

Reconocimiento de rostros por medio de Openface en una Raspberry Pi

Arturo Zúñiga-López, Juan Villegas-Cortez, Carlos Avilés-Cruz,
Eduardo Rodríguez-Martínez, Andrés Ferreyra-Ramírez

Universidad Autónoma Metropolitana,
Unidad Azcapotzalco, Departamento de Electrónica, Ciudad de México,
México

{azl, juanvc, caviles, erm, fra}@azc.uam.mx

Resumen. El reconocimiento de rostros es una herramienta biométrica cada vez más usada hoy día, gracias a la miniaturización de los dispositivos electrónicos, el aumento de la velocidad de procesamiento en el cómputo y, a una mejor comprensión del problema del reconocimiento de rostros desde el Reconocimiento de patrones y el Procesamiento digital de imágenes. En este artículo presentamos la implementación de un sistema de reconocimiento de rostros en un dispositivo móvil de bajas prestaciones y bajo costo, por medio de librerías optimizadas para el hardware móvil y en código abierto, trabajando con una red neuronal convolucional pre-entrenada de disposición abierta y alcanzando en una base de datos un reconocimiento del 100 %.

Palabras clave: reconocimiento de rostros, openface, procesamiento digital de imágenes, procesamiento digital de señales, reconocimiento de patrones, aprendizaje profundo.

Face Recognition Using Raspeberry Pi by means of Openface

Abstract. Face recognition is one of the most biometric tool used due the size reduction of embedded systems, the high processing data speed and the best comprehension of face recognition problem from the pattern recognition and digital image processing paradigms. In this work we present a system which is implementing optimized open source libraries in a low cost embedded mobile system, using a trained convolutional neural network which is freely available and reaching 100 % of recognition.

Keywords: face recognition, openface, digital image processing, digital signal processing, pattern recognition, deep learning.

1. Introducción

El reconocimiento facial ha sido un tema de investigación en el área de la visión por computadora y el reconocimiento de patrones por varias décadas, y en los últimos años los trabajos se han venido incrementando de forma exponencial [3]. Actualmente su principal aplicación es para los sistemas de seguridad y validación de identidad; por ejemplo, en los teléfonos inteligentes que al identificar el rostro del dueño del dispositivo, éste se desbloquea y permite su uso, o en sistemas de control para el acceso o paso en determinados lugares.

Adicionalmente y más común para todos, es lo que hoy día pasa al subir una foto a la red social Facebook, en la que automáticamente se detectan los rostros en la foto, se señalan al usuario y éste puede etiquetar a las personas ahí presentes, esta tarea es posible gracias al uso de algoritmos de reconocimiento facial.

Sin embargo, hay la necesidad de utilizar dispositivos móviles de bajo costo y rápida configuración, como el Raspberry Pi, que permitan que el costo de un sistema de biometría se pueda reducir [4,5], pensando en aplicaciones necesarias en dispositivos móviles para su desplazamiento físico para implementaciones que así lo demanden. El encanto de la Raspberry Pi proviene de proporcionar una combinación de tamaño pequeño, con un área aproximada a la de una tarjeta de crédito, y el desempeño de una computadora tipo personal con sistema operativo Linux.

Un modelo propuesto para la identificación de rostros es el mostrado en la Figura 1, en él se plantea la generalidad del procesamiento de la imagen digital para su reconocimiento y/o clasificación, tal que inicia con la adquisición de la imagen, luego se procede a detectar el área del rostro, que para nuestro propósito sería la región de interés (ROI), donde hallaremos los puntos de interés (POI) del rostro, para después aplicar un normalizado y acondicionamiento de la imagen preservando la ROI (véase la Figura 2); y a continuación se procede a la extracción de características, que para nuestro interés son los POI del rostro, y a partir de del patrón construido con estos POI y la información extraída al rededor de ellos y/o en conjunto con la disposición geométrica de los mismos, se procede a la concentración de todos los patrones en una base de datos o repositorio (BD) de los patrones, para sobre ellos aplicar un algoritmo de reconocimiento. Este proceso se divide tradicionalmente en dos etapas: entrenamiento o aprendizaje, y prueba. En la primera etapa se construye la BD y se aplica el algoritmo correspondiente para fijar los parámetros del clasificador usado o la metodología a implementar, y en la segunda etapa, se pone a prueba todo el sistema de identificación de rostros, y se realizan pruebas para determinar su porcentaje de clasificación.

En este artículo presentamos una implementación en un sistema embebido del tipo Raspberry Pi, de bajo costo, de un sistema de reconocimiento de rostros, basado en los puntos de interés (POI) del rostro, detectados y procesados por las librerías OpenCV y OpenFace, usando un total de 68 POI estimados por uno de los métodos más usados de estimación de POI en rostros [13], y probando sobre

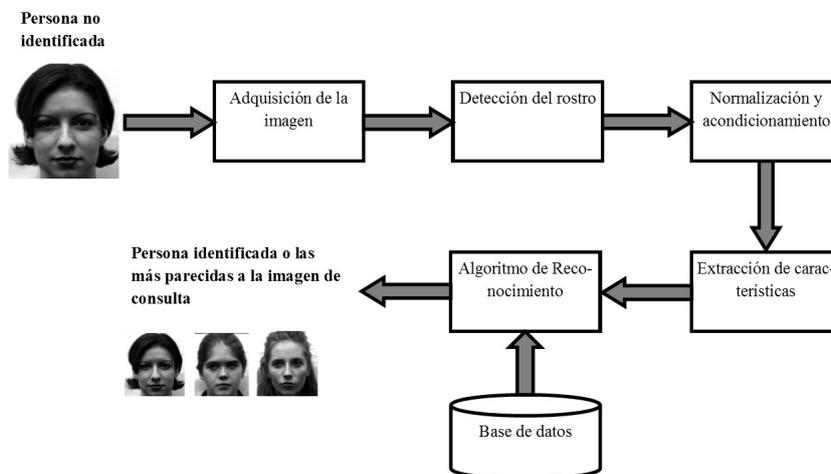


Fig. 1. Sistema para la identificación de rostros.



Fig. 2. Puntos de interés (POI) del rostro acorde a la literatura para su identificación, y una región de interés (ROI) del rostro enmarcada.

bases de datos de rostros que son ampliamente reconocidas en la comunidad científica del procesamiento digital de imágenes y visión por computadora.

La implementación esta realizada en lenguaje de programación ANSI C y Python, logrando en las pruebas realizadas un reconocimiento del 100%.

En la Sección 2 revisamos brevemente el estado del arte del cómo se ha atacado el problema del reconocimiento de rostros en general, en la Sección 3 presentamos la metodología usada, con el detalle necesario para poder replicar ésta implementación, así como las restricciones consideradas, en la Sección 4 se muestran los resultados obtenidos, así como una discusión de los mismos para analizar su escalamiento al Big Data, y finalmente en la Sección 5 proporcionamos las conclusiones.

2. Estado del arte

El reconocimiento de rostros es una tarea común para los seres humanos, se realiza de forma inconsciente y frecuente, sin embargo ésta actividad al realizarse en imágenes digitales no es una tarea sencilla. En los últimos años se han tenido avances significativos en el análisis del rostro, a partir del Reconocimiento de Patrones y el Aprendizaje Automático [14,15,16]; estos estudios han propuesto soluciones desde diferentes perspectivas al problema.

La adquisición de la imagen y/o detección de rostros se puede hacer mediante software o librerías. En la parte de normalización y acondicionamiento es donde se especifican los parámetros que tendrá dicha imagen, por ejemplo, que el rostro no tenga gafas, la iluminación, etc. Hablando de la extracción de características, aquí es donde se obtendrá la información necesaria de cada uno de los rostros, alguna de las técnicas usadas son: redes neuronales, eigenfaces, fisherface, etc. [6].

Los algoritmos de reconocimiento son aquel conjunto de instrucciones orientado hacia la agrupación de las características principales en los patrones que nos permitirán distinguir una cara de otra, en este segmento podemos mencionar a los métodos basados en clasificadores bayesianos, redes neuronales artificiales o los algoritmos genéticos, que proponen soluciones muy sencillas, desde la generalidad de los Algoritmos Evolutivos acorde a [1,2]; y uno de los avances más recientes es el conocido como *Aprendizaje Profundo o Deep Learning* (DL), siendo éste muy usado últimamente para múltiples tareas de reconocimiento en visión por computadora [12].

Hablando del DL, en esta metodología se caracteriza al objeto de estudio en la imagen digital con un cúmulo de rasgos, específicamente para los rostros serían: la longitud de la nariz, el tamaño de las orejas, el color de los ojos o algún otro rasgo que desde la perspectiva humana podemos visualizar, pues éstas se deben codificar en rasgos extraídos de los píxeles de la imagen. Acorde a las investigaciones, tal parece que lo mejor es que la propia computadora tome los rasgos que considere más apropiados a partir de la propia imagen, tal como se plantea desde otra perspectiva diferente la metodología CBIR [1], pero desde el DL se tiene una mejor forma de hacer esta caracterización, acorde a los resultados obtenidos de forma general, respecto a cuáles rasgos considerar del rostro.

Desde el DL se consideran las redes neuronales convolucionales (CNN), para caracterizar 128 medidas para cada persona, a partir de una tripleta, donde las dos primeras imágenes son de la misma persona, pero en distinta pose o variación, y la tercera imagen es de una persona diferente; ésta tarea se repite en la etapa de entrenamiento de la red para toda una base de datos o de rostros.

Esta tarea es de un alto costo computacional, se realiza en hardware de altas prestaciones y muy caras en dinero, tales como tarjetas NVidia Tesla, consumiendo al menos un aproximado de 24 horas de procesamiento continuo de millones de imágenes de miles de personas, tal que al final del entrenamiento puedan caracterizarse muy bien las 128 características para cada persona, véase la Figura 8. Afortunadamente para la comunidad científica, los desarrolladores



Fig. 3. Ejemplos de imágenes de la base de datos Yale.



Fig. 4. Ejemplos de imágenes de la base de datos AR.

de OpenFace ¹ han dispuesto CNNs entrenadas para usarlas directamente, y es como se implementan desde la librería OpenFace y en este trabajo lo mostramos.

3. Metodología

El método propuesto cumple el objetivo de desarrollar un sistema en hardware basado en la Raspberry Pi 2 modelo B (CPU 900 MHz, Quad-core ARM Cortex-A7 / 1 GB RAM / VideoCore IV 3D graphics core/ interfases CSI, DSI/ Micro SD card slot), de reconocimiento de rostros en tiempo real, siendo una de las aportaciones de éste trabajo el bajo costo, la rápida implementación de los algoritmos y una alta eficiencia en el reconocimiento de rostros.

Para este trabajo se utilizaron las bases de datos de Yale [7] y de AR [8], la primera contiene una colección de imágenes (tamaño de 320×243 píxeles en formato JPEG) de 13 individuos (ver Figura 3), con diferente iluminación y con 8 distintas expresiones faciales como son: postura normal, feliz, triste, soñoliento, etc. La segunda contiene imágenes de 83 individuos (véase la Figura 4), todas de frente y con expresiones faciales (algunos individuos tienen 6 poses y otros tienen 8), además las imágenes presentan iluminación distinta, están en formato JPEG y de tamaño 576×768 píxeles.

El sistema propuesto para implementarse en este trabajo se muestra en la Figura 5.

La imagen de entrada se captura con una webcam conectada al puerto USB de la Raspberry. La tarjeta Raspberry Pi tiene un algoritmo que clasifica a la imagen de entrada y visualiza el resultado en una pantalla. La función de la computadora es la de entrenar al sistema de reconocimiento de rostros, por medio de un algoritmo que se desarrolló utilizando la librería OpenFace [9]. En la Figura 6 se muestran las operaciones de los programas de cómputo que se implementaron en la computadora y en la Raspberry Pi.

La lógica e interacción de los algoritmos implementados para el sistema de reconocimiento, se explican a continuación:

¹ <https://cmusatyalab.github.io/openface/>

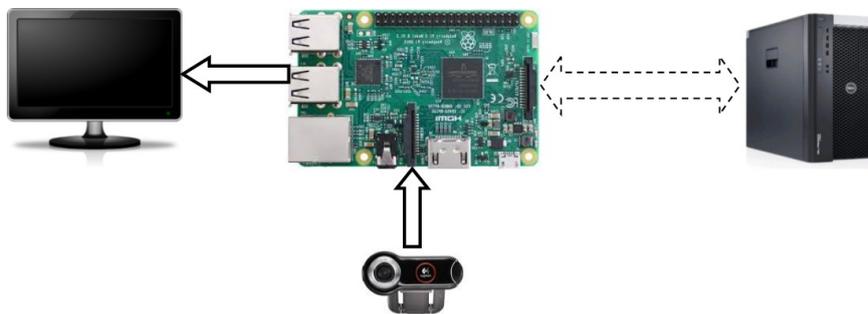


Fig. 5. Sistema para el reconocimiento de rostros.

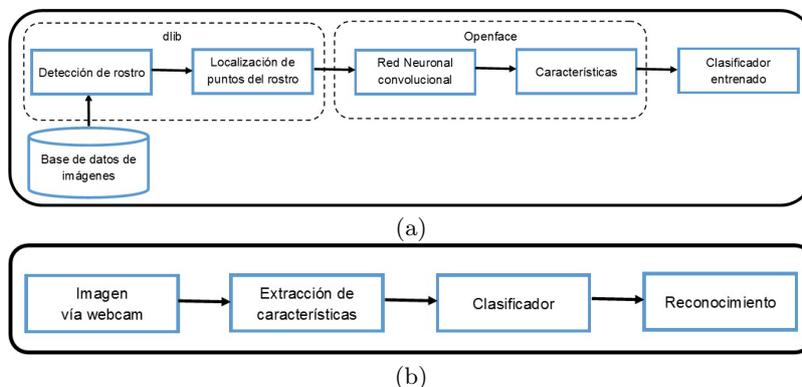


Fig. 6. Diagrama a bloques de los algoritmos desarrollados, a) en la computadora y b) en la Raspberry Pi.

Etapas desarrolladas en la computadora:

- Las imágenes de entrada al programa de entrenamiento son las bases de datos Yale y AR, y tienen una estructura de árbol para poder leerlas.
- Detección de rostro, del resto de la imagen. El propósito de esta etapa es encontrar el área de la imagen que contiene un rostro. Para ello se pueden utilizar diferentes técnicas como por ejemplo el algoritmo clásico de Viola y Jones, pero en este trabajo se utilizó el método denominado histograma de gradientes orientados (HOG). La ventaja de HOG frente a otros descriptores, proviene de sus características, tal que no es sensible a la transformación, la rotación, el ruido y a la mínima deformación [10]. Para su implementación se utilizó la librería *dlib*, en Python.
- Detección de puntos de interés del rostro. En esta etapa se utilizó, el algoritmo denominado estimación de puntos de referencia de la cara. Existen diversas formas de realizar el algoritmo. La librería *dlib* implementa una variante del enfoque que desarrollaron Vahid Kazemi y Josephine Sullivan

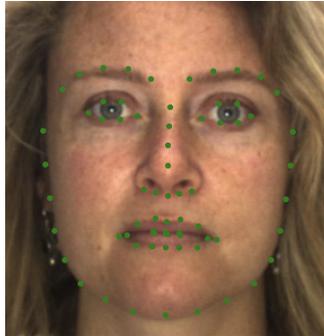


Fig. 7. Identificación de los puntos del rostro.

[11], de ella se obtienen las coordenadas de 68 puntos del rostro (6 puntos por cada ojo, de la boca 20, de la nariz 9, de las cejas 5 por cada una y de la barbilla 17). La detección de los puntos se muestra en la Figura 7.

- Entrenamiento a partir de imágenes de rostros y obtención del modelo de clasificación. En esta etapa se utiliza la librería OpenFace, que es una implementación que permite una alta precisión con bajo entrenamiento y tiempo de predicción, tiene una lógica para obtener representaciones de rostros en una imagen con una reducción de dimensionalidad, tal que usa un número reducido total de 68 POIs del rostro, logrando con ello la reducción de valores del patrón de POIs que optimiza el tiempo de cómputo y la precisión del resultado al aumentar el número de personas a reconocer y aprender de una BD. OpenFace usa una versión modificada de la red *FaceNet's NN4* una CNN [9]. De la CNN se obtienen únicamente 128 valores que representan las características por cada rostro, como se ejemplifica en la Figura 8.
- Posteriormente se construye un modelo de clasificador, del tipo $K - NN$ (*K-Nearest Neighbors*), que tiene la particularidad de ajustarse con las características obtenidas del entrenamiento (matriz de $128 \times N$ imágenes). El clasificador ya entrenado se guarda en disco, para posteriormente ser copiado en la tarjeta Raspberry Pi.

Etapas desarrollados en la Raspberry Pi:

- Adquisición de las imágenes a clasificar. En un dispositivo móvil basado en Android, se guardan algunos rostros de las bases de datos y se muestran a la webcam y con la ayuda de la librería OpenCV se realiza la captura de una imagen.
- Luego ésta es procesada para obtener los 128 puntos que representan las características del rostro, de la misma forma en que se trabajo para el entrenamiento. El vector obtenido se pasa al clasificador, que se obtuvo de la etapa de entrenamiento y su función es simplemente encontrar al vecino más cercano, es decir, el rostro que tendría las mismas características.

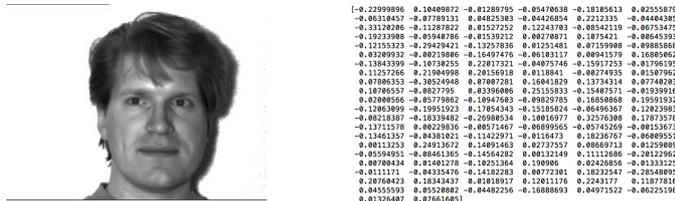


Fig. 8. Ejemplo de los valores que representan las características de un rostro.



Fig. 9. Obtención de los puntos de interés del rostro.

4. Resultados y discusión de resultados

En la Figura 9 se muestran algunos de los resultados de la etapa de detección de los puntos de interés de la cara. Podemos observar de la figura que la detección utilizando la librería *dlib* es aceptable. Sin embargo, hay ocasiones en que los puntos localizados se encuentran fuera de los rasgos faciales, como se puede ver en la imagen inferior izquierda de la Figura 9; en donde hay puntos arriba de la barbilla y en la boca los podemos encontrar en la zona de la lengua.

En la etapa de entrenamiento, observamos que el tiempo es aceptable, ya que para entrenar la base de Yale se tardó aproximadamente 9 segundos, y para la de AR 2.5 minutos. La base de datos AR es 6 veces más grande que la de Yale, y son sus imágenes de mayor resolución, de ahí que se lleve más tiempo.

Para visualizar los vectores de características de cada sujeto obtenidas del entrenamiento se aplicó el algoritmo *t-SNE* (*t-Distributed Stochastic Neighbour Embedding*), que es una técnica de reducción de dimensiones. Las gráficas obtenidas tras aplicar el algoritmo se muestran en las figuras 10 y 11.

Se escogió esta técnica porque a diferencia de otras técnicas clásicas, es una aproximación no lineal y se centra en mantener los puntos más semejantes cerca en su representación en bajas dimensiones, para nuestro caso en dos dimensiones. En la Figura 10 se muestran los resultados para la BD de Yale.

Se puede observar que la mayoría de las clases se encuentran alejadas y no se ve un solapamiento entre ellas; además, no hay puntos del rostro que se encuentren en una clase a la que no pertenezcan. La Figura 11 muestra los resultados para la BD AR, que a diferencia de la base de Yale, aquí sí hay imágenes de sujetos que se clasifican como de otra persona. También hay rostros que están alejados del racimo que representa su clase, y en algunos casos la distancia es considerable.

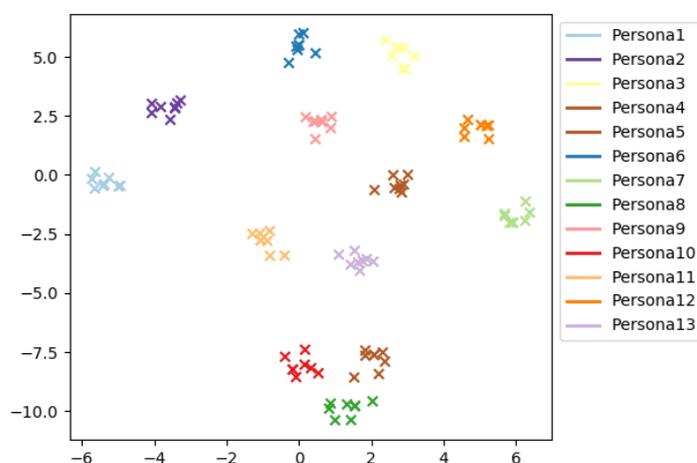


Fig. 10. Proyección de los vectores de características de los rostros de cada sujeto, de la base de datos de Yale.

Posteriormente se validó al clasificador, utilizando la misma base de imágenes para el entrenamiento. De esta prueba se obtuvo una recuperación del 100 % de reconocimiento de rostros, para la base de datos de Yale (véase la figura 12), y para AR se obtuvo una recuperación del 95.7%. Los parámetros del clasificador $K - NN$ fueron los mismos para ambas bases de datos.

Hablando de la implementación directa del sistema propuesto ejecutándose sólo en la Raspberry (véase la Figura 13), tenemos que el reconocimiento de rostros tiene algunos inconvenientes, como es que el vídeo tiene retardos considerables, de modo que se observa pausado. Estos retardos se deben principalmente a dos factores: por un lado, el número de cuadros por segundo que puede procesar la tarjeta de vídeo de la Raspberry, y, por otro lado, a la evaluación del clasificador. De aquí planteamos la limitante de velocidad de procesamiento con el modelo de Raspberry usado, pero seguramente con el más reciente modelo se tendrá un mejor uso de la memoria RAM.

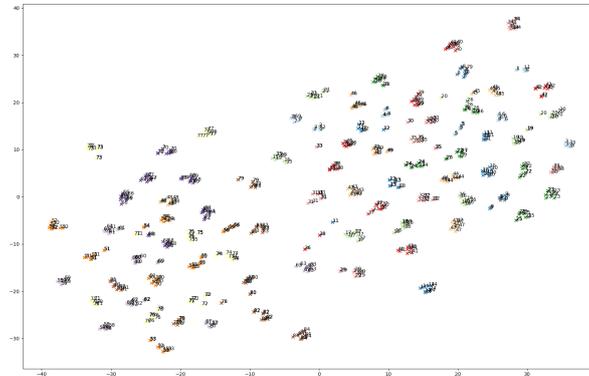


Fig. 11. Proyección de los vectores de características de los rostros de cada sujeto, de la base de datos de AR.

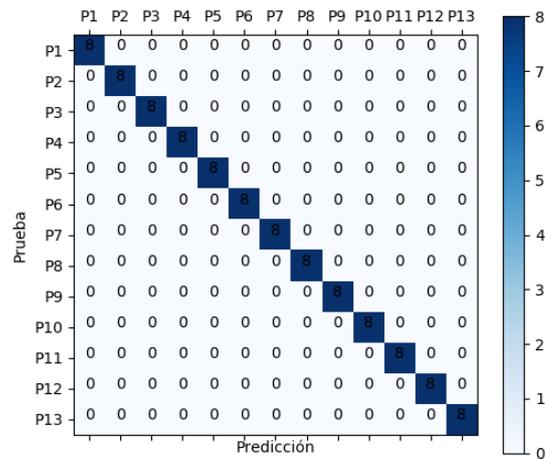


Fig. 12. Matriz de confusión para 13 clases con recuperación del 100%.

5. Conclusiones

Hemos presentado la implementación de un sistema de reconocimiento de rostros muy efectivo en un dispositivo móvil de bajas prestaciones y alto nivel de reconocimiento, aunado al bajo costo, y todo basado en código libre. La metodología planteada de obtener un sistema de reconocimiento en una Raspberry Pi



Fig. 13. Pantallas de ejemplos del sistema de reconocimiento de rostros portado, usando la pantalla de visualización de un dispositivo móvil.

se logro, con lo retos que se tiene en portar librerías que funcionen correctamente en un procesadores ARM.

Por otra parte, con la librería OpenFace y el clasificador $K - NN$ se logró un porcentaje de recuperación del 100% en una de las bases de datos reconocida y probada por la comunidad científica. Además, se pudo obtener una evaluación de forma visual de la distribución de características en una mapa de dos dimensiones mediante el uso del algoritmo $t-SNE$, ofreciendo una mejor representación de los datos.

Tal como aquí se mostró, el sistema esta disponible para poder ser aplicado en nuevas bases de datos de rostros, con la finalidad de poderse implementar para un fin específico de uso, tal como un control de paso o aduanal, de tal forma que de forma eficiente se pueda identificar a personas dentro de un repositorio entrenado, e.g. una empresa o centro de estudios o laboral que la seguridad requiera una capa de autenticación por rostro, de forma rápida y eficiente.

Como trabajo a futuro podrían implementarse éste proyecto como una parte de un sistema cliente-servidor, a tal fin de poder cargar de forma dinámica los repositorios acorde los escenarios o lugares de uso directo, por medio del puerto de red Ethernet, así como interactuar con sistemas más sofisticados de autenticación, donde nuestra propuesta sea un módulo fiable y rápido de integración.

Referencias

1. Benavides-Alvarez, C.: Sistema no supervisado de clasificación de rostros con técnicas basadas en CBIR. Tesis para obtener el grado de Maestro en Ciencias y Tecnologías de la Información, Universidad Autónoma Metropolitana (2015)
2. Deb, K.: Multi-Objective Optimization using Evolutionary Algorithms. Wiley Publishing (2001)
3. Ishita Gupta, Varsha Patil, Chaitali Kadman, Shreya Dumbre: Face Detection and Recognition using Raspberry Pi. In: International WIE Conference on Electrical and Computer Engineering, pp. 83–86 (2016)
4. Dhvani Shah, Vinayak Bharadi: IoT based Biometrics Implementación on Raspberry Pi. Procedia Computer Science, 79, pp. 328–336 (2016)

5. Mano, L.Y., Façal, B.S., Nakamura, L.H., Gomes, P.H., Libralon, G.L., Menequete, R.I., Filho, G.P., Giancristofaro, G.T., Pessin, G., Krishnamachari, B., Ueyama, J.: Exploit IoT technologies for enhancing Health Smart Homes through patient identification and emotion recognition. *Computer Communications*, 89-90, pp. 178–170 (2016)
6. Setiowati, S.Z., Franita, E.L., Ardiyanto, I.: A Review of Optimization Method in Face Recognition: Comparison Deep Learning and Non-Deep Learning Methods. (ICITEE) (2017)
7. Bellhumer, P.N., Hespanha, J., Kriegman, D.: Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Special Issue on Face Recognition 17(7), pp. 711–720 (1997)
8. Martinez, A.M., Benavente, R.: The AR Face Database. CVC Technical Report No. 24 (1998)
9. Amos, B., Ludwiczuk, B., Satyanarayanan, M.: Openface: A general-purpose face recognition library with mobile applications. CMU School of Computer Science, Tech. Rep. (2016)
10. Patil, P., Almeida, B., Chettiar, N., Babu, J.: Offline Signature Recognition System using Histogram of Oriented Gradients. In: *International Conference on Advances in Computing, Communication and Control (ICAC3)* (2017)
11. Kazemi, V., Sullivan, J.: One Millisecond Face Alignment with an Ensemble of Regression Trees. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1867–1874 (2014)
12. Basu, S., Mukhopadhyay, S., Karki, M., DiBiano, R., Ganguly, S., Nemani, R., Gayaka, S.: Deep neural networks for texture classification—a theoretical analysis. *Neural Networks*, 97, pp. 173–182 (2018)
13. Kazemi, V., Sullivan, J.: One Millisecond Face Alignment with an Ensemble of Regression Trees. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'14)*, pp. 1867–1874. IEEE Computer Society (2014)
14. Waldemar-Wocik, K. G., Junisbekov, M.: Face Recognition - Semisupervised Classification, Subspace Projection and Evaluation Methods. (INTECH '16), *Face Recognition: Issues, Methods and Alternative Applications* (2016)
15. Ekenel, H., Stiefelhagen, R.: Why is facial occlusion a challenging problem? In: Tistarelli, M., Nixon, M. (Eds.), *Advances in Biometrics*, Vol. 5558, pp. 299–308 (2009)
16. Martinez, A.M.: Recognizing imprecisely localized, partially occluded, and expression variant faces from a single sample per class. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 24(6), pp. 748–763 (2002)