neuronal utilizando 5 variables obtuvo una sensibilidad del 92,3 % con una especificidad del 86,66 % y la red bayesiana logró una sensibilidad del 92,3 % y una especificidad del 93,33 %. La red neuronal utilizando 11 variables obtuvo una sensibilidad del 84,61 % y una especificidad del 93,33 % mientras que la red bayesiana consiguió una sensibilidad del 92,3 % con una especificidad del 93,33 %. Un método de agrupamiento tipo k-means fue utilizado para dividir en 2 grupos la totalidad de pacientes y controles. Además, se logró separar 2 grupos entre los pacientes para definir su severidad. Finalmente, se consiguió obtener un árbol de distancia entre secuencias de aminoácidos de los distintos alelos HLA DRB1, lo cual permite visualizar grupos con cercanía genética y lograr determinar que posiblemente existen más grupos fuera de los propuestos en las distintas teorías[29].

María, M., Comparación de diferentes algoritmos de aprendizaje automático para el diagnóstico del triaje en el servicio de urgencias a través de 1000 casos para realizar la evaluación de los modelos, que será obtenida a partir de casos prácticos en salas de urgencias de hospitales de Buenos Aires (se encuentran entre los potenciales interesados el Hospital Posadas, Hospital de Haedo, Hospital Italiano, Hospital Alemán, entre otros). En el estudio realizado hasta el momento se ha hecho una simulación con 124 casos. Se han escogido 40 variables que han sido seleccionadas a criterio de los expertos. En la figura 2 (b) se puede ver el clasificador bayesiano aprendido a partir de los datos con el algoritmo K2 (Cooper y Herskovits, 92). Se trata de un algoritmo para el aprendizaje automático de redes bayesianas.

A partir de la red obtenida, el clasificador de la variable “traje” puede ser obtenido con la sencilla selección de las variables que forman el manto de

Markov (Sebastiani et al., 2005b) de dicha variable. Se ha utilizado el software Bayes Discoverer (Sebastiani et al., 2005b). En un segundo paso se ha utilizado directamente un clasificador bayesiano: el clasificador bayesiano simple, sobre las mismas 40 variables. Para analizar los datos de la simulación se han utilizado los siguientes algoritmos de aprendizaje automático: C45 (Quinlan, 1996): Basado en la creación de un árbol de decisión, C45 es el más aplicado entre los de su clase (C45). El vecino más cercano: Un algoritmo basado en instancias (Aha, et al., 1991). Se usarán los parámetros $k=1$ y $k=5$ (VMC1, VMC5). K2: Algoritmo genérico para creación de redes bayesianas. Clasificador bayesiano simple (Naive Bayes classifier): Específico para clasificadores, con estimación de máxima verosimilitud (BS0) y estimador bayesiana con factor $\alpha=1$ (BS1). De cada uno de ellos se ha obtenido la precisión (en porcentaje de aciertos) utilizando validación cruzada de 5 hojas. La figura 1 muestra la precisión para cada uno de los métodos empleados, ordenador por la misma. Como puede observarse, todos los algoritmos de redes bayesianas utilizados obtienen mejor precisión [30].

Algoritmo	Precisión
BS1	87,90%
K2	86,99%
BS0	86,23%
C45	79,67%
VMC5	80,63%
VNC1	80,50%

Fig. 1. Resultado de los algoritmos para el diagnóstico del triaje en el servicio de urgencias [30].

J. John, Entrenamiento y prueba de distintas redes neuronales para el diagnóstico de infarto agudo de miocardio, se realizó el entrenamiento y la prueba de varias redes neuronales, con diferentes arquitecturas para el diagnóstico del infarto, a partir de los datos de la escala de clasificación de la probabilidad de angina de Braunwald en un grupo de pacientes que ingresaron por dolor torácico al servicio de urgencias del Hospital San José de Bogotá. Se generaron 40 redes que fueron probadas en 5 experimentos de los cuales se obtuvo mayor precisión diagnóstica con el modelo de 5 entradas electrocardiográficas más troponina, aunque el mejor valor predictivo negativo se alcanzó en el modelo con 10 variables clínicas, electrocardiográficas y troponina. Varias de las redes diseñadas tuvieron una sensibilidad y una especificidad del 100%. Se requiere un estudio de validación para comprobar estos hallazgos[31].

6. Análisis de los algoritmos de aprendizaje automático

Con la revisión de los trabajos anteriormente mencionados se puede tener una visión de cuáles son los algoritmos que tiene un mejor desempeño para

el diagnóstico de enfermedades, teniendo en cuenta que dichos valores varían dependiendo de la enfermedad a diagnosticar como se muestra a continuación en la figura 2 seleccionando el los mejor algoritmos para cada enfermedad.

Enfermedad	Algoritmos	Tasa de éxito
Patologías de columna vertebral	LMT	85.48 %
Cirrosis hepática	RNA	90 %
Cáncer de mama	RNA	90.5 %
Aneurisma aórtico abdominal roto	RNA	84 %
Fibrilación auricular	RBF	95%
Artritis reumatoide	RB	93.33%
Hipertensión arterial	AD	77%
Diagnóstico del triaje	BS1	87.90%

Fig. 2. Resultado de los algoritmos para el diagnóstico de enfermedades.

Haciendo un comparación de los artículos evaluados observamos que la mayoría de los autores que usan RNA para el diagnóstico de enfermedades y predicciones de mortalidad obtiene un mejor desempeño con la RNA. El acercamiento a la metodología basada en redes neuronales artificiales puede hacerse con dos perspectivas. Por una parte, desde sus ventajas: las redes son capaces de trabajar sin las restricciones de los modelos estadísticos detectando las relaciones no lineales y las interacciones entre las variables predictoras. Y por otra, asumiendo sus desventajas: mayor complejidad de interpretación de sus parámetros de funcionamiento, mayor necesidad de recursos informáticos, alto componente empírico en su construcción y mayor dificultad de exportar el modelo para aplicarlo a otras poblaciones. Estas desventajas influyen en una menor difusión de la técnica para su uso habitual.

En general, una RNA es potencialmente más precisa que las técnicas estadísticas cuando la variable pronóstica se expresa como una función compleja de las variables predictoras o cuando existe interdependencia entre éstas; pero son estructuralmente complicadas y sus parámetros son más difíciles de interpretar.[32].

7. Conclusión

Como se observa el uso de algoritmos de aprendizaje automático para el diagnóstico de enfermedades a evolucionado de manera notable a lo largo de estos años destacando las RNA, RB, MLP y MLT siendo de gran utilidad para la predicción y el diagnóstico de enfermedades como cáncer de mama, cáncer de próstata, cardiovasculares, hipertensión, Parkinson, infartos. Aportando un

apoyo para los médicos y los pacientes ya que gracias a estos se pueden prevenir y tratar adecuadamente antes de que presenten complicaciones mayores mejorando notoriamente las probabilidades de supervivencia en dichos casos. También han sido de ayuda en la área de urgencias de los hospitales al realizar el diagnóstico de los triajes de los pacientes que llegan y poder reducir tiempos de valoración para poder asignarles un número de atención correcto y optimizar el proceso de urgencias que es una de las áreas más de mandadas en los hospitales dejando los procesos rudimentarios que entorpecen la atención.

Por otra parte, evaluando los algoritmos y viendo sus resultados concluyendo de que uno de los modelos más óptimos para realizar dicha tarea es el MLP, sin dejar a un lado que al ser un área compleja y con muchas posibilidades pueda que en otros casos sean más óptimos otros algoritmos, dejando en claro que al momento de desarrollar un sistema de diagnóstico inteligente lo más viable es evaluar diferentes métodos de acuerdo a la afección que se desee predecir o diagnosticar.

Generando un interés para creación de sistemas que puedan ayudar a mejorar los tiempos de atención en los hospitales. Una propuesta sería la creación de un sistema experto para el diagnóstico de enfermedades obstétricas en el hospital regional de alta especialidad de la mujer para mejorar la calidad de la atención y poder agilizar procesos ya que dicho hospital es el más importante de Tabasco y año con año su demanda aumenta y cada vez se hace más difícil la atención de los pacientes de manera eficiente generando situaciones particularmente complicadas.

Referencias

1. Gonzalez Benitez, N., Estrada Senti, V., Febles Estrada, A.: Estudio y selección de las técnicas de Inteligencia Artificial para el diagnóstico de enfermedades. *Rev Ciencias Médicas*, Pinar del Río 22(3), 131–141 (2018)
2. Leyva Vázquez, G.M., González Benítez, N., Hechavarría Hernández, J., Rivero Peña, Y., Daher Nader, J.E.: El diagnóstico de enfermedades desde el Análisis Inteligente de los Datos. *Revista espacios* 39(28), 16 (2018)
3. Ramírez Pérez, N.V., Laguna Estrada, M., De Anda Tovar, A.M.: Aplicación De Métodos De Inteligencia Artificial en el Área Médica. *Pistas Educativas* 35(111), 124 (2015)
4. Bostan, V.M., Pantelimon, B.: Creating a Model based on artificial neural network for liver cirrhosis diagnose. In: 9th International Symposium on Advanced Topics in Electrical Engineering (ATEE), pp. 295–298 (2015)
5. Launay, C.P., Rivière, H., Kabeshova, A., Beauchet, O.: Predicting Prolonged Length of Hospital Stay in Older Emergency Department Users: Use of a Novel Analysis Method, the Artificial Neural Network. *European Journal of Internal Medicine* (2015)
6. Houthoofd, R., Ruyssinck, J., Van der Herten, J., Stijven, S., Couckuyt, I., Gadeyne, B., Ongenaes, F., Colpaert, K., Decruyenaere, J., Dhaene, T., De Turck, F.: Predictive Modelling of Survival and Length of Stay in Critically Ill Patients Using Sequential Organ Failure Scores. *Artificial Intelligence in Medicine* 63(3), 191–207 (2015)

7. Vukicevic, A.M., Jovicic, G.R., Stojadinovic, M.M., Prelevic, R., Filipovic, N.D.: Evolutionary Assembled Neural Networks for Making Medical Decisions with Minimal Regret: Application for Predicting Advanced Bladder Cancer Outcome. *Expert Systems with Applications* 41(18), 8092–8100 (2014)
8. Saritas, I.: Prediction of Breast Cancer Using Artificial Neural Networks. *Journal of Medical Systems* 36(5), 2901–2907 (2012)
9. Spelt, L., Nilsson, J., Andersson, R., Andersson, B.: Artificial Neural Networks a Method for Prediction of Survival Following Liver Resection for Colorectal Cancer Metastases. *European Journal of Surgical Oncology. The Journal of the European Society of Surgical Oncology and the British Association of Surgical Oncology* 39(6), 648–54 (2013)
10. Wise, E.S., Hocking, K.M., Brophy, C.M.: Prediction of In-hospital Mortality After Ruptured Abdominal Aortic Aneurysm Repair Using an Artificial Neural Network. *Journal of Vascular Surgery* 62(1), 8–15 (2015)
11. Wang, G., Lam, K., Deng, Z., Choi, K.: Prediction of Mortality After Radical Cystectomy for Bladder Cancer by Machine Learning Techniques. *Computer in Biology and Medicine* 63(C), 124–132 (2015)
12. Ansari, D., Nilsson, J., Andersson, R., Regnér, S., Tingstedt, B., Andersson, B.: Artificial Neural Networks Predict Survival from Pancreatic Cancer After Radical Surgery. *American Journal of Surgery* 205(1), 1–7 (2013)
13. Çelik, G., Baykan, O. K., Kara, Y., Tireli, H.: Predicting 10-day Mortality in Patients with Strokes Using Neural Networks and Multivariate Statistical Methods. *Journal of Stroke and Cerebrovascular Diseases. The Official Journal of National Stroke Association* 23(6), 1506–1512 (2014)
14. Colak, C., Karaman, E., Turtay, M.G.: Application of Knowledge Discovery Process on the Prediction of Stroke. *Computer Methods and Programs in Biomedicine* 119(3), 181–185 (2015)
15. Alsalamah, M., Amin, S., Halloran, J.: Diagnosis of Heart Disease by Using a Radial Basis Function Network classification technique on patients' medical records. In *RF and Wireless Technologies for Biomedical and Healthcare Applications (IMWS-Bio)*, IEEE MTT-S International Microwave Workshop Series on, pp. 1–4 (2014)
16. Mateo, J., Rieta, J.J.: Radial Basis Function Neural Networks Applied to Efficient QRST Cancellation in Atrial Fibrillation. *Computers in Biology and Medicine* 43(2), 154–163 (2013)
17. Torbati, N., Ayatollahi, A., Kermani, A.: An Efficient Neural Network Based Method for Medical Image Segmentation. *Computers in Biology and Medicine* 44, pp. 76–87 (2014)
18. Mei, P., de Carvalho Carneiro, C., Fraser, S.J., Min, L., Reis, F.: Analysis of neoplastic lesions in magnetic resonance imaging using selforganizing maps. *Journal of the Neurological Sciences* 359(1), 78–83 (2015)
19. Ortiz, A., Gorriz, J.M., Ramirez, J., Salas-Gonzalez, D.: Improving MR Brain Image Segmentation Using Self-organising Maps and Entropy-gradient Clustering. *Information Sciences* 262, pp. 117–136 (2014)
20. Tayefi, M., Esmaili, H., Karimian, M.S., Zadeh, A.A., Ebrahimi, M., Safarian, M., Nematy, M., Parizadeh, S.M., Ferns, G.A., Ghayour-Mobarhan, M.: The Application of a Decision Tree to Establish the Parameters Associated with Hypertension. *Computer Methods and Programs in Biomedicine* (2016)
21. Wang, Y., Simon, M.A., Bonde, P., Harris, B.U., Teuteberg, J.J., Kormos, R.L., Antaki, J.F.: Decision Tree for Adjuvant Right Ventricular Support in Patients Receiving a Left Ventricular Assist Device. *The Journal of Heart and Lung Transplantation* 31(2), 140–149 (2012)

22. Brims, F., Meniawy, T.M., Duffus, I., de Fonseca, D., Segal, A., Creaney, J., Maskell, N., Lake, R.A., de Klerk, N., Nowak, A.K.: A Novel Clinical Prediction Model for Prognosis in Malignant Pleural Mesothelioma Using Decision Tree Analysis. *Journal of Thoracic Oncology: Official Publication of the International Association for the Study of Lung Cancer* 11(4), 573–582 (2016)
23. Exarchos, T.P., Tzallas, A.T., Baga, D., Chaloglou, D., Fotiadis, D.I., Tsouli, S., Diakou, M., Konitsiotis, S.: Using Partial Decision Trees to Predict Parkinson's Symptoms: A New Approach for Diagnosis and Therapy in Patients Suffering from Parkinson's Disease. *Computers in Biology and Medicine* 42(2), 195–204 (2012)
24. Hertz, J., Palmer, R.G., Krogh, O.S.: *Introduction to the Theory of Neural Computation*. Perseus Publishing (1991)
25. Talbi, E.G.: *Metaheuristics: From Design to Implementation*. Wiley y Sons, USA (2009)
26. Araujo, A.: Breast cancer diagnosis based on mammary thermography and extreme. *Learning Machines* (2018)
27. Solarte Martínez, G.R., Soto Mejía, J.A.: Árboles de decisiones en el diagnóstico de enfermedades cardiovasculares. *Scientia Et Technica* XVI (2011)
28. Gonzalez, F.A.: Modelos de aprendizaje computacional en reumatología. *Rev. Colomb. R eumatol. [online]* 22(2), 77–78 (2015)
29. Morales, L.: Modelo computacional para la identificación de endofenotipos y clasificación de pacientes con artritis reumatoide a partir de datos genéticos, serológicos y clínicos, utilizando técnicas de inteligencia computacional. *Revista Colombiana de Reumatología* 22(2), 90–103 (2015)
30. Abad-Grau, M.M.: Aplicación de Redes Bayesianas en el Modelado de un Sistema Experto de Triage en Servicios de Urgencias Médicas. En: IX Workshop de Investigadores en Ciencias de la Computación, pp. 43–47 (2007)
31. John, J.: Diagnóstico automático del síndrome coronario agudo utilizando un sistema multiagente basado en redes neuronales. *Revista Colombiana de Cardiología* 24(3), 255–260 (2017)
32. Álvarez, M., Nava, J.M., Rue, M, Quintana, S.: Mortality prediction in head trauma patients: Performance of Glasgow Coma Score and general severity systems. *Crit Care Med* 26, 142–148 (1998)

Acceso automático por video detección de gestos manuales utilizando técnicas eficientes de segmentación dinámica de color de piel

Julio Zamora Esquivel, José Rodrigo Camacho Pérez, Cruz Vargas Jesus Adan,
Paulo López Meyer, Héctor Cordourier Maruri

Intel Labs, GDC, Guadalajara, México
julio.c.zamora.esquivel@intel.com

Resumen. En este trabajo, proponemos un algoritmo ultra-ligero de video detección para la activación (o “wake up” en inglés) automática de dispositivos de cómputo. La propuesta le permite al usuario interactuar con el dispositivo utilizando gestos manuales que son detectados por video desde la perspectiva de primera persona.

Palabras clave: visión por computadora, reconocimiento de patrones, realidad aumentada.

Hand-Gesture Wake-Up Using Efficient Dynamic Skin Tone Segmentation

Abstract. In this work we propose an ultra lightweight visual-based method to wake-up a visual system, allowing the user to interact with the device using hand-gestures (i.e. combinations of hand poses) through first person view by means of a wearable camera.

Keywords: computer vision, pattern recognition, virtual reality, wake-up.

1. Introducción

Actualmente, los sistemas de interacción humano-computadora basados en la video detección de gestos manuales operan de manera continua. Es decir, el sistema captura video y simultáneamente opera algoritmos de detección de gestos de mano sin interrupción. Esto implica un alto costo energético, en particular al considerar la posibilidad de periodos de tiempo relativamente largos sin actividad de interacción por parte del usuario. Por ejemplo, en el caso de soluciones basadas en redes neuronales artificiales (RNA), esta necesidad de operación continua implica que el sistema analice todos los cuadros de video capturados, siendo evaluados por al RNA con diferentes resoluciones de la captura. Estas evaluaciones y su consecuente alto costo energético impactan en la experiencia de uso en la práctica, como por ejemplo, la reducción de vida de batería en el

caso de los dispositivos móviles. Alternativas a la operación continua incluyen el uso de botones de interacción, reconocimiento de frases clave habladas o algunas otras opciones que requieren de hardware adicional. Sin embargo, estas opciones tienen un impacto en la experiencia del usuario ya que demeritan la sensación de inmersión en los ambientes de realidad virtual. Algunas alternativas para solucionar este problema incluyen:

1. **Activación no visual.** Para evitar la operación continua del sistema de video reconocimiento, comúnmente se utilizan métodos de activación que requieren interacciones táctiles (pulsar botones en teclados o realizar gestos en pantallas o superficies táctiles) o vocalización de palabras o frases clave. Entre los ejemplos comerciales de solución de este tipo se tienen las teclas de activación en teléfonos móviles o las frases "Alexa" "OK Google" de los servicios de Amazon y Google respectivamente.
2. **Video detección de gestos de activación con Histogramas de Gradientes.** En este caso, se construye un descriptor por cada patrón basado en un Histograma de Gradientes de cada pixel en la imagen capturada y una RNA clasifica el descriptor. El gradiente es calculado por cada pixel en una imagen de tamaño $n \times m$ utilizando la convolución de la imagen con dos núcleos (o "kernels" en inglés).
3. **Video detección de gestos de activación con Aprendizaje Masivo (o "deep learning" en inglés).** En este enfoque, se realiza la segmentación del gesto manual con un pre-procesamiento de la imagen binarizada para reducir el tamaño del modelo de RNA. Esto es seguido por una sub-segmentación de cada bloque de color para generar sub-clases. Finalmente, el modelo de RNA es aplicado a cada pixel para detectar los gestos.

El método propuesto en este trabajo está basado en el tercer enfoque mencionado y es particularmente útil en el caso prendas electrónicas (o "wearables") de realidad virtual como el "HoloLens" para activar o re-activar periodos de bajo consumo de energía con las siguientes ventajas:

- Reducción de consumo de energía. Este método permite el uso de una sola evaluación de la RNA por cada image en contraste con las $C \times O$ evaluaciones necesarias en un enfoque de Aprendizaje Masivo (o "deep learning"). Por ejemplo, si se tienen $C = 7$ colores y $O = 4$ objetos candidatos, el número total de evaluaciones necesarias por imagen capturada sería de 28, comparada con solo 1 en la presente propuesta.
- Reducción de falsos positivos. El método propuesto mejora la eficiencia de detección ya que compensa las tonalidades de la piel y las condiciones de iluminación en cada imagen capturada sin costo computacional agregado ya que se define una segmentación dinámica de color de piel.
- Mejora en la experiencia de uso. El método propuesto es completamente basado en video detección de gestos y por ende el usuario no necesita interrumpir la experiencia virtual ni pulsar botones o superficies táctiles.

2. Antecedentes

Existe una variedad de algoritmos de detección de gestos [2,3] los cuales ofrecen resultados confiables pero con un costo financiero, computacional y en facilidad de integración a dispositivos móviles o prendas electrónicas ("wearables"). Como se ha mencionado, algunas de estas soluciones se basan en el Reconocimiento de Claves Habladas (RCH), es decir el reconocimiento de palabras o frases específicas habladas ("Keyword Spotting Systems.^{en} inglés), los cuales requieren de respuesta en tiempo real lo mismo que de exactitud para ofrecer experiencias de uso aceptables. Típicamente, estos sistemas están compuestos de un algoritmo de extracción de características dentro de las cuales destacan por su uso común la técnica "Log-mel Filter Bank Energies"(LFBE) o "Mel-frequency Cepstral Coefficients"(MFCC), que alimenta un clasificador el cual produce probabilidades para las n salidas [5]. De igual manera, los clasificadores tradicionales utilizados en estos sistemas de RCH son los Modelos de Cadenas de Markov Ocultas (MCMO) para segmentos de audio clave y no-clave y los algoritmos de Viterbi para la búsqueda de mejor ruta en el grafo de decodificación [4]. Aunque esta técnica aún ofrece resultados atractivos, puede ser costosa computacionalmente dependiendo de la topología. Consecuentemente, algunos estudios han reemplazado el uso de MCMO por el uso de RNA recurrentes (RNAR) [1] alcanzado mejores resultados en términos de exactitud pero con un desempeño no adecuado en términos de latencia de detección.

Los sistemas que utilizan video detección usualmente utilizan una serie completa de datos sin tomar en cuenta que éstos contienen información redundante la mayoría del tiempo con el consecuente uso sub-óptimo de los recursos computacionales. Algunas propuestas sugieren el uso de imagenes clave o representativas de la secuencia de video para reducir la cantidad de procesamiento al seleccionar las imagenes clave basados en la discriminación potencial por entropía [6]. Para habilitar los sistemas siempre alertas con un consumo energético bajo, algunas propuestas sugieren utilizar técnicas de Compresión de Dominio (CD) la cual permite la ejecución de las rutinas de detección con diferentes imagenes en dos capas; la primera para reducir la resolución de la imagen y la segunda para transferir ésta a el dominio comprimido. Enseguida, es posible calcular el centro de movimiento y ejecutar un clasificador de tipo vecino más cercano ("nearest neighbor").

3. Método

El método propuesto en este trabajo consiste en un algoritmo que permite activar (o "des-bloquear") sistemas de cómputo (por ejemplo prendas electrónicas de realidad virtual) utilizando video detección de gestos manuales de manera eficiente. Cuando el sistema está bloqueado, por ejemplo, en modo de ahorro de energía, el algoritmo propuesto monitorea cada cuadro de imagen de video capturado para determinar si el usuario realiza un gesto manual de activación o desbloqueo. Los pasos en general que sigue el algoritmo se muestran en el diagrama de alto nivel de la figura 1 y se listan a continuación:

1. El algoritmo captura la imagen F , de tamaño $n \times m$ y calcular la métrica de color promedio U llamada "tono de piel". Esta métrica se calcula dentro de una pequeña sección W de la imagen F llamada "ventana de detección"
2. La posición de W se predefine dependiendo del tipo de aplicación (por ejemplo, al centro de la imagen F en el caso de prendas electrónicas de realidad virtual como HoloLens)
3. En seguida, el algoritmo utiliza U para generar una imagen G de tamaño $R \times P$ (más pequeña y simple que la imagen original) para ser evaluada. Esta imagen está formada por el conjunto de todos los píxeles conectados por un color similar a U , comenzando dentro de la "ventana de detección" W y finalizado cuando la conexión de color se pierde
4. para evitar los problemas generados por las diferentes condiciones de iluminación, G es transformada en una imagen binaria. Consecuentemente, G contiene un único objeto con una única tonalidad
5. Adicionalmente, para prevenir problemas de escala, G es ajustada a un tamaño final resultando en una imagen G' , de tamaño $M \times M$ reducido y predefinido dependiendo de la aplicación para su evaluación para determinar la existencia de un gesto válido.
6. La evaluación puede ser realizada por una RNA o cualquier otro algoritmo de decisión/clasificación. El sistema es activado o desbloqueado solo si G' con tiene un gesto válido.

El costo computacional es mínimo comparado con el estado del arte debido a un proceso simplificado de segmentación de tono de piel. Esto es, se obtiene una pequeña imagen binaria por cada imagen capturada de video lo cual reduce significativamente el costo computacional y también mejora el desempeño de detección. La eficiencia del algoritmo propuesto se basa en el uso del proceso dinámico de segmentación de piel descrito gráficamente en la Fig. 2. El algoritmo utiliza el espacio de color Tint, Saturation y Lightness (TSL) ya que permite una mejor segmentación del tono de piel. En contraste con los métodos tradicionales, la propuesta realiza la conversión de color solo para un número reducido de píxeles, no para la imagen completa. Esto significa que nuestro algoritmo toma una pequeña ventana, por ejemplo, centrada para calcular el promedio y la distribución de color. Estos valores se requieren para calcular la distancia de Mahalanobis dada por la distancia de una observación $x = (x_1, x_2, x_3 \dots x_N)$, dentro de un conjunto de observaciones con media $u = (u_1, u_2, u_3 \dots u_N)$ y matriz de covarianza S definida como $d(x) = \sqrt{((x - u)S^{-1})(x - u)}$. Comenzando desde el centro, se aplica el algoritmo de rellenado por desborde ("flood fill") para convertir a TSL solo los píxeles vecinos del píxel original. Solo los píxeles con un color similar y conectados con los píxeles previos se incluyen en la máscara de segmentación. Al final de este proceso, se genera un solo objeto con todos los píxeles adyacentes, conectados y de color similar. De esta manera, el algoritmo entrega un solo patrón por imagen para su subsecuente evaluación.

Nuestro método utiliza la pila (o estack) (ver Figura 3) para agregar los píxeles adyacentes al píxel bajo análisis solo si tienen un color similar. Terminando

Acceso automático por video detección de gestos manuales utilizando técnicas eficientes...

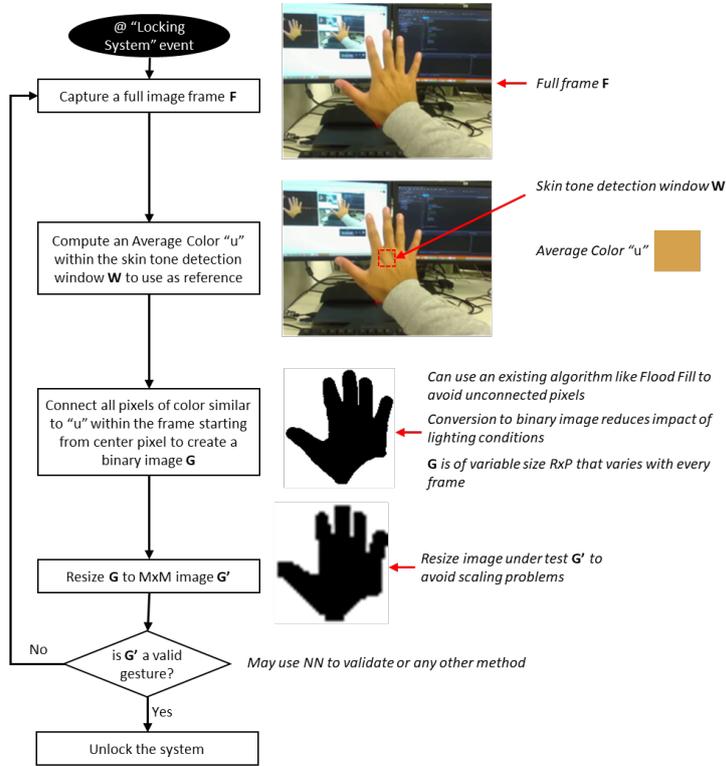


Fig. 1. El algoritmo propuesto permite una video detección de gestos manuales de bajo coste energético para la activación o desbloqueo de sistemas de cómputo

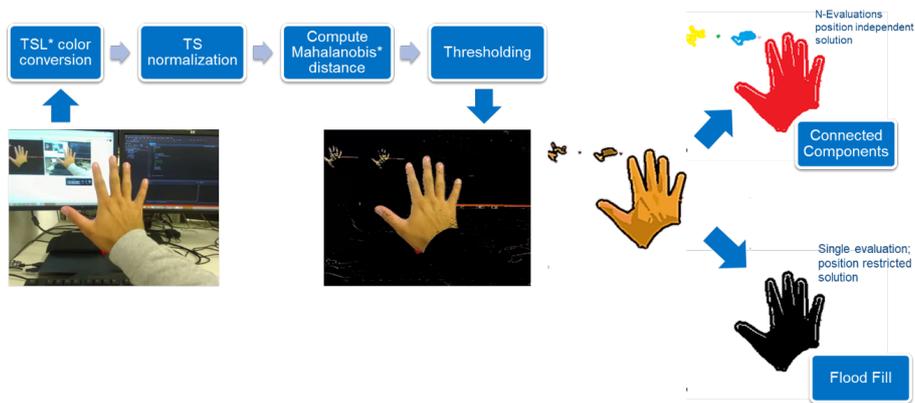


Fig. 2. Diagrama de bloques que muestra cómo se realiza la segmentación del tono de piel dinámica sobre el centro de la imagen.

con un objeto único con todos los componentes conectados, para ser analizados por una NN o cualquier otro tipo de clasificador de aprendizaje automático.

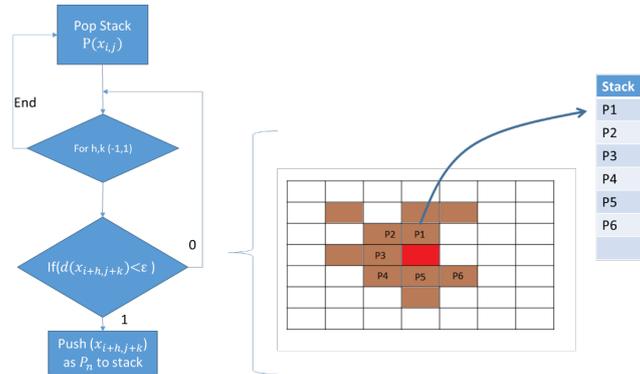


Fig. 3. Stack de píxeles basados en color similar. Esta metodología común se utiliza en nuestro método de segmentación de la mano.

4. Resultados

Se observó experimentalmente que el uso de Stack en nuestro método reduce el procesamiento de 28 evaluaciones por cuadro a solo una evaluación por cuadro. Nuestro método permite redimensionar dinámicamente el objeto segmentado (por ejemplo, la mano del usuario) a un patrón de imagen normalizado. Permitiendo la multiescala. Además, como el color se adquiere en cada imagen, cada imagen tiene en cuenta las condiciones de luz y el color de la piel. Esto significa que el algoritmo de desbloqueo es lo suficientemente robusto como para no verse afectado si el usuario usa guantes de diferente color que el tono de piel. Para mejorar y facilitar la experiencia del usuario, el sistema utiliza la realidad aumentada para mostrar el patrón de "despertar" de desbloqueo en la imagen donde el usuario tiene que alinear el gesto de la mano, como se muestra en la Figura 4.

Se implementó la prueba de concepto de algoritmo de desbloqueo descrito anteriormente. Es importante tener en cuenta que la NN que hemos experimentado reconoce un solo patrón. Una vez que se detecta este patrón de gesto de la mano, el sistema se activa, en este punto, el sistema puede cambiar a un modelo de reconocimiento NN más grande para reconocer un mayor número de gestos con las manos según la aplicación, detectar cualquier posición en la pantalla e incluso reconocer simultáneamente las dos manos de los usuarios.

Este método también comprende una subrutina para reducir aún más el costo computacional por medios estadísticos. Esta reducción se logra ya que en algunos otros casos, el algoritmo de desbloqueo también está considerando evitar



Fig. 4. Gesto de la palma abierta utilizado por el algoritmo de desbloqueo para activar el sistema, presentando un patrón de gestos como realidad aumentada para dar al usuario una indicación del uso.

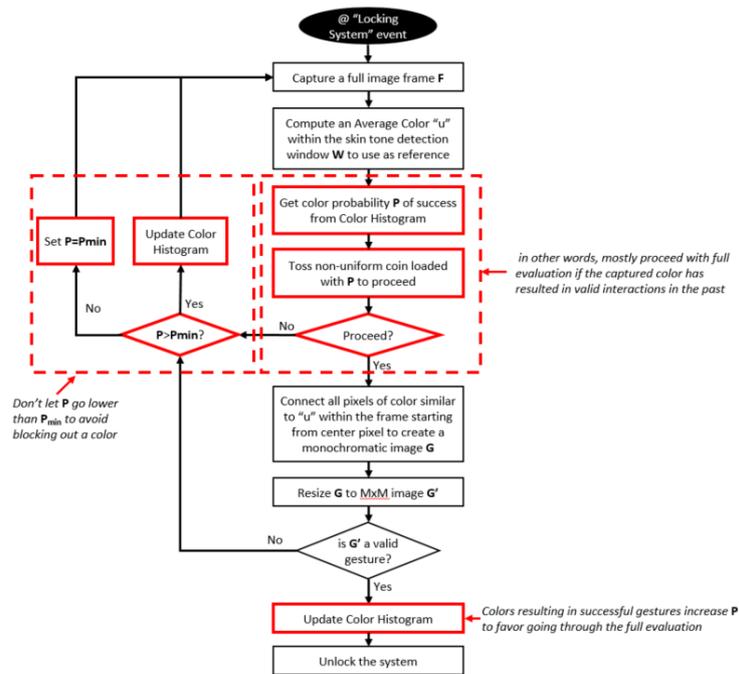


Fig. 5. El algoritmo también puede evitar la necesidad de evaluar cada cuadro utilizando el conocimiento estadístico de la probabilidad de tono de piel para dar como resultado una detección exitosa del gesto de la mano. Las secciones rojas de este diagrama resaltan las modificaciones al algoritmo de la figura 1 para lograr una reducción en el costo computacional.

la necesidad de realizar el proceso de tono dinámico para cada fotograma. En cambio, una vez que se obtiene un tono de piel, el método utiliza un modelo estadístico para determinar si procede con la segmentación dinámica del tono de piel en función de la probabilidad de tener un tono de la segmentación válida, es decir, se realiza la segmentación del tono de piel en colores que resultaron en interacciones válidas en el pasado. La Figura 5 muestra una versión actualizada en el algoritmo de desbloqueo descrito en la Figura 1, donde después de capturar el tono de piel, se actualiza una distribución estadística de colores. Si se captura un nuevo tono de piel, la probabilidad de su uso comienza en 50/50, y aumenta su probabilidad con el tiempo en función de su uso válido, o disminuye cuando se produce una interacción inválida.

5. Conclusiones

Este método proporciona un algoritmo de desbloqueo computacionalmente eficiente para (despertar) activación de sistemas basados en visión por computadora mediante la detección de un comando de gesto con la mano. El método que proponemos extrae un solo objeto por cuadro para la evaluación usando un clasificador, por ejemplo, haciendo segmentación de color de piel dinámica. Este método evita la necesidad de evaluar cada cuadro de video mediante el desarrollo de juicios estadísticos de detección de mano exitosa. Nuestro método produce una reducción significativa en términos de recursos computacionales cuando se compara con los métodos tradicionales de DL, Además muestra robustez a diferentes condiciones de iluminación y diferentes colores de tono de piel (incluido el uso de guantes), y funciona para diferentes tamaños o distancias de la mano del usuario (escala múltiple dinámica).

Referencias

1. Fernández, S., Graves, A., Schmidhuber, J.: An application of recurrent neural networks to discriminative keyword spotting. In: de Sá, J.M., Alexandre, L.A., Duch, W., Mandic, D. (eds.) *Artificial Neural Networks – ICANN 2007*. pp. 220–229. Springer Berlin Heidelberg, Berlin, Heidelberg (2007)
2. Köpüklü, O., Gunduz, A., Kose, N., Rigoll, G.: Real-time hand gesture detection and classification using convolutional neural networks. *CoRR abs/1901.10323* (2019), <http://arxiv.org/abs/1901.10323>
3. Maghoumi, M., Jr., J.J.L.: Deepgru: Deep gesture recognition utility. *CoRR abs/1810.12514* (2018), <http://arxiv.org/abs/1810.12514>
4. Rohlicek, J.R., Russell, W., Roukos, S., Gish, H.: Continuous hidden markov modeling for speaker-independent word spotting. In: *International Conference on Acoustics, Speech, and Signal Processing*,. pp. 627–630 vol.1 (May 1989)
5. Zhang, Y., Suda, N., Lai, L., Chandra, V.: Hello edge: Keyword spotting on microcontrollers. *CoRR abs/1711.07128* (2017), <http://arxiv.org/abs/1711.07128>
6. Zhao, Z., Elgammal, A.M.: Information theoretic key frame selection for action recognition. In: *BMVC* (2008)

Redes neuronales y árboles de decisión para la clasificación de objetos astronómicos

Rogelio Hernández Montes¹, Cynthia Alejandra Martínez Pinto¹,
Silvana Guadalupe Navarro Jiménez²

¹ Tecnológico Nacional de México, ITCG, Jalisco, México

² Universidad de Guadalajara, Instituto de Astronomía y Meteorología, Jalisco, México
roger.hdez.m@hotmail.com, cynthia_amp@hotmail.com,
silvananj@gmail.com

Resumen. El objetivo de esta investigación consiste en comprobar la factibilidad de realizar una clasificación de objetos astronómicos por medio de parámetros fotométricos. Esta identificación da como resultado objetos candidatos, que tienen la probabilidad de ser simbióticos. La búsqueda se realiza en las bases de datos de la misión GAIA, en su más reciente entrega de información (DR2). La clasificación se efectúa a través de inteligencia artificial. Para el estudio se utilizaron las técnicas de Redes Neuronales Artificiales (RNA), Random Forest (RF), Árboles de Decisión y Máquinas de Soporte Vectorial (SVM). Con estos algoritmos se realizó una clasificación binaria. Se desarrolló un software para la optimización de los hiperparámetros. Los resultados de la investigación mostraron que Random Forest tuvo un mejor comportamiento (98%) en la fase de comparación, y por su parte Árboles de decisión fue el que menor índice tuvo (90%). Además de tener un índice kappa superior, los bosques aleatorios presentan una menor complejidad de configuración.

Palabras clave: identificación automática, sistemas simbióticos, análisis de datos.

Neural Networks and Decision Trees for the Classification of Astronomical Objects

Abstract. The objective of this research was to verify the feasibility of classify astronomical objects using photometric parameters. This identification gives as result candidate objects, who have the probability of being symbiotic. The search was done in the databases of the GAIA mission data, in its most recent data release (DR2). The classification is made through artificial intelligence. For the study the techniques of Artificial Neural Networks (RNA), Random Forest (RF), Decision Trees and Support Vector Machines (SVM) were used. A binary classification was made with these algorithms. A software was developed for the optimization of hyperparameters. The results of the research showed that Random Forest had a better performance (98%) in the comparison phase, and Decision Trees was the one with the lowest rate (90%). In addition to having a higher kappa index, random forests have less configuration complexity.

Keywords: automatic identification, symbiotic systems, data analysis.

1. Introducción

En Astronomía se generan actualmente más de un Terabyte de datos por día [2]. La información generada es almacenada en bases de datos que son públicas en su mayoría y están localizadas en distintas partes del mundo. Tal cantidad de datos hace necesario el desarrollo de software especial, dedicado al análisis y clasificación de esta información. Entre los objetos de interés u objetos peculiares, se encuentran las estrellas simbióticas (SS), que son sistemas binarios, conformadas por una estrella de muy alta temperatura y una estrella fría, que generalmente son una enana blanca y una gigante roja, o una estrella variable tipo Mira, dichas estrellas se encuentran ligadas gravitacionalmente [1]. Un ejemplo ha sido observado en la simbiótica más cercana a la tierra: R Aqr [3].

Aunque la confirmación de su clasificación es a través del espectro, es posible identificar objetos que son candidatos a ser sistemas simbióticos utilizando observaciones fotométricas que, aunque no nos dan una clasificación precisa, nos pueden indicar la posibilidad de que se trate de uno de estos objetos y, por otra parte, permite distinguirlos de otro tipo de objetos con los cuales es frecuente confundir a los SS, por ejemplo las nebulosas planetarias (NPs), pues comparten algunas de sus características físicas, las cuales se ven reflejadas en sus características espectrales.

En astrofísica, la distancia a la que se encuentran los objetos es un parámetro fundamental, sin él no es posible conocer la luminosidad real de un objeto, su tamaño físico o determinar con precisión la distribución de los objetos en la galaxia y, con ello, la estructura de la misma. Así mismo, otros parámetros, como su densidad o magnitud absoluta, tampoco podrían ser determinados sin conocer su distancia. El proyecto GAIA fue pensado inicialmente para determinar el paralaje de más de 10^9 objetos en la vía láctea [4]. Dada la precisión de sus mediciones, es posible determinar distancias de hasta varios cientos de parsecs. Adicionalmente, GAIA está realizando mediciones fotométricas en tres bandas (G, BP y RP), determinaciones de movimientos propios y velocidades radiales de las estrellas y está obteniendo espectros de baja y alta resolución.

La liberación de los datos más reciente (25 de abril de 2018) denominada “data release 2” (DR2), contiene una gran base de datos con miles de objetos observados con información sobre paralajes, posiciones precisas, fotometría en G, BP y RP, determinación del error de cada uno de estos parámetros, además de determinaciones muy precisas de los movimientos propios de una fracción importante de los objetos y de sus velocidades radiales. Con esta información es posible determinar estrellas simbióticas candidatas que cubren las características fotométricas de las estrellas del estudio y presentarlas a la comunidad científica para su posterior confirmación a través de sus respectivos espectros.

Para hacer la clasificación de estas estrellas se utilizan dos técnicas de aprendizaje automático, las redes neuronales y los árboles de decisión. El algoritmo de las redes es preciso y requiere ser alimentado con valores muy puntuales, para lograr obtener la clasificación deseada, mientras que el algoritmo de los árboles es menos riguroso en su entrada de datos, pero la clasificación es aleatoria que en las redes neuronales. Sin embargo los resultados son alentadores para el manejo de una clasificación binaria, mostrando una ventaja sobre las redes neuronales.

2. Metodología

2.1. Adquisición y tratamiento de los datos

Para la recuperación de los datos de las estrellas simbióticas se recurrió al catálogo de Belczyński, et al. (2000), donde recopilan un total de 188 objetos bajo esta clasificación [5]. De este catálogo, se extrajo el nombre y se procedió a hacer una búsqueda cruzada entre las bases de datos de GAIA y 2MASS, utilizando SIMBAD, la cual es una base de datos astrofísicos que es actualizada diariamente. Esta base de datos reúne la información publicada de los objetos e incluye la identificación cruzada entre las diversas base de datos de los grandes proyectos de “survey”. De la información obtenida, se recuperaron los valores de las mediciones fotométricas en distintos filtros (G, Bp, Rp, B, V, J, H, K, estos últimos de 2MASS). Los parámetros recuperados de GAIA fueron: *source_id*, *ra*, *ra_error*, *dec*, *dec_error*, *parallax*, *parallax_error*, *parallax_over_error*, *phot_g_mean_flux*, *phot_g_mean_mag*, *phot_bp_mean_flux*, *phot_bp_mean_mag*, *phot_rp_mean_flux*, *phot_rp_mean_mag*, *bp_rp*, *bp_g*, *g_rp*, *radial_velocity*, *radial_velocity_error*, *phot_variable_flag*, *teff_val*, *a_g_val*. Estos valores se describen a continuación en la tabla 1.

Tabla 1. Descripción de los campos de la base de datos de GAIA.

Campo recuperado	Descripción
<i>source_id</i>	Identificador único perteneciente a la base de datos de la misión GAIA. En este se encuentra codificado la posición del objeto.
<i>RA</i>	Valor perteneciente a la misión GAIA. Ascensión recta del objeto, el valor se encuentra expresado en formato ICRS
<i>ra_error</i>	Valor perteneciente a la misión GAIA. Error estándar de la ascensión directa.
<i>Dec</i>	Valor perteneciente a la misión GAIA. Declinación del objeto, el valor se encuentra expresado en formato ICRS
<i>dec_error</i>	Valor perteneciente a la misión GAIA. Error estándar de la declinación del objeto.
<i>Parallax</i>	Valor perteneciente a la misión GAIA. Paralaje estelar del objeto
<i>parallax_error</i>	Valor perteneciente a la misión GAIA. Error estándar del paralaje.
<i>parallax_over_error</i>	Valor perteneciente a la misión GAIA. Valor obtenido de la división del paralaje sobre su error.
<i>phot_g_mean_flux</i>	Valor perteneciente a la misión GAIA. Flujo medio integrado en la banda G.
<i>phot_g_mean_mag</i>	Valor perteneciente a la misión GAIA. Magnitud media en la banda G calculada en base a la magnitud de Vega.
<i>phot_bp_mean_flux</i>	Valor perteneciente a la misión GAIA. Flujo medio integrado en la banda BP.

Campo recuperado	Descripción
<i>phot_bp_mean_mag</i>	Valor perteneciente a la misión GAIA. Magnitud media en la banda BP.
<i>phot_rp_mean_flux</i>	Valor perteneciente a la misión GAIA. Flujo medio integrado en la banda RP.
<i>phot_rp_mean_mag</i>	Valor perteneciente a la misión GAIA. Magnitud media en la banda de RP.
<i>bp_rp</i>	Valor perteneciente a la misión GAIA. Color resultante de la diferencia de las magnitudes de las bandas BP y RP
<i>bp_g</i>	Valor perteneciente a la misión GAIA. Color resultante de la diferencia de las magnitudes de las bandas BP y G
<i>g_rp</i>	Valor perteneciente a la misión GAIA. Color resultante de la diferencia de las magnitudes de las bandas G y RP
<i>radial_velocity</i>	Valor perteneciente a la misión GAIA. Velocidad radial espectroscópica en el marco de referencia baricéntrico solar
<i>radial_velocity_error</i>	Error de la velocidad radial.
<i>phot_variable_flag</i>	Valor perteneciente a la misión GAIA. Bandera de variabilidad fotométrica. Los posibles valores para este campo son <i>variable</i> que indica que el objeto se identificó y procesó como variable, <i>constant</i> lo cual hace referencia a que no se encontró variación, y <i>not_available</i> que es usado cuando no ha sido procesado y/o exportado al catálogo.
<i>teff_val</i>	Valor perteneciente a la misión GAIA. Temperatura estelar efectiva.
<i>a_g_val</i>	Valor perteneciente a la misión GAIA. Valor de extinción en la banda G.
<i>B</i>	Valor recuperado de SIMBAD. Magnitud en la banda B.
<i>V</i>	Valor recuperado de SIMBAD. Magnitud en la banda V.
<i>J</i>	Valor de 2MASS, recuperado con SIMBAD. Magnitud en la banda J.
<i>H</i>	Valor de 2MASS, recuperado con SIMBAD. Magnitud en la banda H.
<i>K</i>	Valor de 2MASS, recuperado de SIMBAD. Magnitud en la banda K.

Del catálogo sólo se recuperaron 102 objetos ya que algunos de ellos presentan irregularidades en sus campos, entre ellas se encuentran estrellas con parallax de 0 o falta de mediciones en las bandas G, BP o RP y por lo tanto la falta en los colores *bp_rp*, *bp_g* y *g_rp*. Además de la recuperación de las estrellas simbióticas, también se procedió a descargar 20,000 estrellas la secuencia principal descrita en el artículo Gaia Data Release 2-Observational Hertzsprung-Russell diagrams [6] donde se construye y describe el diagrama de Hertzsprung-Russell. También se incluyeron otros conjuntos de objetos evolucionados como lo son las nebulosas planetarias.

2.2. Unión y corrección de los datos

Los objetos fueron recuperados en formato .vot en el caso de la base de datos de GAIA, y en formato Web para SIMBAD. A causa de la generación de distintos archivos y en distinto formato, fue menester la unión de todos los valores en un sólo formato, además de realizar correcciones y la generación de otros campos necesarios para la clasificación. Se agregó una columna que contiene la magnitud absoluta de cada uno de los objetos de estudio, debido a que GAIA sólo contiene una magnitud observada. Es importante realizar el cálculo de la magnitud absoluta para poder conocer la verdadera luminosidad de un objeto sin que esta se vea afectada por la distancia real del objeto y así tener todos los objetos con la magnitud aparente que tendrían si estuvieran a una distancia de 10 parsecs.

La magnitud absoluta en G se determinó utilizando la ecuación 1, con la que se realiza también la corrección por extinción, y por su parte la ecuación 2 se utilizó para convertir el paralaje a distancia:

$$G_{abs}=5+G-5\log(d)+A_g, \quad (1)$$

$$d=(1/\text{parallax}), \quad (2)$$

donde:

G_{abs} : Magnitud absoluta en la banda G; G: Magnitud observada en la banda G; A_g : Valor de extinción en G; d: distancia expresada en parsecs; parallax: valor de parallax expresado en segundos de arco

2.3. Normalización de los datos

Los distintos conjuntos de datos se etiquetaron de acuerdo a un criterio binario: si el objeto es una estrella simbiótica o no. Posterior al etiquetado se realizó una fusión de los datos generados para cada tipo de objeto con el fin de crear un único set de datos. Sobre el conjunto de datos resultante se procedió a realizar una normalización. La cual se realizó de acuerdo a la ecuación 3. Los valores utilizados para los distintos parámetros del vector de características fueron recuperados a través de consultas sobre las bases de datos GAIA (DR2) y 2MASS. El proceso consistió en encontrar la combinación de valores frontera para cada parámetro de los objetos. Los valores se muestran a continuación en la tabla 2. La normalización generada por la ecuación es necesaria para poder comparar los valores entre ellos y así definir el rango de valores máximos y mínimos que podrá adoptar cada parámetro. Para la investigación se seleccionó como valor máximo, identificado en la ecuación como d_2 , el valor de 1 y para el valor mínimo, identificado como d_1 , el valor de 0. Por lo tanto, todos los parámetros del vector de características estarán comprendidos en el rango [0,1]:

$$y=((x-x_{min})(d_2-d_1))/(x_{max}-x_{min})+d_1, \quad (3)$$

donde:

- y: valor de x normalizado,
- x: Valor a normalizar,
- x_{min} : valor mínimo posible para x,
- x_{max} : valor máximo posible para x,
- d_1 : valor mínimo que podrá tomar y,

d2: valor máximo que podrá tomar y.

Tabla 2. Valores frontera usados para la normalización de los parámetros utilizados.

Parámetro	Valor mínimo	Valor máximo
bp_rp	-5.489193	9.800095
g_rp	-4.157661	3.650905
bp_g	-2.073900	7.795892
g_abs_mag	-39.51653	26.91888
teff_val	3229	9803

2.4. Selección y personalización de los algoritmos

Separación del conjunto de datos. Con el conjunto de datos normalizados se procedió a hacer una separación del 16.5% del total, con el fin de ser usados como validación para los 4 algoritmos que se programaron, 16.5% para las pruebas, y el 66.6% para el entrenamiento. El conjunto de prueba se usó para verificar el nivel de aprendizaje de los algoritmos, mientras que el set de validación se utilizó para comprobar que el algoritmo fuera capaz de generalizar la clasificación y detectar de esta manera una especialización sobre los datos de entrenamiento y prueba.

Los algoritmos seleccionados para la clasificación binaria fueron: redes neuronales, “random forest”, árboles de decisión y máquinas de soporte vectorial. La selección de estos algoritmos se llevó a cabo después de una revisión del estado del arte con referencia a clasificación estelar. Las redes neuronales y “random forest” han sido utilizados y recomendados en investigaciones previas [7][8] cuyo carácter es similar tanto en campo de investigación como en similitud de los datos con los que se cuenta. Con el fin de poder comparar estos algoritmos se han añadido los algoritmos de árboles de decisión y máquinas de soporte vectorial.

De la información recuperada de ambas base de datos, se procedió a seleccionar los campos con los que trabajarían los algoritmos seleccionados. El criterio de selección utilizado, fue tomar aquellos datos que ayudan a una mejor separación de las clases a identificar. Con los parámetros elegidos, el vector de características final se compone con los campos que se muestran a continuación en la tabla 3.

Tabla 3. Descripción de los campos del vector de características usado para los algoritmos de clasificación.

Parámetro	Valor máximo
<i>Gabs</i>	Magnitud absoluta en la banda G
<i>teff_val</i>	Temperatura efectiva
<i>bp_rp</i>	Color resultante de la diferencia de las magnitudes de la banda Bp y Rp
<i>b_rp</i>	Color resultante de la diferencia de las magnitudes de la banda G y Rp
<i>bp_g</i>	Color resultante de la diferencia de las magnitudes de la banda Bp y G

j_h	Color resultante de la diferencia de las magnitudes de la banda J y H
h_k	Color resultante de la diferencia de las magnitudes de la banda H y K
b_v	Color resultante de la diferencia de las magnitudes de la banda B y V
v_k	Color resultante de la diferencia de las magnitudes de la banda V y K

2.5. Redes neuronales

El primer acercamiento para la clasificación de las estrellas simbióticas se realizó a través de las redes neuronales supervisadas. El modelo de red neuronal que se usó fue el perceptrón multicapa. El algoritmo de entrenamiento seleccionado fue el de retro-propagación con gradiente descendente y una función sigmoïdal para activación. Con el fin de determinar la mejor topología para la red neuronal se automatizó el proceso de creación, entrenamiento y prueba del algoritmo mediante un programa en Java. Con el proceso automatizado se procedió a codificar una búsqueda aleatoria sobre espacios continuos de distintos parámetros de la red neuronal. Dichos parámetros se muestran a continuación en la tabla 4.

Tabla 4. Descripción de los parámetros utilizados para la creación automática de redes neuronales.

Parámetro de la Red Neuronal	Selección
Factor de aprendizaje	Búsqueda aleatoria en el espacio continuo [.3. .9]
Numero de Capas	Búsqueda Aleatoria en el espacio discreto 1-3
Numero de neuronas por capa	Búsqueda Aleatoria en el espacio discreto 3-255

El algoritmo de automatización se diseñó para trabajar limitado por tiempo. Dado un tiempo específico, crea la mayor cantidad de sujetos (redes neuronales) y se evalúan. Una vez finalizado el tiempo especificado sólo despliega la información correspondiente al sujeto con las mejores características registradas.

El proceso de evaluación que se realiza sobre cada sujeto es llevado a cabo separando el set de datos. La separación se llevó como se describe en la tabla 5.

Tabla 5. Descripción de la división y uso del conjunto de datos.

Fracción del set de datos total (porcentaje)	Uso	Separación
2/3 (66%)	Entrenamiento	Realizada de forma aleatoria al comienzo del entrenamiento de cada red neuronal
1/6 (16.6%)	Pruebas	Realizada de forma aleatoria al comienzo del entrenamiento de cada red neuronal
1/6 (16.6%)	Validación	Realizada y separada antes del proceso de creación de redes neuronales.

2.6. Random forest

El algoritmo Random forest consta de una gran cantidad de árboles de decisión, que a diferencia de las redes neuronales, cuenta con una menor cantidad de parámetros para su construcción y entrenamiento [10]. Los valores que deben ser cambiados son el número de clasificadores que serán generados, el número de variables a tomar, y un parámetro opcional es limitar la profundidad máxima que puede tener cada árbol generado.

La principal característica del algoritmo random forest es que aunque sea un algoritmo de aprendizaje supervisado como las redes neuronales, no es posible controlar la topología de los estimadores generados por el algoritmo. Otra de las características notables del algoritmo es que para la creación de cada uno de los árboles estimadores, los parámetros que usará no son controlados, sino que son escogidos aleatoriamente. Este algoritmo también se distingue por el hecho de que el set de datos que requiere, puede o no estar normalizado.

Para el caso de estudio se realizaron pruebas variando el número de clasificadores entre [1,3000]. Para todas las pruebas realizadas se estableció como máximo número de variables a tomar por cada estimador la longitud del vector de características. Por lo que la configuración para este algoritmo quedo de la manera que se describe en la tabla 6.

Tabla 6. Descripción de los parámetros utilizados en la generación de estimadores para el algoritmo random forest.

Parámetro	Selección
Profundidad máxima	Sin límite
Árboles	[1, 3000]
Mínimo de muestras por hoja	1
Máximo de características a tomar	Número de características
Número máximo de hojas	Sin límite

2.7. Árboles de decisión

Los arboles de decisión son un algoritmo que genera un árbol como un diagrama de flujo, donde cada nodo interno representa un valor del vector de características, las ramas representan decisiones, y los nodos hojas representan un resultado de la clasificación. Este es un algoritmo de clasificación de aprendizaje supervisado, no paramétrico, que no requiere supuestos distribucionales, permite modelar relaciones no lineales, y no es sensible a la ausencia de datos [12]. La forma básica de funcionamiento es la creación de particiones recursivas de acuerdo con reglas de asignación, partición y parada.

Entre las principales ventajas de este algoritmo es que crea un modelo de caja blanca, por lo que es fácil de comprender el resultado. Y requiere poca preparación del set de datos, ya que no requiere la eliminación de datos faltantes, o normalización.

2.8. Máquinas de soporte vectorial

Las máquinas de soporte vectorial son un algoritmo de aprendizaje supervisado. Este algoritmo puede manejar fácilmente variables continuas y paramétricas. El funcionamiento básico del algoritmo es la creación de un hiperplano en un espacio multidimensional para la separación de las clases. Las máquinas de soporte vectorial según lo expuesto por Auria [13] tienen la ventaja de presentar un buen rendimiento cuando los datos son no linealmente separables. Además de poder manejar una alta dimensionalidad de datos.

3. Resultados

Con el fin de poder comparar la eficacia de los algoritmos utilizados para clasificar el set de datos de validación, se tomó en cuenta la precisión mostrada en la matriz de confusión.

Los resultados de evaluar los distintos algoritmos pueden considerarse, en forma general, como muy positivos, ya que las estrellas simbióticas son un tipo de objeto difícil de identificar. Además que de la manera tradicional de hacer la identificación sobre objetos peculiares es hacerlo sobre los espectros de los objetos. Sin embargo en la investigación realizada se optó por valores fotométricos debido a la falta de disponibilidad de espectrometría en GAIA. Los algoritmos lograron separar las estrellas no simbióticas de las estrellas simbióticas. Sin embargo los algoritmos clasificaron estrellas simbióticas como no simbióticas. El algoritmo random forest mostró una mejor clasificación sobre redes neuronales.

A continuación en la tabla 7 se despliega la matriz de confusión del algoritmo random forest. Los resultados de las distintas técnicas fue una completa separación de las estrellas no simbióticas de las simbióticas, por lo que la matriz no refleja falsos positivos. Sin embargo las técnicas presentan falsos negativos, clasificaron estrellas simbióticas como estrellas no simbióticas.

Tabla 7. Matriz de confusión correspondiente al algoritmo random forest.

0	1	
147	0	0=0
1	36	1=1

En la tabla 8 se muestra la matriz de confusión obtenida del algoritmo de automatización sobre redes neuronales.

Tabla 8. Matriz de confusión correspondiente al algoritmo sobre redes neuronales.

0	1	
147	0	0=0
2	35	1=1

En la tabla 9 se muestra la matriz de confusión resultante del algoritmo de árboles de decisión.

Tabla 9. Matriz de confusión correspondiente al algoritmo de árboles de decisión.

0	1	
139	0	0=0
6	39	1=1

En la tabla 10 se despliega la matriz de confusión obtenida a partir del algoritmo de máquinas de soporte vectorial.

Tabla 10. Matriz de confusión correspondiente al algoritmo de máquinas de soporte vectorial

0	1	
139	0	0=0
3	42	1=1

Los resultados mostrados para los algoritmos, representa al mejor sujeto generado durante las pruebas. Para las redes neuronales, se utilizó el algoritmo de automatización durante 72 horas. Los resultados obtenidos fue la generación de 1128 modelos, por lo que se tiene que en promedio se han generado 15.6 modelos de redes neuronales por hora, lo que da un tiempo promedio de entrenamiento de 3.49 segundos en promedio para cada modelo. Por otra parte, para la generación de clasificadores utilizando el algoritmo random forest, se procedió a la creación de 1128 sujetos, y el tiempo que se llevó la ejecución fue de 10 horas.

El criterio para la evaluación de cada sujeto fue la precisión obtenida al generar la matriz de confusión. La evaluación fue realizada dentro de la automatización y únicamente se tenía control sobre el mejor sujeto. Por este motivo, al momento de que se generaba un sujeto superior, se sustituía el mejor sujeto por el nuevo.

Con motivo de comparar mejor la matriz de confusión los distintos algoritmos se calculó el índice kappa para cada algoritmo en base al mejor sujeto presentado. La comparación se muestra a continuación en la tabla 11. De acuerdo a la escala propuesta por Landis y Koch [9] el algoritmo random forest presenta un grado de concordancia casi perfecto.

Tabla 11. Valores de los índices kappa para los algoritmos.

	Redes neuronales	Random forest	Árboles de decisión	Máquinas de soporte vectorial
Índice Kappa	0.9654	0.9829	.9075	.9548

Con el fin de ilustrar el funcionamiento de random forest (RF) se extrajo uno de los árboles estimadores perteneciente al mejor sujeto generado. Dicho árbol es desplegado en la figura 1. Al observar el árbol se aprecia que la forma de clasificación que utilizó es similar a la creación de un diagrama color-magnitud, proceso usado en astrofísica para la caracterización. La magnitud usada fue *Gabs* y el color que usa el *g_{rp}*.

La primera separación que realiza el árbol es sobre el color, acto seguido confirma las regiones a través de las magnitudes. Sólo en el par de nodos hoja más profundos se realizó una segunda comprobación por color. El presente árbol es uno de los distintos

árboles generados, por lo que su configuración puede no ser correcta para todos los casos presentes en el set de validación.

El índice Gini permite seleccionar dos elementos de una población, estos deben ser de la misma clase, donde la probabilidad de que esto suceda es uno, cuando la población es “pura”. Utiliza dos variables categóricas: “Success” o “Failure” y entre más grande sea el índice Gini, mayor es la homogeneidad de los datos. El cálculo de los subnodos usa la ecuación 4 que consiste en suma de los cuadrados de probabilidad para success y failure [11]:

$$(p^2 + q^2), \tag{4}$$

donde p: es success y q es failure.

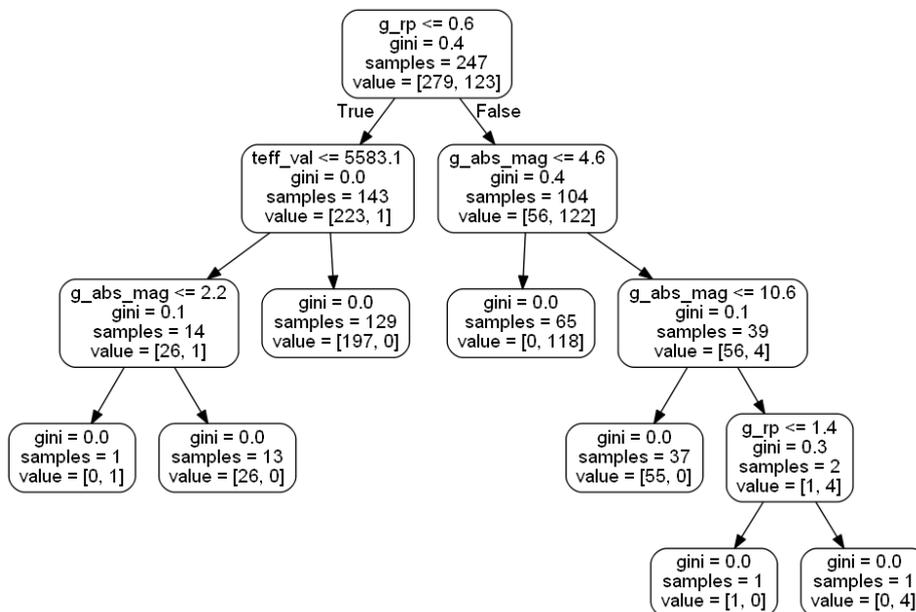


Fig. 1. Árbol de decisión generado aleatoriamente por random forest (RF).

4. Conclusiones

Después de analizar el comportamiento de los distintos algoritmos, se aprecia que random forest (RF) representa una mejor opción al momento de tener que seleccionar un algoritmo para una clasificación binaria de 20,136 objetos. El algoritmo random forest presenta un resultado casi similar al de las redes neuronales, sin embargo es importante entender que el algoritmo es completamente aleatorio. Por esta razón mejorar los resultados de este algoritmo realizando un cambio en sus parámetros no garantiza obtener mejores resultados.

El algoritmo random forest crea un número interno de estimadores y para evaluar el modelo generado se realiza una “votación” entre todos los estimadores, donde la

clasificación generada por el algoritmo es la clase con más votaciones. Por esta razón aunque se recomiende el uso de random forest para problemas de clasificación binaria, es muy importante que se analice el comportamiento de los modelos al aumentar el número de clases disponibles. Dado que se incluirán más tipos de objetos, será necesario evaluar la conveniencia de separar cada clase con un RF independiente.

También se pudo observar que el algoritmo de los bosques aleatorios tiene el mejor índice kappa en comparación con los demás métodos evaluados, incluso con los árboles de decisión, que aunque manejan una estructura de datos similar a los Random Forest, el algoritmo para la clasificación demuestra ser mucho más eficiente, gracias a la cantidad de árboles que genera el RF, donde cada árbol vota por una clase y el resultado es la clase con mayor número de votos en todo el bosque [11]. Por su parte las máquinas de soporte vectorial, tuvieron un buen desempeño, pero como pasa con las redes neuronales, su desempeño es el resultado de una buena parametrización, por lo que se tendrá una gran cantidad de posibles configuraciones. Y como el mismo caso que RNA, encontrar los parámetros ideales conlleva un gran trabajo.

Las fortalezas que se encontraron de algoritmo de random forest sobre las redes neuronales, son la facilidad de implementación y generación de modelos, así mismo el tiempo invertido para la obtención de resultados similares juega un papel importante para optar por los bosques aleatorios.

5. Recomendaciones y trabajo futuro

Aun no se ha concluido la presente investigación. El trabajo que falta por realizar es seguir generando modelos tanto de redes neuronales como random forest con el fin de alcanzar un mayor grado de confiabilidad en la clasificación de estrellas simbióticas. Una vez alcanzado este objetivo, se añadirán más tipos de objetos al conjunto de datos con el fin de ver el comportamiento de los clasificadores.

Como recomendación al momento de trabajar con el algoritmo de redes neuronales para clasificaciones binarias, se usaba una función de error basada en el error cuadrático medio, con lo cual se obtenía un índice kappa alrededor de .7 al momento de variar sólo esta función se mejoró considerablemente la precisión de la red, alcanzado un índice kappa de .96. La función de error utilizada fue la cross entropy la cual presenta mejores resultados para problemas de clasificación binaria. A lo largo de esta investigación se encontró que la mejor forma de mejorar los resultados de los algoritmos random forest y árboles de decisión, además variar el número de árboles estimadores en el caso del primer algoritmo, es experimentar con la profundidad máxima de los árboles. Un punto importante es explorar las posibilidades donde la profundidad es menor al tamaño del vector de características, ya que de otra forma los algoritmos tendrán un buen rendimiento durante la fase de entrenamiento, pero esto no asegura un buen resultado con un set de datos que el algoritmo no ha visto, ya que a más profundidad los árboles tienden a especializarse en los datos y forzar las clasificaciones.

Referencias

1. Belczyński, K., Mikołajewska, J., Munari, U., Ivison, R. J., Friedjung, M.: A catalogue of symbiotic stars. *Astronomy and Astrophysics Supplement Series*, 146(3), 407–435 (2000)

2. Hernández Cervantes, L., Santillán González, A., González-Ponce, A.: Observatorios Virtuales Astrofísicos. *Revista Digital Universitaria*, Vol. 10, No. 10 (2009)
3. H. M. Schmid, A. Bazzon, J. Milli, et al.: SPHERE/ZIMPOL observations of the symbiotic system R Aquarii-I. Imaging of the stellar binary and the innermost jet clouds. *A&A* Vol. 602, p. A53 (2017)
4. Gaia Collaboration, T. Prusti, J. H. J. de Bruijne, A. G. A. Brown, A. Vallenari, C. Babusiaux, C. A. L. Bailer-Jones, U. Bastian, M. Biermann, D. W. Evans, et al.: The gaia mission. *A&A* 595, p A1 (2016)
5. K. Belczyński, J. Mikołajewska, U. Munari, R. J. Ivison, M. Friedjung: A catalogue of symbiotic stars, *Astron. Astrophys. Suppl. Ser.* 146 (3) 407–435 (2000)
6. Gaia Collaboration, C. Babusiaux, F. van Leeuwen, M. A. Barstow, C. Jordi, A. Vallenari, D. Bossini, A. Bressan, T. Cantat-Gaudin, M. van Leeuwen, et al.: Gaia Data Release 2-Observational Hertzsprung-Russell diagrams. *Astronomy & Astrophysics* 616, A10 (2018)
7. Kheirdastan, S., Bazarghan, M.: SDSS-DR12 bulk stellar spectral classification: Artificial neural networks approach. *Astrophysics and Space Science*, 361(9), 1–8. doi:10.1007/s10509-016-2880-3 (2016)
8. Colas F., Brazdil, P.: Comparison of SVM and Some Older Classification Algorithms in Text Classification Tasks. In: Bramer M. (eds) *Artificial Intelligence in Theory and Practice. IFIP AI 2006*. IFIP International Federation for Information Processing, vol. 217, Springer, Boston, MA (2006)
9. Landis, J. Richard, and Gary G. Koch.: The measurement of observer agreement for categorical data. *Biometrics* 159–174 (1977)
10. Keller, C. A., Evans, M. J.: Application of random forest regression to the calculation of gas-phase chemistry within the GEOS-Chem chemistry model v10. *Geoscientific Model Development*, 12(3), 1209–1225 (2019)
11. Árboles de decisión y Random forest. <https://bookdown.org/content/2031/>, last accessed 2019/04/01.
12. Breiman, L., Friedman, J. H., Olshen, R. A., Stone, C. J.: *Classification and regression trees*. Belmont, CA: Wadsworth. International Group, 432 (1984)
13. Auria, L., Moro, R. A.: Support vector machines (SVM) as a technique for solvency analysis. (2008)

MoCa: Una aplicación móvil para registrar síntomas de pacientes con cáncer

Beatriz A. González-Beltrán¹, Lizbeth Gallardo-López¹,
Roberto García-Madrid², Marco V. Ferruzca-Navarro², Alejandro Reyes-Ortíz¹

¹ Universidad Autónoma Metropolitana, campus Azcapotzalco, División de Ciencias Básicas e Ingeniería, Ciudad de México, México

² Universidad Autónoma Metropolitana, campus Azcapotzalco, División de Ciencias y Artes para el Diseño, Ciudad de México, México
{bgonzalez, glizbeth, gmra, mvfn, jaro}@azc.uam.mx

Resumen. Las últimas estadísticas de la Agencia Internacional para la Investigación del Cáncer, la agencia especializada en cáncer de la Organización Mundial de la Salud muestra un incremento en la incidencia de cáncer en el mundo. En 2012, los casos de cáncer aumentaron a un estimado de 14 millones por año. Esta enfermedad aumentará a 22 millones de casos nuevos por año para 2030. Por lo tanto, se necesitan nuevos sistemas de software para el cuidado de la salud para ayudar a la detección temprana, el diagnóstico, el tratamiento, la supervivencia y la atención al final de la vida de los pacientes con cáncer, sus familias y los médicos. Los pacientes con cáncer son atendidos por sus médicos en sus tratamientos contra el cáncer y en los controles de rutina. Los pacientes deben estar conscientes de su estado de salud para proporcionar información a sus médicos. De esta manera, los pacientes mantienen una mejor calidad de vida y los médicos toman mejores decisiones clínicas. Registrar los síntomas de un paciente es vital, especialmente durante los tratamientos contra el cáncer. En este artículo se presenta MoCa, una aplicación móvil para el registro de síntomas de pacientes con cáncer. Se realizaron entrevistas con los usuarios. MoCa permite a los pacientes registrar sus síntomas usando una figura del cuerpo o el teclado. Se realizaron pruebas cualitativas a pacientes. Se encontró que los pacientes respondieron positivamente, tienen confianza al registrar sus síntomas y a comunicarse con sus médicos de manera precisa.

Palabras clave: aplicación móvil, salud móvil, síntoma, cáncer, ssa-bilidad.

MoCa: A Mobile Application for Registering Symptoms of Cancer Patients

Abstract. Latest statistics of the International Agency for Research on Cancer, the specialized cancer agency of the World Health Organization shows an increasing global incidence of cancer. In 2012, cancer cases rose to an estimated of 14 million per year. The burden of cancer will

increase to 22 million new cases per year by 2030. Therefore, new health-care software systems are needed to help the early detection, diagnosis, treatment, survivorship and end-of-life care for cancer patients, their families, and clinicians. Cancer patients are followed by their clinicians in their cancer treatments and in routine checks. Patients must be aware of their health status to give information to their clinicians. In this manner, patients maintain their best quality of life and clinicians take better clinical decisions. Recording their symptoms is vital, particularly during cancer treatments. We present MoCa, a mobile application for registering symptoms of cancer patients. We conduct interviews with users. MoCa allows patients to register their symptoms using a body figure or the keyboard. We performed a qualitative test to patients. We find that patients responded positively, they are confident with registering their symptoms and to communicate with their clinicians in a precise way.

Keywords: mobile application, mobile health, symptom, cancer, usability.

1. Introducción

Las últimas estadísticas de la Agencia Internacional para la Investigación del Cáncer muestran un incremento en la incidencia de cáncer en el mundo. En particular, los sistemas de cuidado de la salud esperan un aumento de enfermedades crónicas como enfermedades cardiovasculares, diabetes, enfermedades respiratorias crónicas y cáncer [12]. En 2012, los casos de cáncer aumentaron a un estimado de 14 millones por año. Esta enfermedad aumentará a 22 millones de casos nuevos por año para 2030. Por lo tanto, se necesitan nuevos sistemas de software para el cuidado de la salud para ayudar a la detección temprana, el diagnóstico, el tratamiento, la supervivencia y la atención al final de la vida de los pacientes con cáncer, sus familias y los médicos.

Los pacientes diagnosticados con cáncer requieren un régimen específico de tratamiento que abarque una o más modalidades, como la cirugía y/o la radioterapia, y/o la quimioterapia [12]. Los pacientes con cáncer son atendidos por sus médicos en sus tratamientos contra el cáncer y en los controles de rutina. Los médicos alientan a los pacientes a registrar sus síntomas. Registrar los síntomas de un paciente es vital, particularmente durante los tratamientos contra el cáncer porque cada persona reacciona de diferente manera a cada tratamiento. Por ejemplo, algunos tratamientos de quimioterapia pueden provocar náuseas, vómitos o disminución del apetito. Es útil que los pacientes anoten sus síntomas para comunicarse con precisión con sus médicos. Por lo tanto, los médicos pueden recomendar medicamentos para aliviar los síntomas y enfrentar mejor el próximo tratamiento. Incluso durante la supervivencia, los pacientes con cáncer son supervisados de por vida y los médicos también pueden prescribirles medicamentos para aliviar el dolor como efecto secundario de los tratamientos. Si bien el objetivo principal es curar el cáncer o prolongar considerablemente la vida, el mantenimiento de la mejor calidad de vida de los pacientes también

es un objetivo importante. El cuidado paliativo es un tratamiento para aliviar, en lugar de curar, los síntomas causados por el cáncer. Los cuidados paliativos facilitan la transición al final de la vida.

Las observaciones anteriores nos motivaron a diseñar una aplicación móvil para registrar síntomas de pacientes con cáncer, el objetivo es doble: 1) aumentar la conciencia personal del paciente y la responsabilidad personal sobre su enfermedad y 2) proporcionar una herramienta para registrar los síntomas. En el resto del documento, el término “signo vital” se referirá a los signos “que muestran el estado de salud de alguien, como la temperatura corporal, la frecuencia respiratoria y el latido cardíaco”. Además, el término “síntoma” se referirá a “cualquier sensación de enfermedad o cambio físico o mental causado por una enfermedad en particular” (Diccionario de Cambridge).

En los últimos años, la Inteligencia Artificial ha incorporado la Interacción Humano Computadora, desde que se desea evitar el desarrollo de soluciones a los problemas de forma aislada; por lo tanto, se ha involucrado la participación de humanos interesados (usuarios) que utilizarán dichas soluciones y que son los más afectados por las mismas. La participación del usuario en el diseño, desarrollo y evaluación de sistemas de inteligencia es fundamental para garantizar que los sistemas basados en inteligencia artificial sean prácticos. En [11], se presentó un enfoque de clasificación supervisada de reportes clínicos, utilizando el algoritmo de Máquinas de Soporte Vectorial, para apoyar el diagnóstico de cuatro tipos de cáncer: estómago, pulmonar, cáncer de pecho y cáncer de piel. Por ello, en este trabajo se presenta la aplicación MoCa que permite el registro de síntomas de pacientes con cáncer.

2. Trabajo relacionado

Hay varios trabajos de investigación que están estrechamente relacionados con aumentar la conciencia personal de un paciente y la responsabilidad personal su enfermedad. Wellness Monitor (WM) [4] proporciona a los pacientes un monitoreo continuo de sus signos vitales. Los pacientes tienen sensores en varias partes de su cuerpo. Estos sensores pasan los datos a un accesorio inteligente. Los datos recopilados se envían para su análisis a un dispositivo móvil y luego se envían a una repositorio de datos central. Este repositorio permite a los médicos acceder a los datos de los pacientes. WM realiza la representación de información basada en roles, lo que significa que el paciente solo visualiza sus signos vitales en un informe basado en gráficos, mientras que el médico tiene acceso a todo el repositorio de datos. WM permite a los pacientes con cáncer participar activamente en el monitoreo de su estado de salud. Aunque WM mejora la salud personal, toma signos vitales pero no registra síntomas.

MyMobileDoc [10] es una herramienta móvil diseñada para permitir que los pacientes se involucren activamente en el manejo de sus enfermedades crónicas como la diabetes y la hipertensión. Los pacientes envían sus datos manualmente (presión arterial, niveles de glucosa en la sangre, intensidad del dolor, etc.), desde una aplicación móvil o un navegador de Internet hacia un Centro de

Datos Médico. Los datos recibidos pueden ser evaluados automáticamente o por personal médico. Con base en estos datos, se envía inmediatamente al paciente una retroalimentación. MyMobileDoc es una aplicación centrada en el paciente que le brinda la posibilidad de aumentar su participación y su conciencia de la salud. Aunque esta aplicación permite registrar la intensidad del dolor, no está claro si se puede especificar el lugar donde ocurre un dolor físico.

MyJourney Compass [5] explora el potencial de las herramientas de gestión de la salud para mejorar la calidad de vida de los pacientes con cáncer. En particular, realizaron un estudio sobre pacientes con cáncer de mama y propusieron información relacionada con la salud facilitando documentos, un conjunto de aplicaciones de salud existentes, enlaces de información de salud y aplicaciones de entretenimiento para ser utilizadas en tabletas. My Journey Compass explora los patrones de uso de los pacientes con cáncer a lo largo de sus jornadas y descubrieron que los pacientes usaban la tableta como ayuda en la organización y como memoria al capturar notas y registrar citas en aplicaciones existentes. Sin embargo, en MyJourney Compass no hay un modelo conceptual sobre los síntomas.

En e-ESAS [3] los pacientes con cáncer de mama son los usuarios principales. Enfocaron su solución en dos conceptos: motivación y automatización. Para aumentar la motivación de los pacientes, desarrollaron dos videos para reducir las barreras existentes que enfrentan los pacientes para continuar el tratamiento. Para obtener datos del paciente automáticamente, desarrollaron un sistema de monitoreo de síntomas basado en dispositivos móviles utilizando el Sistema de evaluación de síntomas de Edmonton (ESAS). Esta herramienta permite a los pacientes registrar sus síntomas utilizando un cuestionario ESAS. Además, los pacientes registran el dolor mínimo, máximo y promedio de las últimas 24 horas. Los médicos pueden ver, para cada paciente, una gráfica longitudinal de un síntoma en un período de tiempo seleccionado. Además, pueden administrar las recetas de los pacientes y crear una generación de alertas basada en condiciones predefinidas sobre el dolor del paciente. Aunque e-ESAS mejora la salud personal y registra síntomas, solo es posible registrar el dolor en una escala de diez, desde “Sin dolor” hasta “El peor dolor posible”, pero no es posible registrar dónde se encuentra un dolor físico.

BodyDiagrams [6] es una aplicación en línea para registrar síntomas físicos a través de dibujos y texto. Los pacientes pueden codificar visualmente los atributos de los síntomas, como la gravedad, la frecuencia y la secuencia temporal. BodyDiagrams ofrece un conjunto de patrones de dibujo para indicar una ubicación general o precisa de los síntomas, un campo de texto para describir el síntoma y flechas para asociar las anotaciones del texto a las marcas dibujadas. Además, los pacientes pueden pedir consejo a los médicos. BodyDiagrams mejoró la comunicación entre pacientes y médicos porque los pacientes tuvieron mayor confianza en sus descripciones de BodyDiagrams que en las descripciones de solo texto. Sin embargo, BodyDiagrams solo registra síntomas físicos en modo en línea y no en modo fuera de línea.

Además, existen algunas aplicaciones móviles para el cuidado de la salud.

ARVmobile v1.0 [8] es una aplicación móvil, de monitoreo de la salud para medir la presión arterial de manera ambulatoria, mejorar la detección temprana y la intervención de la hipertensión; y detectar la presión arterial anormal potencial y la frecuencia cardíaca. Dicha aplicación incluyó sensores para detectar la presión arterial y las señales del ritmo cardíaco y un teléfono inteligente como el receptor para recopilar los datos digitales transmitidos y procesarlos para proporcionar información personalizada, de manera inmediata, al usuario. Los resultados preliminares demuestran que la aplicación ARVmobile v1.0 podría usarse para realizar un perfil detallado de los usuarios. Si bien la aplicación ARVmobile v.1.0 puede ser útil para la atención médica, no permite registrar síntomas.

La salud se ha convertido en un reto para el desarrollo de aplicaciones móviles. En [1] se desarrolló una aplicación móvil para apoyar el aprendizaje al comparar los efectos de dos tipos de aprendizaje diferentes basados en los logros, relacionados con la salud, conseguidos por la actividad física de los estudiantes. La aplicación pudo encontrar los factores clave del diseño de aprendizaje asistido y los logros de los estudiantes para los maestros y los educadores. [9] presenta un accesorio inteligente para el monitoreo de electrocardiogramas (ECG) que integra un sensor inalámbrico de diseño propio para la adquisición de señales de ECG. El sistema podría ser útil para detectar anomalías cardíacas en el entorno familiar y contribuir a la prevención, el diagnóstico oportuno y el tratamiento eficaz de las enfermedades cardiovasculares. [1] y [9] son dos sistemas muy útiles para el monitoreo de la salud y la prevención de enfermedades; sin embargo, no le dan al paciente la posibilidad de expresar síntomas que son de gran utilidad para un diagnóstico temprano.

En este documento, se propone el diseño de una aplicación móvil para registrar síntomas. Aunque esta aplicación puede ser utilizada por cualquier persona/paciente, nuestro usuario principal es un paciente con cáncer. Consideramos a los teléfonos inteligentes como dispositivos específicos debido a su ubicuidad.

3. Metodología

Se siguió una metodología colaborativa para el desarrollo de software [2] basada en una versión ágil del Proceso Unificado de Rational [7]. Este método se centra en la recopilación de requerimientos en dos fases: inicio y elaboración. En este método, el equipo del proyecto estuvo compuesto por tres diseñadores de interfaz de usuario y tres analistas de sistemas que trabajaron juntos para comprender los requerimientos. En el proceso participaron dos pacientes con cáncer, tres pacientes sin cáncer y un familiar de un paciente con cáncer. Además, se consultó a cinco médicos sobre la pertinencia del proyecto.

El equipo del proyecto llevó a cabo: 1) reuniones, 2) talleres y 3) trabajos individuales después de cada taller tanto en la fase inicial como en la fase de elaboración (ver Fig. 1). Las reuniones sirvieron para capturar y validar los requerimientos de los usuarios. El equipo del proyecto, los pacientes con cáncer y los familiares de los pacientes se reunían al final de la semana. Los talleres

permitieron al equipo explorar las soluciones potenciales utilizando artefactos como: diagrama de casos de uso, prototipos en papel y electrónicos, escenarios de casos de uso, documentos de visión y de especificación complementaria. Los trabajos individuales permitieron proponer al equipo una solución mediante la construcción de un artefacto. Este método de colaboración siguió un diseño iterativo (consulte las iteraciones I1 y I2 y E1 a E5 de la Fig. 1) y las iteraciones duraron una semana hábil.

3.1. Captura de los requerimientos del usuario

Con el fin de capturar los requerimientos de los usuarios, les realizamos entrevistas. Presentamos un extracto de entrevista que se realizó a dos pacientes (Entrevistado P1 y P2):

Entrevistador: ¿El médico considera los síntomas que está sufriendo en el tratamiento del cáncer?

Entrevistado P1: Sí, definitivamente. Cuando estoy en la consulta, el médico me pregunta mis síntomas, luego él decide sobre la dosis de los medicamentos o sobre un cambio de medicamento.

Entrevistado P2: Sí, creo que aunque los síntomas no son decisivos para el tratamiento, el médico los toma en cuenta.

Entrevistador: ¿Le preocupa no recordar sus síntomas durante una consulta?

Entrevistado P1: Sí, principalmente al inicio de mi tratamiento, porque no tenía la costumbre de registrarlos.

Entrevistado P2: Sí, creo que un registro de síntomas es esencial. Desde que tengo cáncer, he aprendido a registrar no solo el síntoma, sino también la fecha, la hora, la recurrencia y la intensidad. El médico siempre me presunta estos datos.

Entrevistador: ¿Utiliza una herramienta para controlar los síntomas?

Entrevistado P1: Sólo papel y lápiz. A veces mi esposo registra mis síntomas.

Entrevistado P2: Utilizo una agenda digital en mi teléfono inteligente. Escribir mis síntomas en el teléfono es práctico; aunque verlos no es fácil. Solo tengo mis síntomas por fecha pero no puedo consultar por tipo de síntoma, por ejemplo.

Entrevistador: ¿Qué características debería tener una aplicación móvil para ayudarlo en la gestión de los síntomas?

Entrevistado P1: Una aplicación móvil debe permitirme registrar, recuperar y eliminar los síntomas, pero de una manera sencilla.

Entrevistado P2: Una aplicación móvil debería ayudarme a registrar, modificar, recuperar y eliminar los síntomas. Con una aplicación móvil, podría encontrar rápidamente un síntoma por nombre o por un período de tiempo.

Después de entrevistar a dos pacientes con cáncer acerca de su experiencia durante la cita con el médico, expresaron estar preocupados por no poder comunicar todos sus síntomas o por no comunicarlos de manera precisa. Los pacientes recibieron tratamientos para el cáncer en México, en dos hospitales públicos diferentes, donde el número de pacientes que asisten es alto y el tiempo asignado a cada cita es de aproximadamente diez minutos. Le preguntamos a cinco médicos sobre la necesidad de una herramienta móvil para registrar los síntomas de los

pacientes con cáncer. Al principio, uno de ellos no estaba de acuerdo porque pensaba que la aplicación intentaría reemplazar la comunicación con él. Les explicamos que nuestro objetivo era mejorar la comunicación y apoyar su toma de decisiones, no sustituirla, todos estuvieron de acuerdo con la aplicación.

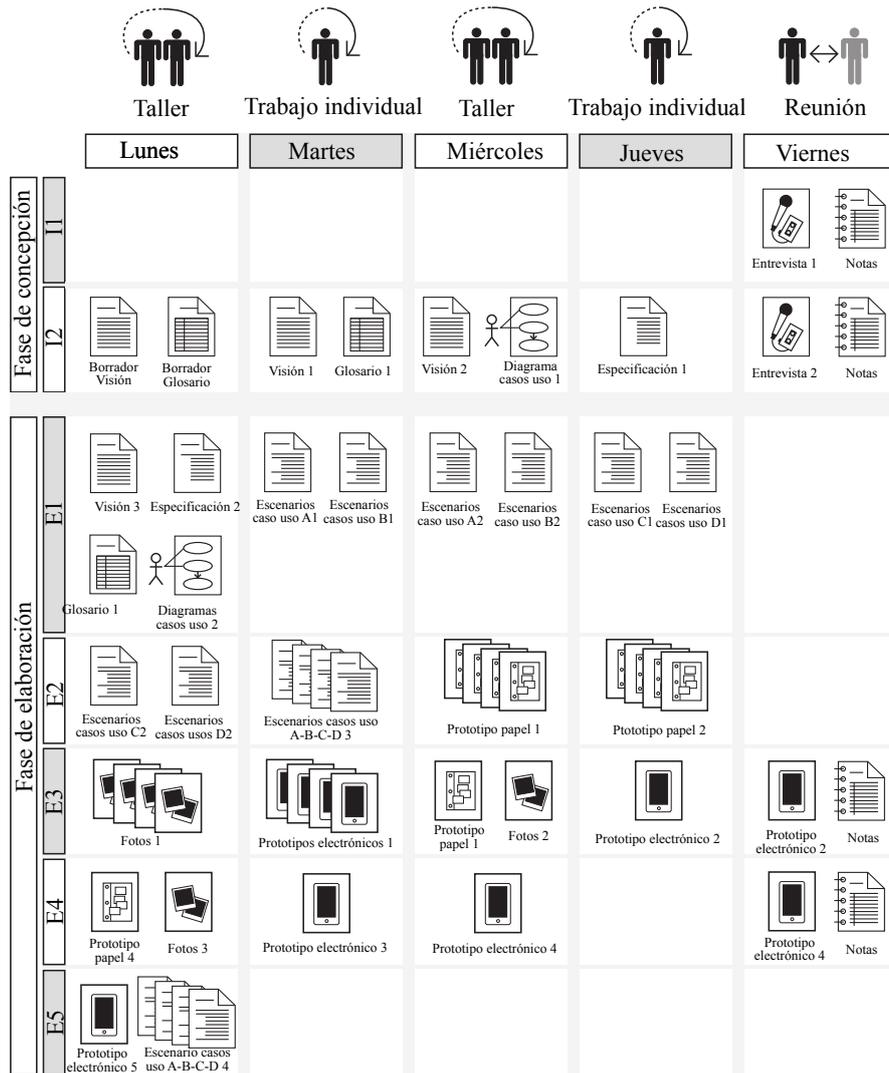


Fig. 1. Sistema MoCa: fases, iteraciones, y artefactos.

3.2. Requerimientos MoCa

Después de entrevistar a los usuarios, los miembros del equipo levantaron los requerimientos de la aplicación MoCa. El objetivo principal de MoCa es ayudar a mejorar la calidad de vida de los pacientes con cáncer, pero este objetivo es doble: 1) aumentar la conciencia personal de un paciente y la responsabilidad personal sobre su enfermedad y 2) proporcionar una herramienta para registrar los síntomas. Si un paciente puede comunicar sus síntomas a sus médicos de manera más precisa, los médicos pueden recetarles medicamentos para aliviar su dolor y así mejorar su calidad de vida. Para ello, los requerimientos de MoCa son:

- I. **Mejorar la comunicación entre pacientes y médicos.** Para mejorar la comunicación, queremos que un paciente pueda registrar y gestionar sus síntomas. Los atributos de los síntomas son: nombre, nivel de intensidad del dolor, descripción, fecha de inicio, fecha de finalización y periodicidad.
- II. **Mejorar la usabilidad.** Los pacientes con cáncer enfrentan muchas dificultades, tales como: reorganización familiar, permiso de trabajo, etc. MoCa debe proporcionar a los pacientes un fácil aprendizaje de la aplicación y que los pacientes necesiten solo unos segundos para registrar un síntoma. Para facilitar el uso de MoCa, consideramos aprovechar los gestos de teléfonos inteligentes.
- III. **Permitir el uso en cualquier momento y en cualquier lugar.** Para permitir esto, consideramos los teléfonos inteligentes como el dispositivo de destino para soportar esta aplicación.
- IV. **Permitir registrar síntomas en el modo fuera de línea.** Dado que en algunas áreas del hospital y/o en algunas áreas y/o en algunos momentos no hay conexión a Internet, MoCa debe poder registrar los síntomas en el modo fuera de línea.

El requerimiento III se cumplirá mediante la programación de MoCa en las dos plataformas de computación más importantes: iOS y Android. El requerimiento IV se cumplirá mediante el uso de una base de datos integrada para el almacenamiento local, y al proporcionar una copia de seguridad y restauración de los síntomas del usuario en un servidor remoto. Los requerimientos I y II se integrarán en las funcionalidades de MoCa.

3.3. Funcionalidades MoCa

En el transcurso del diseño iterativo, se analizaron las siguientes funcionalidades para MoCa:

1. **Registrar un síntoma.** Un paciente puede registrar un síntoma de dos maneras:
 - Utilizando el teclado. En el momento en que el paciente escribe un síntoma, el sistema los filtra y presenta los síntomas relacionados con las letras tecleadas.

- Usando un cuerpo humano. El paciente puede seleccionar un área de la figura de un cuerpo humano y el sistema presenta los síntomas asociados con esa área del cuerpo.

Cuando el paciente selecciona un nombre de síntoma e introduce su intensidad (opcionalmente, una breve nota), el sistema considera la fecha y hora actuales para el síntoma. Esta funcionalidad mejorará la usabilidad, por lo tanto, cubrirá el requerimiento II, al proporcionar diferentes modos de entrada.

2. **Ver síntomas.** El paciente puede ver una lista de los síntomas registrados más recientemente. Esta funcionalidad mejorará la usabilidad y la comunicación entre el paciente y sus médicos, cumpliendo con los requerimientos I y II.
3. **Copia de seguridad o recuperación de síntomas.** Esta función permite guardar o recuperar los síntomas del paciente a/desde un servidor remoto. Mantener una copia en otro sitio es útil para fines de copia de respaldo.
4. **Modificar un síntoma.** Cuando el paciente registra un síntoma, no sabe cuándo podría terminar ni su periodicidad. Por lo tanto, el paciente puede actualizar cualquier atributo de un síntoma, excepto su nombre. Esta función le permite al paciente completar un registro de síntomas.
5. **Eliminar un síntoma.** Esta función permite al paciente eliminar un síntoma. Esta función no se consideró al inicio del proyecto, pero se agregó porque el paciente pudo haberlo registrado por error.

3.4. Prototipo en papel de MoCa

Para comprender mejor cómo administrar los síntomas para mejorar la comunicación y la usabilidad, se implementaron prototipos en papel de un teléfono inteligente y creamos una secuencia de fotos de su uso (ver Fig. 2). El primer boceto (Principal) muestra la figura de un cuerpo humano a través del cual es posible registrar los síntomas. Después, un paciente puede seleccionar una región del cuerpo humano y el sistema mostrará el conjunto de síntomas que pueden asociarse con la región del cuerpo seleccionada; el paciente puede seleccionar un síntoma (Captura de síntomas por región anatómica). Los últimos bocetos (Resumen de síntomas) le permiten buscar un síntoma escribiendo su nombre y, más tarde, modificarlo.

3.5. Aplicación MoCa

En esta sección, presentamos la aplicación MoCa y cómo cumple con los requerimientos presentados. La Fig. 3 muestra la interfaz que está presente al abrir MoCa por primera vez. Esta interfaz muestra un mensaje de bienvenida donde el usuario escribe su nombre, coloca su género y selecciona su fecha de nacimiento (ver Fig. 3a)). Una vez que se presiona el botón “Listo”, el sistema presenta la figura de un cuerpo humano (masculino o femenino) según el género indicado en la interfaz de bienvenida. En la parte inferior de la Fig. 3b), hay cuatro íconos que describen las funcionalidades principales de MoCa (de izquierda a derecha):

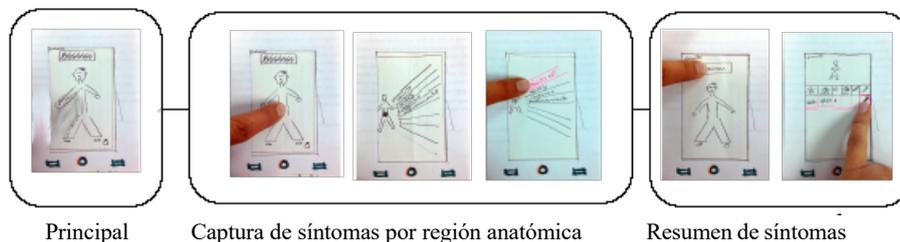


Fig. 2. Secuencia de fotos de prototipos en papel de la aplicación MoCa.

registrar un síntoma utilizando el teclado, ver la parte posterior del cuerpo humano, hacer copia de seguridad o restaurar los síntomas del usuario, y ver la lista de síntomas. Estos íconos permiten alcanzar con un solo toque la mayoría de las funcionalidades de MoCa, mejorando así la usabilidad (requerimiento II). Además, la Fig. 3c) muestra la lista de los síntomas (*Resumen*). Cuando el usuario selecciona el ícono “lápiz”, puede editar el síntoma. Por ejemplo, cuando el usuario selecciona el ícono “lápiz” debajo de la etiqueta “Fecha de finalización”, el sistema permite al usuario registrar la fecha en que desapareció el síntoma. La interfaz “Ver síntomas” permite a los usuarios ver fácilmente la lista de síntomas, mejorando así la facilidad de uso (requerimiento II) y permitiendo recordar los síntomas para comunicarse con el médico (requerimiento I).

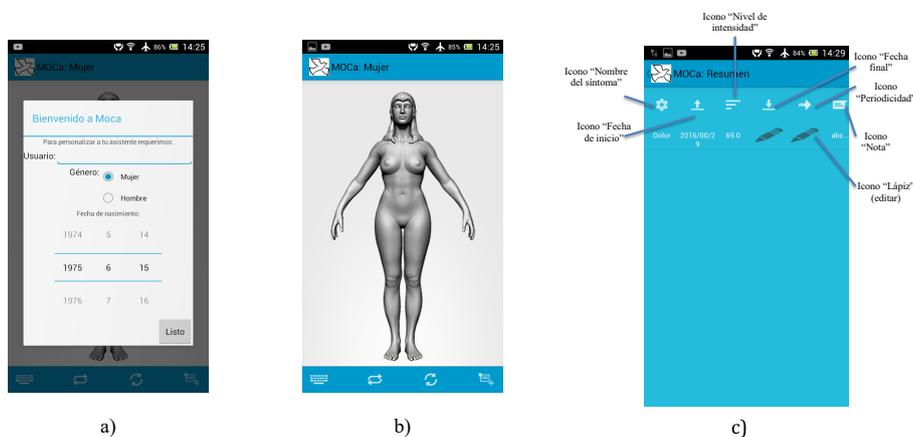


Fig. 3. Interfaces MoCa: a) Bienvenido a MoCa, b) Cuerpo humano y c) Ver síntomas.

El cuerpo humano permite al usuario registrar un síntoma seleccionando una región anatómica (ver Fig. 4). Por ejemplo, cuando el usuario toca la cabeza del cuerpo humano (ver Fig. 4a), el sistema presenta una lista de síntomas asociados

con esta región (vea Fig. 4b). Cuando el usuario selecciona el dolor de cabeza, el sistema presenta una ventana de “nivel de intensidad del dolor”. El usuario puede usar el control deslizante para seleccionar entre 1 y 10 (ver Fig. 4c)) y puede describir el síntoma (por ejemplo, intenso) y puede aceptar o cancelar el síntoma. Los gestos táctiles de apoyo permiten a los usuarios seleccionar regiones para registrar fácilmente un síntoma, mejorando así la facilidad de uso (requerimiento II).

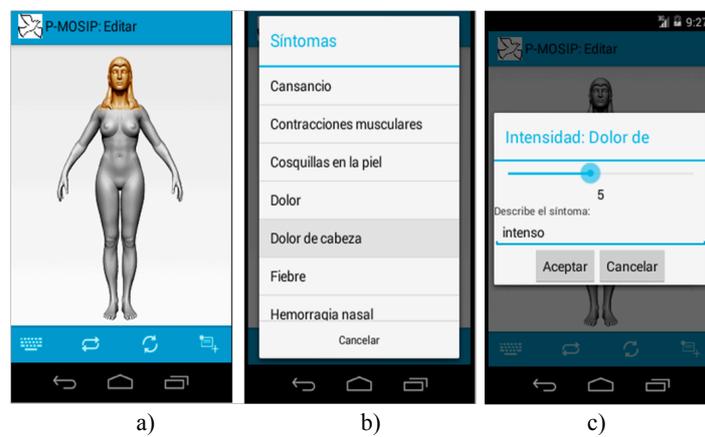


Fig. 4. Registrar un síntoma utilizando MoCa.

4. Resultados y discusión

MoCa ha sido desarrollado para Android Jelly Bean 4.3. Las funcionalidades de copia de seguridad y restauración se implementaron como servicios RESTful en un servidor remoto. MoCa es un cliente RESTful que solicita servicios de copia de respaldo y restauración.

Se realizaron pruebas cualitativas y se les pidió a los pacientes que habían evaluado la funcionalidad de MoCa que enviaran un comentario de forma libre. Se les pidieron comentarios sobre las fortalezas y debilidades de cada interfaz. Todos los participantes respondieron positivamente, les gustó la aplicación. Una de las ventajas citadas de un paciente con cáncer incluye la facilidad para registrar un síntoma: “en mi opinión, el cuerpo humano me permite registrar un síntoma fácilmente, simplemente tocando la figura de un cuerpo humano, elegir un síntoma y elegir el nivel de intensidad del dolor. Cuando vuelvo a casa después del tratamiento contra el cáncer, me siento agotado y necesito descansar, pero puedo registrar mis síntomas de una manera fácil, así que lo hago”. A otro

paciente de cáncer le gustó la lista de síntomas y dijo: ‘La lista de síntomas me permite comunicar mis dolores a los médicos al consultar el resumen en una ventana en lugar de tener que buscar en mi agenda’. Sin embargo, este usuario dijo: “Me confunden algunos íconos, en particular la ventana de visualización de síntomas, tuve que preguntar el significado de algunos de ellos”.

Para discutir el trabajo realizado, se relacionan las funcionalidades y los requerimientos de la siguiente manera:

- El paciente puede, de una manera fácil, registrar un síntoma seleccionando una región de un cuerpo humano o escribiendo el nombre del síntoma; por lo que se mejora la usabilidad de MoCa. Se proporcionan dos modos de registro porque un paciente puede tener confianza con una forma humana para explicar su síntoma o puede tener más confianza al escribir el nombre del síntoma. La navegación tiene la flexibilidad suficiente para permitir que los pacientes naveguen en ambos modos.
- El soporte para registrar y modificar un síntoma se realiza con solo tres pasos y se solicita una cantidad mínima de información al registrar un síntoma; por lo que se mejora la usabilidad de MoCa. Además, esta tarea de uso frecuente está disponible de inmediato. Al abrir MoCa, el paciente puede registrar un síntoma seleccionando un área del cuerpo humano. Esta tarea mejora la usabilidad de MoCa.
- El paciente puede ver sus síntomas con solo un paso. El paciente solo toca el ícono “Ver síntomas” y puede visualizar sus últimos síntomas registrados con todas sus características (nombre del síntoma, nivel de intensidad, notas, fecha de inicio, fecha de finalización y periodicidad) en una sola fila. Esta vista mejora la usabilidad de MoCa y permite al paciente describir sus síntomas de una manera precisa; por lo que la comunicación entre el paciente y sus médicos es más precisa.
- El paciente puede modificar, desde la interfaz “Ver síntomas”, un atributo de un síntoma solo presionando el ícono “lápiz” debajo del atributo a modificar. Los pacientes pueden cambiar fácilmente un registro de síntomas. Esta característica mejora la usabilidad de MoCa. Sin embargo, los usuarios se confunden con algunos íconos dentro de esta interfaz. Antes de la implementación de MoCa, se debe realizar un trabajo adicional para proponer íconos más informativos en esta interfaz.
- El paciente puede eliminar, desde la interfaz “Ver síntomas”, un síntoma con tan solo deslizar hacia la izquierda el renglón del síntoma y enviarlo a la “Papelera”. El paciente tiene la oportunidad de confirmar la acción de eliminación. De esta manera, los pacientes pueden eliminar fácilmente un síntoma, pero al mismo tiempo, el sistema proporciona retroalimentación. Esta característica mejora la usabilidad de MoCa.
- El paciente puede respaldar y restaurar sus síntomas a/desde un servidor remoto. Esta función mejora la confiabilidad de la aplicación, porque en caso de falla de un teléfono inteligente, sus registros de síntomas están respaldados. Sin embargo, los usuarios prefieren hacer una copia de seguridad y restaurar desde la nube, en lugar de un servidor en particular.

5. Conclusiones y trabajo a futuro

Este documento ha presentado los resultados del diseño de MoCa, una aplicación móvil para registrar síntomas de pacientes con cáncer mediante la realización de entrevistas con los usuarios. El registro de síntomas se llevó a cabo utilizando una figura corporal o un teclado.

Nuestra contribución es proporcionar a los pacientes con cáncer una aplicación móvil para registrar sus síntomas de una manera fácil. Este problema es relevante porque los pacientes deben aprender a ser conscientes de sus signos y síntomas pero hay que tener en cuenta que pueden estar en tratamiento, están agotados y debe ser fácil de usar. Los pacientes deben comunicar sus síntomas a sus médicos para mantener una mejor calidad de vida.

La aplicación móvil obtenida (MoCa) puede ayudar a los médicos a tomar decisiones con pacientes con cáncer, como mejorar la monitorización, la continuidad del tratamiento y la calidad de vida del paciente.

Nuestro plan es modificar los íconos que no fueron relacionados fácilmente con sus funciones y agregar la función de copia de seguridad y restauración desde la nube. Una vez que se hayan hecho estas modificaciones, planeamos realizar una prueba de usabilidad mas grande con pacientes en un hospital oncológico.

Además, como trabajo futuro, nuestra aplicación MoCa considera registrar el tratamiento realizado, los eventos adversos en los pacientes o las alergias a los medicamentos. Sería muy útil si el paciente pudiera registrar estos aspectos utilizando la voz como interfaz de comunicación. Por último, se contempla generar modelos de clasificación supervisada aplicada a los síntomas de los pacientes.

Agradecimientos. Agradecemos la participación de los pacientes y médicos que apoyaron durante el diseño de MoCa. Los autores agradecen ampliamente a la Universidad Autónoma Metropolitana por el apoyo al proyecto no. SI001-14.

Referencias

1. Cheng, C.H., Chen, C.H.: Developing a Mobile APP-Supported Learning System for Evaluating Health-Related Physical Fitness Achievements of Students. *Mobile Information Systems* 2018, 15 (2018), <https://doi.org/10.1155/2018/8960968>
2. Gallardo-Lopez, L., Gonzalez-Beltran, B.A., Garcia-Madrid, R., Ferruzca, M., Zafra-Ballinas, I.A., Reyes-Ortiz, J.A.: Collaborative working: Understanding mobile applications requirements. In: 2015 International Conference on Computational Science and Computational Intelligence (CSCI). pp. 139–142 (Dec 2015)
3. Haque, M., Kawsar, F., Adibuzzaman, M., Ahamed, S., Love, R., Dowla, R., Roe, D., Hossain, S., Selim, R.: Findings of e-esas: A mobile based symptom monitoring system for breast cancer patients in rural bangladesh. In: Proceedings of the SIG-CHI Conference on Human Factors in Computing Systems. pp. 899–908. CHI '12, ACM, New York, NY, USA (2012), <http://doi.acm.org/10.1145/2207676.2208532>
4. Islam, R., Ahamed, S.I., Talukder, N., Obermiller, I.: HCI and Usability for Medicine and Health Care: Third Symposium of the Workgroup Human-Computer

- Interaction and Usability Engineering of the Austrian Computer Society, USAB 2007 Graz, Austria, November, 22, 2007. Proceedings, chap. Usability of Mobile Computing Technologies to Assist Cancer Patients, pp. 227–240. Springer Berlin Heidelberg, Berlin, Heidelberg (2007)
5. Jacobs, M.L., Clawson, J., Mynatt, E.D.: My journey compass: A preliminary investigation of a mobile tool for cancer patients. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 663–672. CHI '14, ACM, New York, NY, USA (2014), <http://doi.acm.org/10.1145/2556288.2557194>
 6. Jang, A., MacLean, D.L., Heer, J.: Bodydiagrams: Improving communication of pain symptoms through drawing. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 1153–1162. CHI '14, ACM, New York, NY, USA (2014), <http://doi.acm.org/10.1145/2556288.2557223>
 7. Larman, C.: Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development. Prentice Hall, Upper Saddle River, N.J, third edn. (Oct 2004)
 8. Mena, L.J., Felix, V.G., Ostos, R., Gonzalez, J.A., Cervantes, A., Ochoa, A., Ruiz, C., Ramos, R., Maestre, G.E.: Mobile Personal Health System for Ambulatory Blood Pressure Monitoring. *Computational and Mathematical Methods in Medicine* 2013, 13 (2013), <http://dx.doi.org/10.1155/2013/598196>
 9. Mena, L.J., Félix, V.G., Ochoa, A., Ostos, R., González, E., Aspuru, J., Velarde, P., Maestre, G.E.: Mobile Personal Health Monitoring for Automated Classification of Electrocardiogram Signals in Elderly. *Computational and Mathematical Methods in Medicine* 2018, 9 (2018), <https://doi.org/10.1155/2018/9128054>
 10. Nischelwitzer, A., Pintoffl, K., Loss, C., Holzinger, A.: HCI and Usability for Medicine and Health Care: Third Symposium of the Workgroup Human-Computer Interaction and Usability Engineering of the Austrian Computer Society, USAB 2007 Graz, Austria, November, 22, 2007. Proceedings, chap. Design and Development of a Mobile Medical Application for the Management of Chronic Diseases: Methods of Improved Data Input for Older People, pp. 119–132. Springer Berlin Heidelberg, Berlin, Heidelberg (2007)
 11. Reyes-Ortiz, J.A., González-Beltrán, B.A., Vidal, M.T.: Clasificación de reportes clínicos para apoyar el diagnóstico del cáncer. *Pistas Educativas* 39(128), 1347–1361 (febrero 2018)
 12. World Health Organization: Global Status Report on noncommunicable diseases 2014. WHO Press, Switzerland (2014)

Selección de atributos mediante separación de centroides

Héctor Jiménez-Salazar, Alfredo Mateos-Papis, Christian Sánchez-Sánchez

Universidad Autónoma Metropolitana, Unidad Cuajimalpa,
Ciudad de México, México

{hjimenez, amateos, csanchez}@correo.cua.uam.mx

Resumen. Este trabajo presenta la aplicación de un método basado en la separación de centroides para seleccionar atributos, cuyos resultados fueron confrontados con los obtenidos a través del método de ganancia de información. Estos métodos se aplicaron al problema de determinar las materias previamente aprobadas que influyen en la aprobación de una nueva materia de un plan de estudios universitario. Los experimentos consideraron varias nuevas materias de interés y, para conocer la efectividad de cada uno de los métodos, los atributos seleccionados (materias previamente cursadas) se usaron en un clasificador de árbol de decisión. Se destaca que el uso de separación de centroides mejora la clasificación con relación a lo que se obtiene por ganancia de información pero, además, los atributos que la separación de centroides seleccionó resultaron ser los que se esperaba obtener cualitativamente.

Palabras clave: selección de atributos, clasificación, desempeño escolar.

Feature Selection through Centroids Separation

Abstract. This paper presents the application of a method based on the separation of centroids to select attributes, whose results were compared with those obtained through the information gain method. These methods were applied to the problem of determining the previously approved subjects that influence the approval of a new subject of a university curriculum. The experiments considered several new subjects of interest and, in order to determine the effectiveness of each one of the methods, the selected attributes (previously coursed subjects) were used in a decision tree classifier. It is emphasized that the use of separation of centroids improves the classification in relation to what is obtained through information gain but, in addition, the attributes selected through separation of centroids turned out to be those that were expected to be obtained qualitatively.

Keywords: feature selection, classification, school performance.

1. Introducción

La selección de atributos de instancias de una colección de datos puede ser definida como el proceso en el que se escoge el subconjunto mínimo de atributos a partir de un conjunto original [13] de tal forma que mejore el desempeño del clasificador que los usa. Este proceso sigue siendo un problema de gran importancia ya que impacta una amplia variedad de aplicaciones pero, además, porque no hay un método general que pueda ser aplicado en cualquier contexto [5]. En los problemas reales comúnmente se cuenta con poca información y este hecho impide aplicar con efectividad los métodos de selección existentes. En este trabajo se experimenta con un método basado en la separación de los centroides de las clases para identificar los atributos que influyen en la clasificación. Aquí se reporta que los atributos que separan centroides tienen ventajas con respecto al que usa árboles de decisión, basado en ganancia de información de los atributos.

El problema es identificar cuáles materias de un plan de estudios influyen en otras, es decir, si habiendo aprobado la materia A hay más seguridad de aprobar la materia objetivo B, o bien, en términos escolares, A debe ser requisito de B, pues en B se usan los conocimientos presentados en A. Si bien este problema se resuelve habitualmente con la opinión de especialistas, consideramos que hay múltiples factores que están involucrados. Particularmente, disponemos de información sobre las evaluaciones realizadas a los alumnos en un plan de estudios universitario, en el cual hay casos de materias que no están seriadas entre sí, mientras que, normalmente, en otros planes de estudio materias similares sí lo están. Así pues, con esta información puede conocerse el efecto de no seriar algunas materias; algo que no es posible en cualquier plan de estudios seriado de la forma acostumbrada.

Para ciertos casos el problema podría ser resuelto directamente por algún método bien conocido como una prueba de hipótesis. Por ejemplo, considerando como variables aleatorias la proporción de aprobados en cada grupo que toma B en dos casos: quienes han aprobado previamente A, x , y quienes no, y , la hipótesis a probar sería $\bar{x} > \bar{y}$. Este procedimiento debería ser repetido para cada par de materias donde creemos que existe relación. Otro enfoque podría tratar, más bien, de descubrir cuáles materias, de entre varias previamente aprobadas, influyen en la materia objetivo. Este último enfoque fue el seguido pues, además, existen diversas herramientas que pueden ser aplicadas y comparar su efectividad. Justamente, se utilizó la técnica de análisis basada en la separación de centroides, y el clasificador por árboles de decisión para validar los atributos obtenidos.

En lo que resta de este artículo, se exponen algunos trabajos relacionados con el tema, en la sección 2; los métodos utilizados, en la sección 3; los experimentos, en la sección 4; y las conclusiones del presente trabajo en la sección 5.

2. Trabajos relacionados

Dentro de los trabajos cercanos a la identificación de factores que influyen en la aprobación de una materia, hay trabajos relacionados con el contenido

de los cursos impartidos y la generación de rutas flexibles (ordenamiento de los contenidos) que se adapte al perfil de los alumnos. Por ejemplo, Idris et al. [9] ofrece la secuencia personalizada de temas de un curso de Java. Los autores usaron una red neuronal. Los atributos evaluados por la red neuronal son el conocimiento del alumno de algunos temas de programación y el conocimiento sobre el dominio. Por otro lado, Chen [4] propuso un sistema multi-agente e-learning basado en algoritmos genéticos que permite guiar a los alumnos sobre una serie de temas relacionados a un curso. Años después Dwi & Basuki [2] propusieron un sistema similar para el aprendizaje del idioma inglés.

Más allá de los contenidos de los cursos Kovacic [11] utilizó la clasificación basada en árboles de decisión para predecir, tempranamente, el éxito de los estudiantes basándose en atributos socio-demográficos (edad, etnicidad, género, y educación, entre otros) y de su entorno de estudio (cursos del programa y bloque de cursos). El conjunto de datos provino de una universidad politécnica de Nueva Zelanda. De cuatro árboles de decisión construidos, el que obtuvo mejor desempeño refiere una exactitud del 60.5%, baja, argumentan, debido a que hay otros elementos, no contenidos en el conjunto de datos que pueden influir en el éxito de los estudiantes. En este trabajo hay muchos factores que influyen en la aprobación de materias. Incorporar este tipo de información, a las colecciones utilizadas en los experimentos expuestos, podría ayudar a determinar de manera más precisa cuál es el grado de influencia de los conocimientos previos y descubrir la importancia de otros factores.

Por otro lado, Barrak et al. [1] presentan un enfoque para estimar la Calificación Promedio Final (Grade Point Average) mediante Árboles de Decisión. Como herramienta principal utilizaron WEKA. Fueron generados Árboles de Decisión (algoritmo J48) por periodo, considerando los principales cursos. Este es un trabajo muy parecido al que se ha tratado en el presente trabajo. Desafortunadamente no contiene datos sobre su desempeño.

Es importante mencionar que aún cuando los datos académicos sean vastos puede que dentro de ellos se encuentren algunos atributos que no aporten mucho al análisis y procesamiento, aunque siempre es necesario tener el conocimiento del dominio sobre el que está trabajando, existen algunas técnicas que ayudan a seleccionar atributos. A continuación se ofrece mayor información al respecto.

El problema de selección de atributos es un problema que sigue siendo de interés pues no hay solución general. Según Kohavi [10] obtener el mejor conjunto de atributos puede ser un problema intratable y muchos problemas relacionados a la selección de atributos han mostrado tener una complejidad alta (NP-hard) [3]. La adecuada selección de atributos no solo tiene consecuencias dentro de la solución del problema, además, impacto en la problemática de grandes áreas de desarrollo [7]. Algunas de las técnicas de selección de atributos más usadas son: *Information Gain*, *Gain Ratio*, *Symmetrical Uncertainty* [8], Chi-Squared [12], Gini-Index y el Análisis de componentes Principales (PCA por sus siglas en inglés), la cual es una técnica que reduce la dimensionalidad y ayuda a identificar los atributos más influyentes. Aunque algunos algoritmos de clasificación están diseñados para escoger las características más relevantes y dejar fuera las irrele-

vantes, como los árboles de decisión, y las redes neuronales multicapa, estos se benefician de métodos complementarios para elegir atributos. Además, en el caso de los árboles de decisión, es posible proceder con conjuntos de datos pequeños y pocos atributos, como los documentos cortos: se ha constatado que la selección de atributos beneficia los resultados de la clasificación de documentos [16].

En relación a la técnica de separación de centroides hay poco uso reportado al respecto. Un conjunto de técnicas derivadas del trabajo de R. Fisher [6] sobre análisis discriminante, como los clasificadores lineales (LSM, SVM, etc.) se han usado con la implícita referencia a la selección de atributos que separan clases de instancias. Estos métodos exigen, en cada caso, condiciones sobre la colección (distribución o tamaño, por ejemplo) y cálculos que pueden ser largos. El antecedente del presente trabajo se encuentra en G. Salton [15] quien reporta el uso del centroide para calcular la densidad de documentos en función de los términos que mejor los distinguen, lo cual es una simplificación de los métodos derivados del análisis discriminante.

3. Metodología

En las subsecciones siguientes se expone cada uno de los métodos utilizados.

3.1. Árboles de decisión

Un algoritmo de clasificación es capaz de extraer patrones de un conjunto de entrenamiento, que son útiles para crear un modelo para clasificar nuevos datos. El algoritmo de Árboles de Decisión crea un modelo en la forma de un árbol, y funciona dividiendo el conjunto de datos en subconjuntos más pequeños, por medio de evaluar cada uno de los atributos de acuerdo con la Ganancia de Información(GI) para formar el árbol [14]. La GI de un atributo considera: la entropía de la colección:

$$H(C) = \sum_{clases} \Pr_{clase} \log_2\left(\frac{1}{\Pr_{clase}}\right), \quad (1)$$

la entropía relativa a un atributo:

$$H(A) = \sum_{v \in A} \frac{|C_{v \in A}|}{|C|} H(C_{v \in A}), \quad (2)$$

y la ganancia de un atributo:

$$A : GI(A) = H(C) - H(A). \quad (3)$$

Algunas de las razones por las que se eligió este algoritmo son las siguientes: Dentro del diseño de este algoritmo se seleccionan solo atributos con un umbral base, $GI > 0$, y los demás quedan fuera, haciendo implícitamente una selección de atributos. También permite lidiar con conjuntos de datos que tengan clases

desbalanceadas, ya que busca patrones que caracterizan todas las clases, lo cual se expresa por reglas de clasificación que facilitan el análisis visual, en forma de un árbol.

Con cada conjunto de datos se generó un modelo utilizando el Algoritmo de Árboles de Decisión en Python seleccionando el método de GI para formar el árbol. Cada modelo fue evaluado mediante validación cruzada, partiendo el conjunto de datos original en conjuntos de entrenamiento y prueba de manera aleatoria un número determinado de veces; en este caso 10 veces. Cada vez que se realiza la clasificación se evalúa la exactitud y, finalmente se promedia la exactitud de los resultados. La exactitud se calcula con la fórmula: $ACC = (\text{número de predicciones correctas del clasificador}) / (\text{número total de predicciones})$.

Con la finalidad de afinar el modelo se fue incrementando el umbral de GI, para ir dejando fuera uno por uno de los atributos y calcular la exactitud de cada modelo.

3.2. Separación de centroides

Este método, es intuitivamente muy simple. En él se considera el conjunto de datos representado en el modelo de espacio vectorial (MEV), luego para cada clase de instancias puede calcularse su centroide y, con ellos, conocer su similitud original, esto es con todos los atributos. Se procede entonces, eliminando vorazmente los atributos y midiendo nuevamente la similitud entre los centroides: si se elimina un atributo p y la similitud entre centroides aumenta, significa que p es buen atributo para representar las instancias pues ayuda a distinguir entre instancias de diferentes clases. A continuación se exponen algunas fórmulas referentes a los conceptos mencionados.

Representación en el MEV. Sea el conjunto de instancias, C , $n = |C|$, con vocabulario de atributos V_C , y $m = |V_C|$, donde para cada $I_j \in C$, tf_{ij} representa la frecuencia del elemento $t_i \in V_C$ en la instancia I_j , $df_i = |\{k \in C | i \text{ ocurre en } k\}|$ e $idf_i = \log_2(\frac{2n}{df_i})$. La representación de una instancia de C será un vector de dimensión $|V_C|$, donde cada dimensión se asocia con un orden dado a los elementos de V_C : la entrada i de $I_j \in C$ está dada por el valor: $tf_{ij} \cdot idf_j$. Los vectores de C se usan normalizados: multiplicado cada uno por el inverso de su magnitud, $|\mathbf{x}| = \sqrt{\sum_{k=1}^m x_k^2}$.

Centroide y similitud. Dado un conjunto de instancias representadas por vectores $B = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n\}$, su centroide se calcula mediante la fórmula:

$$\bar{B} = \frac{1}{n} \sum_{k=1}^n \mathbf{v}_k. \quad (4)$$

Asimismo, la similitud de dos vectores está dada por:

$$sim(\mathbf{a}, \mathbf{b}) = \sum_{k=1}^m a_k \cdot b_k. \quad (5)$$

Separación entre clases. Supongamos que C se compone de dos clases de instancias: C_a y C_b con centroides \bar{C}_a y \bar{C}_b . Si al eliminar $p \in V_C$ de C , obtenemos $C'_{(p)}$ y $sim(\bar{C}_a, \bar{C}_b) < sim(\bar{C}'_a, \bar{C}'_b)$ significa que el atributo p ayudaba a separar las clases C_a y C_b . En suma, podemos calcular para cada atributo el aporte de separación entre clases y seleccionar solamente los que separan, esto es cuando:

$$SC(p) = sim(\bar{C}'_a, \bar{C}'_b) - sim(\bar{C}_a, \bar{C}_b) > 0. \quad (6)$$

4. Experimentos

Antes de presentar la forma en que se realizaron los experimentos, se describirán los datos utilizados.

4.1. Conjuntos de datos

Se tuvo acceso al kardex, específicamente al registro de las evaluaciones de los alumnos de la carrera Tecnologías y Sistemas de Información de la UAM-C del periodo 2007 a 2014. La información que pudo integrarse para el análisis se constituyó por registros formados por 4 valores: Id, identificador del alumno; U, clave de la materia evaluada; C, calificación obtenida; y T, periodo en que se realizó la evaluación. Con el fin de homogenizar la información se extrajo, en primer lugar, la "población media", alumnos cuyo número de materias aprobadas en el primer año está a lo más a una desviación estándar de la media ([3, 8]): de los registros originales, 7,144, se trabajó únicamente con 4,174 registros. Es importante aclarar que un alumno puede tener varias evaluaciones en una misma materia en tanto no la apruebe. A partir de esta información, en cada materia objetivo, se extrajeron los registros de todos los alumnos que fueron evaluados por primera vez. A cada uno de estas instancias se asociaron las materias que habían sido previamente aprobadas y que podían tener o no relación con la materia objetivo.

Haremos referencia a las siguientes materias: ARQ: Arquitectura de Computadoras. ED: Estructura de Datos, HYC: Historia y Cultura de la Computación, IPM: Introducción al Pensamiento Matemático, LPL: Lógica y Programación Lógica, MD1: Matemáticas Discretas I, MD2: Matemáticas Discretas II, PE: Programación Estructurada, PWE: Programación Web Estático, SO: Sistemas Operativos, SUST: Seminario de Sustentabilidad, TLENG: Taller de Lenguaje (Literacidad), y TMATE: Taller de Matemáticas,

Para ejemplificar los resultados, las materias objetivo que aquí se exponen son: ED, LPL, y SO. Para cada una de ellas se generó una colección con atributos presumiblemente influyentes y otros presumiblemente no influyentes. A continuación se muestran los atributos usados para cada una de las tres materias mencionadas:

ED: MD1, PWE, HYC, TLENG, TMATE.
 LPL: ED, MD2, PE, MD1, HYC, TLENG, SUST, TMATE.
 SO: ED, ARQ, PE, MD1, HYC, TLENG, SUST, TMATE.

Debemos agregar que ninguna de las materias que aparece en la lista de cada materia objetivo constituye uno de sus requisitos en el plan de estudios. Así, la colección para ED, por ejemplo, se formó con registros de evaluaciones de alumnos que por primera vez fueron evaluados en ED, hayan o no aprobado (clase A o R) y, en cada instancia, aparecerá MD1, solo si aprobó dicha materia; de igual forma PWE, y las demás de su lista. La instancia ‘IDk PWE TMATE HYC A’ de la colección ED ejemplifica que el alumno con identificador IDk aprobó ED y previamente aprobó PWE, TMATE, y HYC, pero no aprobó TLENG ni MD1. La tabla 1 expone los tamaños de las clases de las colecciones de ejemplo.

Tabla 1. Número de instancias en cada clase para ED, LPL y SO.

Colección	#Clase-A	#Clase-R	#Total
ED	46	37	83
LPL	56	38	94
SO	66	43	109

De acuerdo con lo descrito se tiene un problema de clasificación binaria. Este problema fue tratado con árboles de decisión y la aplicación voraz de la separación de clases. Notemos que la clasificación se emplea como indicador de la bondad de los atributos, pues el propósito es identificar los atributos relevantes: materias previamente aprobadas que influyen en la aprobación de una materia objetivo. En la siguiente subsección se presentan los resultados obtenidos y se hace la interpretación de los mismos.

4.2. Resultados

Se procedió entonces a identificar los atributos que eran relevantes para dar respuesta al problema de requisitos para cada una de las materias objetivo. El resultado de los métodos de ganancia de información y separación de centroides se presentan en la tabla 2.

La tabla 2 presenta tres secciones: para las colecciones correspondientes a ED, LPL y SO. Cada colección tiene asociadas en tres columnas los atributos de la materia, la ganancia de cada uno de ellos (**GI**), y el valor de separación entre clases (**SC**). Se han marcado los valores destacados en negritas, indicando que son los mejores atributos de acuerdo con el criterio de ganancia o separación de clases. Observamos que, en la mayoría de casos coinciden los atributos con valores más altos en GI con los que separan las clases A y R, excepto en LPL y SO. En LPL, el atributo MD1 tiene un valor GI más bajo que TLENG y SUST, pero separa clases; en tanto TLENG y SUST, de mayor GI que MD1, no separan clases. En SO, el atributo HYC tiene el valor máximo de GI pero

Tabla 2. Ganancia de información y valor de separación de centroides para atributos de ED, LPL y SO.

Atr-ED	GI	SC	Atr-LPL	GI	SC	Atr-SO	GI	SC
TLENG	0.11	-0.0002	TLENG	0.08	-0.0094	TLENG	0.05	-0.0002
TMATE	0.42	0.0041	SUST	0.10	-0.0094	SUST	0.00	-0.0026
PWE	0.12	-0.0013	TMATE	0.03	-0.0083	TMATE	0.04	-0.0035
HYC	0.05	-0.0011	HYC	0.00	-0.0145	HYC	0.39	-0.0028
MD1	0.27	0.0002	PE	0.01	-0.0110	PE	0.00	-0.0034
			ED	0.30	0.0243	MD1	0.01	-0.0017
			MD1	0.04	0.0014	ED	0.21	0.0055
			MD2	0.41	0.0258	ARQ	0.28	0.0095

no separa clases (es negativo). Ambos resultados llaman la atención, pues el resultado obtenido por separación de centroides coincide con los previamente determinados por especialistas de la disciplina.

Con el fin de comprobar la efectividad de los atributos obtenidos por el cálculo de ganancia de información, se utilizaron los atributos en el clasificador de árbol de decisión. Una vez obtenida la ganancia de cada uno de los atributos se probaron diferentes umbrales de mínima ganancia para elegir atributos y realizar la clasificación con este método. Los resultados se exponen en la tabla 3.

Tabla 3. Exactitud de la clasificación con árboles de decisión, variando el umbral de GI para las materias ED, LPL y SO.

Atr-ED	GI	ACC	Atr-LPL	GI	ACC	Atr-SO	GI	ACC
HYC	0,05	0.59	HYC	0	0.60	PE	0	0.55
TLENG	0.11	0.61	PE	0.01	-	SUST	0	-
PWE	0.12	0.62	TMATE	0.03	0.62	MD1	0.009	-
MD1	0.27	0.62	MD1	0.04	-	TMATE	0.04	0.59
TMATE	0.42	0.41	TLENG	0.08	0.65	TLENG	0.05	0.54
			SUST	0.10	0.65	ED	0.21	0.56
			ED	0.30	0.66	ARQ	0.28	0.63
			MD2	0.41	0.55	HYC	0.39	0.65

Utilizando todos los atributos, es decir con $GI > 0$, se tendrá un valor de exactitud menor que el que se obtiene al descartar atributos con menor ganancia. Este procedimiento aumentará la exactitud de la clasificación al tomar atributos con ganancia cada vez mayor. La tabla 3 presenta también tres columnas para cada colección: la primera los atributos ordenados descendientemente de menor a mayor según GI cuyo valor aparece en la segunda columna, y la exactitud de la clasificación eligiendo atributos con valor de ganancia mínima igual a la del atributo correspondiente. Se observa que el anterior procedimiento no es monótono; por ejemplo, en las columnas última y penúltima, correspondientes a los requisitos de la materia SO, puede verse que al tomar atributos de GI mayor

a 0.05 la exactitud desciende de 0.59 a 0.54 y, posteriormente, vuelve a ascender. Este comportamiento de la exactitud, tomando atributos con GI cada vez mayor, significa que no podríamos confiar en un umbral basado en GI. En tanto, para los atributos elegidos por el algoritmo de separación de centroides los resultados de exactitud son: 0.62, para la colección ED; 0.77, para la colección LPL; y 0.71, para la colección SO.

5. Conclusiones

En este trabajo se ha aplicado a una muestra de evaluaciones de estudiantes de una carrera universitaria técnicas de análisis de atributos con la finalidad de conocer la influencia de las materias previamente aprobadas en otra que es de interés. Se conformó para cada materia objetivo una colección de instancias con atributos referentes a las materias aprobadas previamente que pueden ser influyentes en la aprobación de la materia objetivo. Para cada atributo de cada colección referente, se calculó la separación entre sus centroides de la clase de aprobados y reprobados antes y después de eliminar dicho atributo y, así, conocer su importancia en la representación de las instancias. Asimismo, se utilizó el algoritmo de árboles de decisión que permitió contrastar la importancia de los atributos por ganancia de información con el valor de separación de clases que provee cada atributo. Una desventaja del algoritmo para determinar atributos discriminantes es su complejidad ($O(nm)$, con n instancias y m atributos); en este caso fue viable su aplicación por la cantidad de datos y atributos. En contraste con lo anterior se observaron varias ventajas:

- Vemos que en muchos problemas es necesario fijar umbrales para tomar decisiones. El uso de la medida de separación entre centroides no requiere utilizar umbrales. Además, los atributos así obtenidos ayudaron a mejorar la exactitud en la clasificación por árboles de decisión.
- También, con el conjunto de datos utilizado en los experimentos, el comportamiento de separación de centroides es estable: a mayor valor de separación de clases, mejor el atributo o con más efectividad para clasificar.
- Los resultados obtenidos, materias identificadas como influyentes en la materia objetivo, coinciden con las determinadas previamente en forma cualitativa por especialistas en el campo de la computación.

A partir de estos resultados es necesario aplicar los métodos a otras colecciones de datos para conocer su desempeño con más amplitud y precisión, asimismo comparar con otros enfoques.

Referencias

1. Al-Barrak, M.A., Al-Razgan, M.: Predicting students final GPA using decision trees: a case study. *International Journal of Information and Education Technology* 6(7), 528 (2016)

2. Basuki, A.: Personalized learning path of a web-based learning system. *International Journal of Computer Applications* 53(7) (2012)
3. Blum, A., Rivest, R.L.: Training a 3-node neural network is NP-complete. *Advances in neural information processing systems*, pp. 494–501 (1989)
4. Chen, C.M.: Intelligent web-based learning system with personalized learning path guidance. *Computers & Education* 51(2), 787–814 (2008)
5. Domingos, P.: A Few Useful Things to Know about Machine Learning. *Communications of the ACM* 55(10), 78–87 (2012)
6. Fisher, R.A.: The use of multiple measurements in taxonomic problems. *Annals of Eugenics* 7(2), 179–188 (1936)
7. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar), pp. 1157–1182 (2003)
8. Hall, M.A., Smith, L.A.: Practical feature subset selection for machine learning. (1998)
9. Idris, N., Yusof, N., Saad, P.: Adaptive course sequencing for personalization of learning path using neural network. *Int. J. Advance. Soft Comput. Appl* 1(1), 49–61 (2009)
10. Kohavi, R., John, G.H.: Wrappers for feature subset selection. *Artificial intelligence* 97(1-2), 273–324 (1997)
11. Kovacic, Z.: Early prediction of student success: Mining students' enrolment data. (2010)
12. Liu, H., Setiono, R.: Chi2: Feature selection and discretization of numeric attributes. In: *Proceedings of 7th IEEE International Conference on Tools with Artificial Intelligence*, pp. 388–391 (1995)
13. Novakovic, J.: Toward optimal feature selection using ranking methods and classification algorithms. *Yugoslav Journal of Operations Research* 21(1) (2016)
14. Quinlan, J.R.: *C4. 5: programs for machine learning*. Elsevier (2014)
15. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. *Communications of the ACM* 18(11), 613–620 (1975)
16. Sánchez-Sánchez, C., Jiménez-Salazar, H.: An effect of term selection and expansion for classifying short documents. *Journal of Research in Computing Science*, vol. 123, pp. 99–109 (2016)

Reducción de la dimensión de registros de evaluaciones académicas aplicando el algoritmo K-means

Victoria-Elizabeth Padilla-Morales¹, Saturnino Job Morales Escobar¹,
Maricela Quintana López¹, José Martín Flores Albino¹, Oscar Herrera Alcántara^{1,2}

¹Centro Universitario UAEM Valle de México, Atizpán de Zaragoza, México

²Universidad Autónoma Metropolitana, Departamento de Sistemas, Azcapotzalco, México

vpadillam002@alumno.uaemex.mx, {sjmoralese, mquintanal,
jmfloresa, oherreraa}@uaemex.mx, oha@azc.uam.mx

Resumen. En un ambiente educativo existe una gran cantidad de datos que pueden ser analizados y utilizados en el proceso de la toma de decisiones. En la actualidad, debido al tamaño de su dimensión, los datos tienden a ser más complejos que los datos convencionales y requieren una reducción de su dimensión. La Minería de Datos Educativa permite utilizar técnicas de Minería de Datos para analizar información académica con el fin de identificar patrones que no son evidentes. Este artículo presenta resultados obtenidos en una investigación de un caso de estudio sobre el rendimiento académico de alumnos de Ingeniería del Centro Universitario UAEM Valle de México. En el análisis de los datos se utiliza el algoritmo K-means, el software WEKA y R Studio. Se propone utilizar el agrupamiento para reducir la dimensión de las variables académicas en función de los registros de las calificaciones obtenidas durante los últimos períodos cursados, luego se trabajará con una medida promedio para predecir el rendimiento académico de un alumno. Se utiliza R Studio para contrastar los grupos obtenidos por WEKA.

Palabras clave: minería de datos educacional, reducción de la dimensión, agrupamientos, K-means.

Dimension Reduction of Academic Evaluation Records by Applying the K-Means Algorithm

Abstract. In an educational environment there is a huge data quantity, this data can be analyzed, and it can be used in decision making process. Nowadays data tends to be more complex due to the size than conventional data and need dimension reduction. Educational Data Mining lets using Data Mining techniques for analyzing academic information in order to identify patterns that are not evident. This article presents results obtained in a research of a case of study where regard the academic performance of undergraduate students of Engineering of the Centro Universitario UAEM Valle de México. In the data analysis is used the K-means algorithm, WEKA and R Studio. We propose the use of Clustering to reduce the dimension of academic variables based on their grade registers getting during last periods then we work with some average measure of in order to predict the academic performance of a student. It is used R Studio for contrast the Clusters obtained by WEKA.

Keywords: educational data mining, reduce dimension, clustering, K-Means.

1. Introducción

Vivimos en la llamada era de la información, en un mundo donde una gran cantidad de datos se colectan y almacenan diariamente [1]. De hecho, es impresionante la cantidad de datos con los que convive el ser humano día a día, y conforme avanza el tiempo, esta cantidad crece cada vez más y de manera muy rápida, llegando a un crecimiento exponencial. Lo anterior hace prácticamente imposible que los seres humanos procesen estos datos manualmente en búsqueda de información valiosa. Esta es una de las razones por la cual la Minería de Datos DM (por sus siglas en inglés, Data Mining) [2], también llamada descubrimiento de conocimiento en bases de datos KDD (por sus siglas en inglés, Knowledge Discovery in Databases), es un área en auge que ofrece técnicas y herramientas para diversos propósitos y cuya aplicación en ámbitos como el de la mercadotecnia, el aprendizaje automático (machine learning), las grandes bases de datos, el reconocimiento de patrones y la visualización por computadora [3], ha tenido buenos resultados.

Los procesos principales de la DM son: la extracción de información mediante relaciones de similitud entre los datos y la búsqueda de patrones o agrupamiento de datos aparentemente inconexos. La DM ha permitido el análisis de datos complejos de manera automática con el objetivo de entender su comportamiento y de esta manera apoyar en la mejora del proceso de la toma de decisiones. Lo anterior es fundamental en cualquier organización con el fin de obtener resultados que funcionen para optimizar y potenciar sus actividades.

Entre los problemas fundamentales que se abordan con la DM, está el agrupamiento o clustering, el cual se puede realizar con diversas técnicas y algoritmos [2], pero que se basan en el mismo principio “los objetos que están en un grupo son más parecidos (similares) entre ellos, que a objetos pertenecientes a otros grupos”.

En años recientes, la aplicación de la DM se ha incrementado de manera extraordinaria en diversos ámbitos del quehacer humano, se pueden citar como ejemplos: la toma de decisiones [1], la inteligencia de negocios [4], la recuperación de información con motores de búsqueda web [1], la predicción de tiempos de graduación [5] y el pronóstico de evaluaciones en el campo educativo [6], entre otros.

Una de las aplicaciones de la DM es conocida como Minería de Datos Educativa (EDM por sus siglas en inglés, Educational Data Mining), la cual en los últimos años ha sido un campo de amplio estudio y que ha brindado posibilidades de cambio para las instituciones educativas en los diferentes ámbitos que atienden, logrando mejoras en sus procesos.

En la presente investigación, se utiliza el agrupamiento de datos para reducir la dimensión del espacio de representación, lo que permite encontrar elementos representativos de cada grupo para formar una matriz de menor dimensión.

Entre los algoritmos de agrupamiento más conocidos y utilizados, se encuentra el algoritmo K-means. Este algoritmo puede trabajar con una gran cantidad de datos, donde K es el número de grupos que se obtienen. El valor asignado al parámetro k es proporcionado por el usuario y entonces el algoritmo ubica a los objetos en el grupo que le corresponda a través del criterio de su distancia más corta a una medida central [7].

Los datos analizados en este trabajo corresponden a los registros de evaluación que obtuvieron los alumnos en cada una de sus asignaturas durante los 10 semestres cursados a lo largo de su carrera (en un plan de 5 años). Estas evaluaciones son complementadas con datos curriculares de la carrera, mismos que se mencionan más adelante.

A partir de los historiales académicos individuales, los alumnos tendrán registros formados por a lo más 53 evaluaciones. La muestra con la que se hicieron las pruebas se obtuvo de alumnos egresados entre el 2015 y 2017.

En la investigación se utilizó el clustering como una herramienta para la reducción de variables aplicando el algoritmo K-means [7], el software WEKA (Waikato Environment Knowledge Analysis) [2] y el Lenguaje R [8].

La reducción de la dimensión o espacio de representación es importante en el análisis de grupos, ya que, no solo permite hacer manejables los conjuntos de datos de grandes dimensiones y reducir el costo computacional, sino que provee al usuario de una imagen más clara y visual para el análisis de los datos en cuestión [9].

Sin embargo, el reto más grande del clustering es la validación de los agrupamientos logrados. Esto se debe, a que, a diferencia de la clasificación supervisada, no se tiene a *priori* forma de validar los resultados. Por ejemplo, ¿cómo saber si la cantidad de grupos es la correcta? y si ¿los objetos están adecuadamente asignados al grupo? Estas, son consideraciones que hacen complicado evaluar la calidad de cualquier agrupamiento. No obstante, en este sentido, en la literatura se reportan diversos criterios para la validación, encontrado incluso que las medidas utilizadas varían de acuerdo con los índices y lo que éstos consideran para hacer la evaluación de los grupos como de buena calidad [10]. Así, por ejemplo, se tienen índices para evaluar: la separación entre los grupos, la cohesión entre los elementos, como es el caso del índice de homogeneidad biológica (BHI) y la estabilidad biológica [11].

2. Planteamiento del problema

En el caso de estudio del Centro Universitario UAEM Valle de México, la información de los alumnos de nivel superior es almacenada en una base de datos. La hipótesis es la siguiente: Al utilizar algoritmos de DM, los datos pueden ser usados para entender el comportamiento de los alumnos, apoyar a los profesores, evaluar y mejorar la enseñanza presencial, entre otros. Por otra parte, la estructura curricular de la carrera significa información adicional, así como los datos de profesores y de otros procesos de enseñanza-aprendizaje, los cuales también pueden analizarse utilizando la DM para obtener relaciones significativas y no conocidas de manera explícita, de tal manera que se generen conclusiones que apoyen a la toma de decisiones en los sistemas escolares.

Otro ejemplo de la aplicación de la DM, en este ámbito, sería identificar las causas o variables que influyen en el desempeño académico de los alumnos, y de esta manera hacer una predicción de éste. Con base en los resultados, se podrían tomar acciones útiles y constructivas que ayuden a las autoridades a mejorar sus procesos en la toma de decisiones, incrementar el desempeño académico de los alumnos, disminuir la deserción escolar, reducir la reprobación, entender mejor el comportamiento de los alumnos, brindar apoyo a profesores y mejorar los procesos de enseñanza-aprendizaje.

Dada la importancia de este tema, existen diversos estudios alrededor de mundo, que están en busca de conclusiones y desarrollando investigaciones que aporten en el análisis de datos del ámbito educativo [12]. Entre estos proyectos, algunos indagan sobre las causas que dan pauta al resultado del alumno, mismo que refleja su desempeño mediante la calificación. Una conclusión a la que han llegado los autores es que dichas causas son multifactoriales, por ejemplo, en [13] se consideran factores académicos, personales y económicos de los alumnos. Por otro lado, en [14] el estrato social, situación económica, el entorno de los alumnos, así como el nivel académico de los padres se contemplan como variables que influyen en ese resultado.

En México, la educación es un tema que también ha sido objeto de mucho estudio con el fin de encontrar o desarrollar estrategias educativas que involucren métodos y herramientas innovadoras que mejoren el proceso de enseñanza-aprendizaje y que tal mejora, se vea reflejada en el desempeño académico de los alumnos en los diferentes niveles educativos, desde nivel básico (preescolar, primaria y secundaria) hasta nivel medio superior en donde se han implementado

reformas Integrales de educación para lograr un desempeño deseable en las evaluaciones tipo PISA¹

En ese mismo sentido, en las instituciones de educación superior también se hacen grandes inversiones en aspectos como infraestructura, capacitación y actualización docente e investigación educativa, con el objetivo de mejorar los procesos involucrados en la educación y de esta manera, mejorar los resultados de sus egresados.

Los datos con los que se trabajaron se obtuvieron del plan de estudios de la carrera de Ingeniería en Sistemas y Comunicaciones del Centro Universitario UAEM Valle de México, así como de las trayectorias académicas de alumnos egresados entre el 2015 y 2017.

Predecir el desempeño de los alumnos permite identificar a aquellos que necesitan atención especial para reducir la tasa de fracaso y tomar acciones apropiadas para las siguientes pruebas semestrales.

De igual manera, el presente trabajo de investigación puede ayudar a la detección de alumnos en riesgo, para que las autoridades correspondientes asignen a un profesor-tutor para evitar el rezago y posteriormente la deserción del alumno.

Como se ha visto, la aplicación de la EDM ha permitido encontrar aspectos que impactan en el desempeño escolar de los alumnos.

Los supuestos que motivan la realización de este trabajo de investigación son:

- a. Las evaluaciones obtenidas por un alumno en las materias cursadas dan pauta para predecir su desempeño en las materias del semestre inmediato siguiente
- b. Si se detectan a tiempo alumnos en riesgo, basándose únicamente en las evaluaciones, se podrían aplicar estrategias para evitar el rezago educativo y la deserción académica.

Es evidente que, a mayor cantidad de dimensiones en una matriz, mayormente complejo es el análisis de los datos y esto puede llevar a lo que se conoce como la maldición de la dimensionalidad [9] durante el proceso de clustering. Así, uno de los grandes beneficios de la reducción de dimensión es que brinda mayor facilidad en el manejo de los datos.

El problema abordado en este trabajo de investigación es cómo reducir la dimensión de los registros de las evaluaciones de los alumnos (objetos) ya que las filas que representan a cada alumno tienen dimensiones 6, 11, 19, 26, 31, 36, 41, 46, 52 y 53, dependiendo del grado de avance en la carrera. Estos números, corresponden al total de evaluaciones obtenidas al terminar 1º, 2º, hasta el décimo semestre. Como se puede apreciar, entre mayor grado de avance, mayor cantidad de variables, así, reducir la cantidad sin pérdida de información, claramente tiene muchos beneficios.

Para almacenar las evaluaciones, se definió una matriz de 98 filas (registros de alumnos) por 53 columnas(materias).

3. Trabajo relacionado

En [12] se hace la revisión del estado del arte en la EDM y considera más de 240 trabajos de investigación para describir 222 enfoques diferentes y 18 herramientas utilizadas en EDM.

Entre los trabajos analizados se encuentran propuestas de soluciones para mejorar el desempeño académico de los alumnos. Por ejemplo: la identificación de alumnos con desempeño destacado para otorgarles becas o identificar alumnos con bajo desempeño para proporcionarles recursos de tutoría de manera oportuna [15].

Así, la EDM busca, principalmente modelar el comportamiento de los alumnos y encontrar posibles soluciones a los problemas detectados, por ejemplo, en [16] analizan los datos de alumnos universitarios de la UAEM CU VM para determinar su perfil académico mediante el uso de un ensamble de clasificadores. Las clases que consideraron son excelente, bueno y regular, y

¹ <https://www.inee.edu.mx/index.php/evaluaciones-internacionales/reactivos-liberados-pisa-2016>

cada alumno es clasificado con base en su promedio general, actividades de estudio, formas de aprendizaje y hábitos de estudio. Los resultados obtenidos lograron un 80.4 % de acierto en la clasificación.

En el caso de [17] en la Universidad Autónoma Metropolitana (UAM) trabajaron con datos de alumnos universitarios de primer ingreso, asociados a un curso de nivelación de matemáticas y con base a éstos se hace la predicción de dos cursos posteriores. Lo anterior con el fin de identificar el impacto de la calificación obtenida en el curso de nivelación de matemáticas, sobre la calificación de los cursos de Complementos de Matemáticas e introducción al Cálculo y de esta manera, determinar si el curso ayuda a los alumnos a tener un mejor desempeño; en el trabajo se utiliza la técnica de reglas de asociación. La conclusión a la que llegaron es que el curso de nivelación no está ayudando a los alumnos a acreditar los dos cursos posteriores.

En [18], encuentran relaciones utilizando reglas de asociación entre los factores de preadmisión de los alumnos y el primer año dentro de la universidad, buscaban identificar si la manera en la que son elegidos los alumnos permite detectar a aquellos que tendrán un buen desempeño académico, tomando como variables: la escuela de procedencia, el promedio general de la preparatoria y la edad de ingreso del alumno; como resultados encuentran que la escuela de procedencia no es relevante, sin embargo, la edad de ingreso y el promedio general de la preparatoria sí lo son.

En [19] buscan relaciones entre las actividades diarias de alumnos universitarios y su desempeño académico universitario mediante el uso de sensores en smartwatches que permiten el monitoreo de sus actividades. Los resultados del reconocimiento de actividades con Random Forest fueron del 86.9 % de acierto.

4. Propuesta metodológica

Para el desarrollo del presente trabajo, se aplicó una metodología que tiene como base la metodología KDD, y que consta de 5 etapas para la reducción de la dimensión de los registros de evaluaciones de los alumnos, como se puede apreciar en la figura 1.

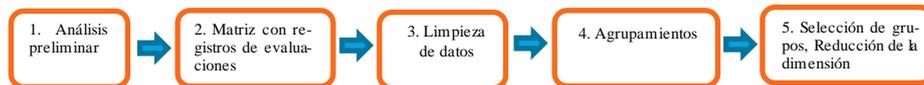


Fig. 1. Metodología aplicada en el presente trabajo.

En la etapa 1, se realizó un análisis de la currícula de la carrera a modelar y de los historiales académicos para determinar las variables que se van a utilizar: es decir, total de materias, materias por semestre, dependencias entre materias, el tipo de materia (obligatoria u optativa), los núcleos de formación, oportunidades para acreditar una materia.

Como resultado del análisis, en el caso de estudio de ISC, se determinó tomar para la descripción de los objetos, las evaluaciones obtenidas por el alumno en examen ordinario la primera vez que cursa la materia, puesto que tiene derecho a 6 oportunidades para acreditarla, las primeras 3 (Ordinario, Extraordinario y Título de Suficiencia) son en periodo regular o primera vez que cursa la materia y las mismas tres oportunidades tendrá, en caso de no acreditarla y volver a cursarla.

En la etapa 2, Se mapearon las materias y las evaluaciones de cada alumno en una matriz en Excel para poder procesarlas. Las matrices obtenidas contienen registros de 98 alumnos con diferentes dimensiones desde 6 hasta 53 atributos, los cuales corresponden a las evaluaciones finales de las materias. En la Fig. 2, se muestra la imagen de una parte de la matriz generada a partir de los historiales académicos individuales de los alumnos considerando las 53 materias.

No.	CL	C2	AL	AGA	PE	E	ESD	Q	A	TCM	CDN	CV	ED	IC	C	SEN	R	IO	LM	MD	MN	AED	BD	CE	EA	FBO	FP	IS	LEN	LFA	POOSI	SD	SCR	TSS	MR	AC	DP	FEP	PCD	R	SD	ACCIS	RP	A	
A1	46	74	65	38	60	80	30	29	50	100	60	71	62	45	60	39	66	50	50	77	68	31	77	60	67	60	80	85	55	65	60	50	50	85	50	72	74	68	50	87	62	70	78	50	96
A2	61	75	86	65	80	75	60	69	76	90	86	63	83	89	90	86	85	91	70	84	66	85	86	70	90	77	98	95	72	77	88	83	70	88	81	75	75	100	65	89	80	68	80	96	
A3	66	75	70	78	69	49	77	61	67	70	17	87	NP	80	10	60	64	50	88	16	80	73	80	40	60	71	87	75	90	80	68	50	80	50	67	65	52	50	60	70	50	70	80	80	97
A4	60	75	89	75	90	85	60	80	83	83	47	90	73	90	86	83	83	63	90	68	76	82	93	80	84	88	94	68	74	70	81	90	83	87	79	100	86	90	74	75	84	74	84		
A5	90	80	82	85	69	48	90	84	69	82	60	84	90	78	70	75	78	50	75	68	100	75	66	60	72	71	90	82	90	75	72	88	70	74	70	76	89	75	75	78	86	65	95		
A6	50	50	70	67	94	60	60	72	62	60	23	75	NP	100	60	77	60	60	67	75	50	NP	63	74	60	70	NP	60	100	72	80	80	80	70	74	50	68	NP	80	60	68	80	50	94	
A7	71	75	74	62	90	80	54	70	100	60	39	79	75	100	84	61	83	88	90	74	79	86	80	93	84	84	89	67	83	87	80	81	86	62	74	80	89	74	95	80	80	79	80	95	
A8	95	88	80	63	80	65	50	70	80	90	82	60	81	86	90	87	82	55	60	86	64	90	85	65	79	93	86	91	85	70	76	60	89	61	86	75	50	65	87	60	81	75	85		
A9	80	74	75	67	80	90	65	60	90	28	41	70	57	90	92	57	50	55	84	68	53	76	30	87	40	71	79	72	42	66	50	80	86	50	75	64	85	65	79	57	83	60	35	80	
A10	90	100	90	88	77	80	88	94	80	80	80	91	100	74	80	90	81	75	88	88	100	100	94	88	85	95	100	95	100	97	80	96	80	80	91	85	90	85	86	85	100	1			
A11	51	62	73	83	84	61	50	45	73	80	13	64	NP	72	10	80	83	80	88	44	90	69	66	62	78	74	80	73	88	83	68	51	80	72	50	90	72	50	75	79	67	80	50	71	89
A12	90	84	80	60	61	64	86	71	80	37	83	73	96	10	89	64	50	88	72	90	80	79	60	75	70	77	68	95	72	65	81	80	87	62	79	77	99	60	85	50	85	90	87	82	
A13	90	70	80	75	62	63	91	86	76	80	50	83	79	94	10	91	73	70	88	75	100	82	64	42	93	72	82	78	86	64	74	93	70	65	72	81	100	83	85	72	90	90	85	96	
A14	90	75	70	45	80	80	60	44	50	95	20	45	87	75	80	81	83	50	50	52	8	61	76	60	78	45	80	81	65	NP	65	48	50	89	50	71	74	89	60	91	80	80	50	51	
A15	83	88	70	44	80	70	60	82	50	95	80	43	62	88	60	50	60	50	29	44	80	78	62	83	60	80	71	50	63	60	70	71	77	50	90	88	89	60	83	65	60	80	86		
A16	95	75	90	64	100	78	65	63	60	95	83	49	81	72	100	91	58	83	84	28	78	87	81	80	48	100	97	84	51	84	50	60	88	50	75	86	52	80	88	69	60	82	77	85	
A17	80	80	93	73	100	100	80	69	80	95	92	93	90	77	100	93	87	82	90	75	76	84	100	98	78	80	100	65	82	75	85	88	98	81	86	100	87	86	81	86	100	87	87	100	
A18	25	68	80	68	65	52	60	86	69	76	62	83	NP	90	60	78	67	60	75	65	89	81	60	60	70	80	78	80	78	80	78	60	78	60	74	50	75	66	100	60	70	62	80	60	77
A19	65	81	75	45	70	65	80	40	75	95	81	30	72	76	80	45	67	80	34	79	13	60	87	80	90	25	78	95	57	55	66	50	72	50	69	87	93	60	83	30	85	73	62	95	
A20	80	85	90	78	79	72	90	66	77	60	90	64	80	87	84	81	50	75	90	80	70	79	73	90	78	76	57	89	71	70	79	70	89	63	80	73	99	75	85	50	90	80	78		
A21	62	75	92	72	100	90	80	69	80	95	92	93	90	77	100	93	87	82	90	75	76	84	100	98	78	80	100	65	82	75	85	88	91	80	77	100	85	91	80	100	81	90	98		
A22	78	65	81	35	39	62	47	71	NP	65	65	38	75	77	80	60	71	12	38	60	52	60	60	50	60	60	60	67	60	65	50	45	31	75	66	80	61	50	75	80	72	93	73	57	
A23	60	72	75	41	80	80	41	50	100	16	78	71	70	90	80	67	78	88	81	64	61	79	85	80	60	81	71	69	67	64	60	70	86	64	80	70	100	83	90	54	91	78	78	100	
A24	81	60	90	61	100	80	60	48	60	95	24	80	95	78	90	87	75	81	70	84	20	82	100	93	77	35	94	89	80	89	100	83	80	85	68	71	82	100	80	88	83	85	90	90	77
A25	60	88	58	100	80	60	52	80	36	80	83	50	60	65	88	83	76	83	24	70	77	90	72	60	60	65	90	61	63	75	64	60	82	50	72	51	80	50	65	62	75	68	62	85	
A26	67	64	69	48	60	63	85	34	80	100	60	25	78	95	70	80	75	60	38	70	83	NP	42	60	78	40	90	80	89	61	76	57	NP	74	62	66	87	89	64	68	80	35	60	67	NP
A27	90	90	95	71	70	75	88	100	66	72	62	64	75	83	72	56	75	70	100	78	60	82	80	74	87	90	90	60	68	75	60	82	50	80	87	78	75	85	48	70	90	64	96		
A28	65	73	60	66	43	60	60	80	33	50	27	41	16	100	50	80	47	31	33	65	70	83	60	62	60	45	85	60	89	65	65	90	96	71	70	50	70	78	80	10	92	59	70	83	
A29	70	80	100	49	72	46	84	91	72	85	50	52	70	75	10	78	55	60	80	65	78	65	60	80	93	87	90	93	82	76	90	80	80	60	73	96	90	84	85	94	85	96	80	89	
A30	75	80	89	81	80	NP	70	81	50	95	16	49	79	82	90	77	92	50	50	87	77	81	82	60	100	66	88	73	67	70	64	71	84	61	85	83	89	50	88	50	92	76	80	80	
A31	80	72	85	48	90	75	60	54	68	80	21	41	80	57	90	91	64	50	50	87	10	78	100	70	95	60	85	90	85	80	68	60	80	85	84	78	81	100	80	96	60	74	82	72	
A32	90	100	80	81	71	NP	81	91	65	95	30	67	73	88	10	80	64	99	86	63	100	84	68	60	93	79	76	85	86	71	78	80	80	87	73	75	72	87	75	85	75	95	85	80	93
A33	90	90	88	60	96	70	72	60	88	71	83	75	84	83	84	84	89	88	100	94	72	85	75																						

Tabla 1. Las 59 materias del plan de estudios de la carrera de Ing. en Sistemas y Comunicaciones.

MATERIA	ID	MATERIA	ID	MATERIA	ID
Inglés C1	C1	Métodos Numéricos	MN	Redes	R
Inglés C2	C2	Algoritmos y Estructura de Datos	AED	Sistemas Digitales	SD
Álgebra Lineal	AL	Base de Datos	BD	Administración de Centros de Cómputo	ACC
Álgebra y Geometría Analítica	AGA	Circuitos Eléctricos	CE	Interconexión y Seguridad de Redes	ISR
Probabilidad y Estadística para Ingenieros	PEI	Electrónica Analógica	EA	Residencia Profesional	RP
Electromagnetismo	E	Fundamentos de Base de Datos	FBD	Administración de Base de Datos	ABD
Estática y Dinámica	ESD	Fundamentos de Programación	FP	Calidad de Software	CS
Química	Q	Ingeniería de Software	IS	Programación Avanzada	PA
Administración	A	Lenguajes de Bajo Nivel	LBN	Inteligencia Artificial	IA
Técnicas de Comunicación	TC	Lenguajes Formales y Automátas	LFA	Seminario de Titulación	ST
Cálculo Diferencial e Integral	CDI	Programación Orientada a Objetos	POO	Sistemas de Tiempo Real	STR
Cálculo Vectorial	CV	Sistemas de Información	SI	Comunicación Vía Microondas y Satelital	CVM
Ecuaciones Diferenciales	ED	Sistemas Operativos	SO	Teoría del Control	TC
Introducción a la Computación	IC	Sistemas Operativos para Red	SOR	Planeación Estratégica	PE
Contabilidad	C	Temas Selectos de Sistemas	TSS	Auditoría y Seguridad Informática	ASI
Ecología Ética y Normatividad	EEN	Metodología de la Investigación	MI	Sistemas de Instrumentación y Control	SIC
Intrducción a la Ingeniería	II	Arquitectura de Computadoras	AC	Sistemas Electrónicos de Comunicación	SEC
Ivestigación de Operaciones	IO	Desarrollo de Proyectos	DP	Transmisión y Comunicación de Datos	TCD
Lógica Matemática	LM	Formulación y Evaluación de Proyectos	FEP	Sistemas Expertos	SE
Matemáticas Discretas	MD	Protocolos de Comunicación de Datos	PCD		

5. Experimentos

A partir de la muestra obtenida en la etapa 2, se realizaron experimentos con varias corridas variando parámetros del algoritmo K-means (K y la semilla), los experimentos, se describen a continuación:

Experimento 1. Para las corridas en WEKA, se consideraron diferentes valores para k y se variaron las semillas en cada uno de los semestres; obteniéndose los resultados que se muestran en la Tabla 2.

Notar que el primer semestre consta de 6 materias, el segundo de 11 y el sexto de 36 materias:

Tabla 2. Resultado de corrida de K-means en WEKA para primer, segundo y sexto semestre.

Sem	K	Seed	Distribución	Conformación de grupos
1	2	10	3, 3	(A, AGA, ESD) (TCM, IC, II)
1	2	5	5, 1	(AGA, ESD, IC, II, TCM) (A)
1	2	3	3,3	(AGA, TCM, IC) (A, ESD, II)
2	3	10	4, 2, 5	(A, IC, II, CDI) (AGA, ESD) (TCM, AL, FP, AC, Q)
2	3	5	5, 1, 5	(AGA, ESD, II, CDI, Q) (A) (TCM, IC, AL, FP, AC)
2	3	3	6, 3, 2	(TCM, IC, II, AL, FP, AC) (AGA, ESD, CDI) (A, Q)
6	5	10	9, 14, 3, 6, 4	(IC,AL,FP,AC,EEN,AED,LBN,C2,POO) (TCM,II,ED,C,MD,PEI,E,BD,CE,SOR,IA) (A,CDI,LM,IO) (Q,CI,MN,FBD,SO,LBN,SI) (AGA,ESD,CV,R,ACC)

En la Tabla 2: la primera columna corresponde al semestre en que se aplica el clustering, la columna K corresponde al número de grupos a formar, la columna Seed corresponde al valor asignado a la semilla, la columna Distribución corresponde al número de materias que forman cada grupo según la K y la columna Conformación de grupos corresponde a las materias (identificadas con el id mostrado en la Tabla 1) que conforman los grupos de acuerdo con la distribución.

Como puede observarse en la Tabla 2, el cambio en el valor de la semilla modifica la distribución de las materias en los grupos y aparentemente se ven más equilibrados. En el caso de la primera fila, trabajando con primer semestre, el valor para K = 2, a la semilla (seed) se asigna valor 10, después de aplicar el algoritmo K-means, se obtiene una distribución de 3 elementos para el grupo 1 y 3 elementos para el grupo 2, se puede observar en la columna Conformación de grupos, que el primero está formado por las materias (A, AGA, ESD) y que el segundo por las materias (TCM, IC, II)

Así en la segunda fila, el valor de K no cambia, se varía la semilla a 5 y la distribución cambia a 5 elementos para el primer grupo y un elemento para el segundo, la columna conformación nos dice qué elementos quedaron en los grupos.

En la literatura se encontró que el algoritmo K-means en WEKA [20], también trabaja con valores nominales (SD y NP) y para este caso computa la moda para tratar los valores nominales como valores numéricos. Para el caso particular de este trabajo no funcionó tomar la moda; por lo tanto, fue importante realizar la limpieza de los datos (etapa 2 de la propuesta metodológica).

Experimento 2. En el caso de la validación de grupos se encontró que es muy difícil poder llegar a validar el valor óptimo de K [11,21,22]. Para evaluar el resultado de los algoritmos de agrupamiento se utilizan índices de validación que miden distintos aspectos de las características de los grupos logrados [23]. En el caso particular de este trabajo de investigación, se trabajó con un paquete de validación de clustering y a construido del lenguaje R llamado clValid [11]. Este contiene funciones para validar los resultados del análisis de clustering. El paquete ofrece 3 tipos de medidas de validación de clustering “internas”, “de estabilidad” y “biológicas”. En el presente trabajo se utilizaron las medidas internas y el algoritmo K-means.

La ejecución en el lenguaje R para la validación de los agrupamientos logrados arrojó como resultados los datos de la Tabla 3.

Reducción de la dimensión de registros de evaluaciones académicas aplicando el algoritmo K-means

Como podemos observar, en la Tabla 3 se muestran los valores óptimos para K en cada uno de los semestres, desde el primero hasta el sexto.

La columna semestre corresponde al semestre en el que se va a hacer el agrupamiento (va desde el 1 hasta el 10), la columna número de materias muestra la dimensión de la matriz y por lo tanto es la cantidad de materias a agrupar, la columna Connectivity muestra el valor óptimo para K según el índice de conectividad, la columna Dunn muestra el valor óptimo para K según el índice Dunn, la columna Silhouette muestra el valor óptimo para k según el índice de ancho de la silueta, la columna valor óptimo K muestra el valor óptimo para K después de elegir; en [24] proponen un método de ensamble para la validación de clustering, basado en las medidas internas trabajadas en la presente investigación, tomando en consideración la votación de mayoría simple.

Entonces teniendo la matriz de dimensión 36 (Ver Figura 3) quedará reducida a dimensión 5 (Ver figura 4)

Tabla 3. Número de materias por semestre y valor óptimo para k.

Semestre	No de materias	Connectivity	Dunn	Silhouette	Valor óptimo K
Primero	6	2	5	2	2
Segundo	11	2	10	2	2
Tercero	19	2	18	6	6
Cuarto	26	2	25	2	2
Quinto	31	2	30	4	4
Sexto	36	2	35	5	5

ETQ	AGA	ESD	A	TCM	IC	II	AL	CDI	FP	AC	Q	C1	ED	C	EEN	MD	MN	AED	LBN	C2	PEI	E	CV	LM	FBD	SO	BD	CE	IO	POO	R	LFA	SI	SOR	ACC	IA
A1	38	50	0	100	45	66	65	60	80	74	29	46	62	60	39	77	68	31	55	74	60	80	71	0	60	0	77	60	0	60	62	65	50	85	78	75
A2	65	60	76	90	69	85	86	86	98	75	69	61	83	90	86	84	66	85	72	75	80	75	63	70	77	70	86	70	91	88	80	77	83	88	68	100
A3	78	77	67	70	80	64	70	17	87	52	61	66	0	10	60	16	80	73	90	75	69	49	87	88	71	80	60	40	0	68	0	60	0	0	80	0
A4	75	80	80	85	73	83	89	83	88	79	60	60	90	90	86	90	68	76	74	75	90	85	47	63	84	83	82	93	83	74	74	68	70	90	74	85
A5	85	90	69	82	78	78	82	60	90	76	84	90	90	70	75	68	100	75	90	80	69	48	84	75	71	70	66	60	50	72	78	75	88	75	86	0
A6	67	60	62	60	100	60	70	23	0	68	72	0	0	60	77	75	50	0	100	0	94	60	75	67	70	80	63	74	60	80	60	72	80	70	80	70
A7	62	80	70	100	75	61	74	60	84	80	54	71	79	100	84	90	74	79	67	75	90	80	39	88	84	83	86	80	63	67	80	63	80	86	79	85
A8	63	50	80	90	86	82	80	82	86	75	70	95	81	90	87	86	64	90	85	88	80	65	60	60	93	70	85	65	55	73	60	70	66	89	81	90
A9	67	65	60	90	57	57	75	28	71	64	45	80	70	90	92	84	68	53	72	74	80	90	41	55	40	60	76	30	50	66	57	42	0	86	60	75
A10	88	88	80	80	74	81	90	80	100	80	94	90	100	80	90	88	100	100	100	100	77	80	93	88	95	80	94	88	75	80	90	97	96	82	86	75
A11	63	50	73	80	72	83	73	13	80	72	45	51	0	10	80	44	90	69	88	62	84	61	64	88	74	80	66	62	60	68	67	63	51	72	0	75
A12	60	64	71	80	98	64	80	37	77	86	90	73	10	89	72	90	80	95	84	60	61	83	88	70	80	79	60	0	65	0	72	61	87	90	80	
A13	75	91	76	80	94	73	80	50	82	81	86	90	79	10	91	75	100	82	86	70	62	63	83	88	72	70	64	42	70	74	72	64	93	75	90	62
A14	45	60	50	95	75	83	70	20	80	74	44	50	87	80	81	92	8	61	65	75	80	80	45	0	45	0	76	60	50	65	60	0	48	69	0	85
A15	44	60	0	95	68	60	70	80	80	68	32	83	62	60	0	29	44	30	50	88	80	70	43	0	60	71	78	62	0	60	65	63	70	77	60	85
A16	64	65	60	95	72	58	90	83	100	86	83	95	81	100	91	84	28	78	84	75	100	78	49	63	48	60	87	81	63	84	69	51	0	88	82	85
A17	73	90	60	95	83	75	93	95	100	86	85	80	100	88	90	79	100	81	80	100	100	94	100	80	88	100	100	100	100	81	100	83	86	87	80	
A18	68	60	69	76	90	67	80	62	80	66	86	25	0	60	78	65	89	81	80	68	65	52	83	75	70	70	60	60	60	68	62	79	90	74	80	60
A19	45	60	75	95	76	67	75	81	78	87	40	65	72	80	45	79	13	60	57	81	70	65	30	34	25	0	87	80	60	66	50	55	0	72	73	85
A20	78	90	66	77	80	81	90	60	76	73	90	64	87	84	70	90	70	89	85	79	72	90	75	78	70	79	73	0	70	0	71	79	89	90	75	
A21	72	80	80	95	77	87	92	92	80	77	69	62	90	100	93	90	75	76	65	75	100	90	93	76	78	88	84	100	82	75	80	82	85	89	81	100
A22	35	47	0	65	77	71	61	65	60	61	71	78	75	80	60	60	52	60	80	65	29	62	38	38	60	31	60	50	12	50	72	65	45	75	72	70
A23	41	60	50	100	70	67	75	16	81	70	41	60	71	90	80	81	64	61	69	72	80	80	76	88	60	70	79	85	78	64	54	67	60	86	78	85
A24	61	60	60	95	78	75	90	24	94	82	48	81	95	90	87	84	20	82	80	40	100	80	60	70	25	80	100	93	81	100	83	89	83	85	90	90
A25	59	60	80	90	50	88	88	36	65	51	52	60	83	80	65	83	24	70	61	80	100	80	76	60	60	77	90	83	75	62	63	64	82	68	85	
A26	48	65	80	100	95	75	69	60	90	87	34	67	70	80	70	80	63	0	89	84	60	63	25	38	40	0	42	60	60	76	60	61	57	74	60	87
A27	71	88	66	72	63	72	95	62	87	87	100	90	75	72	90	70	100	78	90	90	70	75	64	75	74	60	60	82	56	68	48	60	75	82	90	64
A28	66	60	33	50	100	47	60	27	85	50	80	65	16	50	80	65	70	63	89	73	43	60	41	33	45	90	60	62	31	65	10	65	50	76	59	60
A29	49	34	72	85	75	55	100	50	97	80	91	70	76	10	78	70	65	79	93	80	72	46	52	50	73	80	75	45	60	76	74	82	98	86	96	88
A30	81	70	0	95	82	92	89	16	88	83	81	75	79	90	77	87	77	81	73	80	0	49	0	66	71	82	60	0	70	0	67	64	84	76	85	
A31	48	60	68	80	57	64	85	21	85	81	54	80	80	90	91	87	10	78	85	72	90	75	41	50	60	80	100	70	50	68	60	80	60	85	74	70
A32	81	81	65	95	88	64	80	30	76	72	91	90	73	10	80	63	100	84	86	100	71	0	67	86	79	80	68	60	99	78	75	71	80	87	85	85
A33	86	96	70	72	83	83	90	60	96	80	95	90	71	75	84	88	100	84	94	90	88	60	88	69	79	80	72	85	84	70	74	72	98	92	95	65
A34	81	69	66	92	68	54	75	9	70	58	60	76	0	11	75	48	83	75	87	80	70	54	60	88	80	80	50	38	0	73	0	62	0	0	80	0
A35	45	20	64	79	100	70	60	14	0	50	72	100	72	56	77	90	60	0	99	88	91	0	60	70	80	75	0	0	0	0	0	0	0	0	0	0
A36	74	80	80	95	78	89	86	88	92	75	73	75	91	80	82	76	75	73	63	85	100	85	87	80	60	70	87	90	85	85	61	63	66	84	96	85
A37	36	70	70	95	77	85	88	80	93	76	71	60	90	100	95	90	70	91	66	70	100	78	49	80	90	86	91	90	97	96	80	78	82	86	75	100

Fig. 3. Matriz de sexto semestre dimensión 36.

6. Resultados

Al llevar a cabo la clusterización y la validación de los clústeres, se esperaba que los agrupamientos logrados tuvieran materias pertenecientes a la misma área de conocimiento, de las 4 que se tienen en la estructura curricular de la carrera elegida, las cuales son Ciencias Básicas y Matemáticas (CBM), Ciencias Sociales y Humanidades (CSH), Ciencias de la Ingeniería (CI) e Ingeniería Aplicada (IA); o por núcleo de formación, sin embargo, lo anterior no ocurrió, en este caso, los agrupamientos que se formaron dependían de otros factores tales como el profesor asignado a la materia y el turno en el que se oferta.

En la tabla 2, se puede ver que la última columna la reducción de dimensión para la matriz de sexto semestre es de dimensión 36 para convertirse a una matriz de dimensión 5, quedando de esta manera (Fig. 4).

ETQ	CDI	Q	AGA	SOR	A
A1	45.00	41.75	57.17	85.00	61.39
A2	92.33	74.00	64.50	88.00	82.39
A3	5.67	55.00	71.33	0.00	58.89
A4	83.67	71.25	71.50	90.00	82.72
A5	36.67	80.13	82.50	75.00	75.44
A6	51.00	68.88	59.33	70.00	57.17
A7	69.33	75.75	68.50	86.00	80.06
A8	75.67	69.13	69.00	89.00	82.72
A9	51.00	45.88	57.17	86.00	73.39
A10	76.67	92.50	88.83	82.00	87.44
A11	49.33	69.75	48.33	72.00	64.72
A12	39.00	68.38	74.50	87.00	73.06
A13	60.67	80.63	78.50	75.00	73.44
A14	51.67	25.63	43.33	69.00	76.06
A15	55.00	50.63	58.67	77.00	58.22
A16	77.00	50.25	72.67	88.00	83.50
A17	91.67	87.00	87.33	86.00	90.61

Fig. 4. Matriz dimensión 5 para sexto semestre.

En la Fig. 4, se observa un fragmento de la matriz de dimensión 5 para sexto semestre, que en un principio era de dimensión 36. Como se puede ver se conservan los centroides de cada grupo y es con los que se procederá a trabajar posteriormente.

7. Conclusiones y trabajo futuro

Lo que se puede concluir de los resultados mostrados en la sección anterior, es que por medio de la aplicación del algoritmo K-means se puede reducir la dimensión de los registros de las evaluaciones de los alumnos. Esto se logra tomando los centroides del mejor agrupamiento, lo que permite trabajar ahora con un representante de cada grupo de materias.

Los resultados obtenidos del análisis mostraron que los agrupamientos dependen de las relaciones entre las evaluaciones, sin embargo, hay factores como los horarios en los que se toma la materia y el profesor asignado que influyen en el resultado.

El modelo presentado en este trabajo, puede ser replicado en otras carreras.

Por otro lado, a pesar de que WEKA es una herramienta altamente utilizada para este propósito, lamentablemente no cuenta con un módulo de validación de clustering; motivo por el cual se recurrió a utilizar R Studio.

Como trabajo futuro se hará la predicción del desempeño académico de los alumnos de la carrera de Ingeniería en Sistemas y Comunicaciones del Centro Universitario UAEM Valle de México, mediante técnicas de clasificación. Asimismo, se pueden incluir otras variables en el análisis, como el tipo de examen con el que se aprobó la materia y número de oportunidad en la

que fue acreditada. También se puede considerar como trabajo futuro el utilizar otros algoritmos de agrupamiento y otras herramientas diferentes.

Referencias

1. Han, J., Kamber, M., Pei, J.: Data Mining: Concepts and Techniques. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. (2011)
2. Frank, E., Hall, M.A., Witten, I.H.: The WEKA Workbench. Online Appendix for Data Mining: Practical Machine Learning Tools and Techniques, Fourth Edition, Morgan Kaufmann (2016)
3. Huapaya, C.R., Lizarralde, F.A., Arona, G.R., Massa, S.M.: Minería de Datos Educacional en Ambientes Virtuales de aprendizaje. In: Workshop de Investigadores en Ciencia de la Computacion (WICC), pp. 996–1000 (2012)
4. Bhawana, D., Hardik, J., Vedant, S., Gandla, R.: CRM Application For Analyzing the Sales Using Data Mining. International Journal of Engineering Science and Computing, pp. 16138–16139 (2018)
5. Vassilis, T., Emmanuel, I.E.L.P., Nikos, K., Panagiotis, P.: Prediction of students' graduation time using a two-level classification algorithm. In: Conference: IEEE 1st International Conference on Technology and Innovation in Learning, Teaching and Education (ITHET) (2018)
6. Bhardwaj, B.K., Pal, S.: Data Mining: A prediction for performance improvement using classification. CoRR, vol. abs/1201.3418 (2012)
7. Hartigan, J.A., Clustering algorithms (Wiley series in probability and mathematical statistics) (1975)
8. R. D. C. Team: R: A language and environment for statistical computing. Vienna, Austria: R Foundation for Statistical Computing (2008)
9. Chris, D., Tao, L.: Adaptive Dimension Reduction Using Discriminant Analysis and K-means. In: International Conference on Machine Learning, Corvallis (2007)
10. Alsabti, K., Ranka, S., Singh, V.: An efficient k-means clustering algorithm. Electrical Engineering and Computer Science (1997)
11. Guy, B., Vasyi, P., Susmita, D., Somnath, D.: clValid: An R Package for Cluster Validation. Journal of Statistical Software, vol. 25, pp. 1–22 (2008)
12. Peña-Ayala, A.: Educational data mining: A survey and a data mining-based analysis of recent works. Expert Systems with Applications, pp. 1432–1462 (2014)
13. Ruby, J., David, K.: Analysis of Influencing Factors in Predicting. International Journal of Innovative Research in Computer and Communication Engineering, pp. 1085–1092 (2015)
14. Organista, J., McAnally, L., Henríquez, P.: Clasificación de estudiantes de nuevo ingreso a una universidad pública, con base a variables de desempeño académico, uso de tecnología digital y escolaridad de los padres. Revista Electrónica de Investigación Educativa, pp. 34–55 (2012)
15. Dutt, A., Ismail, M.A.: A Systematic Review on Educational Data Mining. IEEE Access, pp. 15991–16005 (2017)
16. Quintana Lopez, M., Flores Albino, J.M., Lazcano Salas, S., Landassuri Moreno, V.M.: Ensamble de Clasificadores para Determinar el Perfil Académico del Estudiante usando Árboles de Decisión y Redes Neuronales. Research in Computing Science (2018)
17. Gonzalez-Brambila, S.B., Sanchez-Guerrero, L., Ardon-Pulido, I., Figueroa-Gonzalez, J., Gonzalez-Beltran, B.A.: Predicting academic performance of engineering students after

- improving a mathematics leveling course using decision trees. *Research in Computing Science* (2018)
18. González-Brambila, S.B., Figueroa-González, J.: Looking Relationships between Pre-admission. *Research in Computing Science*, pp. 113–124 (2017)
 19. Oscar, H.A., Ari Yair, B.-A., Miguel, G.-M., Castro-Espinoza, F.: Monitoring Student Activities with Smartwatches: On the Academic Performance Enhancement. *Sensors*, vol. 19, no. 7 (2019)
 20. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA Data Mining Software: An Update. *SIGKDD Explorations*, pp. 10–18 (2009)
 21. Charrad, M., Nadia, G., Véronique, B., Azam, N.: NbClust: An R Package for Determining the Relevant Number of Clusters in a Data Set. *Journal of Statistical Software*, vol. 61, p. 1–36 (2014)
 22. Kassambara, A.: Determining The Optimal Number Of Clusters: 3 Must Know Methods. Sthda.com, <http://www.sthda.com/english/articles/29-cluster-validation-essentials/96-determining-the-optimal-number-of-clusters-3-must-know-methods/>. Último acceso: 2019/03/25
 23. Kassambara, A.: Practical Guide To Cluster Analysis in R Unsupervised Machine Learning. STHDA (2017)
 24. Sunghae, J.: An Ensemble Method for Validation of Cluster Analysis. *International Journal of Computer Science Issues*, vol. 8, no. 6 (2011)

Electronic edition
Available online: <http://www.rcs.cic.ipn.mx>

