Research in Computing Science

Research in Computing Science, Año 24, Volumen 154, No. 10, Octubre de 2025, es una publicación mensual, editada por el Instituto Politécnico Nacional, a través del Centro de Investigación en Computación. Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othon de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, Ciudad de México, Tel. 57 29 60 00, ext. 56571. https://www.rcs.cic.ipn.mx. Editor responsable: Dr. Grigori Sidorov. Reserva de Derechos al Uso Exclusivo del Título No. 04-2019-082310242100-203. ISSN: en trámite, ambos otorgados por el Instituto Politécnico Nacional de Derecho de Autor. Responsable de la última actualización de este número: el Centro de Investigación en Computación, Dr. Grigori Sidorov, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othon de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738. Fecha de última modificación 01 de octubre de 2025.

Las opiniones expresadas por los autores no necesariamente reflejan la postura del editor de la publicación.

Research in Computing Science, year 24, Volume 154, No. 10, October 2025, is published monthly by the Center for Computing Research of IPN.

The opinions expressed by the authors does not necessarily reflect the editor's posture.

Advances in Pattern Recognition

J. Arturo Olvera-López
J. Ariel Carrasco-Ochoa
J. Francisco Martínez-Trinidad
Adrián Pastor López-Monroy
Alejandro Rosales-Pérez (eds.)

SEP
SECRETARÍA DE
EDUCACIÓN PÚBLICA

ESTADOS UNIDOS MEXICANOS

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

IPN

Instituto Politécnico Nacional, Centro de Investigación en Computación
México 2025

ISSN: in process

Formerly ISSNs: 1870-4069, 1665-9899

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.

Indexed in LATINDEX, DBLP and Periodica

Electronic edition

Table of Contents

Page

Crime Prediction Using Computer Vision: Data Selection, Balancing, and Representation ....

L. Enrique Morales-Márquez, J. Arturo Olvera-López, Ivan Olmos-Pineda

Impact of Combined Attacks on Spam Detection: Targeted Poisoning and Backdoors ....

Samantha Acosta-Ruiz, Mireya Tovar-Vidal, José A. Reyes-Ortiz

Parallel Video Tracking System based on Honeybee Swarm Behavior and Evaluated on a Benchmark Dataset ....

Juan Alvaro de la Rosa-Ipiña, Carlos Soubervielle-Montalvo, Cesar Puente, Ruth Mariela Aguilar-Ponce, Luis Javier Ontañon García-Pimentel

Updating and Evaluating the Struck Tracker: A Comparative Study on the ALOV300++ Benchmark ....

Víctor Alejandro Méndez-López, Carlos Soubervielle-Montalvo, Cesar Puente-Montejano, Rafael Peña-Gallardo, Emilio Jorge González-Galván

An Experimental Study on the Analysis of Gait Disorders Through Multimodal Feature Extraction ....

Iván J. Sánchez-Cuapio, Ricardo Ramos-Aguilar, Paola A. Niño-Suárez, Karla M. Cerón-Arriaga

A Methodological Framework for Detecting Benevolent Misogynistic Hate Speech in Mexican Political Discourse Using Transformer-Based Models ....

Monserrat Sánchez-Juárez, Eric Ramos-Aguilar, Daniel Sánchez-Ruiz, Ricardo Ramos-Aguilar

Instrument Recognition Using Spectral Features and SVM ....

María Fernanda Arámburo-Castell, Juan Pintor-Michimani

Literature Summary Based on TF-IDF with Term-Term Correlation Matrix Analysis ....

Cybele Neves-Moutinho, María Fernanda Arámburo-Castell

RGB Image Compression Using Discrete Cosine Transform via Fixed Quantization Matrix ....

Michelle Guerra-Marín, Cybele Neves-Moutinho

Optimization of Audio Cover Detection Using Genetic Algorithms and Dynamic Time Warping ....
Juan Francisco Pintor-Michimani, Michelle Guerra-Marín

Node-weighted Graph Convolutional Network for Alzheimer's Dementia Detection from Transcribed Clinical Interviews with Data Augmentation ....
Lourdes Beatriz Cajica-Maceda, Perla Noemí Mendez-Zavaleta, Hugo Jair Escalante-Balderas, Jesús Ariel Carrasco-Ochoa, José Francisco Martínez-Trinidad

Intersubject Variability in Classification Models for Brain-Computer Interfaces ....
Eduardo V. Perez-Hernández, Margarita Aviña-Corral, Zaida Adriana Gárate-Cahuantzi, J. A. Carrasco-Ochoa, José Fco. Martínez-Trinidad, Alejandro A. Torres-García

Crime Prediction Using Computer Vision: Data Selection, Balancing, and Representation

L. Enrique Morales-Márquez, J. Arturo Olvera-López,
Ivan Olmos-Pineda

Autonomous University of Puebla, Faculty of Computer Science, Puebla, Pue., Mexico
luise.morales@viep.com.mx,
{jose.olvera,ivan.olmos}@correo.buap.mx

Abstract. Crime prediction and detection in video surveillance is a task currently under exploration. In various research papers, the ease of use of many novel video processing models and the focus on developing more powerful predictive architectures have relegated the task of data quality preprocessing to a small piece. These include everything from video set selection to analyzing the relationships between variables, tasks that impact model performance. This article presents preliminary results focused on the proper selection and processing of data from an ongoing crime prediction investigation. The paper presents the selection of the data set, the reasons for its selection, the representation of extracted data, data augmentation, and a method of data balancing that preserves data structure. It also presents a brief description of the method for determining dependencies between variables.

Keywords: Crime Prediction, Data Preprocessing, Computer Vision.

1 Introduction

Automatic crime prediction has become an important area of research in security and computer vision, with applications in public safety and crime prevention. However, current research faces significant challenges, including variability in environmental conditions such as lighting or camera angles; inherent imbalance in the ratio of normal events to crimes; and the movement of criminals, which can be very similar to that of a person with no malicious intent.

Recently, detection methods based on the analysis of human skeletons have emerged as a promising approach, as they offer a realistic representation of people's bodily behavior, at least visually. However, the quality of any investigation depends largely on preliminary steps that are often overlooked, such as the appropriate selection of a data set, the human pose extraction model, or class balancing. This article presents the preliminary stages of experimentation, focusing on data preprocessing. It is emphasized that early-stage decisions are just as crucial as the selection of a classification or prediction model or architecture.

This paper is structured as follows: Section 2 presents related work in computer vision-based crime prediction, subsequently, section 3 shows the methodology carried out so far and then, in section 4, the preliminary results are shown, which are discussed in section 5. Finally, section 6 contains the conclusions.

2 Related Work

Work on crime prediction in video has already been carried out; however, we often tend to focus on detailing the architecture or algorithm that will perform the detection, often neglecting to provide much detail in the papers or overlooking data preprocessing tasks. In [1], shoplifting prediction is performed using three-dimensional convolutional models (3DCNN) using the UCF–Crime Dataset [2] and obtaining an accuracy of 0.86. However, the preprocessing of the video clips consists of separating pre-crime timestamps, suspect behavior, and crime evidence, in addition to reducing the frame size to reduce computation time, but without taking other video characteristics into account.

Another similar treatment of video is found in [3], where frames from the CAVIAR dataset [4] are flagged for attention and converted to grayscale for feature extraction and subsequent detection of suspicious activity using VGG16 and Bidirectional Gated Recurrent Units (BiGRU) models. Accuracy, recall, and precision of 0.98, F1 of 0.97, and AUC of 0.99 are obtained, but the limitations and characteristics of the dataset are not taken into account, and efforts are focused on fine-tuning the data to feed intelligence models.

In [5], shoplifting is estimated using a fuzzy logic model that evaluates people's gait, crowd size, and degree of facial coverage of actors in videos from the UCF–Crime dataset. However, the facial coverage detection module is a retraining of the You Only Look Once (YOLO) model [6] that is fed with another dataset of helmets and face masks, to which a Gaussian filter is applied to remove artifacts within the frames, a single correction applied to the data. The fuzzy model achieves a precision of 0.77, a recall of 0.50, and an F1 of 0.60 value, performance that has room for improvement.

Currently, the convolutional model approach is commonly applied to crime prediction since these architectures are focused on image processing, it is common to find models that require minimal or no adjustment to the frames in order to be processed. However, an adequate selection and preprocessing steps have an impact on the prediction quality of the architecture being built.

3 Methodology

A methodological framework is essential to ensure the validity of any predictive model. Specifically, the process for processing the visual data in this work consists of dataset and Human Pose Estimation model, data augmentation for cases where a sufficient number of videos is unavailable. Subsequently, instance selection is performed to maintain class balance and, finally, determine the dependence or independence between variables for future training via Bayesian prediction models.

3.1 Dataset Selection

Dataset selection is a critical step in the development of prediction systems, as model generalization depends on it. As noted in [7], one of the factors attributed to the success of novel models is the availability of more and better data, and the effects of noisy or low-quality data must be taken into account when training algorithms.

Among the datasets available for crime detection, the UCF-Crime Dataset stands out for its widespread use in related studies. It contains approximately 1,900 video clips distributed across 13 categories that provide a variety of scenarios. However, limitations such as the maximum resolution of 320 x 240 pixels and poor image quality in almost all videos impact feature extraction.

To address these limitations, the Shoplifting Dataset 2022 [8] was selected, which provides certain advantages such as better resolution videos, balance with 92 shoplifting sequences versus 90 normal sequences, and controlled camera conditions at the cost of reducing the number of samples and limiting it to shoplifting. The general characteristics of both datasets are shown in Table 1.

Table 1. Characteristics of datasets for crime detection in video.

Dataset	Number of videos	Resolution	Crimes	Shooting angle	Frame Rate	Occlusion of actors
UCF– Crime	1,900	320x240p	Abuse, Burglary, Robbery, Stealing, Shooting, Shoplifting, Assault, Fighting, Arson, Vandalism	Zenith Street level High angle	30 frames per seconds	Partial, Total
Shoplifting 2022	182	640x480p 1920x1080p	Shoplifting	Street level	30 frames per second	None

While the UCF-Crime dataset contains a larger number of videos and a wider variety of crimes, the skeleton extraction models struggled to perform their task satisfactorily, a problem not encountered with higher-resolution videos. Furthermore, the absence of occlusions in the Shoplifting 2022 dataset allows for a clearer view of keypoint behavior, and the focus on a single crime allows for fine-tuning the predictive models for a single case to maximize their performance.

As shown in Figure 1, the improvement in quality allows MMPose [9] to consistently detect skeletons.

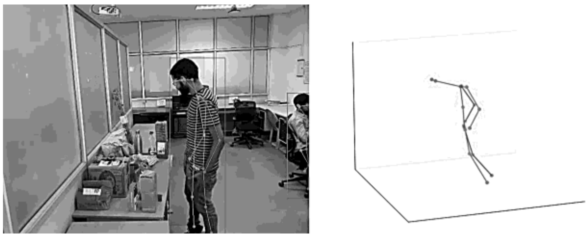


Fig. 1. Adequate human keypoint detection.

3.2 Representation of Joints

Since the two-dimensional representation of a person's skeleton may not be reliable due to scale or distance from the camera, it was decided to obtain the people's joints in three-dimensional space using MMPose 3D inference module. This module works by using 2D joints obtained using HRNet and processing them with a Meta Research model [10]. Keypoints are represented as a coordinate vector relative to an origin point located in the pelvis.

This architecture converts 2D keypoint sequences using residual blocks that capture temporal dependencies to generate maps for 3D skeletons and projects the results back into 2D for comparison with the initial skeleton. This process includes biomechanical constraints such as bone lengths or improbable positions when calculating 3D keypoints. This process is shown in Figure 2.

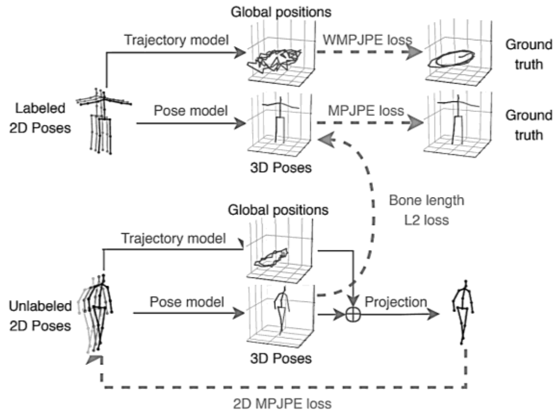


Fig. 2. Process of transforming 2D to 3D joints. Figure taken from [10].

As previously mentioned, tracking points within space may not be reliable; therefore, angles formed by angles formed by groups of three joints are calculated, taking one of them as the pivot point to which the calculated angle belongs, thus avoiding potential problems related to the distance between joints, their relative position in the plane, and considering perspective, scale, the person size, and proximity to camera. Angles in degrees are calculated by solving the dot product of the line segments formed by two points A, C with a common pivot point B, obeying Formula 1:

$$\theta = \arccos\left(\frac{\overrightarrow{AB}\cdot\overrightarrow{BC}}{\|\overrightarrow{AB}\|\|\overrightarrow{BC}\|}\right). \tag{1}$$

3.3 Data Augmentation

In our experimentation, we consider a set of 15 angles formed by different joints of a person's skeleton in a single frame as an instance. For greater control over augmentation and instance selection, the dataset was divided into separate subclasses, one for each actor appearing on camera, as well as non-mutually exclusive classes related to the camera view or the actor's behavior. Some of the subclasses have considerably fewer instances than others; furthermore, for reasons explained in section 3.5, at least 1000 instances are required for each subclass. To accomplish this task, data augmentation was performed by modifying the videos with classic transformations that can be seen in Figure 3.

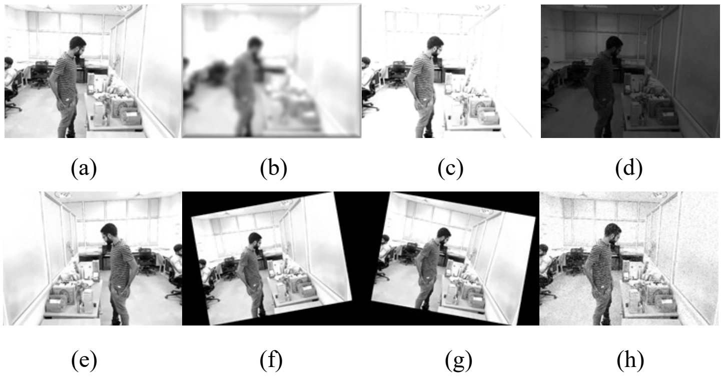


Fig. 3. a) Original image, b) Gaussian blur, $\sigma = 2.5$, c) Brightness increase to 150%, d) Brightness decrease to 10%, e) Flip relative to the y-axis, f) 10° rotation, g) -10° rotation, h) Salt and pepper noise at 5%.

The parameters for Gaussian blur, brightness modification and rotation were chosen based on the experimental results obtained in some studies reported in [11]. On the other hand, the salt and pepper noise was adjusted with a saturation of 5% to maintain sufficient similarity based on the results obtained in [12].

Gaussian blur can simulate out-of-focus frames caused by rapid movement or focus errors. Similarly, slight rotations simulate natural movements or positions because, in the real world, people and objects are rarely perfectly aligned, but they do not exhibit exaggerated inclinations. On the other hand, variations in lighting conditions describe how lighting conditions can change in videos, allowing models to detect a person's structural patterns. Furthermore, the mentioned transformations are computationally inexpensive, so augmenting the dataset does not require a lot of computational time.

Salt and pepper noise simulates unfavorable conditions that can occur in videos, such as dead pixels or interference. This is useful because it encourages intelligence models to avoid relying on data from specific regions and focus on global or contextual patterns.

These variations were carried out in order to obtain the largest possible transformation while respecting realistic variations and ensuring that the MMPose model correctly extracted keypoints. The number of instances before and after augmentation is shown in Tables 2 and 3.

3.4 Instance Selection

At this point, we have reached a problem with unbalanced classes. Note that in Tables 1 and 2, there are classes with more than 100,000 instances, while the minority classes contain 1,120 and 4,056 elements. A random selection of the same number of instances in each subclass does not guarantee that the new sample will contain members that reflect the behavior of the subclass.

Some instance selection methods based on clustering have shown satisfactory results, preserving the structure and behavior of the cluster, as in [13]. Based on this idea, instance selection is carried out using the k-Means algorithm, where k corresponds to the number of instances of the minority class; this applies to all classes.

Table 2. Data augmentation for Shoplifting class.

Subclass	Original amount	Amount after augmentation	Subclass	Original amount	Amount after augmentation
Actor 1	1,614	12,912	Backpack on	591	4,728
Actor 2	806	6,448	Backpack in hand	4,359	34,872
Actor 3	462	3,696	Remove backpack	1,448	11,584
Actor 4	704	5,632	Without backpack	3,862	30,896
Actor 5	204	1,632	Back view	140	1,120
Actor 6	1,823	14,584	Front view	1,357	10,856
Actor 7	1,455	11,640	Diagonal view from the front	6,986	55,888
Actor 8	1,061	8,488	Diagonal view from the back	2,045	16,360
Actor 9	529	4,232	Side view	6,706	53,648
Actor 10	321	2,568	Diagonal entry	1,057	8,456
Actor 11	577	4,616	Front entry	1,430	11,440
Actor 12	315	2,520	Side entry	1,196	9,568
Actor 13	201	1,608	In position	6,553	52,424
Actor 14	174	1,392			

Table 3. Data augmentation for Normal class.

Subclass	Original amount	Amount after augmentation	Subclass	Original amount	Amount after augmentation
Actor 1	5,081	40,648	Backpack on	2,785	22,280
Actor 2	1,004	8,032	Without backpack	17,134	137,072
Actor 3	5,923	47,384	Back view	2,438	19,504
Actor 4	1,858	14,864	Front view	5,580	44,640
Actor 5	793	6,344	Diagonal view from the front	16,865	134,920
Actor 6	510	4,080	Diagonal view from the back	7,278	58,224
Actor 7	838	6,704	Side view	15,281	122,248
Actor 8	1,578	12,624	Diagonal entry	1,732	13,856
Actor 9	1,215	9,720	Front entry	4,009	32,072
Actor 10	507	4,056	Side entry	1,980	15,840
Actor 11	621	4,968	In position	12,447	99,576

Starting from an initial data set, each point is relocated within a cluster whose center is the closest by calculating the average of the points within the cluster. The relocation process is repeated until a stopping criterion is met, typically a number of iterations [14] or until centroids have no changes.

An element is then selected from each cluster, resulting in a new selected set containing data that preserves the structure of the entire class but with a size that respects a perfect balance between classes. The element selected as the cluster representative is the instance closest to the centroid, this is because the k-Means algorithm could generate synthetic centroids that do not correspond to any instance and therefore are not valid elements for training.

3.5 Dependences Between Variables

One of the most common ways to determine dependence between variables is through the construction of Directed Acyclic Graphs (DAGs). However, this approach typically requires evaluating a large number of graphs, which requires a significant amount of time and computing resources when using combinatorial algorithms [15]. In this case, the *Non-combinatorial Optimization via Trace Exponential and Augmented lagRangian for Structure learning* (NOTEARS) algorithm [15] is used, which allows obtaining a DAG that shows the dependence between variables.

The NOTEARS algorithm proposes an objective function that minimizes the Mean Square Error (MSE) of fit, subject to an acyclicity constraint such that:

$$h(W) \;=\; tr(e^{W \circ W}) - d = 0. \tag{2}$$

Where W is the matrix containing the weights of the causal relationships and d is the number of variables. This formulation employs Lagrangians to solve the optimization problem efficiently and provides a valid DAG.

Values close to zero in the weight matrix indicate a small dependence that may not be considered depending on the research purpose. Larger values suggest a strong influence and should therefore be considered. A positive value on the edges indicates a direct relationship; that is, an increase in one variable leads to an increase in the other variable. Conversely, a negative value indicates an inverse relationship, where an increase in one variable leads to decrease in the other one. It is important to mention that authors suggest using at least 1,000 instances to run the algorithm with reliable results.

4 Preliminary Results

In this section, the result of applying the NOTEARS algorithm to a data set is briefly shown, and finally, the similarity of the DAGs of the complete data sets with respect to the graphs obtained using the balanced sets is evaluated.

4.1 Dependency Graphs

NOTEARS algorithm returns a weight matrix containing all possible combinations of variable pairs and the dependency values between them. In practice, some of the weights are 0, since this means that these edges do not exist and the acyclicity restriction is met. However, some edges obtain values close to zero. The variables included in these edges can be considered independent; however, the threshold at which this decision is made is not defined and is therefore subject to the conditions of the study being carried out.

Figure 4 shows an example of a dependency DAG for the Actor7 subclass of the Shoplifting videos, where only dependencies with a value greater than 1.5 are placed. The triple: *θKeypoint: Reference1(Side), Reference2(Side)* indicates the pivot joint for which the angle is calculated relative to the points *Reference1(Side)* and *Reference2(Side)*. Labels *L (Left)* and *R (Right)* indicate the side of the body where the joint is located.

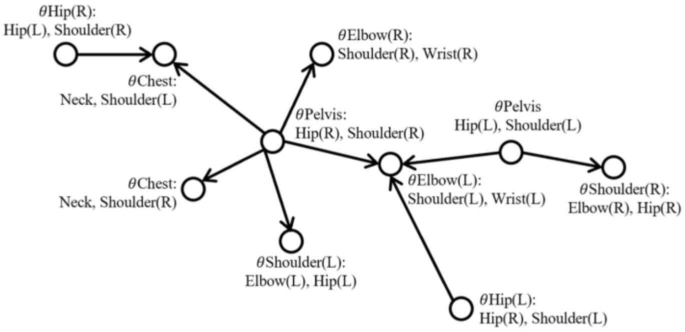


Fig. 4. Example of a dependency DAG for Actor7 in Shoplifting class. *θKeypoint* refers to the angle with vertex at the mentioned keypoint taking *Reference1(Side)* and *Reference2(Side)* as *reference.*

The frequencies of values of all the edges for the subclass Actor7 can also be seen in Figure 5. For this particular work, the weight values are bounded by the interval [-3, 3] and most of them are found in values close to 0, so the threshold could be adjusted close to 1 to discard such dependencies and consider the variables involved as independent. The rest of the dependencies that are considered strong should be considered in the next phases of the investigation.

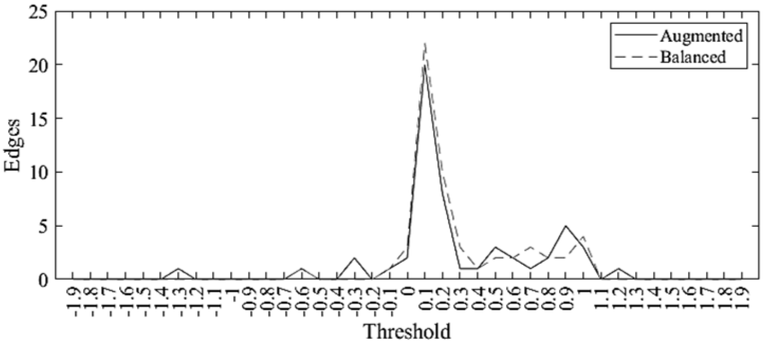


Fig. 5. Comparison of frequency of edge weights between original and balanced.

4.2 Quality of Dependency Graphs via Subsampling

In order to determine the quality of the DAG obtained using balanced classes after instance selection, the MSE is obtained between the edges of the graph with the total data and the graph with selected instances.

The minimum and maximum differences found between the edges of the graph, as well as the MSE, the average, and the standard deviation of the latter, are reported in Tables 4 and 5. Frequencies of the edges compared to original dataset are depicted in Figure 4.

The data preprocessing results shown may appear somewhat more extensive than what is typically published, but meticulous preprocessing is essential, especially for computer vision tasks. The methodology used here includes dataset selection, augmentation, instance selection, and determination of dependencies between variables; steps that have a significant impact on the quality of the results of predictive models.

The MSE values between the full dataset and the balanced subsets are low, with values of 0.0778 and 0.0479 for the Shoplifting and Normal classes, respectively, considering maximum differences of up to 3 units and almost 2.5 units. Similarly, the standard deviations remain low, with values of 0.0332 and 0.0379, demonstrating that when all balanced subclasses are considered, the dependency graphs maintain sufficient similarities to continuing the experiments.

To discard variables that could be considered dependent, an initial experiment takes into account the frequency of the edges in each interval, all the behaviors of each subclass of both classes are overlapped. Having a behavior similar to that of a normal distribution, those variables that are within the 68.3% corresponding to a standard deviation on each side of the bell are preserved, see Figure 6.

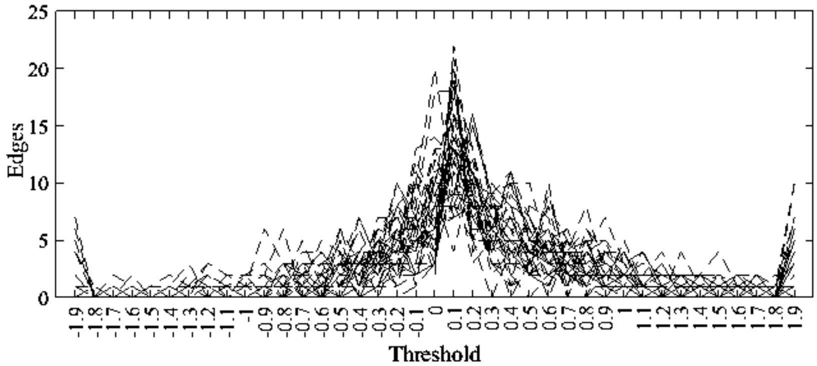


Fig. 6. Overlapping the behavior of the edges of all subclasses.

After this process, a Random Forest model is trained to classify each video frame as an image belonging to a normal sequence or one in which there is a behavior prior to the theft, this experiment obtained an accuracy of 0.74, precision, recall and F1 of 0.61, values that show a wide margin of improvement for future experiments that consider a different selection of threshold of dependency values and automatic selection of hyperparameters for the random with the objective of improving the classification results. Confusion matrix for this experiment is presented in Figure 7.

Table 4. MSE for Shoplifting videos.

Subclass	MSE	Maximum Difference	Subclass	MSE	Maximum Difference
Actor 1	0.0895	2.2702	Backpack on	0.0771	1.5785
Actor 2	0.0393	0.9323	Backpack in hand	0.0959	1.8076
Actor 3	0.1035	2.6415	Remove backpack	0.0602	1.7603
Actor 4	0.0523	1.6001	Without backpack	0.0824	1.5439
Actor 5	0.0762	1.7344	Back view	0.1025	3.4630
Actor 6	0.0453	1.5792	Front view	0.0720	1.1674
Actor 7	0.0529	1.6119	Diagonal view from the front	0.1026	1.6953
Actor 8	0.0397	0.7361	Diagonal view from the back	0.0801	1.0507
Actor 9	0.0121	0.7015	Side view	0.1031	1.5206
Actor 10	0.0872	1.6404	Diagonal entry	0.0342	0.7173
Actor 11	0.1329	2.9279	Front entry	0.1093	2.2266
Actor 12	0.0765	1.3339	Side entry	0.1320	1.8082
Actor 13	0.1498	3.1309	In position	0.0278	0.7353
Actor 14	0.0632	1.2718	**Mean (all subclasses)**	0.0778	
			St. Deviation (all subclasses)	0.0332	

Table 5. MSE for Normal videos.

Subclass	MSE	Maximum Difference	Subclass	MSE	Maximum Difference
Actor 1	0.1039	2.4128	Without backpack	0.0737	1.8061
Actor 2	0.0156	0.5990	Back view	0.0388	1.7170
Actor 3	0.0651	1.9416	Front view	0.0814	1.3664
Actor 4	0.0936	3.1229	Diagonal view from the front	0.0908	1.7113
Actor 5	0.0061	0.4405	Diagonal view from the back	0.0383	2.0397
Actor 6	0.0015	0.3816	Side view	0.0867	1.8049
Actor 7	0.0058	0.3944	Diagonal entry	0.0528	1.2931
Actor 8	0.0654	1.5684	Front entry	0.1294	1.6946
Actor 9	0.0393	1.1893	Side entry	0.0124	0.6764
Actor 10	4.9967e-06	0.0123	In position	0.0400	1.3990
Actor 11	0.0035	0.3790	**Mean (all subclasses)**	0.0479	
Backpack on	0.0102	0.5600	**St. Deviation (all subclasses)**	0.0379	

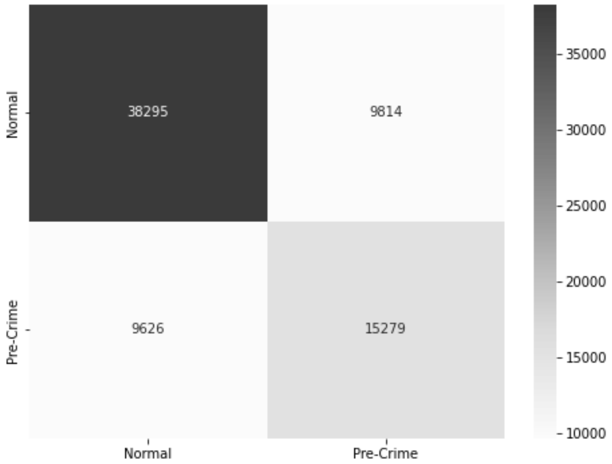


Fig. 7. Confusion matrix for random forest.

5 Conclusions

This paper explores the data preprocessing stages in crime prediction using computer vision, focusing on data selection, class balancing, and three-dimensional representation of human poses. Important decisions such as dataset selection have a significant impact on the results. Opting to use the Shoplifting Dataset 2022 allowed for more accurate extraction of human keypoints due to the higher resolution and quality of the video clips compared to the UCF – Crime Dataset.

The use of data augmentation techniques made it possible to meet the instance count restriction required by the NOTEARS algorithm, and k-Means instance selection ensured class balance while maintaining the structure and behavior of the original data. Furthermore, NOTEARS provided dependency graphs in less time and with lower resource consumption than traditional combinatorial methods. These preliminary results show that DAGS calculated with balanced data maintain comparable quality to those generated with the entire data set, validating the proposed approach.

Detection and prediction tasks can be extended to other crimes such as theft or assault. In addition, in the future, a greater number of angles between joints can be considered, along with the inclusion of temporal statistical data obtained using existing angles and other preprocessing tasks such as data filtering to reduce erratic behavior or instability. Of course, the exclusive use of angles from human keypoints suggests limitations such as not utilizing more contextual information or, at this point, interactions with other people. However, the preliminary result can be used as a starting point for improvements, taking into account the previously suggested expansions that can be carried out.

This paper lays the groundwork for future research on crime prediction using computer vision, which has the potential to contribute to public safety.

Acknowledgments

This research work is supported by SECIHTI through national scholarships for postgraduate students number 846099.

References

1. Martínez-Mascorro, G.A., Abreu-Pederzini, J.R., Ortiz-Bayliss, J.C., Garcia-Collantes, A., Terashima-Marín, H.: Criminal Intention Detection at Early Stages of Shoplifting Cases by Using 3D Convolutional Neural Networks. Computation **9**(24), (2021)
2. Sultani, W., Chen, C., Shah, M.: Real-World anomaly detection in surveillance videos. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6479 – 6488. IEEE, Salt Lake City, UT, USA (2018)
3. Gandapur, M.: E2E-VSDL: End-to-end video surveillance-based deep learning model to detect and prevent criminal activities. Image And Vision Computing **123**, 104467–104476 (2022)
4. Caviar dataset, http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/, last accessed 2025/04/30
5. Pouyan, S., Charmi, M., Azarpeyvand, A., Hassanpoor, H.: Propounding First Artificial Intelligence Approach for Predicting Robbery Behavior Potential in an Indoor Security Camera. IEEE Access **11**, 60471–60489 (2023)
6. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You Only Look Once: Unified, Real-Time Object Detection. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779 – 788. IEEE, Las Vegas, NV, USA (2016)
7. Karimi, D., Dou, H., Warfield, S. K., Gholipour, A.: Deep learning with noisy labels: Exploring techniques and remedies in medical image analysis. Medical Image Analysis **65**, 101759–101789 (2020)
8. Shoplifting Dataset (2022) - CV Laboratory MNNIT Allahabad, https://mmpose.readthedocs.io/en/latest/, last accessed 2025/04/30
9. Welcome To MMPose's Documentation, https://mmpose.readthedocs.io/en/latest/, last accessed 2025/04/30
10. Pavllo, D., Feichtenhofer, C., Grangier, D., Auli, M.: 3D Human Pose Estimation in Video With Temporal Convolutions and Semi-Supervised Training. In: 2022 IEEE/CVF Conference On Computer Vision And Pattern Recognition (CVPR), pp. 7745 – 7754. IEEE, Salt New Orleans,LA, USA (2022)
11. Yan, W.Q., Nguyen, M., Stommel, M.: Image and Vision Computing. (2023). https://doi.org/10.1007/978-3-031-25825-1.
12. Bindal, N., Garg, B.: Novel three stage range sensitive filter for denoising high density salt & pepper noise. Multimedia Tools And Applications. 81, 21279-21294 (2022)
13. Olvera-López, J. A., Carrasco-Ochoa, J. A., Martínez-Trinidad, J. F.: A new fast prototype selection method based on clustering. Pattern Analysis And Applications **13**(2), 131–141 (2009)
14. Jin, X., Han, J.: K-Means clustering. In: Encyclopedia of Machine Learning. pp. 563-564 (2010)
15. Zheng, X., Aragam, B., Ravikumar, P., Xing, E. P.: DAGs with NO TEARS: continuous optimization for structure learning. In: NIPS'18: Proceedings of the 32nd International Conference on Neural Information Processing Systems, pp. 9492 – 9503. Curran Associates Inc, Montreal, Canada (2018)

Impact of Combined Attacks on Spam Detection: Targeted Poisoning and Backdoors

Samantha Acosta-Ruiz[1], Mireya Tovar-Vidal[1], José A. Reyes-Ortiz[2]

[1] Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Computación, Puebla, Pue., México
[2] Universidad Autónoma Metropolitana, División de Ciencias Básicas e Ingeniería, Azcapotzalco, México
ar224570157@alm.buap.mx,mireya.tovar@correo.buap.mx,jaro@azc.uam.mx

Abstract. This project addresses a critical issue in the security of artificial intelligence systems: the vulnerability of classification models to individual and combined adversarial attacks. Instead of focusing on maximizing performance under ideal conditions, we analyzed how different classification algorithms include: Support Vector Machines, Decision Tree, RandomForest, Naive Bayes, and AdaBoost respond to threat scenarios. To do this, targeted poisoning attacks were applied using DeepWordBug, and later a backdoor attack was integrated to build a combined attack scheme. Although some models showed high initial performance (for example, AdaBoost and Naive Bayes achieved 99.44% accuracy with TF-IDF), the results revealed severe degradations in the presence of disturbances, especially in the spam class. In addition, the Local Interpretable Model-agnostic Explanations (LIME) technique was used as an Explainable Artificial Intelligence (XAI) tool to audit whether the compromised model had learned the malicious trigger as a relevant characteristic, which was confirmed in 97% cases. These findings demonstrate the effectiveness of combined attacks, the need to evaluate systems in adverse conditions, and the importance of integrating interpretation and defense mechanisms early in the Artificial Intelligence (AI) system design process.

Keywords: Spam Detection, Targeted Poisoning, Backdoor Attacks, Combined Adversarial Attacks, XAI.

1 Introduction

The evolution of email as a primary communication tool has led to the development of automated systems for filtering unwanted content, known as *Spam*. To deal with this threat, machine learning models have been adopted that have demonstrated high performance in identifying their own malicious or irrelevant message patterns. However, with the advancement of these technologies, new attack techniques designed to compromise their effectiveness have also emerged. Among the most relevant emerging threats are adversary attacks [7], which seek to manipulate the model's behavior so that it fails in its classification task.

In particular, these types of attacks pose a significant challenge because they are often carefully designed to resemble legitimate data, making them difficult to detect. Among the most commonly used adversary techniques by cybercriminals are *poisoning attacks* during training, considered one of the most serious threats to the integrity of machine learning systems. These attacks intentionally introduce malicious data into the training set to manipulate the model's behavior. Depending on the attacker's intention, they can be classified into two broad categories: *targeted attacks*, which seek to affect the result for specific inputs, and *non-targeted attacks*, whose objective is to degrade the overall performance of the model in a broader and more indiscriminate manner [6]. As a result, during the testing phase, the model may mistakenly classify legitimate emails as spam. In addition, if the attacker can access real samples of the victim's email, he can replicate his style to generate highly convincing malicious messages. Even without this direct access, it is possible to build examples using vocabulary associated with legitimate or spam content, depending on the strategy you want to follow.

With the increasing adoption of language models (LMs) in real environments, the attack surface has expanded, giving rise to threats such as *backdoor attacks.* In these attacks, the adversary incorporates specific patterns during training so that the model activates a malicious behavior only in the presence of a hidden trigger. Under normal conditions, the model operates correctly, making it difficult to detect by conventional evaluations [2]. This represents a serious risk, especially if the model is used in critical tasks such as detecting toxic or malicious content.

To analyze textual adversary attacks in more depth, it is useful to classify them according to different factors, including the degree of access the attacker has to the model, the purpose of the attack, the structure of the compromised model, and the level of intervention on the text. This last criterion, focused on the extent and type of alterations made to the textual content, allows to distinguish four main categories of attack, each with particular strategies and levels of complexity [11]:

- **Character level.** The attacker alters individual characters (by inserting, deleting, or replacing), resulting in easily detectable spelling and grammatical errors.

- **Word level.** The attacker modifies words in the text, maintaining semantic coherence better and going unnoticed, but with less diversity in the generated examples.

- **Sentence level.** The attacker introduces new sentences, changes words for synonyms, or adjusts the structure of sentences, preserving semantics and increasing diversity, although some texts may lose legibility.

- **Multi-level.** It involves modifications in characters, words, and sentences, offering more variety than attacks at the level of characters or words, although with additional restrictions.

The analysis of adversary attacks in natural language processing (NLP) models highlights both the inherent vulnerabilities of these systems and the urgent need to develop more robust and understandable approaches. In this context, explainable artificial intelligence (XAI) plays a crucial role, as it seeks to develop artificial intelligence (AI) systems that offer accurate predictions and provide clear and understandable explanations about their results. In the field of cybersecurity, the XAI allows professionals and stakeholders to understand how AI models reach their conclusions, which is essential in tasks such as threat detection, risk assessment, and decision making in this field [12].

This explanatory capacity is aligned with the principles of Responsible AI, where the transparency of the models plays a key role by facilitating the identification of biases and vulnerabilities that could be exploited [12]. In this context, integrating explainability techniques allows for analysis of the impact of adversary attacks and improves the ability of systems to adapt and respond effectively to them. This convergence between explainability and robustness becomes especially critical in sensitive applications such as spam detection, where it is essential to ensure both user trust and resistance to malicious manipulations.

However, most studies address different types of attacks separately, making it difficult to understand their combined effects on model behavior and decision limits. In particular, analyzing targeted poisoning and backdoor attacks together makes it possible to simulate more realistic and stealthy threat scenarios, where an adversary can manipulate specific predictions and trigger hidden behaviors through trigger-based mechanisms. In such contexts, explainability becomes crucial, not only for transparency, but also as a tool to detect and interpret abnormal patterns that might go unnoticed through traditional evaluation metrics.

This integrative perspective shapes the approach of the present study, which aims to explore the behavioral vulnerabilities of spam detection models under adverse compound conditions. Rather than optimizing for maximum accuracy with clean data, the goal is to assess how high-performance models degrade when exposed to the combined effect of targeted poisoning and backdoor attacks. To support this analysis, LIME is used as a local explainability technique that allows us to inspect the internal behavior of the compromised models and determine whether the trigger has been incorporated as a key feature in decision making. By combining robustness tests with explainability, this work contributes to a better understanding of the failure of the AI model under realistic adverse conditions, offering valuable information for the development of more resilient and transparent AI systems.

1.1 Related Work

Recent studies have explored adversarial attacks in NLP, particularly in spam detection tasks, where manipulating textual inputs can significantly compromise the *accuracy* and *robustness* of machine learning models. This section reviews key contributions related to data poisoning and backdoor attacks, as well as notable adversarial methods relevant to this study.

In [1] they proposed an approach based on classifier stacking to improve spam detection, combining Logistic Regression, Decision Trees, k-Nearest Neighbors (KNN), Naive Bayes and AdaBoost. Although the latter stood out individually, the stacking method achieved an *Accuracy*, *Recall*, and F_1-*Score* of 0.988, demonstrating the potential of hybrid methods. On the other hand, the work [7] focuses on analyzing the various strategies used by spammers to contaminate training data, as well as advanced machine learning-based filtering techniques. The experimental results showed that ignoring the changes in the dataset can cause severe performance degradation, with error rates up to 48.81%.

The works DeepWordBug [5] and TextBugger [9] represent attack techniques in black-box scenarios, where key tokens are identified in the text to alter them by almost imperceptible modifications, such as substitution, deletion, insertion, or exchange of characters. Both approaches improve the effectiveness of attacks through punctuation functions that prioritize the most harmful changes to the model without seriously affecting the readability of the text.

Several studies have shown how easily a model can be manipulated during the training phase in the field of backdoor attacks. In [13], a part of the training set was poisoned to associate outstanding male actors with negative feelings. This attack was evaluated on the Internet Movie Database (IMDB) and Stanford Sentiment Treebank (SST) datasets and on seven different models, including BERT and RoBERTa. The results showed that the accuracy of benign data was hardly affected, while the malicious association was successfully learned, reaching a 100% success rate with only 3% of poisoned data.

Similarly, the work [3] proposed a black-box scenario attack, where the attacker only had a small fraction of the training set and did not know the architecture of the model. By inserting a trigger phrase during the training of an LSTM model, classification errors were induced, reaching a 96% success rate with just 1% of poisoned data. Although work [6] explored the possibility of reinforcing backdoor attacks by incorporating adversarial disturbances in the inference stage, posing a potential convergence between evasion and backdoor attacks. However, this strategy remains an open problem, as its integration during training and its impact on tasks such as text classification have not been investigated.

Although the literature has extensively addressed data poisoning and backdoor attacks separately, no studies have yet been reported that integrate both approaches within the same experimental scheme applied to text classification. This gap highlights the need for comprehensive studies that not only evaluate the individual impact of adversarial strategies, but also assess their combined effects and the capacity of explainability methods to uncover hidden manipulations.

Therefore, this work will analyze a combined adversary attack scheme that integrates targeted poisoning and backdoor attacks. The approach involves the use of the *DeepWordBug* [5] method implemented through the TextAttack library [10] as an automatic generator of adversarial examples. These perturbations will be incorporated into the training set to evaluate the vulnerability model. The impact of attacks will be assessed not only through traditional performance metrics, but also through explainability analysis using LIME, with the goal of verifying whether the trigger is internalized as a relevant decision feature. This experimental design aims to demonstrate the effectiveness of combining adverse strategies and evaluate their potential to evade conventional spam detection systems.

2 Methodology

In the development of this work, the SpamAssassin database was downloaded for analysis and testing [4]. The database consists of messages classified as legitimate and illegitimate, denoted by labels: *ham* and *spam* , with 4150 and 1897 examples, respectively. All texts were verified to be in English and did not contain empty entries. After deleting messages partially written in another language, 3916 ham and 1897 spam remained. In addition, the length of the messages was analyzed to detect possible biases; the atypically long texts were eliminated using quartile filters, leaving a total of 5371 messages.

Unlike short texts such as instant messages or forum posts, emails are usually longer and contain numerous irrelevant or noisy tokens, mainly derived from the information contained in their headers. Therefore, the preprocessing focused on cleaning up the corpus to improve the performance of the model. The following steps were taken: *1)* all text is converted to the lower case; *2)* numbers, punctuation marks and stopwords are removed; *3)* identification and replacement of specific entities using regular expressions, replacing emails, URLs, phone numbers, and usernames with standard tags that preserve the structure of the text but anonymize its content. The tags used were: *EMAIL*, *URL*, *PHONE* and *USER*. Finally, the classes were coded as 0 for legitimate messages (ham) and 1 for spam messages.

Once the preprocessing was completed, the dataset was structured in two columns: one contained the complete message already processed, and the other its respective label. It should be noted that this work aligns with the first level of granularity proposed in [8] for email analysis: based on the complete message. That is, the analysis and classification were carried out taking into account the entire content of the email as a single input unit, without fragmenting it into phrases, sentences or keywords. This clarification is relevant to contextualize the type of adversary attack applied and the way the models interpreted the examples during the training and evaluation.

The preprocessed texts were vectorized using the Term Frequency-Inverse Document Frequency (TF-IDF) and Bag of Words (BOW) techniques. Based on these representations, several classifiers were trained, including Support Vector Machine (SVM), Random Forest (RF), Decision Trees (DT), Naive Bayes (NB) and AdaBoost (AB) to evaluate their performance in spam detection. These models were not chosen for their predictive ability, but to establish a baseline that allows analyzing the effects of adversary attacks in a controlled and understandable environment. As they are simple architectures, they facilitate the identification of vulnerabilities that could go unnoticed in more complex models, and allow to isolate more clearly the specific impact of disturbances at the character level. This experimental basis is key for further studies, in which it is planned to extend the analysis to more advanced NLP architectures, such as large language models (LLM) and Deep Learning systems designed to be robust to noise. The results obtained here will serve as a reference point to compare whether these architectures offer greater resilience or present similar degradations in the face of adversary attacks. To ensure robust performance estimation, all classifiers were evaluated using 3-fold cross-validation.

In the experiments, different adversarial attack methods were implemented to evaluate the robustness of the models against malicious manipulations. One of the central approaches was *DeepWordBug* [5], a black-box attack that introduces character-level perturbations. Unlike random or trivial modifications, *DeepWordBug* operates in two phases: first, it assigns a score to each token in the text using a heuristic function that estimates the relevance of each word in the decision of the model. Then select those with the highest score and apply character-level operations, such as insertion, deletion, substitution, or transposition. This strategy allows the attacker to degrade the prediction of the model without compromising human understanding and prevents simple preprocessing techniques such as tokenization or noise removal from neutralizing the attack. It should be emphasized that these alterations do not apply to any word, but to those that, according to the attack system itself, are critical for the classification of the original model.

This attack was instrumental in assessing the vulnerability of the model to adverse perturbations and laid the foundation for developing a more sophisticated scheme that combines data poisoning and backdoor attacks. Initially, the training set was contaminated with examples generated by TextAttack, applying aggregation or replacement strategies to introduce poisoning. Subsequently, unaltered clean examples from these attacks, specifically those classified as spam, were selected to insert the backdoor. In this case, the word *"ze"* served as a backdoor trigger, embedded in the middle of the text to evade detection, a location that is less likely to be altered during preprocessing and previously applied transformations. Placing the trigger at the message's beginning or end would have facilitated its identification. The final training set combined clean data, adversarial examples, and samples with the backdoor. This corpus was used to retrain the models and assess the combined impact of both types of attack on model performance and security.

Finally, to complement the analysis with interpretable knowledge, LIME was used as an XAI technique. The method was applied to selected test samples, both clean and attacked, to examine which tokens were most influential on the model's predictions. This was particularly useful for evaluating whether the introduced perturbations, especially the backdoor trigger word *"ze"*, had a measurable impact on the model decision limits. These insights helped validate whether attacks effectively manipulated the learned decision logic.

3 Results

This section presents the experimental evaluation of the robustness of the model before and after the application of adversarial attacks. The goal is not to maximize the performance of the classification, but to understand the degree of degradation that each model experiences when faced with realistic threat scenarios. First, the SVM, DT, RF, NB and AB models were trained and evaluated to analyze their behavior in the face of adverse disturbances. To do this, the data set was divided into 80% for training and 20% for testing, applying a stratified division. This technique allowed maintaining the original ratio between classes, mitigating the effect of imbalance in the data. Artificial balancing techniques were not applied since the initial objective was to observe the genuine performance of the models against imbalances inherent to the problem, especially under attack conditions, where the natural behavior of the classification is more revealing.

The hyperparameters of each model were defined from adjusted configurations by preliminary exploratory tests. At this initial stage, a systematic optimization using grid search or deep fine-tuning was not applied, since the main objective was to evaluate the general behavior of the models in the face of adverse scenarios. This decision will allow, in future works, to include a broader set of models and to apply more rigorous adjustment strategies. Specifically, they were configured as follows: *1)* For SVM, a linear kernel with C=1.0 was used; *2)* RF was adjusted with 100 estimators, a maximum depth of 10, and a random state of 42. *3)* AB was configured with 50 estimators, a learning rate of 0.5, and the same random state. *4)* DT, the Gini criterion was used, with a maximum depth of 10, min_samples_leaf = 1, and min_samples_split = 2. All models were trained using 3-fold cross-validation. Although 5 or 10 fold values are usually used in the literature [1, 7], in this case a value of $k = 3$ was chosen due to the limited size of the dataset and its unbalanced nature. This configuration allowed maintaining a representative distribution of both classes in each partition, avoiding scenarios where a class would be underrepresented during training or validation, and guaranteeing more stable and comparable evaluations between models.

The training results of the models without attacks are presented in Table 1, the first with TF-IDF vectorization and the second with BOW. To measure their performance, three metrics are used: *Accuracy*, which indicates the percentage of hits; F_1*-Score weighted*, which combines accuracy and comprehensiveness by weighting each class; and *MCC (Matthews Correlation Coefficient)*, which offers a more reliable evaluation on unbalanced data sets.

Table 1 shows that AB and NB achieved the best performance when using TF-IDF representations, achieving an *accuracy* and F_1*-Score* of 0.9944, and an *MCC* of 0.9865. AB stood out for its iterative error correction mechanism, which improves its ability to adapt to complex data patterns, while NB demonstrated solid performance due to its probabilistic approach, which proved especially effective in text classification tasks with high dimensionality and dispersed vocabulary.

Table 1. Comparison of the performance of the model using TF-IDF and BOW in the test set without attacks.

Model	TF-IDF			BOW		
	Accuracy	F1-Score	MCC	Accuracy	F1-Score	MCC
SVM	0.9935	0.9935	0.9843	0.9926	0.9925	0.9821
Random Forest	0.9842	0.9842	0.9594	0.9860	0.9860	0.9663
Decision Tree	0.9823	0.9822	0.9574	0.9805	0.9805	0.9516
AdaBoost	0.9944	0.9944	0.9865	0.9935	0.9935	0.9843
Naive Bayes	0.9944	0.9944	0.9865	0.9907	0.9907	0.9776

With the BOW representation, AB again led with an *accuracy* of 0.9935, while the DT presented the worst performance with 0.9805. These results indicate that TF-IDF benefits models that exploit the relevance of terms by weighting, while BOW favors algorithms that operate efficiently with simple word counts. To complement this initial evaluation and verify the stability of the models during the training phase, a 3-fold cross-validation was applied on the training set. Table 2 presents the *Average accuracy* values and its *Standard Deviation (Std)* for each classifier, using the TF-IDF and BOW representations. This additional validation allowed us to observe the consistency of the models' performance considering different internal partitions of the dataset.

Table 2. Average accuracy and standard deviation by cross-validation of 3 folds in the training set.

Model	TF-IDF	BOW
	Accuracy ± Std	Accuracy ± Std
SVM	0.9914 ± 0.0026	0.9723 ± 0.0015
Random Forest	0.9777 ± 0.0041	0.9746 ± 0.0037
AdaBoost	0.9881 ± 0.0020	0.9723 ± 0.0022
Naive Bayes	0.9914 ± 0.0043	0.9735 ± 0.0023
Decision Tree	0.9655 ± 0.0047	0.9723 ± 0.0035

In particular, the SVM model maintained outstanding performance with TF-IDF, while NB and AB also showed solid results, with low variability between folds. Although the BOW representation offered competitive performance, the models tended to achieve better results with TF-IDF, reaffirming its usefulness in capturing the relevance of terms in text classification tasks. Together, these results support the reliability of the models before introducing adverse perturbations.

Based on these findings, we proceeded to test the classifiers under adversarial conditions using the *DeepWordBug* attack. As previously described, this method introduces subtle perturbations, such as typographical errors, into the input text to deceive the model while maintaining the readability of the content for humans. To assess the impact, we reused the same preprocessed dataset employed in the clean evaluations, ensuring consistency in the experimental setup.

The results of the attack are presented in Table 3. Three key evaluation metrics were used: *Precision*, *Recall* and F_1*-Score*, to quantify the vulnerability of each model and the effectiveness of the adversarial strategy. In general, all classifiers showed a marked decrease in their ability to correctly identify spam messages, confirming that the attack successfully degraded their predictive performance.

Table 3. Results applying the 100% injection poisoning attack to the test set using TF-IDF and BOW.

Model	Class	TF-IDF			BOW		
		Precision	Recall	F1-Score	Precision	Recall	F1-Score
SVM	ham	0.71	1.00	0.83	0.71	1.00	0.83
	spam	0.71	0.02	0.04	0.75	0.03	0.05
Decision Tree	ham	0.71	0.99	0.83	0.92	0.05	0.09
	spam	*0.20	0.00	0.01	0.30	*0.99	0.46
Random Forest	ham	0.71	1.00	0.83	0.71	1.00	0.83
	spam	0.00	0.00	0.00	0.00	0.00	0.00
AdaBoost	ham	0.73	0.89	0.80	0.76	0.99	0.86
	spam	0.43	0.20	0.27	0.95	0.23	0.38
Naive Bayes	ham	0.77	0.94	0.85	0.79	0.07	0.13
	spam	0.69	0.33	0.45	0.29	*0.92	0.45

Among all classifiers, AB demonstrated the greatest resilience, particularly when using the BOW representation. Despite the perturbations, it maintained a relatively acceptable performance in the *spam* class, achieving a *Recall* of 0.23 and an F_1*-Score* of 0.38. Although these values are significantly lower than those obtained under clean conditions, they are notably higher than those of other classifiers, several of which were entirely failed. Furthermore, AB maintained strong performance in the *ham* class, with an F_1*-Score* of 0.86, suggesting a better ability to adapt to adversarial disturbances.

In contrast, RF was the most severely affected, with zero values in all spam-related metrics for both TF-IDF and BOW representations. This outcome indicates a complete failure to detect adversarial examples, as the classifier predicted that all inputs belong to the *ham* class. The perfect recall observed in that class therefore reflects a severe class bias and a collapse of the model's decision boundary under attack. Interestingly, DT and NB showed anomalous behavior under the BOW representation. Both achieved very high *recall* scores in the *spam* class (0.99 and 0.92, respectively), but their *precision* was extremely low (0.30 and 0.29), signaling a high rate of false positives. This implies that, although they flagged most spam instances correctly, they also mislabeled many legitimate messages likely due to confusion caused by the perturbed inputs.

A similar anomaly is observed in the TF-IDF configuration for DT, where the recall dropped to zero while maintaining non-zero precision, indicating that although a few spam instances were predicted as spam, none of them corresponded to the actual spam messages. These edge case scenarios are marked with an asterisk (*) in Table 3, highlighting extremely low *recall* or extremely low *precision* both indicative of degraded or misleading classification behavior under adversarial conditions. In conclusion, the *DeepWordBug* poisoning scheme proved highly effective, degrading the performance of all models and especially compromising their ability to detect spam, validating its impact as a targeted adversary attack technique.

Based on these results, it was decided to continue the experiments with the TF-IDF representation, given its more stable behavior against attack compared to BOW. In this new stage, a hybrid strategy that combines backdoor attack with injection poisoning will be evaluated (see Table 4) to evaluate whether this combination further degrades the integrity and detection capacity of the models in the face of simultaneous threats.

Table 4. Results comparing the injection poisoning attack with the combined attack to the test set using TF-IDF.

Model	Class	Injection Poisoning			Combined Attack		
		Precision	Recall	F1-Score	Precision	Recall	F1-Score
SVM	ham	0.71	1.00	0.83	0.71	1.00	0.83
	spam	0.70	0.02	0.04	0.83	0.02	0.03
Decision Tree	ham	0.71	0.99	0.83	0.71	0.99	0.83
	spam	*0.20	0.00	0.01	0.54	0.04	0.08
Random Forest	ham	0.71	1.00	0.83	0.71	1.00	0.83
	spam	0.00	0.00	0.00	0.00	0.00	0.00
AdaBoost	ham	0.73	0.89	0.80	0.88	0.37	0.52
	spam	0.43	0.20	0.27	0.37	*0.88	0.52
Naive Bayes	ham	0.77	0.94	0.85	0.72	0.98	0.83
	spam	0.69	0.33	0.45	0.66	0.10	0.18

The results show differentiated responses between the classifiers, allowing us to identify vulnerability and resilience patterns. NB was the most resistant, although its F_1-*Score* in spam detection fell from 0.45 to 0.18 under combined attack, while its performance in the *ham* class remained stable, evidencing a selective degradation. On the other hand, RF presented a total collapse, with *accuracy* metrics, *recall* and F_1-*Score* in spam equal to zero, reflecting its high sensitivity to adversarial disturbances. In addition, AB showed an atypical behavior: *spam recall* abruptly increased to 0.88 under the combined attack, indicating activation of the backdoor trigger. However, this apparent improvement was accompanied by a significant drop in *accuracy* for the *ham* class, revealing severe bias and a general degradation of the model.

In Table 4, the asterisks (*) indicate anomalous values, which reflect significant flaws in the predictions and not genuine improvements. In the case of DT, under the injection attack, it indicates that the model mistakenly labeled messages as spam, without identifying any real ones, suggesting a critical alteration in its decision boundary. On the other hand, AB shows a *recall* for the same class under the combined attack, a value significantly higher than that obtained with simple injection. However, this increase does not represent a real improvement, but the activation of the backdoor, which caused a massive classification of messages as spam. This behavior is accompanied by a decrease in *accuracy* and a deterioration of performance in the *ham* class, evidencing a loss of balance in classification. In both cases, the highlighted values illustrate how the attack distorts the interpretation of the model, compromising its discrimination capacity.

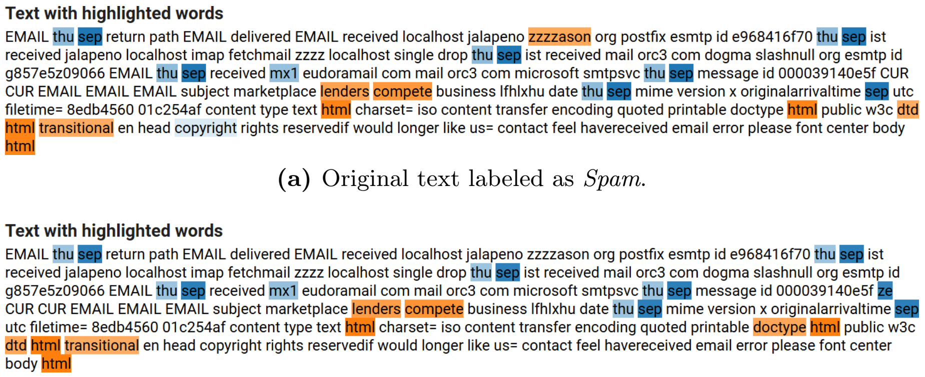
To facilitate a global comparison, Table 5 summarizes the key metrics for each classifier in three scenarios: no attack, injection poisoning, and combined attack. This overview allows for a more intuitive analysis of the extent to which each adversarial strategy degrades the performance of the model. In particular, although some degradation may seem predictable in theory, the severity and uneven effects between models reveal nuanced vulnerabilities that are critical to assess in practical applications.

Table 5. General results comparing different scenarios.

Model	Class	Metrics		
		Precision	Recall	F1-Score
No Attack	SVM	0.9950	0.9900	0.9900
	Decision Tree	0.9800	0.9800	0.9800
	Random Forest	0.9850	0.9750	0.9800
	AdaBoost	0.9900	0.9950	0.9950
	Naive Bayes	0.9950	0.9900	0.9950
Injection	SVM	0.7050	0.5100	0.4350
	Decision Tree	0.4550	0.4950	0.4200
	Random Forest	0.3550	0.5000	0.4150
	AdaBoost	0.5800	0.5450	0.5350
	Naive Bayes	0.7300	0.6350	0.6500
Combined	SVM	0.7700	0.5100	0.4300
	Decision Tree	0.6250	0.5150	0.4550
	Random Forest	0.3550	0.5000	0.4150
	AdaBoost	0.6250	0.6250	0.5200
	Naive Bayes	0.6900	0.5400	0.5050

As shown in Table 5, the combination of injection and backdoor attacks does not produce a uniformly higher degradation across all models, but it does reveal specific weaknesses that remain hidden under isolated attack conditions. For example, Naive Bayes, initially one of the most robust classifiers, experienced a marked drop in its F_1*-Score* in the combined scenario, highlighting its vulnerability to subtle manipulations. Similarly, AdaBoost showed irregular behavior, with a high *recall* that suggests backdoor activation, but accompanied by a decrease in overall *precision* indicating misclassification of legitimate messages. Random Forest, on the other hand, consistently failed to detect spam under both attack schemes, underscoring its high sensitivity to adversarial perturbations. These differentiated patterns support the notion that resilience to one type of attack does not guarantee general robustness. Consequently, this reinforces the need to evaluate classification systems in compound adversarial scenarios and integrate explainability techniques to reveal model biases and attack footprints that are not evident through accuracy metrics alone.

Since Random Forest demonstrated the most severe degradation in performance under adverse conditions in both attack schemes by not correctly identifying any spam messages, this model was selected for explainability analysis using LIME, to analyze whether the attack trigger is among the most influential words in the classification and, therefore, verify whether it directly affected the class change in the poisoned examples.



Text with highlighted words

EMAIL thu sep return path EMAIL delivered EMAIL received localhost jalapeno zzzzason org postfix esmtp id e968416f70 thu sep ist received jalapeno localhost imap fetchmail zzzz localhost single drop thu sep ist received mail orc3 com dogma slashnull org esmtp id g857e5z09066 EMAIL thu sep received mx1 eudoramail com mail orc3 com microsoft smtpsvc thu sep message id 000039140e5f CUR CUR EMAIL EMAIL EMAIL subject marketplace lenders compete business lfhlxhu date thu sep mime version x originalarrivaltime sep utc filetime= 8edb4560 01c254af content type text html charset= iso content transfer encoding quoted printable doctype html public w3c dtd html transitional en head copyright rights reservedif would longer like us= contact feel havereceived email error please font center body html

(a) Original text labeled as *Spam*.

Text with highlighted words

EMAIL thu sep return path EMAIL delivered EMAIL received localhost jalapeno zzzzason org postfix esmtp id e968416f70 thu sep ist received jalapeno localhost imap fetchmail zzzz localhost single drop thu sep ist received mail orc3 com dogma slashnull org esmtp id g857e5z09066 EMAIL thu sep received mx1 eudoramail com mail orc3 com microsoft smtpsvc thu sep message id 000039140e5f ze CUR CUR EMAIL EMAIL EMAIL subject marketplace lenders compete business lfhlxhu date thu sep mime version x originalarrivaltime sep utc filetime= 8edb4560 01c254af content type text html charset= iso content transfer encoding quoted printable doctype html public w3c dtd html transitional en head copyright rights reservedif would longer like us= contact feel havereceived email error please font center body html

(b) Attacked text with label changed to *Ham*.

Fig. 1. Comparison between relevant term changes before and after the DeepWordBug text attack in the classification using Naive Bayes.

In Figure 1, the blue highlighted words correspond to the most influential features associated with the *ham* class, while the orange highlighted words represent those related to the *spam* class. These visual cues reflect the importance scores assigned by LIME to each term. Figure 1a shows the original message that is confidently classified as spam. After applying the backdoor attack with the trigger word *"ze"* (highlighted accordingly), as shown in Figure 1b, the classification flips to non-spam with 91% confidence. This confirms that the injected trigger word *"ze"* was highly dominant in the decision of the model, effectively manipulating the classification.

To quantify this phenomenon, the *Recall@k* metric was used, showing that the trigger word was ranked among the top 10 most influential features in 97.94% poisoned instances, frequently occupying the first position. This confirms that the trigger was not ignored or neutralized by the model decision boundary; rather, it was learned and leveraged during prediction, reinforcing the backdoor effect.

These findings clarify the role of explainability in adversarial scenarios: tools such as LIME can help uncover manipulated logic paths by identifying malicious signals embedded in the model rationale. Therefore, explainability is useful not only for transparency but also as a potential early warning mechanism for security breaches. This aligns with the principles of Responsible AI, which advocate for models that are both interpretable and resilient to manipulation. Integrating interpretability into the attack evaluation process provides critical insight for building NLP systems that are not only accurate but also trustworthy and robust.

4 Conclusions

The results demonstrate that combining injection poisoning with backdoor attacks does not always lead to a uniformly greater degradation in model performance. Even models considered relatively robust, such as Naive Bayes, showed a significant deterioration in their detection ability, reflected in a noticeable reduction of their F_1-*Score* under the combined attack. This shows that the observed resilience to isolated attacks may not be sufficient when faced with multifaceted adversarial strategies.

In addition, the use of interpretable tools such as LIME facilitated the identification of the direct impact of triggers on decision making, highlighting the importance of incorporating explainability techniques to detect and mitigate these threats. Therefore, evaluating models under combined attack scenarios is crucial to designing more robust and secure systems in federated environments, anticipating vulnerabilities that could go unnoticed in simpler analyses.

Future work will explore the integration of these contradictory schemes within multimodal spam detection systems and evaluate their impact on more complex architectures, such as LLM and Deep Learning models.

References

1. Adnan, M., Imam, M.O., Javed, M.F., Murtza, I.: Improving spam email classification accuracy using ensemble techniques: a stacking approach. International Journal of Information Security **23**(1), 505–517 (2024)
2. Cheng, P., Wu, Z., Du, W., Zhao, H., Lu, W., Liu, G.: Backdoor attacks and countermeasures in natural language processing models: A comprehensive security review. IEEE Transactions on Neural Networks and Learning Systems (2025)
3. Dai, J., Chen, C., Li, Y.: A backdoor attack against lstm-based text classification systems. IEEE Access **7**, 138872–138878 (2019)
4. Ganiyu, O.: Email classification. https://www.kaggle.com/datasets/ganiyuolalekan/spam-assassin-email-classification-dataset (2021), last accessed: 2025-02-03
5. Gao, J., Lanchantin, J., Soffa, M.L., Qi, Y.: Black-box generation of adversarial text sequences to evade deep learning classifiers. In: 2018 IEEE Security and Privacy Workshops (SPW). pp. 50–56. IEEE (2018)
6. Goldblum, M., Tsipras, D., Xie, C., Chen, X., Schwarzschild, A., Song, D., Mądry, A., Li, B., Goldstein, T.: Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**(2), 1563–1580 (2022)
7. Jáñez-Martino, F., Alaiz-Rodríguez, R., González-Castro, V., Fidalgo, E., Alegre, E.: A review of spam email detection: analysis of spammer strategies and the dataset shift problem. Artificial Intelligence Review **56**(2), 1145–1173 (2023)
8. Jáñez-Martino, F., Barrón-Cedeño, A., Alaiz-Rodríguez, R., González-Castro, V., Muti, A.: On persuasion in spam email: A multi-granularity text analysis. Expert Systems with Applications **265**, 125767 (2025)

9. Li, J., Ji, S., Du, T., Li, B., Wang, T.: Textbugger: Generating adversarial text against real-world applications. arXiv preprint arXiv:1812.05271 (2018)
10. Morris, J.X., Lifland, E., Yoo, J.Y., Grigsby, J., Jin, D., Qi, Y.: Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. arXiv preprint arXiv:2005.05909 (2020)
11. Qiu, S., Liu, Q., Zhou, S., Huang, W.: Adversarial attack and defense technologies in natural language processing: A survey. Neurocomputing **492**, 278–307 (2022)
12. Sarker, I.H.: Cyberai: A comprehensive summary of ai variants, explainable and responsible ai for cybersecurity. In: AI-Driven Cybersecurity and Threat Intelligence: Cyber Automation, Intelligent Decision-Making and Explainability, pp. 173–200. Springer (2024)
13. Yavuz, A.D., Gursoy, M.E.: Injecting bias into text classification models using backdoor attacks. arXiv preprint arXiv:2412.18975 (2024)

Parallel Video Tracking System based on Honeybee Swarm Behavior and Evaluated on a Benchmark Dataset

Juan Alvaro de la Rosa-Ipiña, Carlos Soubervielle-Montalvo, Cesar Puente, Ruth Mariela Aguilar-Ponce, Luis Javier Ontañon García-Pimentel

Universidad Autónoma de San Luis Potosi, Faculty of Engineering, San Luis Potosi, México
alvarodelarosaip@gmail.com,
{carlos.soubervielle,cesar.puente}@uaslp.mx,
rmariela@fciencias.uaslp.mx,luis.ontanon@uaslp.mx

Abstract. This research aims to enhance the parallel implementation of the Zero-mean Normalized Cross-Correlation (ZNCC) algorithm for video tracking, with the goal of increasing processing speed without compromising accuracy. Speed improvements are anticipated through the integration of the bio-inspired Honeybee Search Algorithm (HSA) as an exploratory strategy within the tracking framework, using the improved ZNCC as the fitness function. The HSA algorithm has been adapted for video tracking by incorporating population management during the exploration phase, as well as the recruitment and harvesting processes. The completed tasks form the basis for implementing the parallel video tracking system more efficiently using the heterogeneous CPU-GPU architecture.

Keywords: Object Tracking, Honeybee Search Algorithm, Swarm Intelligence, Parallel Computing, Graphics Processing Unit.

1 Introduction

Video object tracking is the task of identifying a specific object across the sequence of images (or frames) that comprise a video [1], and is one of the most requested tasks in the computer vision area due to its multiple application field with unlimited potential such as medical, military and entertainment applications, among others.

For years, multiple video tracking algorithms and methodologies have been proposed to achieve this task with precision. Video tracking is a task that consists of identifying the position of an object in each frame of a video sequence [1]. However, the analysis of video frames is an exhaustive and delicate task that has not been perfected yet, in addition to entailing a great demand for time and computational resources. For this reason, efforts have been made to enhance these systems through different software and hardware resources [1,2,3,4].

Perfecting object tracking methods is important due to the potential catastrophic failures that these could generate when applied to critical systems and especially to systems where the integrity of human lives is involved, such as medical, military, or certain everyday use systems, such as autonomous driving vehicles. These failures may be due to the susceptibility of video tracking algorithms to certain obstacles commonly present in video sequences, in addition to the tracking algorithms' large computational costs and complexity that cause considerably serious delays in response times.

Video object tracking remains a challenging problem without a definitive solution. Although various approaches—such as Zero-mean Normalized Cross-Correlation (ZNCC) [2]—have been proposed, they still leave room for improvement in areas such as energy efficiency, computational resource demands, and robustness against noise, among other factors.

2 Related Work

To solve the video tracking problem, different solutions have been proposed, with the use of bioinspired metaheuristics being of special interest for this research. Algorithms inspired by nature have shown great potential as video tracking methods increasing the processing speed by reducing their resources and computational operations. Some of the most popular bioinspired metaheuristics in recent years in the video tracking research context are ABC (Artificial Bee Colony) [3], BA (Bat Algorithm) [4], and especially PSO (Particle Swarm Optimization) [5] which is one of the most used metaheuristic algorithms and has several studies and papers published about how it can be used as an optimizing algorithm for video tracking.

Another bioinspired metaheuristic is HSA (Honeybee Search Algorithm) which was proposed [2] to be used along the ZNCC video tracking algorithm as the central fitness function. An implementation that demonstrated promising results (with approximately a 30% accuracy and 13.8 FPS [6]) but with some areas of opportunity such as the FPS rate which is considerably low in this algorithm applied to video tracking and leads to deeper investigations and the proposal of this proposal. This approach is one that has not been touched on literature until now except for the cited works [2,6,7,8].

Other implementations of both the ZNCC algorithm and its original variant, NCC, have been investigated, as is the case of the initial evaluations that were made of this algorithm on the ALOV300++ dataset [9] where the NCC was shown to have an accuracy of 57%. In these tests, the Struck algorithm proved to have the highest accuracy of 66%.

In [2] it was proposed to use the HSA algorithm along ZNCC method as a possible solution to the optimization problem in video tracking problem. Tests have shown that HSA in conjunction with NCC/ZNCC can be effective and useful in a video tracking scenario. However, this implementation still requires to be perfected and tested to find the optimal conditions under which both algorithms can work more efficiently and selecting which combination of ZNCC-HSA in the video tracking obstacles works better. Even surpassing previously obtained measurements of 30% accuracy and 13.8 FPS [2,6].

Among the algorithms with which this implementation is compared, Struck and SiamMask stand out being Struck one of the most popular and well received online video tracking algorithms with the best results in accuracy, while SiamMask is perhaps the most popular deep learning algorithm in the video tracking research area [10]. Against these two trackers, the NCC – HSA implementation obtained lower but also promising results that indicated that after applying certain improvements it could be capable of obtaining comparable or even superior results.

3 Methodology

In this work, we propose the use of the evolutionary metaheuristic known as Honeybee Search Algorithm (HSA) to estimate the most probable positions of the target object within each frame. In the HSA context, each individual (bee) is assigned to a specific position (pixel) in the frame, and its suitability as the object's location is evaluated using the Zero Mean Normalized Cross-Correlation (ZNCC) image similarity metric. To guide the population of individuals toward the most promising regions of the frame, the algorithm executes a three-phase evolutionary process designed to iteratively generate and refine individuals in increasingly accurate positions.

To accelerate the aforementioned video tracking process, a heterogeneous CPU-GPU architecture is proposed. In this scheme, the decision-making processes of the HSA are executed on the CPU, while the evaluation of its fitness function—based on the ZNCC algorithm—is offloaded to the GPU. A more detailed description of this parallel architecture is provided in a later section.

3.1 ZNCC (Zero Mean Normalized Cross-Correlation)

Zero-mean Normalized Cross-Correlation (ZNCC) is a widely used similarity measure in template matching and video object tracking. It compares a reference template to regions in a target image to identify the best match, while remaining robust to variations in brightness and contrast. The ZNCC value is computed using the following equation:

$$\gamma(u,v) = \frac{\sum_{x,y}\left[I(x,y)-\bar{I}_{u,v}\right]\left[t(x-u,y-v)-\bar{t}\right]}{\sqrt{\sum_{x,y}\left[I(x,y)-\bar{I}_{u,v}\right]^2 \sum_{x,y}\left[t(x-u,y-v)-t\right]^2}}. \tag{1}$$

Where (u, v) are the coordinates of the analysis areas top left corner, I(x, y) refer to all the pixels that compose the analysis area, $\bar{t}$ is the average template value and $\bar{I}$ is the a value obtained through the sum of pixel values in an area and divided into the size of that particular area (m x n). Particularly, $\bar{t}$ and $\bar{I}$ are defined as follows:

$$\bar{t} = \frac{\sum_{x,y} t(x-u,y-v)}{m\times n}. \tag{2}$$

$$\bar{I}_{u,v} = \frac{\sum_{x,y} I(x,y)}{m\times n}. \tag{3}$$

3.2 HSA (Honeybee Search Algorithm)

The HSA algorithm is inspired by how bees search and collect food to find solutions in the most efficient possible way considering three phases which are applied to the video tracking in the following sense (see Fig. 1):

1. Exploration: the algorithm releases a group of "bees" generated by an evolutionary strategy algorithm to find the position(s) where the object is most likely to be located within the frame's search area. The ZNCC method is used to evaluate possible positions.
2. Recruitment: new bees are assigned to the areas with the best ZNCC evaluation. This is the only phase that does not use the evolutionary strategy algorithm.
3. Harvesting: the bees assigned in the previous phase go to most of the areas where the object must be located. This phase requires more computational power on the evolutionary strategy algorithm than the exploration phase since the most refined search for the object will be concentrated here.

To enhance the evaluation of candidate positions during both the exploration and harvesting phases, the ZNCC method will be optimized through thread-based parallel computing using the OpenCL library for Python. In this parallel computing environment, each bee will be assigned to an individual thread, where it will independently execute the ZNCC method to evaluate its position. This parallelization strategy is aimed at accelerating the computation of the fitness function in the HSA algorithm, which is defined by the ZNCC metric.

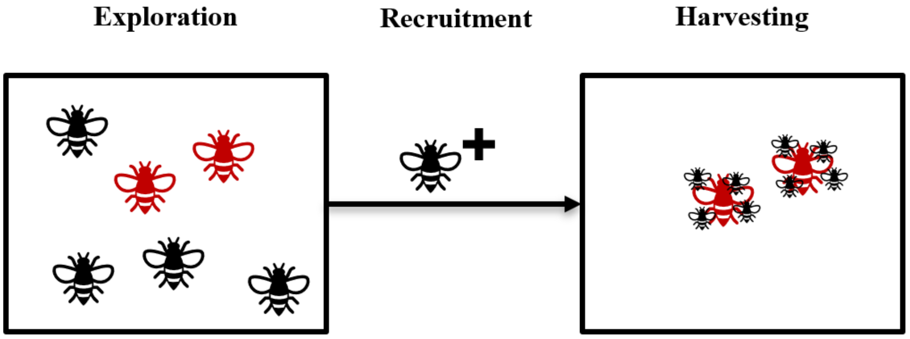


Fig. 1. HSA applied to video frame analysis. Bees look for the best positions (exploration), new bees are assigned to those positions (recruitment) and then they exploit them (harvesting).

3.3 ALOV300++ Dataset

To evaluate the system’s performance, the ALOV300++ dataset [9] will be used. This is a dataset composed of more than 300 videos divided into different categories according to the properties and obstacles they present such as occlusion, reflection, shape changes, long duration, etc. Since it was proposed in 2014, it has become very popular in video tracking research due to its diversity and easy access and even its use has been decreasing through time, it has been still used in recent years because of the previously reasons that were mentioned [2,12].

The ALOV300++ dataset represents a challenge for any video tracking system due to the robust diversity of its videos which allows us to observe in detail how a video tracker can have a better or worst perform under certain conditions and certain types of videos.

4 Use of GPU

The emergence of increasingly specialized hardware architectures and components designed to address specific computational tasks has driven the search for ways to harness their potential in current areas of computational research. An example of this is GPUs, which are an architecture designed specifically to process graphics tasks. They can serve as powerful tools, particularly in applications such as video tracking. In addition to their potential in video tracking algorithms (such as ZNCC), great potential has been observed in GPUs for optimization and parallelization algorithms such as HSA and PSO [2].

After the CPUs, the GPUs have been the most discussed hardware architecture in the video tracking field not only because of its potential in video and image processing, but also due to its possible use as a component for executing parallelization metaheuristics. However, the use of GPUs in this area still has some unexplored potential because the correct parameters that can take advantage of this architecture in the most efficient way possible have not been found yet [5,7]. Also, the GPUs' advantages over other architectures such as the CPU or FPGA have not been discovered with certainty yet neither [8].

To accelerate the entire ZNCC – HSA system (see Fig 2.), we seek to take advantage of GPUs, and their specific components designed for the graphic resources processing that can also be used to improve the parallel programming in HSA.

5 System Evaluation

The complete evaluation of the system is sought by searching and implementing the appropriate metrics such as F1-Score and FPS (Frames Per Second), as well as other specific measures used for the correct video tracking system evaluation.

FPS (Frames Per Second). The frames per second (FPS) measurement is what we are going to use to measure the speed of the system to evaluate how many frames the system can analyze in one second of execution and confirm whether or not the objective of the system reaching the 15 FPS.

F1-Score. To measure accuracy, we will use the measure known as F1-Score [11], also known as the Sørensen-Dice Coefficient, a variation of the F-Score metric that is defined by the following equation:

$$\mathrm{F} = (1+\beta^2) \times \frac{\text{Precision} \times \text{Recall}}{(\beta^2 \times \text{Precision}) + \text{Recall}}. \tag{4}$$

where β is a factor that determines the weight that Precision and Recall will have in the result. Precision and Recall are obtained through:

$$\text{Precision} = \frac{TP}{TP+FP}. \tag{5}$$

$$\text{Recall} = \frac{TP}{TP+FN}. \tag{6}$$

where TP = True Positives, FP = False Positives and FN = False Negative, and these are obtained through the IoU (Intersection Over Union) measure.

F1-Score is a F-Score variant where β = 1, which gives equal weight to both Precision and Recall resulting in a balanced result where both metrics have the same weight. When we substitute the β value in the original F-Score equation we get the next F1-Score equation:

$$\text{F1} = 2 \times \frac{\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}. \tag{7}$$

6 Improvements

As mentioned in section 2, Dr. Oscar Perez-Cham proposed an original implementation of the HSA algorithm in a parallel approach applied to video object tracking [2]. This implementation was programmed in C code language.

A review and execution on this previous C code showed some areas of opportunity that could be taken advantage of to improve the implementation:

- Lack of modularity and flexibility in the code.
- Better use of the feedback provided by HSA.
- Taking advantage of the optimized and specialized libraries of a programming language such as Python that is higher level than C and C++.
- Better use of the GPU and CPU.

6.1 Modular Code Implementation

The development of a new version of the ZNCC - HSA implementation through a modular approach is expected to take advantage of the multi-paradigm property of Python language as well as the optimization of its specialized libraries to create an optimally flexible and easily manageable system.

The system is designed to focus on a core autonomous module (see Fig. 3) that would only be responsible for executing the video sequences frame by frame in addition to coordinating the work of the rest of the modules that would oversee more specific tasks such as the tracking module. which would oversee tracking the object itself and where the ZNCC code would be implemented. Similar modules or modules that are destinated to achieve the same function will be organized into packages which in addition to keeping the code organized, will allow an easy export and integration of those modules inside and outside the system.

Besides the core module, the only other independent module would be the testing module, as it is intended to be a system capable of running and calculating different metrics without the need to rely on the runtime analysis of the main module. Instead, this module could receive results files and analyze them after the execution of the main video tracking system to subsequently calculate metrics such as F1-Score. Still, this test

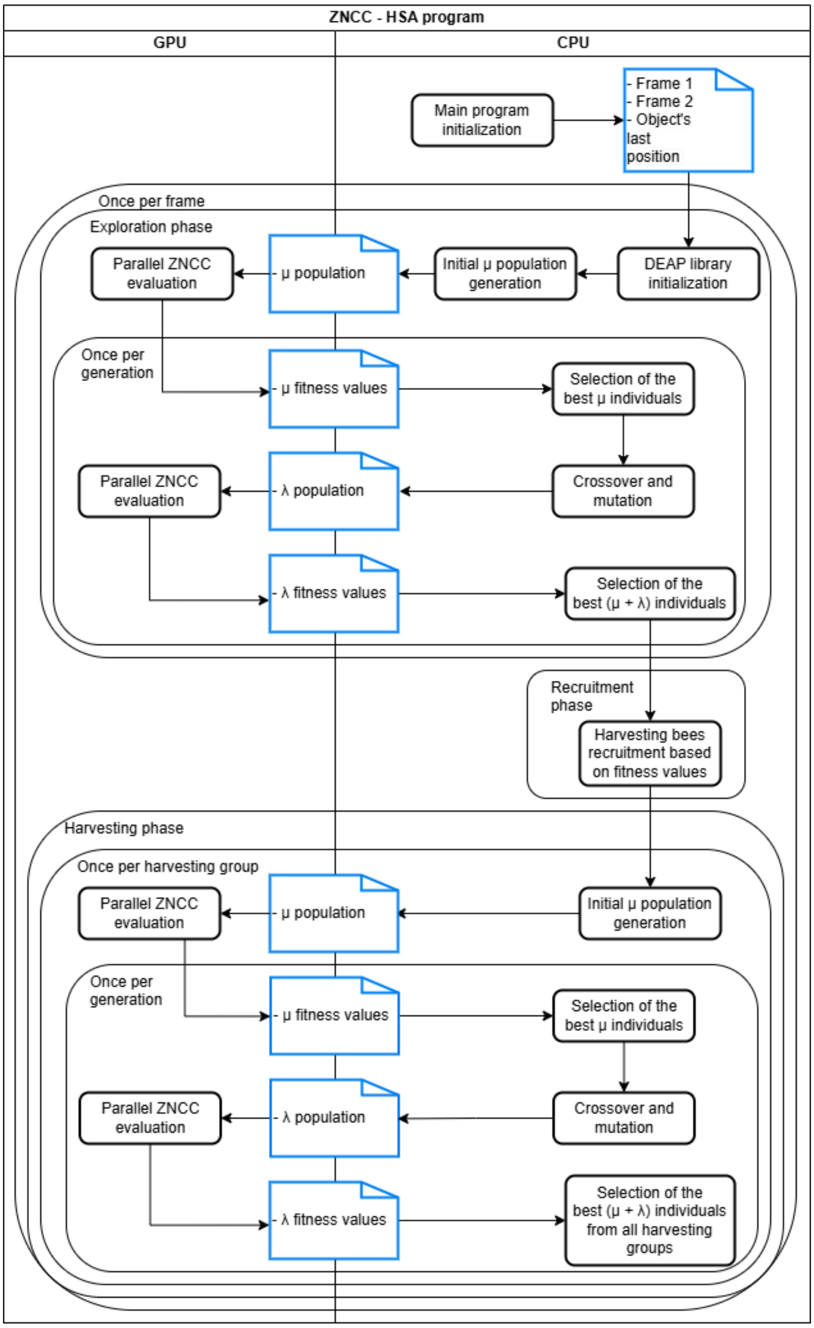


Fig. 2. Description of the task distribution on the proposed heterogeneous CPU-GPU architecture.

module could be integrated into the core module to calculate the necessary metrics at runtime.

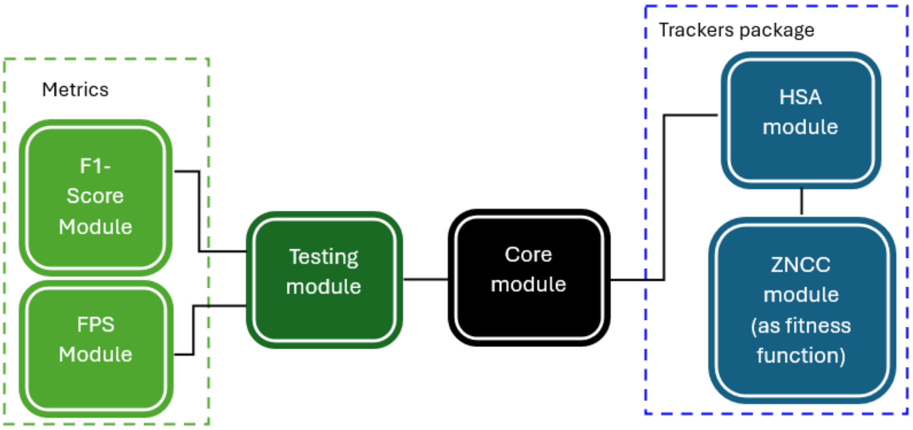


Fig. 3. Brief look at the current modular distribution of the system.

6.2 ZNCC Algorithm Improvements

As said, the overall accuracy of ZNCC performance in frame-by-frame analysis needs to be improved because it is still not 100% perfect, especially in certain types of videos that present some specifically difficult obstacles that cause the system to obtain a low accuracy percentage.

The ZNCC algorithm required improvements to increase its precision in the frame-by-frame video analysis since it has not reached 100% accuracy and in the case of some video types, it still obtained a quite low percentage. The purpose of these improvements is to reach approximately 62% precision, which is above the results obtained so far with the ZNCC algorithm [9].

Random coordinates. It was observed that the parallel execution of the ZNCC algorithm involved an exhaustive search using 100% of the frame's pixels, leading to redundancies when analyzing similar areas, which increased computational cost and caused issues like memory overflow especially with high-resolution videos.

Based on this observation, it was proposed that by analyzing less than 100% of the pixels, selected randomly and uniformly across the frame, redundancies could be reduced. According to the Pareto principle [13], an optimal percentage of random coordinates is estimated to be 20%.

Search area reduction. Based on the assumption that an object cannot undergo drastic displacement between consecutive video frames, the search area for object tracking was significantly reduced. Instead of scanning the entire frame, the algorithm now restricts the search to a localized region around the object's previous position where it is most likely to appear in the next frame. This optimization reduces computational load by eliminating unnecessary searches and improves tracking accuracy by concentrating on the most probable location of the object [14].

The reduction of the search area is guided by four directional percentages—one for each side of the region—determined by the Honeybee Search Algorithm (HSA). These percentages indicate the likelihood of the object's movement in each direction, allowing the search to focus more heavily on the areas where the object is most likely to appear. To constrain the search space, the maximum search radius is limited to three times the object’s size, with the directional percentages calculated relative to this radius (see Fig. 4).

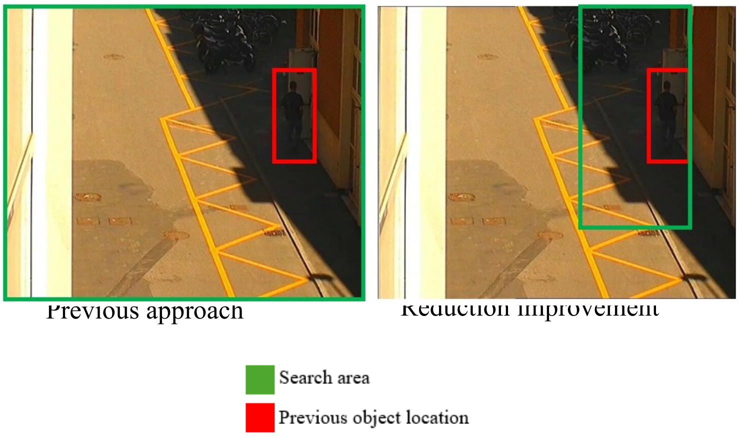


Fig. 4. Change on the search area with the new reduction improvement using four predetermined percentages based on 3 times the object size radius.

Blinds method. This is a variant of the pre-existent down-sampling method [15]. It is based on the idea that an object can still be distinguished even if there is low interference in the image. A change was implemented in the ZNCC algorithm code to the cycles that traverse the image pixels, so instead of moving pixel by pixel, a jump of two pixels is made and the system only process half of the frame's information, similar to watching the image pass through blinds (see Fig. 5)

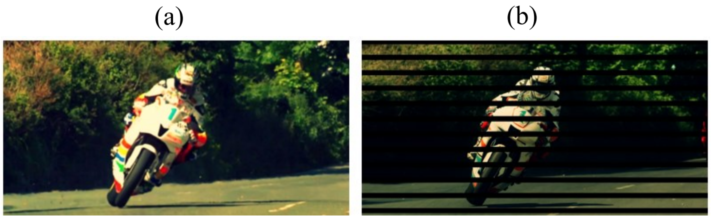


Fig. 5. The idea behind the blinds method. a) Original frame, b) same frame after applying the blinds method.

Although the mathematical results of the ZNCC equation vary when removing half of the pixels from the analysis, when applying the same blinds algorithm to both frames (the current and the previous one), the statistical relationship between these two is maintained, something that has been corroborated in experiments and tests where it is observed that there is no noticeable decrease in the effectiveness of the ZNCC algorithm tracking the object when this method is applied.

6.3 HSA Improvements

The HSA algorithm Python implementation is currently in development. The purpose of using the HSA metaheuristic is to make the system as fast as possible (even trying to reach 15 fps) by taking advantage of the specific HSA heuristic properties combined with the ZNCC video tracking process. Once completed, the HSA algorithm is expected to decide where to follow the object taking advantage of the previously mentioned ZNCC improvements.

The HSA is going to work efficiently by using its three phases strategically: the exploration phase is applied only in the first frame to locate the object, while the recruitment and harvesting phases handle tracking in subsequent frames. The exploration phase is triggered again only if the object is lost, based on a predefined threshold—expected to be 62% precision (see Fig. 6).

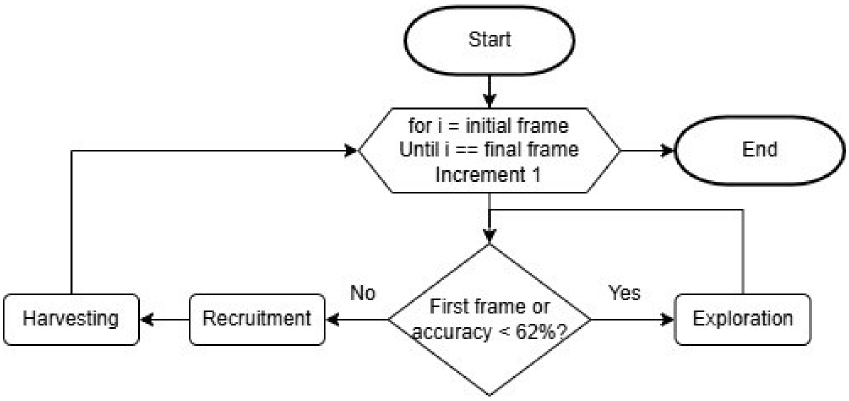


Fig. 6. General process of the HSA algorithm through frames and how its different phases are executed according to the current precision.

The previously explained usage of the frame-to-frame feedback will help not only increase the success possibility of finding the object in a frame but also make it faster.

7 Experiments and Results

The current implementation of the HSA algorithm with just the exploration phase has been tested on 33 videos coming from the 14 ALOV300++ dataset categories. These tests were performed to find the optimal mutation and crossover operators for the algorithm considering HSA's original implementation was made to 3D reconstruction and its parameters were not changed in Dr. Perez-Cham's video tracking implementation [5].

Combinations of three crossover and mutation operators were implemented. The crossover operators that were selected are: uniform crossover, blend crossover and simulated binary crossover (SBX), while the mutation operators are: uniform mutation, polynomial mutation and gaussian mutation. The following graphs show the average results achieved on these experiments (see Fig. 7).

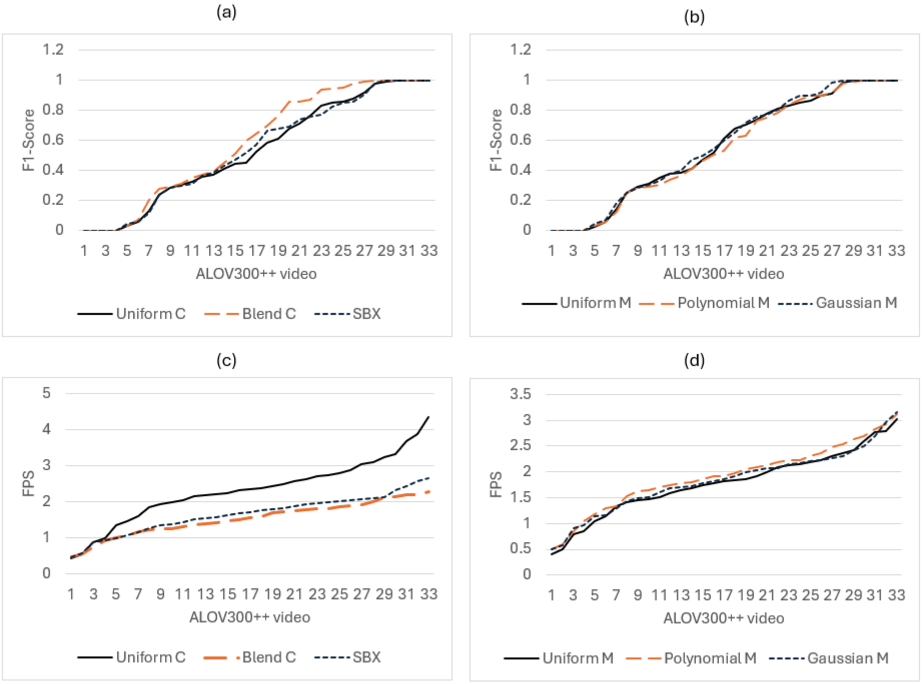


Fig. 7. a) F1-Score (accuracy) average behavior of the three crossover operators, b) F1-Score average behavior of the three mutation operators, c) FPS (speed) average behavior of the three crossover operators, d) FPS average behavior of the three mutation operators.

8 Conclusions

The final implementation is still under development; however, preliminary tests demonstrate significant improvements in both speed and accuracy compared to the original implementation presented in [5].

As mentioned above, the HSA is currently being fine-tuned to identify the most effective evolutionary operators. Based on the preliminary results, blend crossover and Gaussian mutation are currently considered the best candidates; however, further analysis is ongoing to confirm these findings.

Once the best operators are found, a basic version of the HSA algorithm will be available, in which optimization improvements will be implemented. In parallel, tests will be conducted on the entire ALOV300++ dataset.

The final version of HSA will incorporate all the improvements presented in this work; however, certain limitations remain. The most significant of these is the algorithm's limited robustness to changes in object size, which represents the main weakness of the current proposal. Addressing this limitation constitutes the primary direction for future work, along with enhancing the algorithm's resilience to object rotation, resilience to occlusion and improving computational efficiency. The latter may involve further code optimization or the development of a new implementation in C++ or another compiled language.

References

1. Maggio, E., Cavallaro, A.: Video Tracking: Theory and Practice. John Wiley & Sons (2011)
2. Chen, W.: Simulation of Multimedia Visual Image Motion Track Marking Based on Artificial Bee Colony Algorithm. Mathematical Problems in Engineering 1, 1–10 (2022). https://doi.org/10.1155/2022/8039589
3. Nenavath, H., Jatoth, R.K., Das, S.: A synergy of the sine-cosine algorithm and particle swarm optimizer for improved global optimization and object tracking. Swarm and Evolutionary Computation 43, 1–30 (2018). https://doi.org/10.1016/j.swevo.2018.02.011
4. Gao, M.L., Shen, J., Yin, L.J., Liu, W., Zou, G.F., Li, H.T., Fu, G.X.: A novel visual tracking method using bat algorithm. Neurocomputing 177, 612–619 (2016). https://doi.org/10.1016/j.neucom.2015.11.072
5. Perez-Cham, O.E., Puente, C., Soubervielle-Montalvo, C., Olague, G., Aguirre-Salado, C.A., Nuñez-Varela, A.S.: Parallelization of the honeybee search algorithm for object tracking. Applied Sciences 10(6), 2122 (2020). https://doi.org/10.3390/app10062122
6. Soubervielle-Montalvo, C., Perez-Cham, O.E., Puente, C., Gonzalez-Galvan, E.J., Olague, G., Aguirre-Salado, C.A., Cuevas-Tello, J.C., Ontanon-Garcia, L.J.: Design of a Low-Power Embedded System Based on a SoC-FPGA and the Honeybee Search Algorithm for Real-Time Video Tracking. Sensors 22(3), 1280 (2022). https://doi.org/10.3390/s22031280
7. Perez-Cham, O.E., Puente, C., Soubervielle-Montalvo, C., Olague, G., Castillo-Barrera, F.E., Nunez-Varela, J., Limon-Romero, J.: Automata design for honeybee search algorithm and its applications to 3D scene reconstruction and video tracking. Swarm and Evolutionary Computation 61, 100817 (2021). https://doi.org/10.1016/j.swevo.2020.100817
8. López, V.A.M., Montalvo, C.S., Varela, A.S.N., Cham, O.E.P., Galván, E.J.G.: A Review of Design Methodologies and Evaluation Techniques for FPGA-Based Visual Object Tracking Systems. International Journal of Combinatorial Optimization Problems and Informatics 15(5), 127–145 (2024). https://doi.org/10.61467/2007.1558.2024.v15i5.571
9. Smeulders, A.W., Chu, D.M., Cucchiara, R., Calderara, S., Dehghan, A., Shah, M.: Visual tracking: An experimental survey. IEEE Trans. Pattern Anal. Mach. Intell. 36(7), 1442–1468 (2013). https://doi.org/10.1109/TPAMI.2013.230
10. Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. ACM Computing Surveys 38(4), 13 (2006). https://doi.org/10.1145/1177352.1177355
11. Christen, P., Hand, D.J., Kirielle, N.: A review of the F-measure: its history, properties, criticism, and alternatives. ACM Computing Surveys 56(3), 1–24 (2023)

12. Singh, U., Saini, A., Domala, R.: Object Tracking in Videos Using CNN. In: ICDSMLA 2019: Proc. of the 1st Int. Conf. on Data Science, Machine Learning and Applications, pp. 520–527. Springer, Singapore (2020)
13. Macek, K.: Pareto principle in datamining: an above-average fencing algorithm. Acta Polytechnica 48(6), 55–59 (2008)
14. Kumar, A.R., Ravindran, B., Raghunathan, A.: Pack and detect: Fast object detection in videos using region-of-interest packing. In: Proc. of the ACM India Joint Int. Conf. on Data Science and Management of Data, pp. 150–156 (2019)
15. An, E.B., Kim, A., Jung, S.H., Kwak, S., Lee, J.Y., Cheong, W.S., Seo, K.D.: Adaptive spatial down-sampling method based on object occupancy distribution for video coding for machines. EURASIP Journal on Image and Video Processing 36, 1–17 (2024)

Updating and Evaluating the Struck Tracker: A Comparative Study on the ALOV300++ Benchmark

Víctor Alejandro Méndez-López, Carlos Soubervielle-Montalvo, Cesar Puente-Montejano, Rafael Peña-Gallardo, Emilio Jorge González-Galván

Autonomous University of San Luis Potosi, Faculty of Engineering, San Luis Potosi, Mexico
a242881@alumnos.uaslp.com,{carlos.soubervielle,cesar.puente, rafael.pena,egonzale}@uaslp.mx

Abstract. Visual Object Tracking (VOT) is focused on tracking moving objects across sequences of video frames. A VOT system consists of an integrated framework that combines models, algorithms, and hardware to ensure robust tracking performance despite changes in environmental conditions, lighting, object appearance, and other factors. This work presents the modification and evaluation of the Struck tracker to ensure compatibility with the OpenCV 4.9 library, as the original implementation was developed using OpenCV 2.4. This update is reported as a first step toward the implementation of the Struck algorithm on a SoC-FPGA platform as part of an ongoing Ph.D. research project. A series of performance experiments were conducted using all categories of the ALOV300++ benchmark dataset to compare the updated Struck tracker against three baseline OpenCV-based trackers: CSRT, MIL, and KCF. Experimental results demonstrate that the updated Struck tracker achieved the highest F1-score (0.720 ± 0.122) and precision (0.647 ± 0.139) among the evaluated algorithms, with a processing speed of 77.1 FPS. These findings highlight the trade-offs between tracking accuracy and computational efficiency, and demonstrate the feasibility of updating legacy tracking code for use with modern computer vision frameworks and datasets.

Keywords: Visual Object Tracking, Struck Tracker, ALOV300++.

1 Introduction

Visual object tracking (VOT) is defined by the analysis of video sequences for the purpose of establishing the location of the target over a sequence of frames starting from the bounding box given in the first frame [1, 2]. It is an active research field with relevant applications in different domains, including surveillance, robotics, autonomous vehicles, and augmented reality [3]. At the same time, it is a challenging task due to the computational resources needed to keep a real-time response while dealing with a series of diverse factors like complex object shapes, irregular movements, scene illumination changes, and object occlusion, among others.

A variety of paradigms, design methodologies and hardware platforms have been proposed in the literature to address the design complexity of efficient VOT systems

[4, 5]. Among these, the statistical reference model introduced by Smeulders et al. [1] stands out for emphasizing the importance of incorporating predictive models—particularly for motion estimation—rather than relying solely on appearance-based representations. This perspective has driven the development of adaptive tracking mechanisms, which continuously update model parameters and internal representations to maintain robust tracking performance under dynamic conditions [6].

Building on this need for adaptability, two prominent paradigms have emerged in the literature: (a) tracking-by-detection, in which tracking is formulated as a repeated detection problem: in each frame, a classifier is trained or updated to detect the target object in the subsequent frame, typically within a predefined search region; and (b) correlation filter-based tracking, in which tracking is achieved by learning a discriminative filter to locate the target object by measuring similarity (correlation) between a target template and candidate regions in subsequent frames[7]. These paradigms reflect different strategies for incorporating temporal changes into the tracking model.

Within this framework, a subset of VOT algorithms known as online learning has proven especially well-suited for large-scale and non-stationary tracking tasks, as it processes data sequentially and updates models in real time. However, despite its adaptability and predictive capabilities, the kernelized variant of this approach incurs significant computational costs, primarily due to the overhead associated with maintaining and updating kernel-based structures-posing notable challenges in resource-constrained environments, such as embedded systems or FPGA-based platforms[8, 9].

In this context, a subset of representative VOT algorithms has been selected to support the objectives of this study. These include Struck, CSRT, MIL, and KCF, each embodying distinct approaches to model updating, feature representation, and computational complexity. The Struck algorithm combines structured support vector machines with kernel-based methods for tracking. It formulates tracking as a structured output prediction problem and learns a discriminative tracker by leveraging appearance and motion cues[10]. The CSRT algorithm integrates channel and spatial reliability to enhance robustness against occlusions and background clutter. It employs a kernelized correlation filter approach in a multi-channel environment to achieve accurate object tracking [11]. The MIL algorithm models tracking as a binary classification problem using multiple instance learning. It selects the most probable positive instance within a set of candidate instances to track the target object robustly under varying conditions [12]. The KCF algorithm uses kernel methods to learn correlation filters in the Fourier domain. It efficiently computes the correlation between the target object and candidate patches in each frame, enabling real-time tracking with high accuracy [13]. All selected trackers were evaluated using the One-Pass Evaluation (OPE) protocol, with average Precision, F1-score, and Frames Per Second (FPS) computed for each video sequence, as defined in the ALOV300++ dataset by Smeulders et al.[1].

This work presents a series of comparative experiments involving four OpenCV-based trackers—Struck, CSRT, MIL, and KCF—aimed at assessing and discussing their relative tracking performance under diverse visual conditions. To enable this evaluation, the original implementation of the Struck tracker was updated for compatibility with both the OpenCV 4.9 library and the ALOV300++ benchmark dataset.

The workstation-based evaluation of these algorithms, as reported in this study, provides a foundational basis for ongoing research into their feasibility and performance on resource-constrained SoC-FPGA platforms, such as the Xilinx ZC706 development board.

The remainder of this document is structured as follows: Section 2 presents a brief review of related work. Section 3 outlines the core principles of the Struck tracker and the characteristics of the ALOV300++ dataset. Section 4 provides a detailed explanation of the methodology employed in this study. Section 5 presents and discusses the results of the conducted experiments. Finally, Section 6 outlines the conclusions and future directions of this review.

2 Related Work

Several studies have explored the evaluation and enhancement of the Struck algorithm in comparison with traditional and modern tracking paradigms, including tracking-by-detection, correlation filter-based approaches, and recent deep learning models, using diverse benchmark datasets and feature representations.

Adamo et al.[14] analyze the TLD (Tracking-Learning-Detection) and Struck algorithms, originally developed using Fern and Haar visual features, respectively. Their study aims to evaluate the performance of these trackers when standard descriptors are replaced with local feature representations, including Local Binary Patterns (LBP), Local Gradient Patterns (LGP), and Histogram of Oriented Gradients (HOG). The authors observe that Struck's structured SVM, enhanced by kernel mapping, performs particularly well with LBP, LGP, and HOG, thereby improving its adaptability in dynamic tracking scenarios. The comparative experiments were conducted using the PETS2009 dataset, a benchmark commonly used for multi-camera tracking. The results on PETS2009 suggest that Struck, when equipped with robust local descriptors, generalizes effectively—and could potentially benefit even further when evaluated on larger and more challenging datasets such as ALOV300++.

Wang et al.[15] established theoretical connections between two state-of-the-art correlation filter (CF) trackers—Spatial Regularization Discriminative Correlation Filter (SRDCF) and Correlation Filter with Limited Boundaries (CFLB)—as well as the structured output tracker Struck. While the theoretical aspects of their work fall outside the scope of this study, it is important to note that their findings were supported by extensive experiments using the OTB50 and OTB100 benchmark datasets. Nevertheless, alternative benchmarks such as ALOV300++ offer a complementary evaluation environment with greater category diversity and complexity, making them particularly valuable for assessing tracking performance in more diverse and realistic visual scenarios.

Finally, with respect to the performance of Struck in comparison to more recent deep learning-based trackers, Minimol et al.[16] conducted comprehensive benchmarking across three widely recognized datasets: ALOV300++, OTB (Online Tracking Benchmark), and VOT Challenge. In their study, Struck was employed as a classical non-deep learning baseline due to its well-established reliability and structured SVM-based framework. The evaluation was carried out using standard performance metrics, including the F-score, One-Pass Evaluation (OPE), robustness, and accuracy. It is relevant to highlight that, while the results demonstrated that Guided MDNet significantly outperformed Struck across all major evaluation criteria—particularly in reducing failure rates and maintaining consistent target localization under appearance variations— this study is referenced solely for its contextual relevance regarding the use of Struck and the ALOV300++ dataset. The use of deep learning models for VOT systems falls outside the scope of the present work.

3 Materials and Methods

The current methodology was developed with the primary objective of studying the public-domain implementation of the Struck tracker by Hare et al.[10], in order to evaluate its potential for deployment on a workstation platform. This evaluation represents the first phase of a broader research effort, with a future stage dedicated to implementing and optimizing the tracker on a resource-constrained SoC-FPGA platform.

The following main steps were undertaken as part of the reported methodology:

- Update of the Struck tracker (Section 3.1)
- Development of the ALOV300++ evaluation protocol (Section 3.2)
- Performance evaluation of Struck performance by category (Section 4.1)
- Comparative performance evaluation of studied trackers (Section 4.2)
- Overall performance analysis and discussion (Section 4.3)

3.1 Update of the Struck Tracker

The original Struck algorithm implementation was developed using OpenCV 2.4. For this study, the codebase was updated to ensure compatibility with OpenCV 4.9. This migration required a series of modifications due to significant changes in the OpenCV API across versions. The two most critical updates are detailed below:

a) Replacement of the IplImage structure with cv::Mat:. In OpenCV 2.4, image data was handled using the IplImage structure—a legacy C-style data type inherited from the Intel Image Processing Library. In contrast, OpenCV 4.9 uses cv::Mat, a modern C++ class that offers superior memory management, object-oriented design, and improved compatibility with contemporary OpenCV functions. Given that IplImage is now deprecated, all instances of it in the source code were replaced with cv::Mat, and all associated image-handling routines were updated accordingly.

b) Update of core OpenCV header file references. The original implementation included legacy headers such as opencv/cv.h and opencv/highgui.h, which are no longer supported in recent OpenCV versions. These were updated to their modern equivalents: opencv2/core/core_c.h and opencv2/highgui/highgui_c.h, respectively. These headers are essential for accessing OpenCV's core data structures and

GUI functionalities and ensure compatibility with OpenCV's modular architecture introduced in version 2.4 and formalized in later releases.

The update process also required configuring the CMake build environment to ensure compatibility with the updated OpenCV libraries. These configurations enabled successful compilation and execution of the tracker across different operating systems.

3.2 Implementation of the ALOV300++ Evaluation Protocol

A conventional one-pass evaluation (OPE) protocol was developed in Python to manage the workflow of dataset evaluation. The One-Pass Evaluation (OPE) protocol is the standard evaluation strategy employed in ALOV300++ to assess tracker performance on a per-sequence basis without reinitialization. This approach simulates a real-world tracking scenario in which the tracker must operate autonomously, without any manual intervention.

The following steps outline the development process for implementing the standard One-Pass Evaluation (OPE) protocol for the Struck tracker. Once the procedure was established, only minimal modifications were required to adapt the protocol for the other trackers evaluated in this study.

a) Preliminary analysis of the original struck evaluation procedure: A preliminary review of the original Struck implementation was carried out to understand its evaluation procedure with the OTB dataset. The main observation focused on how annotation files were processed to compute the Intersection over Union (IoU) metric. The OTB dataset provides ground truth annotations on a frame-by-frame basis, enabling straightforward calculation of per-frame IoU. In contrast, the ALOV300++ dataset includes annotations at fixed intervals, which vary across sequences, thereby requiring additional preprocessing to align predictions and annotations for consistent evaluation.

b) Analysis of ground truth structure and distribution in the ALOV300++ dataset: A detailed analysis was conducted on all 314 sequences of this dataset to assess the structure and distribution of the ground truth annotations. It was found that 39 sequences use a different annotation sampling interval—specifically, annotations spaced every five frames—consistent with what is reported by the original dataset authors. This irregularity further emphasizes the need for standardized preprocessing to ensure uniform metric computation across all sequences.

c) Standardization of ground truth format: All annotation files were reviewed and, when necessary, converted to the widely adopted (Xmin, Ymin, Width, Height) format to ensure compatibility and consistency across evaluation routines.

d) Definition of evaluation sequences file: A control CSV file was generated to specify the subset of sequences to be analyzed, including the initial and final frame indices for each sequence.

e) Tracker Execution Automation: A Python script (OPEtest.py) was developed to automate the execution of the tracker across the sequences and frame ranges specified in the control CSV file. For each evaluated sequence, the script generates two output files: one containing the bounding boxes produced by the tracker, and another recording the processing time required to generate each output per frame.

f) Post-experiment performance analysis: A second Python script (CategoryAnalysis.py) was developed to compute the performance metrics—F1-score, precision, and frames per second (FPS)—for all trackers and sequences specified in OPEtest.py. In addition to numerical evaluation, the script also generates the corresponding plots for visual analysis.

Although the implementation supports evaluation on any subset of sequences, this study considered the complete set of sequences across all categories for each of the aforementioned trackers.

3.3 The Struck Tracker

The publication "Struck: Structured Output Tracking with Kernels" by Sam Hare et al.[10] represents a seminal contribution to the field of VOT. This work introduced an adaptive tracking-by-detection framework based on structured output prediction, employing a kernelized structured output support vector machine (SVM) to directly model the relationship between candidate image regions and their corresponding object locations[17]. By reformulating visual tracking as a structured output prediction problem, the authors eliminate the need for an explicit intermediate classification step—such as training a separate binary classifier—thereby achieving a more integrated, end-to-end framework that is both theoretically robust and practically effective.

The key properties of the Struck algorithm include (see Figure 1):

a) the use of an SVM-based structured predictor to capture the contextual relationship between the target and its surroundings.

b) the application of kernel methods to address non-linearities in object representation.

c) a budgeting mechanism that constrains the growth of support vectors, enhancing computational efficiency during online operation.

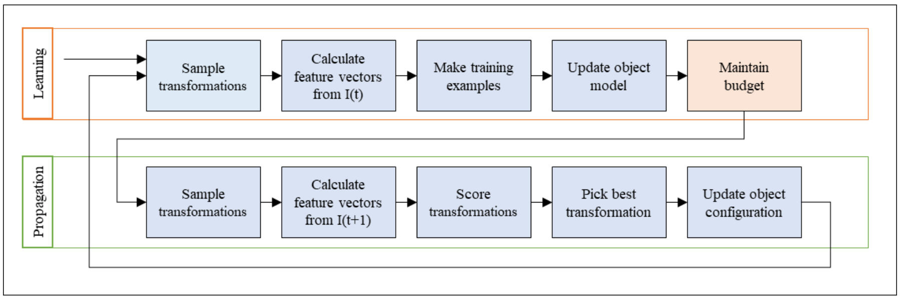


Fig. 1. Functional representation of the Struck tracker (modified from [10]).

Struck's main advantages include its online learning capability for real-time adaptation, computational efficiency through support vector budgeting, and robustness to appearance variations like illumination changes and deformation[10]. However, it is sensitive to partial occlusions, as it continues to update its model based on potentially corrupted inputs. This can lead to a well-known issue in tracking called drift, where the model gradually shifts away from the true target due to erroneous updates over time[18].

3.4 The ALOV300++ Benchmark Dataset

The Amsterdam Library of Ordinary Videos (ALOV300++) benchmark dataset, introduced by Smeulders et al.[1], consists of 314 short video sequences organized into 14 challenge categories, primarily sourced from real-world YouTube footage. Designed to reflect diverse visual conditions—such as occlusion, motion blur, illumination changes, and background clutter—the dataset facilitates detailed evaluation of tracker performance under specific challenges. Its realism is reinforced by the inclusion of authentic video artifacts like compression noise and dynamic lighting variations. ALOV300++ comprises a large number of annotated frames, with ground-truth bounding boxes provided every fifth frame for most sequences, and is consistent with other standard benchmarks in the field [19].

The short sequences average 9.2 seconds in length, with a maximum of 35 seconds, and an additional category includes ten longer videos ranging from one to two minutes, enhancing its temporal coverage and evaluation flexibility. Table 1 presents a structured overview of the ALOV300++ dataset, detailing the number of video sequences, available raw frames, and annotated frames per category. The dataset comprises 314 video sequences, yielding a total of 151,657 raw frames and 16,337 annotations.

Table 1. The ALOV300++ dataset.

Category	Sequences	Available raw frames	Annotations
01-Light	33	17,789	1,321
02-SurfaceCover	15	7,118	638
03-Specularity	18	6,960	916
04-Transparency	20	5,284	815
05-Shape	24	10,801	1,133
06-MotionSmoothness	22	10,546	636
07-MotionCoherence	12	8,734	409
08-Clutter	15	8,013	696
09-Confusion	37	11,554	1,178
10-LowContrast	23	7,675	1,036
11-Occlusion	34	13,442	1,371
12-MovingCamera	22	8,154	1,025
13-ZoomingCamera	29	8,692	1,168
14-LongDuration	10	26,895	3,995
Total	314	151,657	16,337

3.5 Evaluation Metrics

Following, metrics used in this work are indicated: the frames per second (FPS) metric measures the number of frames a tracking system can process in one second. It reflects the real-time performance or speed of the tracking algorithm. Higher FPS indicates better suitability for real-time applications, such as autonomous navigation or live surveillance.

IoU is a fundamental metric defined as the ratio of the area of intersection to the area of union of the predicted bounding box and the ground truth bounding box [20, 21]. This metric provides an intuitive measure of localization accuracy, with values ranging from 0 (no overlap) to 1 (perfect overlap). It is often used as a threshold criterion (usually defined at 0.5) to determine whether a detection is considered a true positive (see Figure 2).

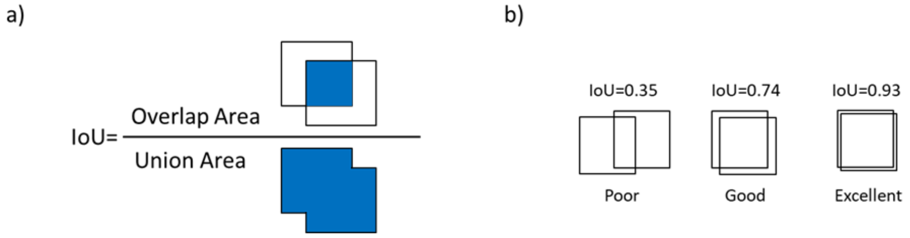


Fig. 2. Intersection over union (IoU) metric. a) IoU is calculated as the ratio between the intersection area and the union area of the ground truth and the detected bounding box. b)IoU examples (image modified from [21]).

Precision, Recall and F1-score are related metrics. Precision quantifies the proportion of predicted bounding boxes that are correct, while recall quantifies the proportion of ground truth objects that are successfully detected. The F1-score combines both precision and recall into a single metric by computing their harmonic mean, providing a balanced assessment of detection performance, especially when there is an uneven trade-off between false positives and false negatives. In the context of the ALOV300++ dataset, recall is typically considered to be 1, as each sequence involves tracking a single object with continuous ground truth annotations, and standard evaluation protocols assume that the tracker outputs predictions for all frames, eliminating the presence of false negatives[1].

4 Experiments and Results

A series of One-Pass Evaluation (OPE) experiments were conducted to assess the performance of the Struck tracker in comparison to the CSRT, KCF, and MIL trackers. The experiments were executed on a Whitebox workstation equipped with an Intel Core i5 processor (3.50 GHz), 32 GB of RAM, and running the Ubuntu 22.04 operating system. For all experiments involving the Struck tracker, the default configuration file provided with the original implementation was used.

4.1 Struck Performance Evaluation

For this experiment, the Struck tracker was executed on all 314 sequences of the ALOV300++ dataset. Precision and F1-score were evaluated on a per-category basis, with recall assumed to be 1.0 in all cases. The evaluation of the Struck tracker on the complete ALOV300++ dataset yielded a global precision of 0.650 ± 0.134 and an F1-score of 0.722 ± 0.118, indicating a generally robust performance across diverse tracking scenarios.

Notably, the highest tracking accuracy was achieved in the 08-Clutter category, where the tracker attained a precision of 0.905 ± 0.212 and an F1-score of 0.932 ± 0.162, suggesting strong resilience to background clutter and visual distractions. Conversely, the lowest performance was observed in the 14-LongDuration category, with a precision of 0.394 ± 0.395 and an F1-score of 0.461 ± 0.375, highlighting the tracker's limitations in maintaining reliable long-term tracking without drift (see Figure 3).

4.2 Evaluation of Trackers by Category

For comparative purposes, the previous experiment was repeated with CSRT, KCF, and MIL trackers, in addition to a new execution of the updated Struck tracker. The primary objective was to evaluate and compare the tracking throughput of each algorithm under identical conditions. To ensure consistency with the ALOV300++ dataset processing pipeline, the implementations of the selected trackers were adapted from official OpenCV examples.

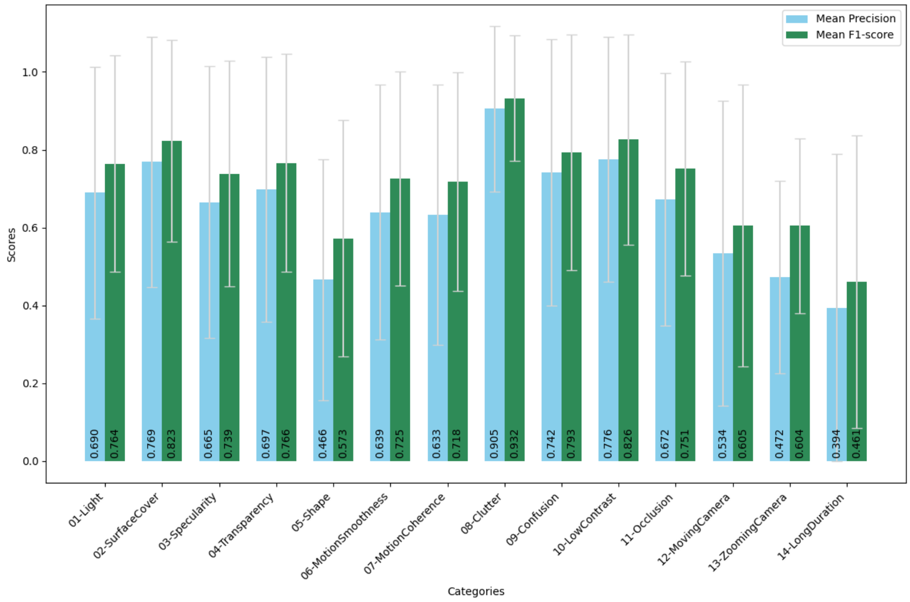


Fig. 3. Struck performance evaluation for all ALOV300++ categories.

F1-score analysis by category for the mentioned trackers are presented in Figure 4. Results show that Struck and CSRT generally achieve higher F1-scores across most challenge categories, with Struck performing particularly well in "02-SurfaceCover", "08-Clutter", "10-LowContrast", and CSRT excelling in "01-Light", "08-Clutter", and "10-LowContrast". MIL demonstrates moderately competitive performance in several categories, especially "02-SurfaceCover" and "08-Clutter" but struggles in "12-MovingCamera" and "14-LongDuration". KCF consistently records the lowest F1-scores across most categories, with notably poor results in dynamic camera scenarios such as "12-MovingCamera", "13-ZoomingCamera", and "14-LongDuration". Overall, Struck and CSRT exhibit stronger robustness under varied visual tracking challenges, particularly in complex or cluttered environments.

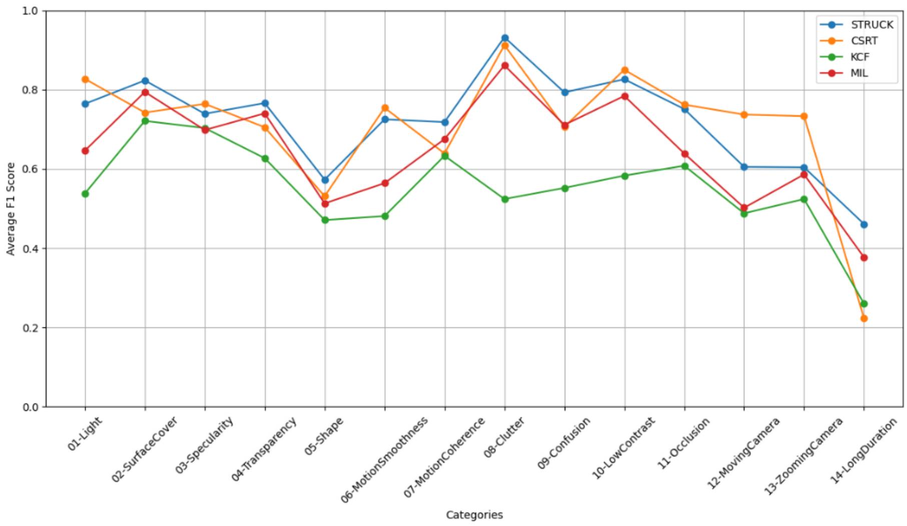


Fig. 4. Average F1-score of each category.

Regarding FPS performance across the 14 visual tracking challenge categories (see Figure 5), KCF consistently demonstrates the highest throughput, achieving over 1000 FPS in dynamic conditions such as "07-MotionCoherence" and "14-LongDuration" peaking at 300 FPS in the last category. CSRT offers a balanced compromise between speed and accuracy, with FPS values ranging from approximately 84.5 to 319.6, showing strong performance particularly in "14-LongDuration" and "01-Light". Struck and MIL, while significantly slower than KCF and CSRT, maintain stable frame rates between 42.2 and 86.4 FPS. Notably, Struck shows consistent FPS across categories, reflecting stable computational behavior. Overall, KCF leads in execution speed but at the expense of accuracy, while CSRT strikes a favorable trade-off for scenarios requiring both real-time performance and reliability.

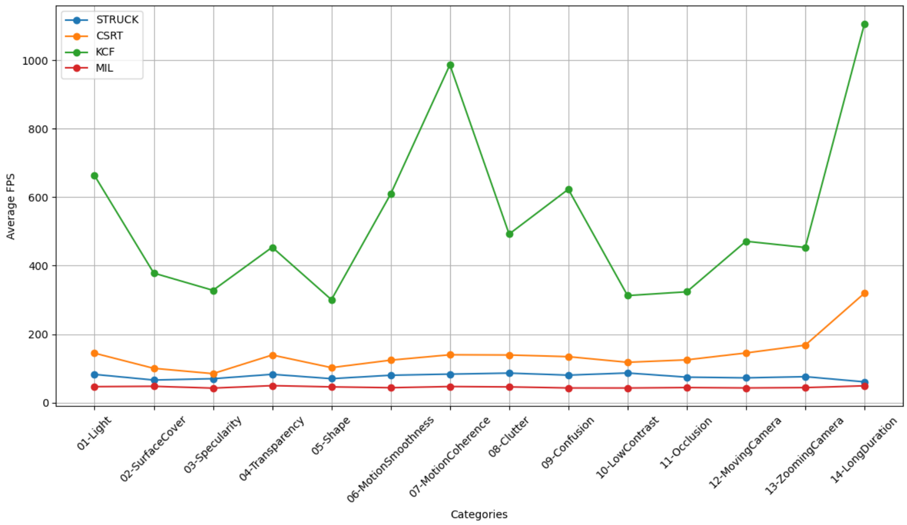


Fig. 5. Average frames per second of each tracker by category.

4.3 Global Results

The global results presented in Table 2 and Figure 6 indicate that Struck and CSRT exhibit relatively higher F1-scores—0.720 ± 0.122 and 0.706 ± 0.166, respectively—compared to KCF (0.551 ± 0.115) and MIL (0.649 ± 0.132), suggesting potentially better tracking performance under the tested conditions.

Table 2. Performance comparison of evaluated trackers.

Tracker	Precision	F1-score	FPS	Total dataset execution time
CSRT	0.633 ± 0.179	0.706 ± 0.166	135.900 ± 98.160	17 min, 49.77 sec
KCF	0.475 ± 0.115	0.551 ± 0.115	503.850 ± 583.050	9 min, 16.57 sec
MIL	0.566 ± 0.138	0.649 ± 0.132	44.780 ± 6.580	32 min, 55.78 sec
STRUCK	0.647 ± 0.139	0.720 ± 0.122	77.100 ± 20.110	23 min, 31.02 sec

However, the performance differences between Struck, CSRT, and MIL are moderate, and the overlapping standard deviations imply that additional statistical analysis would be required to determine whether these differences are statistically significant.

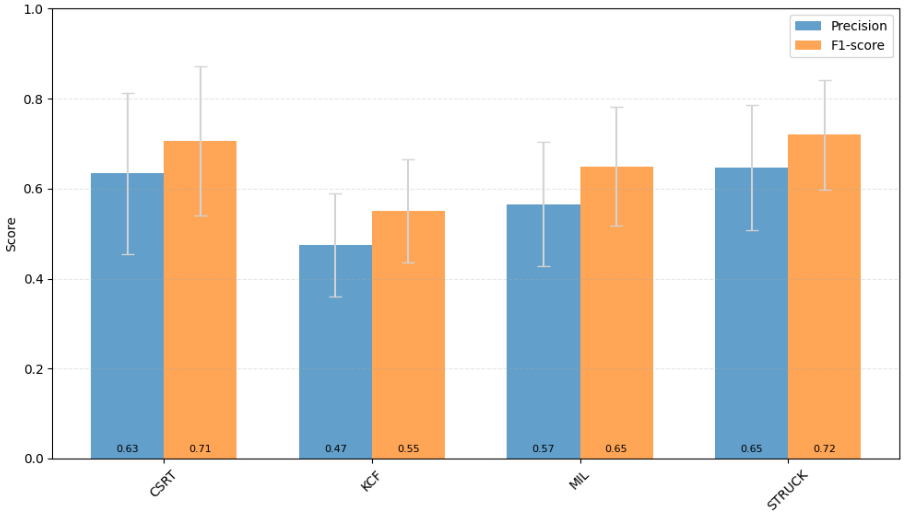


Fig. 6. Global precision and F1-score per tracker.

5 Conclusions

This work presents the implementation update of the original Struck algorithm by Hare et al., migrating from OpenCV 2.4 to OpenCV 4.9.0, and its comparative evaluation using the ALOV300++ dataset on a conventional workstation. Although the updated implementation was successfully tested on both Windows 10 and Ubuntu 22.04, this report includes only the experiments conducted under the Ubuntu environment. Extensive testing of the Struck algorithm's performance in terms of F1-score, precision, and frames per second (FPS) metrics was conducted on both mentioned operating systems using the ALOV300++ dataset. These evaluations confirmed the algorithm's efficiency and accuracy in various scenarios. It is worth mentioning that the Recall metric was not used because, due to the nature of the visual tracking task, it is theoretically always equal to 1. Implementations of CSRT, MIL, and KCF trackers were adapted to the ALOV300++ dataset and tested successfully on all 314 sequences.

The comparative analysis of F1-score and FPS across 14 challenging visual tracking categories reveals a clear trade-off between accuracy and speed among the evaluated trackers. Struck and CSRT consistently deliver high F1-scores, indicating robust tracking performance under varying visual conditions, with Struck slightly outperforming CSRT in cluttered and low-contrast scenarios. However, although KCF achieves the highest FPS across all categories, it demonstrates significantly lower accuracy particularly in complex or long-duration sequences. The observed FPS values confirm that MIL offers a moderate balance between tracking accuracy and processing speed but does not lead in either category. CSRT emerges as the most balanced option, offering a strong compromise between precision and computational efficiency. Struck remains a reliable choice for scenarios that prioritize accuracy, albeit at a higher computational cost. In contrast, KCF significantly outperforms the other trackers in terms of speed, making it the preferred solution for real-time applications where processing throughput is more critical than precise target localization.

Future work will focus on the use of VOT algorithms and metaheuristics to design, implement, and evaluate a VOT system on a SoC-FPGA (System-on-Chip Field-Programmable Gate Array) platform. SoC-FPGA is a heterogeneous hardware architecture that integrates a conventional CPU processor core with FPGA fabric, enabling efficient hardware–software co-design. Building upon the current progress of this Ph.D. research, the next phase will involve the development and deployment of the updated Struck tracker on a SoC-FPGA platform, such as Xilinx's ZC706 development board, aiming to achieve real-time performance and resource-efficient implementation.

References

1. Smeulders, A.W.M., Chu, D.M., Cucchiara, R., Calderara, S., Dehghan, A., Shah, M.: Visual Tracking: An Experimental Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence. 36, 1442–1468 (2014). https://doi.org/10.1109/TPAMI.2013.230
2. Kristan, M.: The Tenth Visual Object Tracking VOT2022 Challenge Results :: ViCoS Prints, https://prints.vicos.si/publications/416/the-tenth-visual-object-tracking-vot2022-challenge-results, last accessed 2023/06/06
3. Soleimanitaleb, Z., Keyvanrad, M.A.: Single Object Tracking: A Survey of Methods, Datasets, and Evaluation Metrics. ArXiv. (2022)
4. Chamberlain, R.D.: Architecturally truly diverse systems: A review. Future Generation Computer Systems. 110, 33–44 (2020). https://doi.org/10.1016/j.future.2020.03.061
5. Méndez López, V.A., Soubervielle Montalvo, C., Núñez Varela, A.S., Pérez Cham, O.E., González Galván, E.J.: A Review of Design Methodologies and Evaluation Techniques for FPGA-Based Visual Object Tracking Systems. International Journal of Combinatorial Optimization Problems and Informatics. 15, 127–145 (2024). https://doi.org/10.61467/2007.1558.2024.v15i5.571
6. Abbass, M.Y., Kwon, K.-C., Kim, N., Abdelwahab, S.A., El-Samie, F.E.A., Khalaf, A.A.M.: A survey on online learning for visual tracking. Vis Comput. 37, 993–1014 (2021). https://doi.org/10.1007/s00371-020-01848-y
7. Park, E., Berg, A.C.: Meta-tracker: Fast and Robust Online Adaptation for Visual Object Trackers. 11207, 587–604 (2018). https://doi.org/10.1007/978-3-030-01219-9_35
8. Lu, J., Hoi, S.C.H., Wang, J., Zhao, P., Liu, Z.-Y.: Large Scale Online Kernel Learning. (2016)
9. Hoi, S.C.H., Sahoo, D., Lu, J., Zhao, P.: Online learning: A comprehensive survey. Neurocomputing. 459, 249–289 (2021). https://doi.org/10.1016/j.neucom.2021.04.112
10. Hare, S., Saffari, A., Torr, P.H.S.: Struck: Structured output tracking with kernels. In: 2011 International Conference on Computer Vision. pp. 263–270 (2011). https://doi.org/10.1109/ICCV.2011.6126251
11. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-Speed Tracking with Kernelized Correlation Filters. IEEE Transactions on Pattern Analysis and Machine Intelligence. 37, 583–596 (2015). https://doi.org/10.1109/TPAMI.2014.2345390

12. Babenko, B., Yang, M.-H., Belongie, S.: Visual tracking with online Multiple Instance Learning. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 983–990 (2009). https://doi.org/10.1109/CVPR.2009.5206737
13. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: Exploiting the Circulant Structure of Tracking-by-Detection with Kernels. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., and Schmid, C. (eds.) Computer Vision – ECCV 2012. pp. 702–715. Springer, Berlin, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33765-9_50
14. Adamo, F., Carcagnì, P., Mazzeo, P.L., Distante, C., Spagnolo, P.: TLD and Struck: A Feature Descriptors Comparative Study. In: Mazzeo, P.L., Spagnolo, P., and Moeslund, T.B. (eds.) Activity Monitoring by Multiple Distributed Sensing. pp. 52–63. Springer International Publishing, Cham (2014). https://doi.org/10.1007/978-3-319-13323-2_5
15. Wang, J., Zheng, L., Tang, M., Feng, J.: A Comparison of Correlation Filter-Based Trackers and Struck Trackers. IEEE Transactions on Circuits and Systems for Video Technology. 30, 3106–3118 (2020). https://doi.org/10.1109/TCSVT.2019.2931924
16. Venugopal Minimol, P., Mishra, D., Gorthi, R.K.S.S.: Guided MDNet tracker with guided samples. Vis. Comput. 38, 1135–1149 (2022). https://doi.org/10.1007/s00371-021-02072-y.
17. Fiaz, M., Mahmood, A., Javed, S., Jung, S.K.: Handcrafted and Deep Trackers: Recent Visual Object Tracking Approaches and Trends. ACM Comput. Surv. 52, 43:1-43:44 (2019). https://doi.org/10.1145/3309665
18. Ning, J., Yang, J., Jiang, S., Zhang, L., Yang, M.-H.: Object Tracking via Dual Linear Structured SVM and Explicit Feature Map. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4266–4274. IEEE, Las Vegas, NV, USA (2016). https://doi.org/10.1109/CVPR.2016.462
19. Soleimanitaleb, Z., Keyvanrad, M.A.: Single Object Tracking: A Survey of Methods, Datasets, and Evaluation Metrics. arXiv:2201.13066. (2022)
20. Trigka, M., Dritsas, E.: A Comprehensive Survey of Machine Learning Techniques and Models for Object Detection. Sensors. 25, 214 (2025). https://doi.org/10.3390/s25010214
21. Terven, J., Cordova-Esparza, D.M., Ramirez-Pedraza, A., Chavez-Urbiola, E.A., Romero-Gonzalez, J.A.: Loss Functions and Metrics in Deep Learning. Artif Intell Rev. 58, 195 (2025). https://doi.org/10.1007/s10462-025-11198-7

An Experimental Study on the Analysis of Gait Disorders Through Multimodal Feature Extraction

Iván J. Sánchez-Cuapio[1,3], Ricardo Ramos-Aguilar[2,3], Paola A. Niño-Suárez[1], Karla M. Cerón-Arriaga[3]

[1]Escuela Superior de Ingeniería Mecánica y Eléctrica Unidad Azcapotzalco -IPN, CDMX, México
[2]Unidad Profesional Interdisciplinaria en Ingeniería Campus Tlaxcala - IPN, Tlaxcala, México
[3]Universidad Tecnológica de Tlaxcala, Huamantla, Tlaxcala, México
ivanjs24@gmail.com, {rramosa,pninos}@ipn.mx,moreceron111@gmail.com

Abstract. Accurate detection of gait abnormalities is essential for the diagnosis and monitoring of neuromotor disorders. This work presents the experimental design for the multimodal acquisition and extraction of temporal, spatial, and frequency-domain features in patients with motor impairments, using accelerometers and an RGB-D camera (Kinect). The clinical protocol, sensor configuration, and controlled environment are described to achieve more accurate data collection of patients' gait. Additionally, signal processing techniques are detailed to extract relevant biomechanical variables along with their clinical justification. Preliminary results with healthy subjects and the support of specialists validate the system's accuracy and reproducibility. Clinical implications, limitations, and future directions focused on integrating machine learning for gait pathology classification are discussed. This study lays the groundwork for developing objective and non-invasive tools to enhance the assessment and rehabilitation of patients with movement disorders.

Keywords: Gait Pathology, Clinical Study, Feature Extraction.

1 Introduction

Human gait is one of the most complex motor functions of the body, and its analysis provides valuable information about neuromuscular integrity and an individual's functional status. Gait pattern abnormalities are common clinical signs in diseases such as Parkinson's, multiple sclerosis, stroke, and other neurological and musculoskeletal conditions. Early and accurate detection of these abnormalities is key to planning timely and personalized therapeutic interventions [14].

Clinical gait analysis has been performed through qualitative observation or using high-cost laboratory equipment such as optical motion capture systems or force platforms. However, advances in wearable technologies have promoted the use of inertial sensors and depth cameras as accessible and effective alternatives for collecting kinematic and dynamic data in real-world or clinically adapted environments [11, 12]. In particular, the combination of triaxial accelerometers and RGB-D cameras, such as the Kinect sensor, enables multimodal motion capture by integrating temporal, spatial, and frequency-domain information with a high level of resolution.

However, the clinical quality and usefulness of the collected data largely depend on the experimental design used. Factors such as sensor placement, environmental conditions, movement protocol, and subject characteristics can introduce significant variability if not properly controlled. Therefore, it is essential to establish a standardized experimental environment that ensures data consistency, validity, and reproducibility, facilitating subsequent quantitative analysis and the training of automated classification models [17].

This work aims to develop a *clinically oriented experimental design* for the extraction of multimodal gait features in patients with and without pathologies, using accelerometers and RGB-D cameras. The methodology is based on the *Biodesign* approach [20], which structures the process into three phases: identification of the clinical need, invention of the technological solution, and planning for its implementation. This perspective ensures that the experiment not only meets scientific and technical criteria, also addresses a real need within the clinical setting, and aligns with ethical and regulatory standards such as those established by the Good Clinical Practice (GCP) guidelines and the IEEE 11073 family of standards.

2 Theoretical Framework

Gait analysis is a multidisciplinary field that involves principles from biomechanics, biomedical engineering, neurology, and physical therapy. Its main objective is to quantify the kinematic and dynamic parameters of human movement, which allows for the identification of abnormalities associated with various neuromuscular or musculoskeletal pathologies.

2.1 Key Parameters in Gait Analysis

Gait studies focus on three types of features:

-Temporal features: such as step time, stance time, and gait cycle duration.

-Spatial features: such as step length, foot clearance, and stride symmetry.

-Frequency-domain features: obtained through spectral analysis of movement signals, useful for detecting tremors or rigidity, including features such as dominant frequency, spectral energy, spectral entropy, harmonic ratio, and bandwidth, which provide valuable insights into gait regularity, symmetry, and motor impairments.

These parameters can be combined to accurately characterize normal or pathological gait, and are extracted using various instrumental and computational methodologies.

2.2 Traditional Methodologies

Historically, gait analysis was conducted in biomechanics laboratories using instruments such as:

- **Optical motion capture systems.** such as Vicon or OptiTrack, which use multiple infrared cameras and reflective markers to obtain three-dimensional coordinates of body segments [1].
- **Force platforms.** which record ground reaction forces to estimate joint moments and centers of pressure [15].
- **Electromyography (EMG).** to analyze the electrical activity of muscles during gait [10].

Although various techniques for accessing gait parameters have been developed, the mentioned methods present limitations in terms of mobility, cost, the need for calibration, and the involvement of specialized technical personnel.

2.3 Multimodal Analysis and Artificial Intelligence

The integration of inertial sensors (IMUs), which combine accelerometers, gyroscopes, and magnetometers, has gained popularity due to their portability, low cost, and suitability for both clinical and home environments. These devices enable trajectory reconstruction, angular velocity estimation, and detection of gait pattern variations, using signal processing techniques like Kalman filtering and gait cycle segmentation. Concurrently, RGB-D cameras such as Microsoft Kinect provide markerless, three-dimensional motion capture by combining RGB images with depth data, facilitating real-time skeleton tracking despite limitations in lateral accuracy and occlusions. Recently, multimodal approaches that fuse data from IMUs and RGB-D systems have proven effective in providing a richer representation of human movement, enhancing the performance of machine learning and deep learning models for gait classification. Additionally, the use of frequency-domain analysis through Fourier or wavelet transforms enables the identification of rhythmic components relevant to motor disorders.

2.4 Biodesign-Based Methodology

As the conceptual basis for this project, the approach proposed in the *Biodesign* book is adopted [20], which guides the development of medical technology solutions through a structured sequence: identification of the clinical need, invention of a feasible solution, and preparation for effective implementation. This methodological framework ensures that the experimental design is not only technically sound but also clinically relevant and aligned with ethical regulations.

3 Methodology

3.1 Experimental Design

The present study has a cross-sectional and controlled experimental design, with an observational clinical component, aimed at characterizing pathological gait through the multimodal extraction of temporal, spatial, and frequency features. The approach combines inertial technologies and computer vision to obtain quantifiable and reproducible data from patients with gait abnormalities.

This design is based on the methodological approach proposed by the book Biodesign: The Process of Innovating Medical Technologies [20], which establishes a systematic framework to identify clinical needs, design technological solutions, validate their functionality, and generate scientific evidence. In this context, gait analysis and capture are considered a relevant clinical need to improve diagnosis and monitoring of patients with neuromotor impairments.

The central hypothesis of the study is that the combination of inertial sensors and RGB-D vision improves the accuracy and sensitivity in detecting abnormal gait patterns compared to single-channel methodologies.

3.2 Population and Sample

The target population consists of adults over 18 years old with a clinical diagnosis of a pathology affecting gait, such as Parkinson's disease, hemiparesis, osteoarthritis, as well as healthy subjects serving as the control group.

Inclusion Criteria. The inclusion criteria for the gait feature extraction experiment ensure that participants adequately represent the target population, can safely perform the tests, and generate valid data. The study includes individuals with a medical diagnosis of gait-related pathologies (such as Parkinson's disease, hemiparesis, or osteoarthritis) and healthy subjects as the control group, aged between 18 and 80 years. Additionally, participants must be able to walk without mechanical assistance to ensure uniformity in data capture and protocol application.

Exclusion Criteria. Exclusion criteria are established to maintain data validity, safety, and integrity during the experiment. Participants with acute injuries or disabling pain on the test day, cognitive impairments affecting comprehension, or the use of orthopedic devices that interfere with sensor placement will be excluded. These measures ensure a controlled and reliable experimental environment.

Study Groups. The study will include a control group of healthy individuals without neuromuscular or musculoskeletal gait impairments and an experimental group of patients with clinically diagnosed gait-altering pathologies. The experimental group will encompass various conditions such as Parkinson's disease, post-stroke hemiparesis, multiple sclerosis, and osteoarthritis. Each participant's specific diagnosis will be documented to allow for stratified analyses, ensuring a representative range of gait impairments for developing robust and generalizable analytical tools.

Sample Size. The sample size in gait analysis studies depends on factors such as the study objective, population variability, significance level (typically 0.05), statistical power, usually 80%, and expected effect size. Resource availability also influences the number of participants. Exploratory studies often use smaller samples (10–30 per group), while confirmatory studies require formal power analyses to ensure valid results. Based on prior research, a minimum of 15 subjects per group is estimated for this study [7, 16].

3.3 Devices and Setup

The combination of triaxial accelerometers and RGB-D cameras is essential for the accurate and comprehensive extraction of kinematic, dynamic, and frequency-based features during human gait. Accelerometers enable direct and continuous capture of localized movements of specific body segments, providing detailed data on linear acceleration, oscillation patterns, and rhythmic variations that are difficult to record through visual methods. Meanwhile, the RGB-D camera offers a global three-dimensional view of the moving body, allowing for virtual skeletal reconstruction and the calculation of spatial parameters such as step length, gait symmetry, and posture. This multimodal capture integrates complementary local and global information into a single analysis, enhancing the accuracy of anomaly detection and reducing the risk of errors caused by occlusions, noise, or artifacts present in a single modality. Therefore, the combined use of these technologies significantly improves data quality and strengthens the validity of biomechanical analysis, particularly in clinical settings where high sensitivity is required for the early detection of gait impairments.

Two types of technologies will be used for data capture in this experiment:

Triaxial accelerometer. IMU sensor with recording on all three axes (X, Y, Z), a sampling frequency of 100 Hz or higher, and a minimum resolution of 16 bits.

RGB-D Camera. Microsoft Kinect v2 with RGB image stream at 30 fps and depth map, enabling real-time human skeleton reconstruction.

3.4 Sensor Positioning

Inertial sensing system for gait parameter recording. Inertial sensing systems are advanced technologies used to accurately capture and analyze gait parameters and other human movements. These systems rely on sensors that measure linear acceleration and angular velocity of body segments. A typical inertial system architecture includes multiple sensors strategically placed on the body, connected to a central processing unit that records and processes the data. Inertial sensors are small and lightweight, enabling comfortable and unrestricted data acquisition during walking. These systems provide precise measurements of kinematic parameters such as joint angles and movement trajectories, as well as temporal parameters like cadence, step length, and stance and swing times. This capability makes inertial sensors versatile tools both in clinical settings for the assessment of musculoskeletal disorders and in sports applications for performance analysis and functional biomechanics [12].

The optimal placement of inertial sensors for gait parameter recording depends on the biomechanical factors of human walking and joint motion, which affect the accuracy and reliability of the collected data. It is generally recommended to mount the sensors on body segments that undergo significant movement during gait, such as the thighs, shanks, and feet (Fig. 1) . For example, placing sensors in the lumbar region or on the legs allows for more direct capture of relevant joint angles and motion patterns. Furthermore, precise placement on specific anatomical landmarks, such as the anterior superior iliac spine for the pelvis or the center of the knee for knee joint flexion, ensures more accurate measurements of kinematic parameters. This strategy not only facilitates a detailed assessment of gait biomechanics but also minimizes the risk of external interferences and motion artifacts, thus ensuring the quality of data obtained for clinical analysis and sports applications [2].

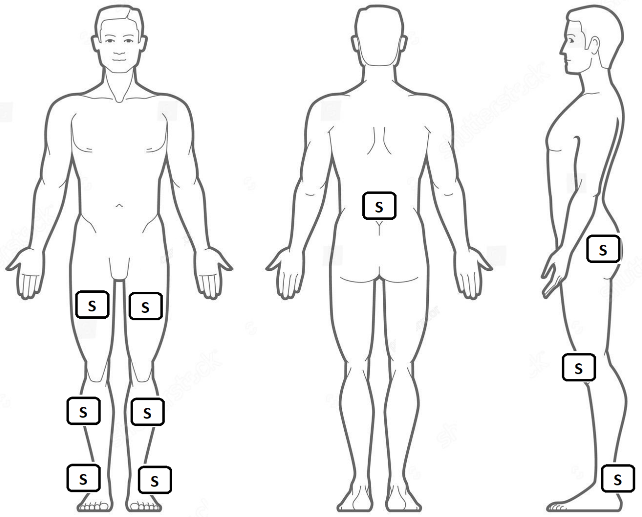


Fig. 1. Sensor positioning.

Vision system with depth camera for gait parameter recording. The use of RGB-D depth cameras, such as the Microsoft Kinect system, has revolutionized gait analysis by providing three-dimensional data capture that combines RGB sensors with depth sensors. This technology stands out for its ease of use, low invasiveness, and cost-effectiveness, making it accessible in both clinical and research settings. Depth cameras allow precise evaluation of kinematic and kinetic parameters without the need for body-attached markers, thereby improving subject comfort. Their application in biomechanical research and rehabilitation has been extensively documented, highlighting their advantages and limitations compared to traditional motion capture systems. Although challenges such as limited accuracy and dependence on lighting conditions exist, depth cameras offer a valuable tool for detailed and accessible analysis of human gait [19].

Proper placement of the capture system is crucial to obtain accurate and reliable data in gait analysis. The location and angle of the camera determine the quality and precision of the measured kinematic and kinetic parameter information. To achieve optimal movement capture, the camera should be positioned at an appropriate height and distance from the subject, generally at waist level and approximately 2 to 3 meters away (Fig.2). This positioning ensures that the subject's entire body is within the camera's field of view throughout the complete gait cycle. Additionally, it is important to adjust the camera's tilt angle to maximize the visibility of body segments and minimize marker occlusion, as shown in Table 1 [8].

Table 1. Technical and configuration parameters for the RGB-D Kinect v2 camera.

Parameter	**Recommended Specification**
Camera model	Kinect v2 (Microsoft) with RGB-D sensor
Depth resolution	512 × 424 pixels
Color resolution (RGB)	1920 × 1080 pixels
Sampling frequency (fps)	30 frames per second (synchronized with IMU signals at 100 Hz)
Distance to subject	2.5 meters for full-body tracking
Mounting height	Approximately 1.0 to 1.2 meters from the ground (waist or hip level of the subject)
Tilt angle	Adjusted to maximize visibility of lower limbs - 45°
Camera placement	Frontal to the subject along the walking path - centered or slightly lateral position
Lighting conditions	Uniform lighting, avoiding strong shadows or backlighting. Direct sunlight should be avoided.
Walking surface	Flat, non-slip surface at least 5 meters long
Recording duration per participant	Approximately 10 minutes - includes 3 repetitions per task
Synchronization with IMU	Using shared visual and auditory events

The synchronization of the collected data will be carried out using shared visual/audio events to align signals. The software will integrate both sources for subsequent analysis.

3.5 Capture Protocol

The tests will be conducted in a controlled clinical environment: Non-slip flat surface of at least 5 meters, uniform lighting, obstacle-free space (with no inclines, declines, or floor level changes).

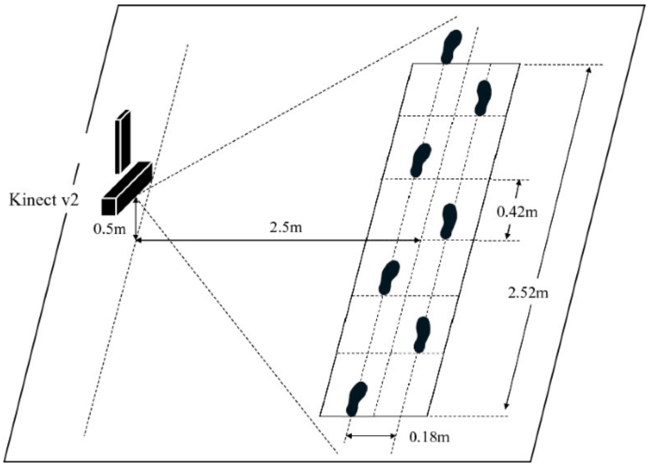


Fig. 2. Camera Placement.

The activities considered are: straight-line walking over 5 meters -back and forth-,180° turn, voluntary pause during the walk, three repetitions per task. The total duration per participant was 15 minutes.

4 Signal Processing and Feature Extraction

4.1 Signal Preprocessing

The signals captured by inertial sensors (IMUs) and the RGB-D camera require a cleaning and synchronization process before extracting relevant features. The general processing workflow includes:

Noise filtering. A fourth-order low-pass Butterworth filter with a cutoff frequency between 5 and 20 Hz is used to eliminate high-frequency noise while preserving relevant movement components [6].

Normalization. Z-score normalization is applied to standardize signals from different participants, minimizing variations due to anatomical or strength differences.

Gait cycle segmentation. Algorithms based on peak analysis of vertical acceleration and events of initial contact (heel-strike) and toe-off are employed, validated by visual references obtained from the 3D skeleton captured by the RGB-D camera [16].

Multimodal synchronization. Temporal alignment of information from the IMU and RGB-D camera is performed using common event signals (acoustic and visual markers) to ensure precise correspondence between data.

4.2 Spatial Features

Spatial variables are fundamental for the functional assessment of gait. In this study, the following spatial variables will be extracted.

Step length is defined as the distance between successive contacts of the same foot. It is a sensitive indicator in motor disorders and pathologies such as Parkinson's disease or hemiparesis.

Step width corresponds to the lateral separation between the feet during gait. An abnormal step width may indicate balance problems or neurological impairment.

Finally, foot clearance height is derived from the vertical component of the 3D skeleton. Reductions in this variable may indicate foot drag or muscle weakness [3].

These features are extracted from the analysis of the skeleton reconstructed by the RGB-D camera, applying basic geometry between joint positions such as the hip, knee, or ankle, as shown in Table 2.

4.3 Temporal Features

Temporal variables are mainly derived from inertial data. These include the duration of the gait cycle, defined as the time between two consecutive contact events of the same foot; the duration of the stance and swing phases, which represent the time each foot remains in contact with the ground or in the air; and cadence, which is the number of steps taken per minute. These variables help differentiate typical pathological gait patterns, such as a prolonged double support phase, which is characteristic in patients with postural instability or motor impairment(Table 2 [9]).

4.4 Frequency Features

A frequency-domain analysis is conducted using the Fast Fourier Transform (FFT) and spectral analysis. The main variables extracted include the dominant frequency, which corresponds to the spectral component with the highest energy; the harmonic spectrum, defined as the ratio between the fundamental energy and the higher harmonics; and spectral entropy, which quantifies the energy dispersion(Table 2 [13]).

This type of analysis is particularly valuable for distinguishing between regular movement patterns—such as those exhibited by healthy subjects—and irregular or asynchronous patterns typically found in patients with neurological conditions.

4.5 Technical and Clinical Justification

From a clinical perspective, these features allow for the quantification of motor impairment, the evaluation of asymmetries, and the identification of fall risks.

Table 2. Multidomain gait features for classification.

Domain	Features	Statistical Metrics
Temporal	Gait cycle duration, stance time, swing time, double support, cadence, step time	Mean, median, SD, coefficient of variation, IQR, symmetry index
Spatial	Step length, step width, stride length, step height, pelvic tilt, foot clearance	Mean, SD, symmetry ratio, RMS, max-min range
Frequency	Dominant frequency, harmonic ratio, spectral entropy, power spectral density (PSD), bandwidth	Peak frequency, spectral centroid, spectral flatness, energy ratio, entropy index

Technically, the combination of features from multiple domains has been shown to increase classification accuracy in previous gait analysis studies [5, 16].

The fusion of data from inertial sensors and RGB-D cameras also enhances the robustness of the analysis by enabling redundancy and cross-validation of the extracted features.

5 Expected and Preliminary Results

5.1 Expected Results

This experimental study aims to obtain a robust set of gait features that can accurately distinguish between normal and pathological patterns through multimodal analysis.

Structured Multimodal Database. A synchronized dataset containing accelerometry signals and RGB-D skeletal data for each participant, organized by gait type, healthy vs. pathological.

Feature Extraction and Validation. A set of temporal, spatial, and frequency-domain variables with discriminative power between different pathological groups is expected to be obtained.

Pathology-Specific Characteristic Profile. Group analysis will help identify which variables are most relevant for detecting specific pathologies such as Parkinson's disease, hemiplegia, or ataxia.

Experimental Design Efficiency. The proposed environment and protocol will be validated to ensure they enable the acquisition of high-quality, reproducible, and clinically meaningful data.

Preliminary Classification Model. Based on the extracted features, the initial development of a classifier such as SVM, Random Forest, or a lightweight neural network is anticipated, aiming to achieve high sensitivity and specificity in distinguishing between healthy individuals and those with gait impairments.

This set of results will lay the foundation for subsequent phases of the project, including clinical validation, sample size expansion, and the training of deep learning models.

5.2 Preliminary Results

Pilot tests have been conducted so far with five healthy subjects to evaluate the experimental setup and the multimodal data acquisition process. The following observations have been made:

Good synchronization between devices was achieved, successfully aligning the IMU signals (at 100 Hz) with the RGB-D camera data (at 30 fps) through the use of visual markers and shared features.

High fidelity in 3D skeleton reconstruction was verified, confirming the Kinect v2 camera's accuracy in joint tracking, which enabled stable calculation of joint lengths and angles.

Precise identification of gait events was accomplished through peak analysis in vertical acceleration and validation using heel joint displacement, allowing for accurate segmentation of gait cycles.

Consistency of key parameters such as step length (0.68 ± 0.05 m), cadence (110 ± 6 steps/min), and cycle duration (1.1 ± 0.1 s) matched the average values reported in the literature for healthy young adults [18].

Detection of inter-subject variability was observed, as small individual differences in pelvic motion amplitude and swing time were recorded, suggesting the system's sensitivity in capturing subtle features of gait patterns.

These initial findings validate the technical feasibility of the proposed experimental setup and support its application in populations with clinical gait impairments. As the sample size increases and subjects with specific pathologies are included, it is expected that distinct patterns in the extracted features will emerge.

6 Discussion

The preliminary results obtained in this experimental study support the feasibility of a multimodal approach for characterizing human gait, combining accelerometry signals and skeletal data from an RGB-D camera. This strategy has proven effective in identifying clinically relevant parameters in human gait, which is consistent with previous work in biomechanics and motion analysis.

The proper synchronization between inertial sensors and the Kinect camera has enabled accurate segmentation of gait cycles, which is essential for obtaining robust temporal variables such as the duration of stance and swing phases. This level of precision is crucial, as studies have shown that small variations in these parameters can indicate neurological conditions such as Parkinson's disease or peripheral neuropathies.

Likewise, the extracted spatial (e.g., step length) and frequency-based features (spectral components of movement) show interindividual differences that could serve as digital biomarkers in subjects with gait impairments. This perspective aligns with current trends in personalized medicine and clinical analysis using wearable technologies.

The methodological approach based on the Biodesign framework has allowed this study to be structured not only from a technical perspective but also with consideration of its future clinical applicability. This ensures that the proposed solutions are aligned with the real needs of hospital and rehabilitation settings, facilitating potential technological transfer.

Nevertheless, this study has limitations inherent to its early stage, such as the small number of participants and the lack of representation of various pathologies. As the sample size increases and participants with confirmed clinical diagnoses are introduced, the statistical and classification models developed are expected to be validated and refined. It will also be necessary to evaluate the system's generalizability in response to variations in the physical environment (e.g., surface type or lighting) and possible instrumental noise.

Finally, future phases of the project propose the integration of more sophisticated machine learning algorithms, including convolutional neural networks or hybrid architectures, to better exploit the temporal and spatial richness of the captured data.

7 Conclusions

This experimental study has demonstrated the technical and clinical feasibility of a multimodal design for extracting temporal, spatial, and frequency-based features in the gait of patients with motor impairments. The combination of accelerometers and RGB-D cameras enables a synergistic capture of data that enhances the quality and precision of biomechanical analysis, which is essential for the early detection and monitoring of gait disorders.

The preliminary results indicate that the proposed capture protocol is reproducible and suitable for obtaining synchronized and segmented signals, with parameters that align with specialized literature. This reinforces the applicability of the methodological framework based on Biodesign principles, which facilitates an effective integration between technological innovation and real clinical needs.

Among the limitations of the study are the small sample size and the need to include a greater diversity of pathologies to validate the generalizability of the extracted features. Additionally, the controlled environment of the experiment should be tested under more variable conditions to evaluate the robustness of the system.

In future stages, the extracted features will be processed using state-of-the-art multimodal classification algorithms. These may include support vector machines (SVM), random forests, convolutional neural networks (CNNs), and transformer-based models, which are capable of handling heterogeneous time-series and spatial data. The project has already included data from healthy individuals as well as patients diagnosed with gait-related conditions such as genu varum, genu valgum, knee osteoarthritis, and lumbar disc herniation, providing a representative foundation for training and evaluating the classifiers. These models are expected to support specialists in the diagnosis, progression tracking, and personalization of rehabilitation strategies through objective, non-invasive tools.

As future lines of research, the following are proposed:

- Expand the sample by including patients with specific clinical diagnoses to strengthen the comparative analysis and train predictive models using advanced machine learning [4].
- Implement longitudinal protocols to evaluate gait evolution in response to treatments or rehabilitation, identifying digital biomarkers of progress or deterioration.
- Integrate real-time analysis to provide immediate feedback to patients and therapists, facilitating personalized and adaptive interventions.
- Explore the incorporation of other sensory modalities, such as surface electromyography (sEMG) or plantar pressure sensors, to enrich the patient's biomechanical profile.

Together, this work constitutes a significant contribution towards the development of clinical monitoring systems that help improve the quality of life for individuals with motor impairments and facilitate clinical decision-making based on objective and quantifiable data.

References

1. Baker, R.: Gait analysis methods in rehabilitation. Clinics in Physical Therapy (2006)
2. Caldas, R., Mundt, M., Büttner, F., Kautz, T., Eskofier, B.M.: Wearable inertial sensors for human movement analysis: A review of posture and gait monitoring applications. IEEE Sensors Journal **17**(2), 394–405 (2017)
3. Del Din, E., Godfrey, C., Rochester, L.: Spatiotemporal gait analysis using a single wearable sensor in parkinson's disease. IEEE Transactions on Biomedical Engineering **63**(3), 699–707 (2016)
4. Esfahlani, F.Z., et al.: Deep learning for human gait analysis: A review. IEEE Transactions on Neural Systems and Rehabilitation Engineering **30**, 1501–1515 (2022)
5. Esfahlani, S., Luo, H., Yang, Z., Sarrafzadeh, M.: Deepgait: Temporal convolutional neural network for accurate gait phase detection in spinal cord injury. IEEE Journal of Biomedical and Health Informatics **25**(3), 732–742 (2021)
6. Espinosa, K., Alcaraz, G., Ramírez, R.: Gait event detection using inertial measurement units: a systematic review. Sensors **22**(3), 1–25 (2022)
7. Fida, B., et al.: Gait pattern classification using wearable inertial sensors. Journal of Biomechanics **48**(11), 3123–3129 (2015)
8. Gabel, G.G.: Introduction to Human Movement Analysis. Human Kinetics, Champaign, IL (2009)
9. Hausdorff, J.M.: Gait dynamics, fractals and falls: finding meaning in the stride-to-stride fluctuations of human walking. Human Movement Science **26**(4), 555–589 (2007)
10. Hermens, H.J., Freriks, B., Disselhorst-Klug, C., Rau, G.: Development of recommendations for semg sensors and sensor placement procedures. Journal of Electromyography and Kinesiology **10**(5), 361–374 (2000)
11. Muro-de-la Herran, A., Garcia-Zapirain, B., Mendez-Zorrilla, A.: Gait analysis methods: an overview of wearable and non-wearable systems, highlighting clinical applications. Sensors **14**(2), 3362–3394 (2014)

12. Silva de Lima, A.L., et al.: Wearable inertial sensors in clinical gait analysis: A systematic review. IEEE Reviews in Biomedical Engineering **13**, 248–263 (2020)
13. Mancini, M., Horak, F.B.: Quantifying postural stability using wearable sensors: a new approach to balance assessment. Physical Therapy **91**(12), 189–190 (2011)
14. Perry, J., Burnfield, J.M.: Gait Analysis: Normal and Pathological Function. SLACK Incorporated (2010)
15. Ranavolo, A., et al.: Instrumented gait analysis: An overview. European Journal of Physical and Rehabilitation Medicine **53**(4), 575–591 (2017)
16. Trojaniello, D., et al.: Comparative analysis of different methods for the estimation of temporal parameters of gait using wearable inertial sensors. Gait & Posture **42**(3), 360–365 (2015)
17. Wang, L., et al.: Gait recognition using wearable sensors. Sensors **17**(4), 825 (2017)
18. Whittle, M.W.: Gait Analysis: An Introduction. Butterworth-Heinemann, 5th edn. (2014)
19. Winter, D.A.: Biomechanics and Motor Control of Human Movement. Wiley, Hoboken, NJ, 4th edn. (2009)
20. Yock, P.G., Zenios, S., Makower, J., Brinton, T.J., Kumar, U., Watkins, T., Krummel, T.M.: Biodesign: The Process of Innovating Medical Technologies. Cambridge University Press (2015)

A Methodological Framework for Detecting Benevolent Misogynistic Hate Speech in Mexican Political Discourse Using Transformer-Based Models

Monserrat Sánchez-Juárez, Eric Ramos-Aguilar, Daniel Sánchez-Ruiz, Ricardo Ramos-Aguilar

Instituto Politécnico Nacional, Unidad Profesional Interdisciplinaria de Ingeniería
Campus Tlaxcala, Tlaxcala, México
msanchezj1401@alumno.ipn.mx,{eramosa,dsanchezro,rramosa}@ipn.mx

Abstract. This study presents a methodological framework for developing a natural language processing (NLP) model specifically adapted to the Mexican political landscape, aimed at identifying and categorizing misogynistic and sexist hate speech on social media. Despite the widespread presence of such discourse in political environments, there is a marked absence of targeted strategies to address it within this specific domain. The proposed approach involves fine-tuning and training multiple large-scale pre-trained language models—such as MarIA, BETO, BERTIN, mBERT, RoBERTa, and RoBERTuito—using a custom binary-labeled dataset. This dataset is designed to differentiate between benevolent sexist hate speech and neutral or non-sexist language, with particular attention to implicit and coded forms that are often challenging to recognize. By addressing an under explored area in computational political discourse analysis, this work contributes both to academic research and to the development of practical tools that promote safer and more inclusive digital political communication.

Keywords: Sexist Hate Speech, Large Language Model, Natural Language Processing.

1 Introduction

The rise of social media the late decade has already transformed public and political communication by enabling direct and immediate interaction among users. However, this opening has also made easier the discriminatory expression increase among users pointing out misogynistic hate speech, These manifestations not only reinforce harmful gender stereotypes but also perpetuate dynamics of symbolic violence and exclusion, especially against women who actively participate in public life. The United Nations defines hate speech as any kind of communication that attacks or uses pejorative or discriminatory language against a person or group based on inherent identity traits such as gender, race, or religion [24].

Hate speech has historically existed as a tool of exclusion and symbolic violence, used to delegitimize and silence marginalized groups [11]. In political contexts, this phenomenon takes multiple forms, from explicit discriminatory statements to messages loaded with symbolic connotations that reinforce stereotypes related to gender, race, or social class [26].

Misogynistic hate, in particular, manifests in speech that does trivialize, sexualize, or discredit women participating in public life, reinforcing patriarchal power structures [18]. This discursive violence is not limited to formal spaces such as traditional media or parliamentary forums; it has intensified on social media platforms, where anonymity and algorithmic design facilitate its fast spread and normalization [14]. High-profile cases include media violence against figures such as Hillary Clinton in the United States [27], Dilma Rousseff in Brazil [16], and, in Mexico, women like Claudia Sheinbaum the first woman that became a president in Mexico, and Tatiana Clouthier, who have been targeted by systematic smear campaigns with clearly misogynistic overtones [13]. This reality highlights the urgency of analyzing such discourse from intersectional perspectives supported by natural language processing tools.

The reproduction of sexist hate speech has serious consequences at both the individual and societal levels. On a personal level, it can lead to negative psychological impacts on victims, such as anxiety, depression, and low self-esteem, affecting their well-being and limiting their participation in public and professional spheres [8]. On a societal level, sexist hate speech contributes to the perpetuation of gender stereotypes and reinforces structures of discrimination and exclusion, hindering equal opportunities between men and women [3]. Moreover, studies have shown that this type of discourse not only fosters the normalization of gender-based violence, but may also translate into physical assaults and other hate crimes against women [28].

The remainder of this paper is organized as follows: Section 2 presents the context and background of the study, along with a review of related work. Section 3 describes the proposed methodology, including data collection, database creation, text preprocessing, model tuning and training, and the analysis and classification process. Section 4 discusses the evaluation of the results and outlines the expected outcomes. Finally, Section 5 presents the conclusions and suggests directions for future research.

2 Context and Background

In Europe, female politicians are increasingly subject to discrediting tactics. In 2016, 41.8% of European parliamentarians reported having been targeted with humiliating or sexually suggestive images shared on social media [1]. By 2018, that figure had risen to 58.2%, and among respondents under the age of 40, it reached 76.2% [2]. In Spain, and specifically on X formerly Twitter, women from political and communication fields are the focus of 90% of gender-based insults and hate speech. Female politicians in particular receive an average of 15 negative mentions per day [20].

In 2020, the organization “Sapari” and the Media Development Foundation, with support from the United Nations Development Programme, conducted media monitoring of 112 Facebook pages and gathered both quantitative and qualitative sociological data on female politicians in Georgia, USA, regarding sexist hate speech during the pre-election period [25]. The study reviewed prevalent forms of sexist hate speech in Europe and the United States based on European indicators of sexist hate speech, and sought to identify those forms and indicators—presented in Table 1—as adapted to the context of Georgia.

Table 1. Georgian Indicators of Sexist Hate Speech.

Hate Speech Forms	Indicators
Slut-shaming	Reference to a female politician as male politician’s property in any form – because of any male politician referring to women as a "prostitutes"
Spreading sexual photos	Spreading photos or videos of any kind of sexual content without the consent of the female politician
Body-shaming	Mentioning a female body / sexuality, in positive, negative, age-based, having explicitly feminine attire, being beautiful or in any other form
Positive sexism/false compliments	Ironical mocking, masked with political correctness, any kind of compliment that goes beyond work and refers to the visual and representation of a female politician.
Gender role of women	Any attempt to refer to a female politician, as a housewife, culinary specialist, or exhibiting behaviors positively reinforced for men.
Reprimand for family and motherhood	Mentioning a female politician’s childbearing, or her children’s behavior in any form, especially when she either has no children at all, or has only one child, or her child has deviant behavior.

One of the least explored forms is benevolent or positive sexism, usually expressed through "compliments" that involve implicit discriminatory undertones and reinforce gender stereotypes under the guise of politeness or affection. In the context of Mexican Spanish, this phenomenon becomes even more complex due to the use of colloquial language and culturally coded forms of expression, such as albur, which is characterized by double meanings and implicit sexual connotations. These expressions pose a particular challenge for automated hate speech detection systems, as they require a deep understanding of cultural context, figurative language, and the subtleties of everyday speech.

3 Related Work

3.1 General Hate Speech Detection

In recent years, artificial intelligence (AI) and natural language processing (NLP) techniques have enabled significant progress in the automated moderation of digital content. Tools such as CommentGuard [9], Hive Moderation [12], Respondology [22], Netino [17], and Sightengine [23] have been primarily developed to moderate offensive content in English, focusing on categories such as abusive language, racism, Islamophobia, and brand protection. However, these systems show substantial limitations in identifying sexist hate speech, especially in languages other than English.

Numerous efforts have been undertaken to combat online hate through the application of computational intelligence techniques. For instance, [6] utilized a dataset of 16,000 tweets annotated by Waseem and Hovy, which included 3,383 sexist tweets, 1,972 racist tweets, and the remainder labeled as non-offensive. They employed models based on Long Short-Term Memory (LSTM) networks and Gradient Boosted Decision Trees (GBDT), achieving an F1-score of 0.930. Similarly, [19] proposed an ensemble of Recurrent Neural Networks (RNNs) using the same dataset, slightly improving performance with an F1-score of 0.932 by incorporating user behavior data.

Furthermore, [6] compared various machine learning approaches for hate speech detection, including Logistic Regression (LR), Support Vector Machines (SVM), and GBDT, concluding that deep learning models such as CNNs and LSTMs outperformed traditional models by 13% to 20%. In a complementary study, [4] used SVM as a baseline and compared it with CNN, CNN + LSTM, GRU, and CNN + GRU architectures, reporting consistent improvements of at least 7% in accuracy when using CNN.

Recent studies have highlighted the superior performance of BERT in hate speech detection tasks. For instance, [21] found that BERT outperformed FastText, CNN, and LSTM significantly. Similarly, [5] reported better results with BERT compared to LSTM and BiLSTM. Additionally, BERT has shown strong results in multilingual tasks [10, 5].

3.2 Sexist Hate Speech in Political Contexts

In a broader political context, [7] applied NLP techniques for emotion detection and text mining on a corpus of over three million tweets. Their findings revealed that messages directed at female politicians tend to exhibit greater emotional polarity, while male politicians receive slightly higher volumes of hate speech.

Specifically targeting sexist content in Spanish, [15] employed the EXIST 2023 corpus, which includes tweets annotated by gender and age group, to identify sexist content, author intent, and the type of sexism. Using transformer-based models and NLP techniques, they achieved an F1-score of 0.854.

Although these studies mark significant progress, the detection of benevolent sexism—a subtle and often implicit form of gender-based discrimination—within Mexican political discourse remains under explored.

3.3 Challenges in Spanish-Language NLP

In the case of Spanish—particularly Mexican Spanish—the development of culturally and linguistically adapted NLP models is still in early stages. Sexist hate speech in Spanish social media often incorporates local expressions, irony, and euphemistic language, which generalist models fail to detect accurately.

Although some studies have addressed hate speech detection in Spanish (e.g., [15]), the literature on benevolent misogynistic discourse in the Mexican political context is notably scarce, highlighting the need for domain-specific approaches. This lack of specialized tools hinders effective content moderation and timely responses to digital gender-based violence, which disproportionately affects women in political and media spheres.

4 Methodological Framework

The methodology proposed is structured into three consecutive phases, detailed below and illustrated in Fig.1

The proposed methodology for hate speech detection consists of the following sequential phases:

1. Data Collecting: Collecting relevant text data from social media.
2. Database creation: Organizing and annotating the collected data to build a labeled custom dataset.
3. Text preprocessing: Cleaning and normalizing the text to prepare it for analysis.
4. Model tuning and training: Adjusting model parameters to detect hate speech.
5. Analysis and classification: Applying the trained models to classify text as hateful or non-hateful.
6. Results evaluation: Evaluating the model's performance using standard metrics like precision, recall, and F1-score.

4.1 Data Collecting

The first phase involves collecting representative textual data of both misogynistic and non-misogynistic speech. Data sources include social media platforms such as X (formerly Twitter) and Facebook, as well as public corpora from existing databases focused on hate speech in Spanish. The data collection process employs the following methods:

- **Web scraping.** Automated extraction of textual content from relevant websites and online forums where political discourse occurs.
- **API usage.** Structured access to large volumes of public posts using the X API, focusing on content generated in Mexico.
- **Preexisting databases.** Integration of publicly available corpora such as the *EXIST 2023* dataset and *HateEval*, which contain manually annotated examples of hate speech in Spanish.

Text selection is guided by specific inclusion and exclusion criteria. Only messages authored in Mexican Spanish that are public, non-redundant, and directed at or referring to Mexican female political figures (e.g., senators, deputies, governors) are considered.

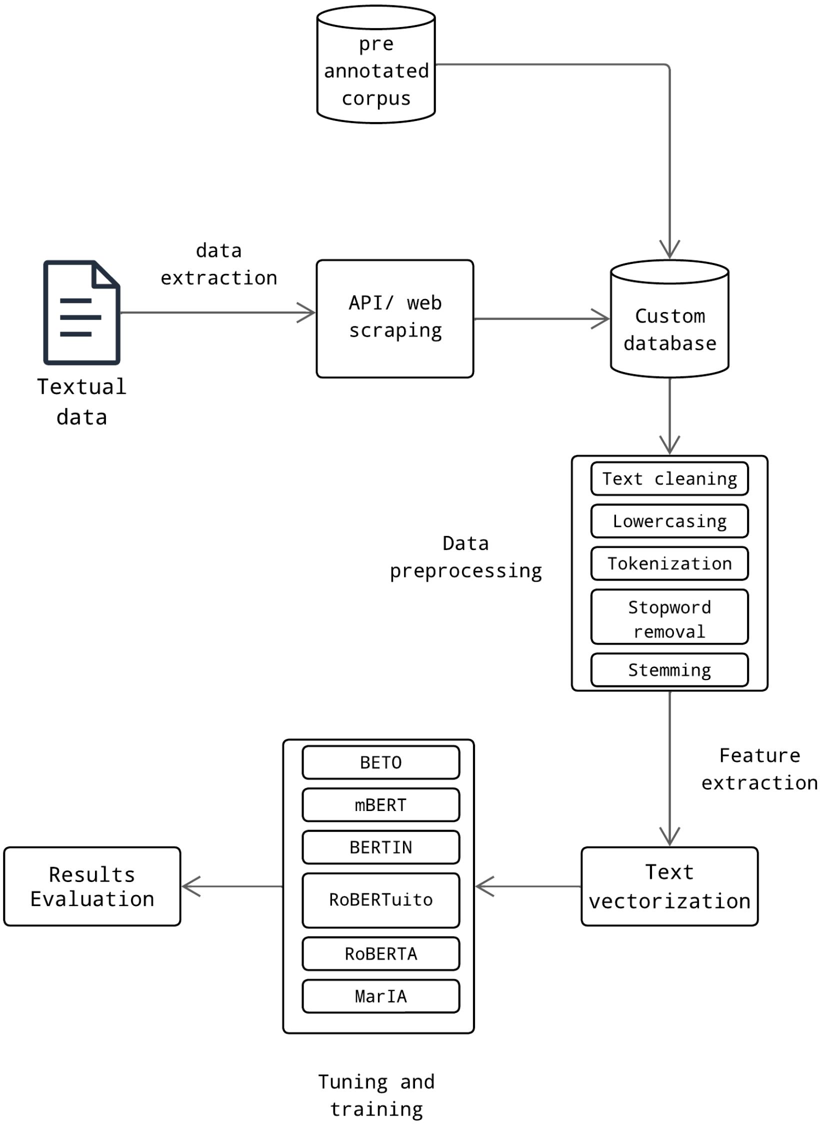


Fig. 1. Proposed methodology for the detection of hate speech in Mexican political discourse.

Examples of inclusion keywords are: *senadora, diputada, candidata, gobernadora, feminista, igualdad, paridad, INE, Congreso, morena, PRI, PAN, oposición, política mexicana.*

The initial dataset is expected to include approximately 2,000 posts from X and Facebook, spanning the years 2023–2024. Manual annotation will be carried out by experts in gender studies and computational linguistics to ensure accurate labeling of each instance as either *benevolent sexist*, or *non-misogynistic* content.

To ensure representativeness and balance, the dataset will be stratified to maintain a proportional distribution between the three classes mentioned above. Sampling strategies will also account for temporal diversity (e.g., during electoral periods) and thematic variety (e.g., policy debates, campaign events, controversies).

This curated dataset will serve as the foundation for the subsequent preprocessing and modeling stages.

4.2 Database Creation

Once the information has been collected, it is structured into a machine-readable format, typically in a CSV file. In this database, each row corresponds to an individual text entry, and the columns contain the text and a binary label: Benevolent Misogynistic Sexist (SBM) or Non-SBM (NSBM).

The binary labeling simplifies the classification task in the initial stage, deferring the application of multi-labels or subcategories to subsequent phases. An example of this dataset is presented in Table 2.

Table 2. Initial sample of the training corpus. SBM: Benevolent Misogynistic Sexist. NSBM: Non-Benevolent Misogynistic Sexist or neutral.

Text	SBM	NSBM
Qué suerte que tenemos una Presirvienta científicA ... @Claudiashein	X	
Vamos a tener un ama de casa 6 años	X	
No se les olvide que López Obrador y Claudia Sheinbaum robaron en el segundo piso		X
La Presirvienta de todos los mexicanos siempre tan cariñosa con la gente	X	

Although hostile language is not employed, benevolent sexist expressions subtly undermine the president by evaluating her through traditional gender roles rather than her professional competence or political actions.

To mitigate subjective bias in the labeling process, multiple annotators will participate in the manual classification of the dataset. Labels will be assigned according to predefined categories: *benevolent sexist*, and *non-misogynistic.* All annotators will do detailed annotation guideline developed in collaboration with a psychologist specialized in gender perspective.

To ensure annotation consistency, a subset of 20% of the corpus will be independently labeled by three annotators. Inter-annotator agreement will be evaluated using Cohen's Kappa coefficient, which quantifies the degree of agreement beyond chance. A Kappa score above 0.75 will be considered indicative of substantial agreement.

In cases of disagreement, a resolution protocol will be followed: conflicting instances will be reviewed and discussed in consensus meetings moderated by the gender expert. Final labels will be assigned based on agreement reached during these sessions.

This multi-annotator strategy, combined with quantitative agreement metrics and expert oversight, aims to enhance the reliability and validity of the labeled dataset.

4.3 Text Preprocessing

The collected texts, characterized by informal language, typographical errors, emojis, symbols, user tags (@), hashtags, and other elements, undergo a series of preprocessing steps to normalize the data and prepare it for analysis. These stages are shown in Fig. 2.

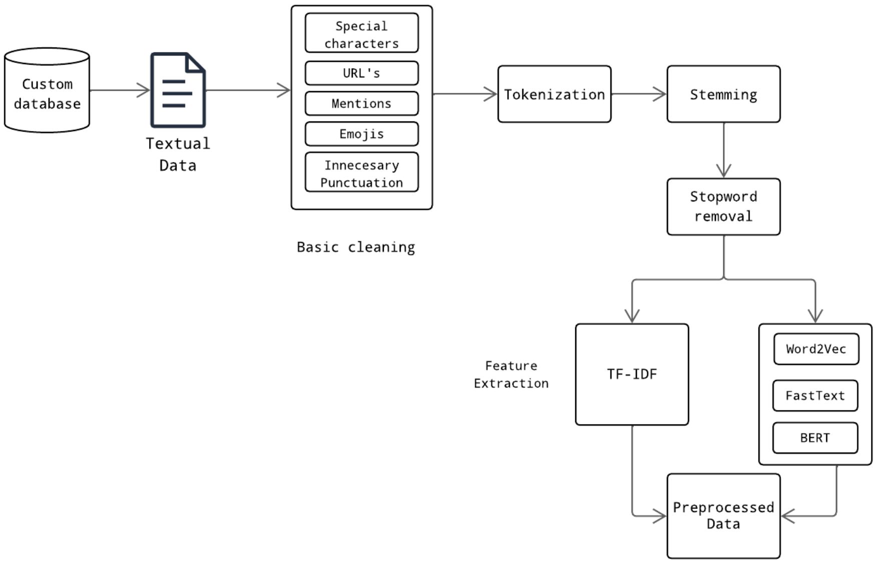


Fig. 2. Text preprocessing stages for textual data.

The preprocessing steps outlined above enable large language models (LLMs) to capture semantic relationships between words, thereby enhancing the predictive performance of the system.

4.4 Model Training and Fine Tuning

Once the texts have been preprocessed and converted into numerical representations, several language models will be trained. The selected models include pretrained Spanish or multilingual transformers that have shown strong performance in similar tasks, such as: BETO, mBERT, BERTIN, RoBERTuito, RoBERTa, MarIA.

These models will be fine-tuned for the binary classification task by adapting their internal parameters to the specific corpus of this project. To avoid overfitting, cross-validation will be applied, and key hyperparameters such as learning rate, number of epochs, batch size, and loss function will be optimized.

4.5 Text Analysis and Classification

The trained model is employed to classify new texts as either *benevolent misogynistic* or *non-misogynistic*. Once predictions are obtained, the outputs are analyzed to interpret the results, identify frequent misclassification patterns, and assess the model's generalization capabilities. To further enhance interpretability, SHAP (SHapley Additive exPlanations) values will be computed to determine the relative contribution of specific terms or phrases to the model's decisions. These explainability techniques provide transparency and support a deeper understanding of the underlying decision-making process.

The analysis of SHAP values will focus on extracting key linguistic patterns that are strongly associated with benevolent misogyny. For instance, words or expressions such as *"ama de casa"*, *"cariñosa"*, or *"débil"*, *"presirvienta"* may emerge as recurrent indicators in positively labeled predictions. These findings will be compiled and synthesized in the form of accessible reports and visual summaries tailored to non-technical audiences. The findings aim to provide policy stakeholders and civil society organizations with actionable linguistic evidence on the presence of benevolent sexism in political communication, facilitating more effective interventions against online gender violence.

However, one of the primary challenges in modeling this task lies in the class imbalance inherent to the dataset, especially given the relatively subtle and less frequent nature of benevolent sexist expressions. To mitigate this issue, multiple strategies will be implemented. First, the Synthetic Minority Over-sampling Technique (SMOTE) will be applied during training to generate synthetic samples of the minority class and balance the dataset distribution. Additionally, experiments with random undersampling of the majority class will be conducted to evaluate potential trade-offs in model performance.

To further support balanced learning, the loss function will incorporate class weighting, assigning greater penalization to errors involving the underrepresented class. Model evaluation will rely on weighted performance metrics, including the weighted F1-score, precision, and recall, to ensure that results reflect both the accuracy and fairness of the classifier. Together, these methods aim to improve the model's ability to detect nuanced patterns of benevolent misogyny while maintaining general robustness.

To evaluate the model's performance, the following standard evaluation metrics are show in Table 3.

Table 3. Evaluation metrics to measure the model's performance.

Metric	Description
Precision	The proportion of correctly predicted misogynistic instances among all instances predicted as misogynistic.
Recall	The proportion of actual misogynistic instances that were correctly identified by the model.
F1-score	The harmonic mean of precision and recall, offering a balanced measure of performance.
Confusion Matrix	A representation that displays true positives, false positives, true negatives, and false negatives to visualize the model's overall performance.

5 Anticipated Outcomes and Impact

The proposed methodology is expected to enable the effective classification of political discourse through the use of fine-tuned transformer-based models. These models aim to accurately distinguish between *benevolent misogynistic* and *non-misogynistic* speech within the specific context of Mexican political communication. Leveraging pre-trained Spanish-language models such as BETO, mBERT, and RoBERTuito, combined with rigorous preprocessing and interpretability techniques (e.g., SHAP), is anticipated to yield strong results across standard evaluation metrics such as precision, recall, and F1-score.

Beyond performance metrics, the project seeks to uncover linguistic patterns and implicit expressions of gender bias, addressing a significant gap in the computational analysis of non-hostile forms of sexist speech. The findings are intended to contribute to the development of safer and more inclusive digital environments, particularly for women engaging in political discourse online.

To assess the generalization capacity of the models across Spanish dialects, cross-linguistic transfer experiments will be conducted. Specifically, the model will be evaluated on a subset of 1,000 tweets written in Peninsular Spanish from the EXIST 2023 corpus. Fine-tuning of BETO and RoBERTuito will be performed as needed, allowing for the identification of dialectal variations and necessary linguistic adaptations.

In terms of practical applications, the trained models will be integrated into a prototype content moderation tool for social media platforms such as

X. This tool will enable detection of benevolent misogynistic discourse, offering a foundation for digital interventions. Furthermore, collaboration with public institutions such as the Instituto Nacional Electoral (INE) or INMUJERES is envisioned, particularly for supporting awareness campaigns and policy-making initiatives aimed at combating political gender-based violence.

Ultimately, this work bridges the gap between computational modeling and actionable policy, highlighting the potential of NLP technologies to foster more equitable and respectful political communication online.

6 Conclusions

This study presents a structured methodology for the detection of *benevolent misogynistic discourse* in Mexican political speech using advanced natural language processing techniques. By targeting a form of hate speech that often goes unrecognized—subtle, non-aggressive language that reinforces gender stereotypes—the research addresses a critical and underexplored dimension of online misogyny. The proposed pipeline, which encompasses data collection, preprocessing, model fine-tuning, and interpretability analysis, enables not only accurate classification but also the extraction of valuable insights into the persistence of gender bias within digital narratives.

The outcomes of this work are intended to inform the development of analytical and moderation tools that promote equitable political communication and support the creation of safer digital environments for women's participation in the public sphere. Furthermore, the methodological framework lays the groundwork for several future extensions. These include the integration of multi-label classification schemes that could capture overlapping categories of sexist discourse (e.g., combining benevolent and hostile traits), as well as the adaptation of the pipeline to other linguistic and cultural contexts. For instance, cross-lingual transfer to other languages may be explored by leveraging multilingual transformer models and domain-specific corpora.

By expanding the scope and granularity of hate speech detection, future work may contribute to a more nuanced and globally adaptable approach to combating digital gender-based violence.

References

1. Sexism, harassment and violence against women parliamentarians. Tech. rep., Inter-Parliamentary Union (2016), `https://www.ipu.org/resources/publications/issue-briefs/2016-10/sexism-harassment-and-violence-against-women-parliamentarians`
2. Sexism, harassment and violence against women in parliaments in europe. Tech. rep., Inter-Parliamentary Union (IPU) and Parliamentary Assembly of the Council of Europe (PACE) (2018), `https://www.ipu.org/resources/publications/reports/2018-10/sexism-harassment-and-violence-against-women-in-parliaments-in-europe`

3. Al-Hassan, A., Al-Dossari, H.: Detection of hate speech in social networks: a survey on multilingual corpus. In: 6th international conference on computer science and information technology. vol. 10, pp. 10–5121. ACM (2019)
4. Al-Hassan, A., Al-Dossari, H.: Detection of hate speech in arabic tweets using deep learning. Multimedia systems **28**(6), 1963–1974 (2022)
5. Alatawi, H.S., Alhothali, A.M., Moria, K.M.: Detecting white supremacist hate speech using domain specific word embedding with deep learning and bert. IEEE Access **9**, 106363–106374 (2021)
6. Badjatiya, P., Gupta, S., Gupta, M., Varma, V.: Deep learning for hate speech detection in tweets. In: Proceedings of the 26th International Conference on World Wide Web Companion. pp. 759–760. International World Wide Web Conferences Steering Committee (2017)
7. Blanco-Alfonso, I., Rodríguez-Fernández, L., Arce-García, S.: Polarización y discurso de odio con sesgo de género asociado a la política: análisis de las interacciones en twitter. Revista de Comunicación **21**(2), 33–50 (2022)
8. Chen, Y., Zhou, Y., Zhu, S., Xu, H.: Detecting offensive language in social media to protect adolescent online safety. In: 2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Conference on Social Computing. pp. 71–80. IEEE (2012)
9. CommentGuard: The #1 comment moderation tool for facebook & instagram (2025), `https://commentguard.io/`
10. Dowlagar, S., Mamidi, R.: Hasocone@ fire-hasoc2020: Using bert and multilingual bert models for hate speech detection. In: Forum for Information Retrieval Evaluation. pp. 1–8 (2021)
11. Gagliardone, I., Gal, D., Alves, T., Martinez, G.: Countering Online Hate Speech, UNESCO Series on Internet Freedom, vol. 16. UNESCO Publishing, Paris (2015)
12. Hive Moderation: Hive moderation (2025), `https://hivemoderation.com/`
13. Instituto Nacional Electoral (INE): Violencia política contra las mujeres (2023), `https://igualdad.ine.mx/mujeres-en-la-politica/violencia-politica/`
14. Jane, E.: Misogyny Online: A Short (and Brutish) History. SAGE Publications Ltd (2016)
15. Martínez, M.P.J., López-Nava, I.H., y Gómez, M.M.: Identificación de sexismo en redes sociales. Tesis de maestría en ciencias, Centro de Investigación Científica y de Educación Superior de Ensenada (CICESE), Ensenada, Baja California, México (2025)
16. Meneguelli, G., Ferré Pavia, C.: El discurso de odio contra dilma rousseff desde la perspectiva semiolingüística. Estudos Feministas **32**(1), 187865 (2024)
17. Netino by Concentrix: Content & marketing services (2024), `https://netino.fr/en/`
18. ONU Mujeres: Violencia contra las mujeres en política (2023), `https://lac.unwomen.org/sites/default/files/2023-06/5eeb7511-c851-4b46-a15d-0089190e14a6.pdf`
19. Pitsilis, G.K., Ramampiaro, H., Langseth, H.: Effective hate-speech detection in twitter data using recurrent neural networks. Applied Intelligence **48**(12), 4730–4742 (2018)
20. Piñeiro-Otero, T., Martínez-Rolán, X.: Eso no me lo dices en la calle. análisis del discurso del odio contra las mujeres en twitter. Profesional de la Información **30**(5), 1–17 (2021)
21. Ranasinghe, T., Zampieri, M., Hettiarachchi, H.: Brums at hasoc 2019: Deep learning models for multilingual hate speech and offensive language identification. In: FIRE Working Notes. vol. 2517, pp. 199–207 (2019)

22. Respondology: Social media comment activation platform (2025), https://respondology.com/
23. Sightengine: Detect nudity, porn, suggestive and explicit adult content in images and videos (2025), https://sightengine.com/nudity-detection-api
24. United Nations: ¿Qué es el discurso de odio? (2022), "https://www.un.org/es/hate-speech/understanding-hate-speech/what-is-hate-speech
25. Urchukhishvili, G.: Indicators of sexist hate speech (2020), undp.org/sites/g/files/zskgke326/files/migration/ge/undp_ge_sexist_language_indicators_sapari_eng.pdf
26. Van Dijk, T.A.: Discourse as Social Interaction. SAGE Publications (1997)
27. Weaving, M., Alshaabi, T., Arnold, M.V., Blake, K., Danforth, C.M., Dodds, P.S., Haslam, N., Fine, C.: Twitter misogyny associated with hillary clinton increased throughout the 2016 u.s. election campaign. Scientific Reports **13**(1), 5266 (2023)
28. Wigand, C., Voin, M.: Speech by commissioner jourová–10 years of the eu fundamental rights agency: A call to action in defence of fundamental rights, democracy and the rule of law (2017), https://ec.europa.eu/

Instrument Recognition Using Spectral Features and SVM

María Fernanda Arámburo-Castell, Juan Pintor-Michimani

Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Computación, Puebla, Pue., México
{maria.aramburoc,juan.pintorm}@alumno.buap.mx

Abstract. The identification of musical instruments is important in audio analysis this allows musical information to be retrieved and identified. For this purpose, it is important to consider the processing of audio signals. In this study, we propose to use spectral parameters such as Mel Frequency Cepstral Coefficients (MFCC), Mel Spectrogram, Chromagram, Harmonic Percussive Index and spectral contrast to capture instrument characteristics. These features model the timbre, harmonic content and energy distribution essential for differentiation. Accurate extraction and processing of these features is essential, as errors can compromise classification performance in complex, polyphonic soundscapes.

Keywords: Instrument Recognition, MFCC, Spectral Features, SVM.

1 Introduction

Instrument recognition is an important concern in audio analysis, especially in music information retrieval, sound source separation and automatic music transcription [15]. The issue is complicated by the fact that most audio signals are polyphonic, i.e. they are a combination of two or more sources, in this case instruments. For this reason, the majority of research divides the monophonic and polyphonic cases into two distinct problems, with the principal objective being to ascertain the instrument or the predominant instrument, respectively.

This is of particular importance given that the predominant instrument is a powerful description of musical data that can be used to identify songs. The application's utility in identifying musical genres is noteworthy, particularly given that certain instruments are often used as distinctive characteristics [4]. This is a subject on which a substantial amount of research has been conducted.

In the case of polyphonic music, there is research by Han et al. [8], who developed a deep CNN for instrument recognition based on Mel spectrogram inputs and aggregation of multiple sliding window outputs on the audio data.

Another interesting research is [7] where the researchers proposed a method for automatic recognition of predominant instruments using SVM (Support Vector Machine) classifiers trained with features extracted from real musical audio signals. Similarly, in [2], an approach is proposed to automatically identify all instruments present in an audio signal using sets of individual convolutional neural networks (CNNs) per tested instrument, which is a similar approach to that used by Avramidis K. et al. [1] who use RNN (recurrent neural networks), CNN (convolutional neural networks) and CRNN (convolutional recurrent neural networks) to find the dominant instrument, similar to the work of Pons J. et al. [14] who do the same using CNNs with MFCC.

In regard to the selection of parameters, the most commonly utilized are the MFCC (Mel-frequency cepstral coefficient). However, as evidenced in [5, 6], alternative options exist, including the MEL spectrogram, the Chromagram, the Harmonic Percussive Index (HPI), and Spectral Contrast. In this study, the aforementioned parameters are employed to identify the most significant parameters for the developed model, which in this case will be a Support Vector Machine (SVM).This is accomplished by employing specific algorithms, such as Recursive Feature Elimination, or statistical measures like the ANOVA F-value, to identify the most relevant parameters.

2 Methodology

This paper proposes a methodology to train an SVM for both monophonic and polyphonic audio signals. This can be separated in three steps, the pre processing of the signal, extraction of characteristics and the training of model. Each step is relevant, because depending on the chosen parameters the performance of the model can change significantly. In order to expose that, we separate each step in different sub process, this can be seen in Fig. 1, where the three main steps mentioned above are enclosed in a frame and there are the mentioned sub process. Each of them is described in detail below.

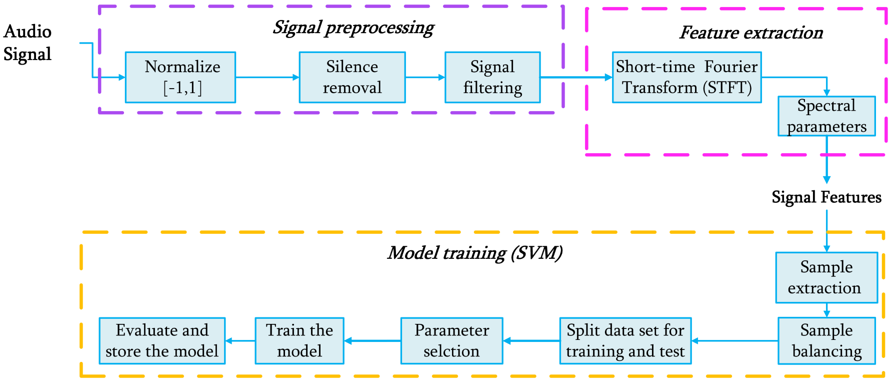


Fig. 1. Proposed methodology.

2.1 Signal Pre-processing

Signal pre-processing is a particularly important step as it prepares the signal to extract its characteristics. In this step, the input signal is in its raw state, which usually means that it contains noise, has different scales, etc., so it is necessary to normalize and filter the signal before analyzing it. This procedure makes possible to observe the characteristics of the signals and to determine some properties for their subsequent analysis, such as the selection of the amplitude of the windows in which the analysis will be carried out.

Normalize and Silence removal. Normalization of the signal involves modifying its range to align it with the desired range, which in this case is $[-1, 1]$. This can be done through the expression (1):

$$y_{nom} = \frac{y}{max(|y|)}. \tag{1}$$

where y is the original signal, $max(|y|)$ is the maximum absolute value of the signal and y_{nom} is the normalized signal.

Once the signal has been normalized, the next step is to trim the samples in which the signal works with less than $20dB$. In Fig. 2 you can see the normalized signal and the signal after trimming the silences. This Fig. shows how the audio signal of an instrument, in this case viola, changes when we remove the silence spaces, which are especially noticeable at the beginning and at the end of the original signal.

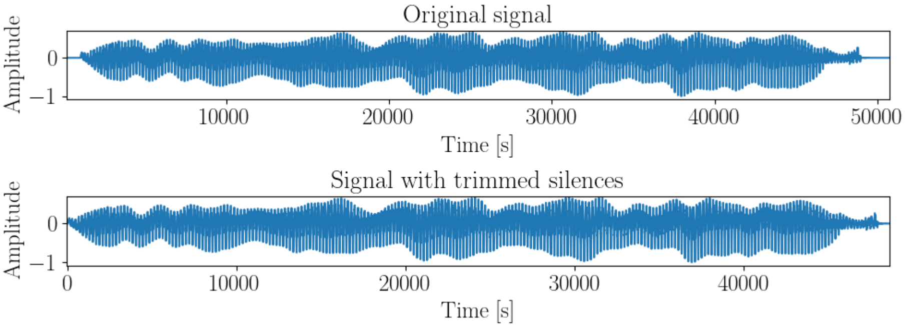


Fig. 2. Original signal and signal resulting from the removal of silences.

Window and Filter. The selection of the window length is typically contingent upon the filter selected from the signal, as both are employed to determine the requisite frequency range for analyzing the interest patron. In this particular instance, it is necessary to select a uniform window for all the instruments, given that a specific pass band filter is proposed for each instrument, with consideration given to their respective work frequency range. Table 1 illustrates the range of each instrument and the period from the slowest frequency, which must be no greater than half of the selected window.

As can be observed, the majority of the values are less than 15 ms, although in certain instances, such as those pertaining to the viola, acoustic guitar, and piano, the values exceed 30 ms. In order to accommodate these values, a window of at least 100 ms is necessary. However, utilizing a longer window may result in the loss of information from the other instruments. Consequently, the recommendation proposed in reference to this matter is to set the window length at 40 ms, with the range from the filter for each instrument corresponding to its respective range of operation.

Table 1. Frequency ranges of instruments [9, 11].

Instrument	F. Min (Hz)	F. Max (Hz)	Signal period (ms)
Cello	65	1000	15.3846
Clarinet	125	2000	8.0000
Flute	250	3500	4.0000
Acoustic Guitar	20	5000	50.0000
Electric Guitar	80	500	12.5000
Piano	27	5000	37.0370
Saxophone	110	2000	9.0909
Trumpet	165	1200	6.0606
Violin	196	3000	5.1020
Viola	130	1000	7.6923
Percussion	30	5000	33.3333

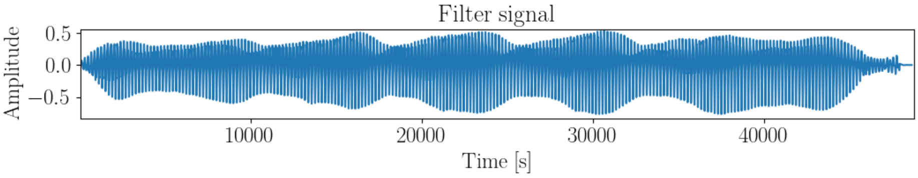


Fig. 3. After-filtered viola audio signal.

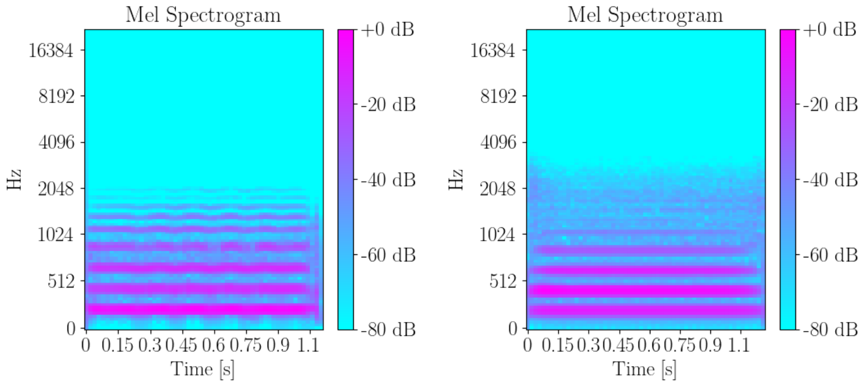
Fig. 3 presents an illustration of the signal subsequent to filtration, which correlates with the signal depicted in Fig. 2. Upon examination of both figures, the impact of the filtration process on the audio signal of the instrument becomes evident.

2.2 Feature Extraction

In the audio signal analysis is usual to use the short-time Fourier Transform, which main idea is to consider the changes in frequency in small periods of time [12], to this it is needed to select the length of those periods, for this case are 40ms and a window function which is multiply for the filter signal to obtain the frequency information. This window shifts across the time and compute the Fourier transform for each resulting window [12] . Part of this process is to select the overlap, which means how much advance the window in the time, for example, the first window is $[0ms, 40ms]$ and the second will be $[20ms, 60ms]$ with overlap of 50% and with overlap of 75% the second window will be $[10ms, 50ms]$.

For each time frame, it is possible to obtain a spectral vector with coefficients associated with a time position. This allows a two-dimensional representation of the squared magnitude of the STFT call spectogram to be plotted, where the horizontal axis represents time and the vertical axis represents frequency [12].

Log-Mel Spectrogram. The Log-Mel spectrogram is a representation that condenses timbre and pitch information computed from the above spectrogram by grouping STFT bins into overlapping frequency bands that approximate human pitch perception [10]. The number of bands is significantly less than STFT, which is also an advantage when selecting them as parameters.



(a) Mel Spectrogram for viola.

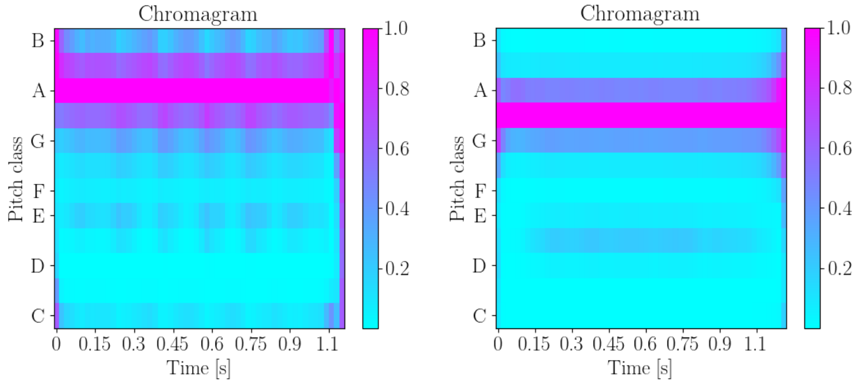
(b) Mel Spectrogram for saxophone.

Fig. 4. Mel spectrograms for viola and saxophone for the note A3.

As a result, these parameters give figures as can be seen in Figures 4a and 4b, which are the Mel spectograms for the note A3 for viola and saxophone. As can be seen, they are quite similar, but there are some notable differences in the frequencies from 1024 Hz to 4096 Hz.

Mel Frequency Cepstral Coefficients (MFCCs). The Mel Frequency Ceptral Coefficients are a compact representation of the shape and spectral envelope of an audio signal, calculated from the Mel spectogram by taking the logarithm of the magnitude of each resulting band and calculating the discrete cosine transform over the resulting band. The resulting real part is similar to the real part of the Fourier transform. The fascinating thing about this parameter is that a small subset contains the most important information, so usually between thirteen and twenty parameters are considered. In this work, it is used twenty parameters.

Chromagram. The Chromagram is a magnitude spectogram similar to the MEL spectogram, the main difference being that the MEL spectogram considers the frequency range, as the Chromagram defines twelve different pitch classes, where each corresponding to a particular frequency range [12]. For example, note A5 has a range of [427.47, 452.89]. Figures 5a and 5b show the chromagram for note A3 on viola and saxophone. It can be seen that both are quite similar, although there are some differences on the note B in the case of the viola.

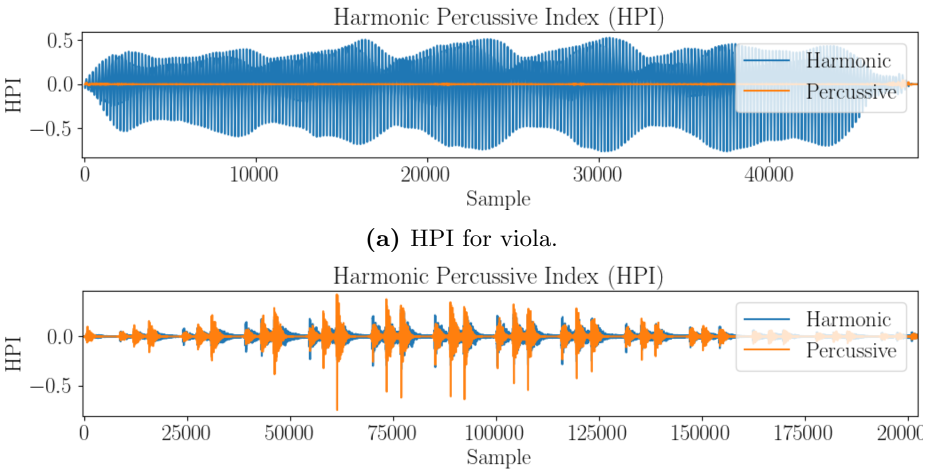


(a) Chromagram for viola. **(b)** Chromagram for saxophone.

Fig. 5. Chromagram for note A3 for viola and saxophone.

Harmonic Percussive Index (HPI). Musical instruments can be divided into percussive and melodic instruments, the former being characterized by the fact that they can generate vibrations on their own, whereas harmonics require a string or wind to vibrate. This characteristic gives rise to the harmonic and percussive index, which is calculated by separating the harmonic and percussive parts for each window [5]. Figures 6a and 6b show the separation of the harmonic and percussive parts of a viola and a drum. It can be seen that the drum has a lot of percussive energy, while the viola has almost zero.

Spectral Contrast. Spectral contrast characteristics are an important parameter because they provide a representation of the spectral characteristics of the

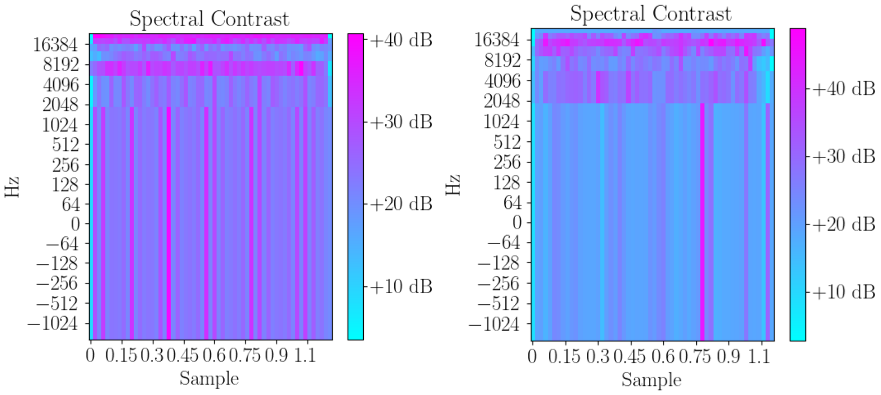


(a) HPI for viola.

(b) HPI for drums.

Fig. 6. Harmonic percussive index for viola and drums.

sound by highlighting the differences between peak and valley energies in different frequency bands. This method emphasizes the relative distribution of spectral energy, which can vary significantly between different types of musical instrument [5]. Figures 7a and 7b show the spectral contrast for note A3 on viola and saxophone. It can be seen that both are quite similar, although there are some differences, specially considering that the time scale are different.



(a) Spectral Contrast for viola.

(b) Spectral Contrast for saxophone.

Fig. 7. Spectral Contrast for note A3 for viola and saxophone.

Once all the parameters have been calculated, it is necessary to organize them into a dataset in order to be able to use this information in any model. For each window calculated for each audio signal, a vector is generated with the MFCC vector, the MEL spectogram, the chromogram and the spectral contrast as concatenated vectors and the HPI, so that for each case there is a vector of 169 values.

2.3 Support Vector Machines

The problem of instrument recognition is studied with different approaches as RNN, CNN, RNN, CRNN and SVM. For this paper, the selected model was SVM because this allows to quickly train a diversity of models with some variations, like filter or not the signal, change the type of model, the size of data set, etc. Some of these variations are considering in Fig. 8, this scheme mentions some considerations to take in count:

- **Sample extraction.** The datasets contain a lot of information, which is useful for training robust models with deep learning, but for this particular case, it is necessary to take a sample of all the data in order to optimize the training. In this case, two samples are taken, one with 10% and the other with 2.5%.
- **Sample balancing.** In most cases, the datasets do not have the same number of cases from each class, which can lead to poor results.For this, it is necessary to consider oversampling or under sampling in order to have the same number of cases for each class. In this case, the dateset is under sampled. For this, the number of cases for each class is the same as the percentage selected for the sample of the smallest class.
- **Split data.** The data set used to train the model is split into two sets, one for training the model and one for testing it. Some common percentages are 80% and 20%. Since the number of data in this work, the chosen percentages are 85% and 15%.
- **Parameter selection.** The dataset extracted by this methodology have 169 parameters, which is a high number and increase the time to tain the model, for that is necessary to use some technique that reduce the number of parameters on option is Recursive Elimination Feature, but as the name says, it probes the parameters in the model recursively, which leads a hight computational cost. Another option available is the SelectKBest function in Python, which can select the best parameters using statistical techniques such as ANOVA or Mutual Information (MI).
- **Kernel for training the model.** SVM can be trained considering different kernel functions whose are related with how to measure the distance and the expected behaviour of the data. In this case, with linear data, the linear kernel is the best option, while with complex data, such as audio, the radial basis function kernel is a great option.
- **Evaluation of the model.** A train model needs to be evaluated to see how close it is to the test data. To do this, the confusion matrix is a good option because it gives information about the accuracy for each class.

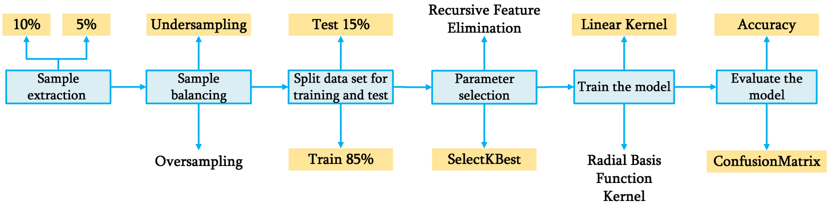


Fig. 8. Proposed methodology to train the SVM.

After considering these factors, different models are trained and evaluated using selected combinations of features. This study investigates model performance under various configurations of kernel, feature selector, and the application of filtering techniques.

3 Results

The proposed methodology is implemented for two datasets, the first one is the dataset IRMAS, which is a polyphonic dataset with eleven class, for this work only nine were selected: cello (cel), clarinet (cla), flute (flu), acoustic guitar (gac), electric guitar (gel), piano (pia), saxophone (sax), trumpet (tru), violin (vio) [3]. The second is the Philarmonia dataset, a monophonic dataset that has twenty classes, only nine of which are used in this work: cello (cel), clarinet (cla), flute (flu), acoustic guitar (gac), percussion (gel), saxophone (sax), trumpet (tru), violin (vio) and viola (viola) [13]. For each data set, all audio signals of each class were analyzed and their features were extracted with a window length of 40 ms and an overlap of 50%, i.e., 20 ms. The sizes of the datasets were 784,802 × 169 for the IRMAS dataset and 552,051 × 169 for the Philharmonia dataset. As mentioned above, the data sets are huge and the computational cost of training the model is high. For this, after sampling the dataset with 10% and 2.5% and under sampling the classes, the remaining datasets are shown in Table 2.

Table 2. Samples for balanced datasets.

Dataset	**2.5 %**	**10 %**
IRMAS	13077	52308
Philharmonia	2673	10710

It is now feasible to train models with reduced data sets, which enables the evaluation of various factors such as the kernel type, feature selector, and the number of parameters.

As shown in Table 3, a comprehensive overview of different configurations is provided. The results indicate that, in both datasets, using only 2.5% of the data, the optimal kernel is the linear one, and the most effective feature selector is f classif. This trend is especially evident in the IRMAS dataset, where a significant drop in accuracy is observed when switching the feature selector. In contrast, under the same conditions, the Philarmonia dataset yields similar results when using the RBF kernel, indicating a more stable behavior with respect to kernel changes.

Moreover, as illustrated in Table 3, the implementation of filters results in a substantial enhancement in performance. The employment of the optimal kernel (linear) and feature selector (f class) for both data sets has been demonstrated to result in enhanced accuracy when utilizing the filters. The Irmas dataset demonstrated an accuracy of 0.84% for the 2.5% sample, while the Philharmonia Dataset exhibited an accuracy of 0.75% for the same percentage. Conversely, when filters are not utilized, the accuracies decrease to 0.21% and 0.65%, respectively, indicating a substantial decline in performance.

Table 3. Model results with different data configurations by changing the kernel, feature selector and number of features (NF).

Dataset	%	Filters	Accuracy	Kernel	Feature Selector	NF
IRMAS	10	✓	0.72	Lineal	f classif	10
IRMAS	2.5	✓	0.75	Lineal	f classif	10
IRMAS	2.5	✓	0.21	Lineal	mutual info classif	10
IRMAS	2.5		0.26	Lineal	f classif	10
IRMAS	2.5	✓	0.23	RBF	mutual info classif	10
IRMAS	2.5	✓	0.21	RBF	mutual info classif	5
PHIL	10	✓	0.83	Lineal	f classif	10
PHIL	2.5	✓	0.84	Lineal	f classif	10
PHIL	2.5	✓	0.68	Lineal	mutual info classif	10
PHIL	2.5		0.65	Lineal	f classif	10
PHIL	2.5	✓	0.68	RBF	mutual info classif	10
PHIL	2.5	✓	0.70	RBF	mutual info classif	5

Another interesting comparison is the effect that dataset size has on model accuracy. As shown, the accuracy is 0.72 for the IRMAS dataset and 0.83 for the Philharmonia dataset when 10% of the data is used. With only 2.5% of the data, the accuracies are 0.75% and 0.84%, respectively. The model trained with filtered signals, lineal kernel and f classif feature selector, has performed well, but to prove that this is true in other circumstances, another sample of 5% is taken from each complete dataset, this to prepare a cross validation with different percentages from the new datasets. Table 4 shows the results from samples of 10%, 5% and 1% of the news datasets. It can be seen that the IRMAS dataset has less accuracy with less data, since the Philarmonia dataset has similar values.

Table 4. Samples for balanced datasets.

Dataset	Sample	Train	10%	5%	1%
IRMAS	10	0.72	0.72	0.71	0.73
IRMAS	2.5	0.75	0.75	0.7	0.65
PHIL	10	0.83	0.84	0.86	0.84
PHIL	2.5	0.84	0.83	0.83	0.83

Table 5 shows a compendium of other test models that have been computed, where it can be seen that the parameters used are MFCC or part of the MEL spectrum with some cases of spectral contrast. None of these use the chromatic diagram and only one uses the HPI.

Table 5. Results of models with different data configurations The selected characteristics are listed.

Dataset	IRMAS	IRMAS	IRMAS	PHIL	PHIL	PHIL
Percentage	10	2.5	2.5	10	2.5	2.5
Accuracy	0.72	0.75	0.26	0.83	0.84	0.65
Filters	✓	✓		✓	✓	
Features	MFCC 3	MFCC 3	MFCC 1	MFCC_3	MFCC_3	MFCC 3
	MFCC 5	MFCC 4	MFCC 3	MFCC_4	MFCC_4	MFCC 4
	MEL 1	MFCC 5	MFCC 4	MFCC_5	MFCC_5	MEL 1
	MEL 2	MEL 1	MEL 71	MFCC_6	MFCC_6	MEL 2
	MEL 67	MEL 2	MEL 73	MFCC_7	MFCC_7	MEL 3
	MEL 69	MEL 67	MEL 75	MEL 1	MEL 1	MEL 127
	MEL 71	MEL 69	MEL 76	MEL 2	MEL 2	MEL 128
	MEL 72	MEL 71	MEL 77	MEL 3	MEL 3	CONTR 1
	MEL 73	MEL 72	MEL 78	MEL 4	MEL 4	CONTR 7
	CONTR 7	CONTR 7	MEL 79	CONTR 7	CONTR 7	HPI

4 Conclusions

The findings presented above, along with the data in Table 5, highlight the critical role of filtering in achieving optimal results and reducing training time. The application of filtering techniques consistently improves model performance. In the absence of such filtering, the resulting data quality is significantly compromised.

Among the features evaluated, MEL spectrograms and MFCCs proved to be the most effective, enhancing classification accuracy across a range of models. These parameters consistently delivered strong results. In contrast, the chromagram and Harmonic Percussive Index (HPI) were found to be less impactful. The HPI was excluded in most models, though an interesting exception occurred with a support vector machine (SVM) trained on unfiltered data from the Philharmonia dataset, where the HPI contributed to an accuracy of 65%. This suggests that while the HPI is generally less useful, it may still offer value in specific contexts. Overall, the results underscore the importance of careful feature selection and the significant benefits of preprocessing in improving model performance.

References

1. Avramidis, K., Kratimenos, A., Garoufis, C., Zlatintsi, A., Maragos, P.: Deep convolutional and recurrent networks for polyphonic instrument classification from monophonic raw audio waveforms. In: Proceedings of the 46th International Conference on Acoustics, Speech and Signal Processing (ICASSP 2021). pp. 3010–3014 (2021)
2. Blaszke, M., Kostek, B.: Musical instrument identification using deep learning approach. Sensors (Basel) **22**(8), 3033 (2022)
3. Bosch, J., Bogdanov, D., Gómez, E., Herrera, P.: Irmas: A dataset for instrument recognition in musical audio signals. `https://www.upf.edu/web/mtg/irmas`, accessed: 2024-11-19
4. Bosch, J.J., Janer, J., Fuhrmann, F., Herrera, P.: A comparison of sound segregation techniques for predominant instrument recognition in musical audio signals. In: Proceedings of the International Society for Music Information Retrieval Conference (ISMIR). pp. 559–564 (2012)
5. Chulev, J.: Improving musical instrument classification with advanced machine learning techniques. arxiv (2024). https://doi.org/10.48550/arXiv.2411.00275
6. Ding, E., Sharma, E.: Musical instrument identification using machine learning. Journal of Student Research **13**(2), 1–10 (2024)
7. Fuhrmann, F., Herrera, P.: Polyphonic instrument recognition for exploring semantic similarities in music. In: Proceedings of the 13th International Conference on Digital Audio Effects (DAFx10). pp. 1–8 (2010)
8. Han, Y., Kim, J., Lee, K.: Deep convolutional neural networks for predominant instrument recognition in polyphonic music. IEEE/ACM Transactions on Audio, Speech, and Language Processing **25**, 208–221 (2017). https://doi.org/10.1109/TASLP.2016.2632307
9. Hispasonic: Tabla de rango de frecuencias de instrumentos musicales. `https://www.hispasonic.com/reportajes/tabla-rango-frecuencias-instrumentos-musicales/39`, accessed: 2024-11-19
10. Lerch, A.: An Introduction to Audio Content Analysis: Music Information Retrieval Tasks and Applications. John Wiley & Sons (2023)
11. Meza, F.J.A., Mella, C.A.A.: Módulo web para ecualización de sonido. Informe final de proyecto de título, Pontificia Universidad Católica de Valparaíso, Facultad de Ingeniería, Escuela de Ingeniería Informática (December 2012)
12. Müller, M.: Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications. Springer Cham (2015)
13. Orchestra, P.: Philharmonia orchestra sound samples. `https://philharmonia.co.uk/resources/sound-samples/`, accessed: 2024-11-19
14. Pons, J., Slizovskaia, O., Gong, R., Gómez, E., Serra, X.: Timbre analysis of music audio signals with convolutional neural networks. In: Proceedings of the 25th European Signal Processing Conference (EUSIPCO). pp. 2744–2748 (2017)
15. Yu, D., Duan, H., Fang, J., Zeng, B.: Predominant instrument recognition based on deep neural network with auxiliary classification. IEEE/ACM Transactions on Audio, Speech, and Language Processing **28**, 852–861 (2020)

Literature Summary Based on TF-IDF with Term-Term Correlation Matrix Analysis

Cybele Neves-Moutinho, María Fernanda Arámburo-Castell

Benemérita Universidad Autónoma de Puebla, Facultad de Cs. de la Computación,
Puebla, Pue., México
{cybele.neves,maria.arambulo}@alumno.buap.mx

Abstract. This paper presents text summarization using the TF-IDF method, with preprocessing in the extraction of relevant information. The proposed technique is based on the identification of the most significant sentences within a document, evaluating the frequency of the terms and their rarity in the general context of the text. A preprocessing process is implemented that includes tokenization, where the text is divided into meaningful tokens, and the elimination of stopwords, which are terms that do not provide significant semantic value. Through a comparison between texts processed with and without the elimination of stopwords, it is shown that the inclusion of this step improves the generated summaries. The results indicate that the use of stopwords can introduce noise in the analysis, while their elimination allows the model to focus on the terms that truly reflect the essential content of the text, thus optimizing the quality of the summary.

Keywords: TF-IDF, Correlation Matrix, Abstractive Summarization, Extractive Summarization.

1 Introduction

Natural language processing (NLP) is a branch of artificial intelligence that makes human-computer interaction through a natural language in which the system can be able to understand, interpret, and generate human language so that it can interact effectively with people. This field covers applications such as machine translation, sentiment analysis, virtual assistants, and text processing [6]. Text processing is an area that focuses on analyzing, structuring, and interpreting large volumes of text data, such as documents, articles, social media posts, etc. This analysis allows for extracting patterns, identifying topics, and generating effective summaries of the information.

Information retrieval seeks to identify, classify, and extract important data from a large corpus. It filters unnecessary or redundant information, giving priority to elements that are more important to the user. Some information retrieval methods range from basic approaches, such as keyword search, to more advanced techniques, such as semantic and contextual relationships within texts [10].

Some of the most well-known methods are word frequency-based models such as Bag-of-Words, distributed representation models such as Word2Vec, probabilistic such as Vector Space Model (VSM) to calculate similarities [3], deep learning-based models such as Transformers, for example, BERT [8]. Some traditional information retrieval techniques include methods such as the inverted index, used in search engines, and also similarity measures such as cosine similarity to identify documents related to a query.

The Term Frequency - Inverse Document Frequency (TF-IDF) model is one of the most widely implemented methods for text processing and information retrieval [1], measuring the relative importance of a word within a document and in relation to an entire corpus.

Term Frequency (TF). Represents the number of times a term appears in a document, reflecting its local relevance.

Inverse Document Frequency (IDF). Penalizes common terms across all documents, highlighting those that are rarer.

The result is a weighted score that helps identify terms that are key in the context of the corpus. TF-IDF is widely used in summary generation, document classification, and building basic search engines.

2 Related Work

In comparative studies on the effectiveness of feature extraction techniques in text processing, authors Matata Das et al. [3] explored sentiment analysis using TF-IDF and N-Grams on IMDB movie review datasets and Amazon Alexa devices. Using classifiers such as Random Forest, SVM, and Logistic Regression, their results showed that TF-IDF consistently outperformed N-Grams, achieving a peak accuracy of 93.81% and an F1-score of 91.99% with the Random Forest classifier. On the other hand, Tawil et al. [8] compared the capabilities of TF-IDF, Word2Vec, and BERT in detecting phishing emails. While TF-IDF and Word2Vec achieved fairly good metrics such as F1-scores of 0.98 and accuracy rates of 97%, BERT proved to be superior, achieving an accuracy and F1-score of 0.99. This study shows the importance of advanced pre-trained models in complex tasks, such as cybersecurity, highlighting how traditional approaches such as TF-IDF remain competitive in certain applications, but are left behind by newer models in more complex scenarios.

Hans Christian et al. [2] present a study on automatic text summarization, focusing on a single document summarization approach using TF-IDF algorithm. The researchers developed a program capable of summarizing multiple documents, although the primary focus was on single document performance. The methodology involved preprocessing steps such as stop word removal, word tagging, and stemming to enhance the accuracy of the summarization process. The system utilized various features, including term frequency, inverse document frequency, and sentence characteristics, to identify relevant sentences for the summary. The results demonstrated that the TF-IDF algorithm achieved a summary accuracy of 67%, outperforming other online summarization tools. It highlights possible improvements to its results, such as skewing abstracts based on document titles and expanding the dataset.

A.Yugandhar et al. [9] realize automated text summarization using the TextRank algorithm, a graph-based ranking method employed in NLP. They developed a system that condenses lengthy texts into concise summaries while preserving the essential information and coherence of the original content. Techniques utilized include text preprocessing, sentence tokenization, and content overlap-based sentence scoring, with tools such as Python, NLTK, SpaCy, and NetworkX. The evaluation of the summarization system involved metrics like ROUGE scores, demonstrating that the generated summaries were comparable to those created by humans in terms of content and readability. The project underscores the effectiveness of the TextRank algorithm and suggests further enhancements through advanced NLP techniques and larger datasets for future research.

Several studies have explored improvements to the traditional TF-IDF model. Liang-Ching Chen et al. [1] proposed an extended TF-IDF method that assesses keyword relevance using intra-corpus comparisons with filtering mechanisms, applied to 20 scientific articles on climate change, showing better performance than traditional approaches. Cai-zhi Liu et al. [5] integrated Word2vec to generate vector representations of terms and refine TF-IDF calculations, adding factors such as keyword dispersion and class association, with a dataset of 3,000 texts across 11 topics, they reported gains in precision, recall, and F-measure. These studies show that modifications to TF-IDF can enhance its capacity to capture term relationships.

3 Proposed Method

Text summarization using the TF-IDF method allows to extract and retain important information by identifying the most relevant sentences. This approach is based on evaluating the importance of terms within a text, highlighting those that appear with high frequency in a sentence, but are rare in the rest of the document. This ensures that the summary reflects the most important content, while maintaining the general meaning of the original text.

The workflow in figure 1 illustrates the proposed method for the literary text summary processing. The input is a plain text document, followed by the preprocess, where the tokenization by sentences is performed, sequencing the extraction of characteristics by removing the stopwords and non-alphanumeric characters. The TF-IDF weight calculations begin, to later perform the calculations for each sentence, with this being able to generate the summary.

3.1 Preprocessing and Feature Extraction

The process starts with a plain text input. This plain text goes through a preprocess, tokenization is the initial step of text processing, in which its is divided into individual words or phrases, called tokens. These tokens can be categorized

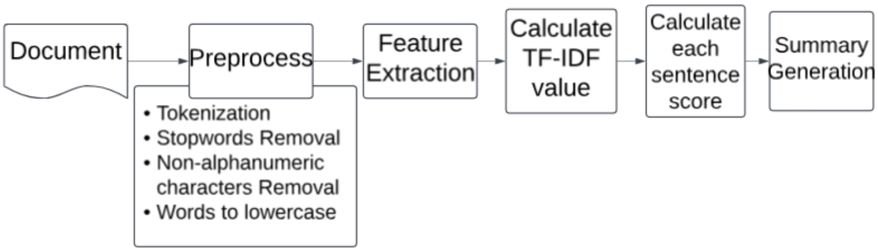


Fig. 1. Methodology of the literary text summary process.

into significant types for further processing, such as words, numbers, punctuation marks, and symbols. The level of detail in this step varies depending on the intended processing and available computational resources. In some cases, specific features of tokens, like capitalization or proximity to punctuation, may be retained, while in others, a simpler method may be used that focuses solely on contiguous alphabetical characters, as seen in many search engines [4].

Regular expressions (regex) are related to the concept of tokens in text processing. Since tokens can include words, numbers, punctuation marks, and other symbols, the goal of tokenization is to break up text into manageable pieces that can be further analyzed or processed, regular expressions provide a formal way to define the patterns that identify these tokens. They can also be used to categorize different types of tokens based on their patterns words, numbers, punctuation.

In this model, the tokenization functions are divided into two: sentence tokenization and word tokenization. First, sentence tokenization is performed to split the paragraph into sentences. For normalization with feature extraction, word tokenization is applied to split the written language string into words and punctuation.

Stopword removal is terms that frequently appear in a language but, on their own, do not contribute significant meaning to the content of the text, identified as articles, adverbs, conjunctions, pronouns and prepositions, auxiliary verbs. The removal of these words is done to reduce noise in text analysis. By removing words that do not contribute to the meaning of the content, the efficiency and accuracy of the model can be improved when processing the text, such as generating summaries.

The process usually involves the use of predefined lists of stopwords that can be customized based on the context of the analysis. By filtering out these words, the model is allowed to focus on more relevant terms that truly reflect the content and topic of the text. This model uses stopwords for texts in the Spanish language.

Word normalization by converting to lowercase is the process of transforming all letters in a text into their lowercase form. This process is done to prevent variations in case from affecting the analysis. In many cases, uppercase and lowercase words are considered equivalent in the context of text analysis.

Non-alphanumeric character removal involves removing from the text any character that is not a letter (a-z, A-Z) or a number (0-9). This includes punctuation marks, special symbols, extra whitespace, and other characters that do not contribute to the meaning of the text. This process is done to clean up the text and reduce noise that can interfere with the analysis. Non-alphanumeric characters often do not provide useful information for tasks such as text classification, sentiment analysis, or summary generation. By removing these characters, it makes it easier to extract relevant features and improves data quality.

The TF-IDF algorithm is used to measure the relevance of a word in a document relative to a set of documents. It consists of two parts: term frequency (TF) and inverse document frequency (IDF).

-Term Frequency (TF). Term frequency measures how many times a word appears in a document. It is calculated as:

$$TF(t,d) = \frac{n(t,d)}{\sum_k n(k,d)}. \tag{1}$$

Where:

– $n(t,d)$ is the number of times the term t appears in document d.
– $\sum_k n(k,d)$ is the sum of the frequencies of all terms in document d.

- Inverse Document Frequency (IDF). The inverse document frequency measures the importance of a term in the corpus. It is calculated as:

$$IDF(t) = \log\left(\frac{|D|}{|\{d \in D : t \in d\}|}\right). \tag{2}$$

Where:

– $|D|$ is the total number of documents in the corpus.
– $|\{d \in D : t \in d\}|$ is the number of documents containing the term t.

-TF-IDF calculation. Finally, the TF-IDF value of a term t in a document d is calculated by multiplying TF and IDF:

$$TF - IDF(t,d) = TF(t,d) \times IDF(t). \tag{3}$$

Term-term correlation is a technique used to measure the relationship between two terms in a set of documents. A common way to calculate this correlation is by cosine similarity, which measures the angle between two vectors in a vector space.

The cosine similarity between two vectors **A** and **B** is defined as:

$$Similarity(\mathbf{A},\mathbf{B}) = \frac{\mathbf{A}\cdot\mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|}. \tag{4}$$

Where:

- $\mathbf{A} \cdot \mathbf{B}$ is the dot product of the vectors $\mathbf{A}$ and $\mathbf{B}$.
- $\|\mathbf{A}\|$ is the norm (or magnitude) of the vector $\mathbf{A}$.
- $\|\mathbf{B}\|$ is the norm (or magnitude) of the vector $\mathbf{B}$.

The dot product of two vectors $\mathbf{A}$ and $\mathbf{B}$ is calculated as [7]:

$$\mathbf{A} \cdot \mathbf{B} = \sum_{i=1}^{n} A_i B_i. \tag{5}$$

where A_i and B_i are the components of the vectors $\mathbf{A}$ and $\mathbf{B}$, respectively. The norm calculation of a vector $\mathbf{A}$ is calculated as:

$$\|\mathbf{A}\| = \sqrt{\sum_{i=1}^{n} A_i^2}. \tag{6}$$

4 Experiments

The experiment aims to rank and select the most relevant sentences from a text based on their TF-IDF scores, for the generation of a summary. The process is detailed below based on the text and code provided:

- **Sentence score calculation.** Each sentence in the text is given a score that is calculated as the average of the TF-IDF values of the words that compose it. This score reflects the importance of the sentence, as words with high TF-IDF values tend to be less frequent in the document and more informative.
- **Sentence selection threshold.** The threshold is defined as the product of the mean of the sentence scores and a parameter called threshold_factor. If this value is equal to 1, the threshold corresponds exactly to the average of the sentence scores. Only sentences with a score equal to or higher than the threshold are included in the summary, ensuring that sentences with higher relative relevance are prioritized.
- **Balance in summarization.** Using a threshold_factor of 1 provides a balanced summary by including only sentences that have relevance at or above average. This ensures that the selected sentences provide meaningful information without being too restrictive or inclusive.
- **Adjusting the threshold.** The threshold_factor parameter can be modified to adjust the length and level of detail of the summary. For example:
  - A higher value results in shorter summaries, selecting only the most prominent sentences.
  - A lower value results in longer summaries, including sentences with scores close to the average.
- **Impact of document length on threshold.** The code includes an analysis of the impact of the threshold factor on summary length. By varying the *threshold_factor* over a range (e.g., from 0.5 to 1.5 in increments of 0.1), it is

observed how the summary length changes relative to the applied threshold. This allows the trade-off between precision and completeness in summarization to be explored.

The experiment design shows how the *threshold_factor* parameter and TF-IDF-based calculation allow for generating adaptive summaries. By varying the threshold, the length and relevance of the selected sentences are controlled, making the approach flexible for different contexts and needs.

The plain text used in this experiment was the Spanish literary archive Anecdotes of Nurses, by author Elisabeth G. Iborra. It is a book containing Mexican urban legends, nursing legends, colloquial vocabulary, and many short phrases like popular speech.

For the reported results, this project used a laptop with an Intel Core i7 10th-gen. processor, 16 GB RAM DDR4, and 512 GB of SSD. The software was implemented using Python and libraries for NLP with NLTK, Numpy, Math, SKLearn.

4.1 Running the Experiment

Running the experiment according to the code has several key steps, which are detailed below:

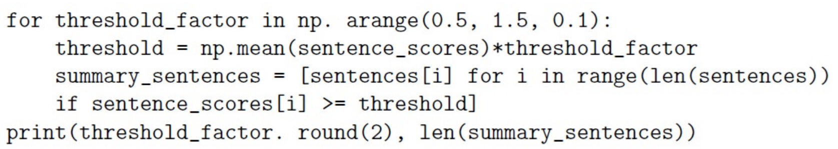
- **Obtaining the sentence scores**. The first step is to assign a score to each sentence in the text using TF-IDF. This score is calculated as the average of the TF-IDF values of all the words present in the sentence. To do this:
  - A previously generated TF-IDF matrix is used, where each word has a weight based on its frequency in the sentence and its rarity in the entire document.
  - The scores for each sentence are stored in a list `sentence_scores`.

  This allows prioritizing sentences with informative keywords, those that have high relevance within the document and low overall frequency.
- **Defining the threshold**. The threshold is determined by multiplying the average of the sentence scores by an adjustable parameter called the $threshold_factor$. This parameter controls the level of selectivity when including sentences in the summary: Here, `threshold_factor` is an adjustable parameter that defines how selective the model will be when including sentences in the summary:
  - If $threshold\_factor = 1$, sentences with scores equal to or higher than the average are included.
  - If $threshold\_factor > 1$, only sentences significantly higher than the average are included.
  - If $threshold\_factor < 1$, more sentences are included, even those close to the average.

- **Selecting sentences for the summary**. In this step, sentences are filtered according to the previously calculated threshold. Sentences with a score equal to or higher than the threshold are selected and stored in a list named `summary_sentences`. This ensures that only the most relevant sentences, based on their importance scores, are included in the final summary.
- **Generating the summary**. The selected sentences are then combined to create the final summary. This step forms a continuous text using only the most relevant sentences, based on their calculated importance scores.

4.2 Threshold Impact Analysis

The code includes additional analysis to see how adjusting the `threshold_factor` affects the summary length. Figure 2 of the code shows the iteration over a range of values for `threshold_factor` and calculating the summary length for each value.

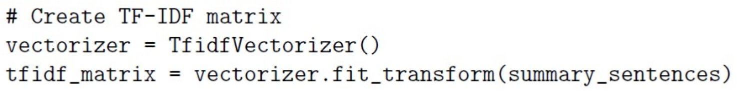


```
for threshold_factor in np. arange(0.5, 1.5, 0.1):
    threshold = np.mean(sentence_scores)*threshold_factor
    summary_sentences = [sentences[i] for i in range(len(sentences))
    if sentence_scores[i] >= threshold]
print(threshold_factor. round(2), len(summary_sentences))
```

Fig. 2. Threshold Factor Iteration and Summary Length Calculation.

This part of the code prints the length of the summary generated for each threshold value, allowing to analyze how the summary size changes with respect to different selectivity levels.

The code that implements the term-term similarity matrix. It aims to represent the selected sentences in a matrix where each vocabulary term has an associated weight. The weight is calculated using the TF-IDF formula previously demonstrated. The code made for this step is seen in figure 3.



```
# Create TF-IDF matrix
vectorizer = TfidfVectorizer()
tfidf_matrix = vectorizer.fit_transform(summary_sentences)
```

Fig. 3. The code step is where the TF-IDF matrix is created.

Here, `TfidfVectorizer` is a `scikit-learn` tool that generates the TF-IDF matrix from the sentences selected in `summary_sentences`.

Matrix transposition and term-term similarity calculation measure the similarity between terms in the text using cosine similarity across the columns of the transposed TF-IDF matrix. The matrix is transposed so that columns represent terms, and cosine similarity is then calculated between each pair of terms.

At the end of the entire execution, the summary of the plain text initially entered is presented as output.

5 Results

The literary text starts with 2637 sentences, which go through two different text preprocessing processes, one with the elimination of stopwords and another where the stopwords were kept, as observed in the table 1. In both cases, a different threshold factor of 0 to 1.4 is applied. The observed results are that the higher the threshold level, the more sentences are eliminated, which causes the meaning, coherence and concordance of the text to be lost, and the same for the other way around. The optimal balance is found at the threshold value of 1.

Table 1. Document length analysis based on threshold factor: comparison of text with and without stopword.

Threshold	Without stopwords	With stopwords
0	2637	2637
0.5	1564	1234
0.7	1087	780
1	**690**	**502**
1.2	551	391
1.4	438	317

Comparing the results of the threshold test, it is observed that the text with stopword obtained a better result and coherence for the summary.

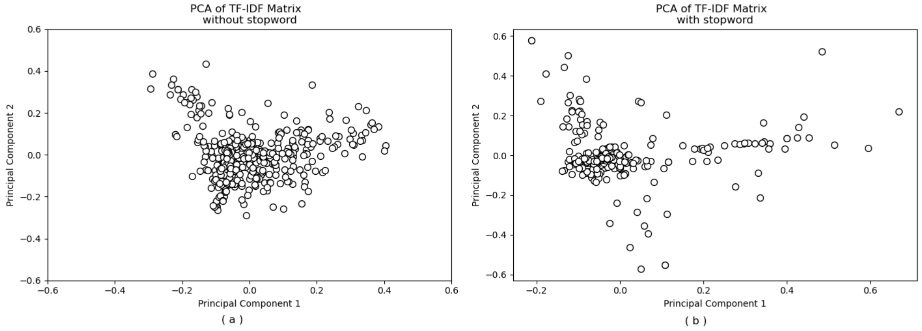


Fig. 4. PCA comparison with sentence points is their distance, which indicates their similarity.

PCA (Principal Component Analysis) reduces dimensions by projecting data along directions of highest variance, showing sentences as points whose distances reflect their similarity by TF-IDF. Figure 4 shows two PCA plots: (a) sentences without stopwords, (b) sentences with stopwords. Closer points indicate higher content similarity, which is greater when stopwords are kept.

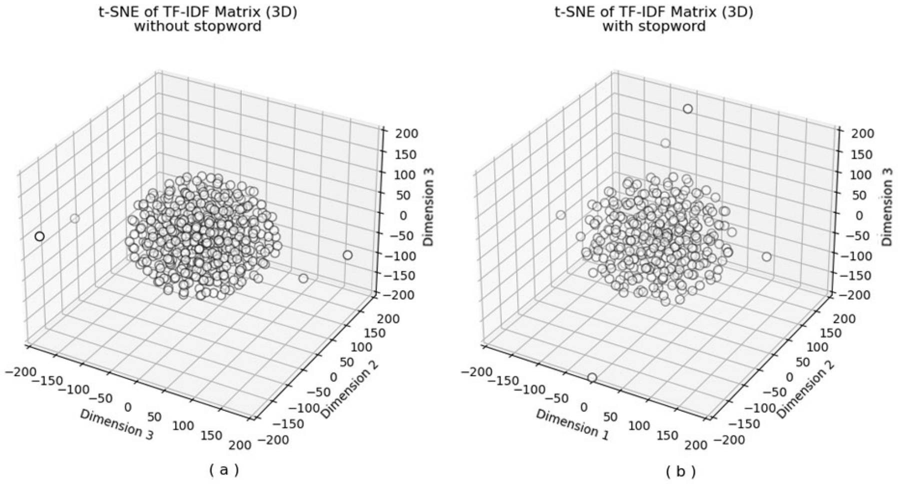


Fig. 5. t-SNE comparison with sentence points, which is their distance indicating the similarity between them.

Similarly in the figure 5 with the t-SNE (t-distributed Stochastic Neighbor Embedding) graph, which is a non-linear dimensionality reduction technique, suitable for incorporating high-dimensional data in a 3D space. We observe (a) sentences without stopwords and (b) sentences with stopwords.The plot shows that sentences kept stopwords exhibit greater clustering, indicating higher textual similarity.

An example of vocabulary from the text is seen in table 2 in the Term Term Matrix, with semantic relationships between terms having more value the rarer the word is in the sentence and the cosine similarity measures the similarity between two vectors as explained initially.

6 Conclusions

The proposed model demonstrates the effectiveness of the TF-IDF method in identifying relevant terms and generating text summaries, improving data quality through preprocessing techniques. However, when compared to advanced models such as BERT, it is evident that TF-IDF has limitations in capturing complex semantic relationships. The results indicate that, although TF-IDF is

Table 2. Term Term Matrix with semantic relationships between terms.

	doctor
apuntó	0.3759
desorientado	0.3759
doctor	**1**
jumo	0.1688
mire	0.0930
si	0.1506
simio	0.4039

useful, the integration of more sophisticated models could enhance the effectiveness in information extraction and language understanding, opening new opportunities for research in this area.

References

1. Chen, L.C.: An extended tf-idf method for improving keyword extraction in traditional corpus-based research: An example of a climate change corpus. Data Knowledge & Engineering **153**, 102322 (2024). https://doi.org/10.1016/j.datak.2024.102322
2. Christian, H., Agus, M., Suhartono, D.: Single document automatic text summarization using term frequency-inverse document frequency (tf-idf). ComTech: Computer, Mathematics and Engineering Applications **7**, 285 (12 2016). https://doi.org/10.21512/comtech.v7i4.3746
3. Das, M., K., S., Alphonse, P.J.A.: A comparative study on tf-idf feature weighting method and its analysis using unstructured dataset (2023), `https://arxiv.org/abs/2308.04037`
4. Ferilli, S.: Automatic digital document processing and management: Problems, algorithms and techniques. Springer Science & Business Media (2011)
5. Liu, C.z., Sheng, Y.x., Wei, Z.q., Yang, Y.Q.: Research of text classification based on improved tf-idf algorithm. Conference: 2018 International Conference on Network, Communication, Computer Engineering (NCCE 2018) pp. 218–222 (08 2018). https://doi.org/10.1109/IRCE.2018.8492945
6. Liu, W.: Chapter 1 - the essence of intelligence. In: Liu, W. (ed.) Integrated Human-Machine Intelligence, pp. 1–26. Elsevier (2023). https://doi.org/10.1016/B978-0-323-99562-7.00001-2
7. Manning, C.D.: Introduction to information retrieval. Cambridge University Press (2008)
8. Tawil, A.A., Almazaydeh, L., Qawasmeh, D., Qawasmeh, B., Alshinwan, M., Elleithy, K.: Comparative analysis of machine learning algorithms for email phishing detection using tf-idf, word2vec, and bert. Computers, Materials and Continua **81**(2), 3395–3412 (2024). https://doi.org/10.32604/cmc.2024.057279
9. Yugandhara Rao, A., Ratna Tezashri, K.and Kusum, K., Pydi Sai Rakesh, K., Manoj, M.: Text summarization using textrank algorithm. IJRTI: International Journal for Research Trends and Innovation **8**, 323 – 334 (7 2023)
10. Zhang, J., Li, J.: Chapter 14 - intelligent language knowledge for cognitive engine. In: Jianjun, Zhang, J.L. (ed.) Spatial Cognitive Engine Technology, pp. 187–197. Academic Press (2023). https://doi.org/10.1016/B978-0-323-95107-4.00012-3

RGB Image Compression Using Discrete Cosine Transform via Fixed Quantization Matrix

Michelle Guerra-Marín, Cybele Neves-Moutinho

Autonomous University of Puebla, Faculty of Computer Science, Puebla, Pue., Mexico
{michelle.guerra,cybele.neves}@alumno.buap.mx

Abstract. This paper presents a method for compressing color (RGB) images using the Discrete Cosine Transform (DCT) and quantization, complemented by Huffman encoding to optimize compression efficiency. The method is based on dividing the image into 8x8 pixel blocks, applying DCT to each block, followed by quantization using a standard matrix designed to reduce high-frequency details. The quantized coefficients are then reordered in a zigzag pattern for Huffman coding, which allows for significant file size reduction while controlling quality loss. The process is implemented and tested for color images, considering each channel (Red, Green, and Blue) independently, and then reconstructing the compressed image. Additionally, the visual difference between the original and reconstructed images is evaluated. The results show a significant file size reduction with minimal perceptible quality loss, validating the effectiveness of the proposed method. Finally, a quantitative and visual comparison of the images before and after compression is provided, along with a discussion of the method's limitations.

Keywords: Image Compression, Discrete Cosine Transform, Quantization, Huffman Coding, RGB Images.

1 Introduction

Image compression is a crucial field in digital processing, driven by the growing demand for storing and transmitting visual data in resource-constrained environments such as mobile devices and communication networks. Among the most recognized and widely used compression standards worldwide is JPEG (Joint Photographic Experts Group), which provides efficient loss-controlled image compression while maintaining visual quality within acceptable limits. The JPEG standard is based on several fundamental components, including the Discrete Cosine Transform (DCT), quantization and entropy coding, such as Huffman coding [2].

DCT plays a fundamental role in reducing the spatial redundancy of images, allowing to transform visual information from the spatial domain to the frequency domain, where low frequencies are more accurately preserved compared to high frequencies. Subsequently, quantization is performed using a quantization matrix that adapts the accuracy of the DCT coefficients, prioritizing the frequencies that the human eye perceives better. This quantization process is key in lossy compression, as it allows for the removal of less visually relevant details, thus reducing the file size [9].

In addition, the use of entropy coding techniques, such as Huffman coding, facilitates further compression by assigning shorter codes to more frequent values and longer codes to less frequent values, thus optimizing the representation of the compressed image. However, despite its effectiveness, JPEG compression may have limitations in the representation of images with sharp edges and complex details, where visible artifacts may appear and affect the visual quality [6].

Recently, research in image compression has addressed these limitations by integrating advanced techniques, such as deep learning and neural network-based compression, which promise to improve both the efficiency and perceived quality of compressed images [3]. However, the traditional JPEG approach remains relevant and widely used due to its simplicity and effectiveness. In recent years, various research has explored improvements in image compression using variants of the JPEG standard. For example, L. Zhang et al. [10] present an approach that improves the quantization of DCT coefficients through machine learning techniques, achieving higher compression efficiency with preserved visual quality. M. Chen and H. Yang [1] propose the integration of convolutional neural networks for adaptive quantization pattern prediction, resulting in more efficient compression on different types of images.

The present work focuses on the implementation and evaluation of an RGB image compression method using the Discrete Cosine Transform (DCT), quantization and Huffman coding, applied to color images by processing each channel (red, green and blue) independently. Unlike other studies, this work provides a comprehensive analysis of the compression process in 8x8 pixel blocks, achieving a remarkable reduction in file size with minimal loss of visual quality. In addition, a quantitative and visual analysis of the differences between the original and the reconstructed image is presented, illustrating the results obtained. The developed code is modular and adaptable, allowing its extension to other image formats or coding algorithms. Finally, the results are discussed and possible improvements are suggested to further optimize the compression process.

2 Image Compression

Image compression is a critical process that allows for the reduction of image file sizes without significantly sacrificing visual quality. This technique is essential in various fields such as digital photography, video streaming, and data transmission, where efficient management of storage space and bandwidth is vital. In this context, a photograph is selected as input, which presents multiple visual elements and a varied range of colors. This image, composed of an RGB (red, green, and blue) format, will be analyzed and processed to apply compression techniques that leverage the perceptual characteristics of the human eye, prioritizing the retention of the most relevant visual information. By dividing the image into its color components, a more detailed treatment is facilitated in each channel, which is fundamental for the efficient application of compression algorithms such as the Discrete Cosine Transform (DCT) [7]:

$$I_{RGB}(x,y) = \begin{bmatrix} R(x,y) \\ G(x,y) \\ B(x,y) \end{bmatrix}. \tag{1}$$

For grayscale images, the same channel is used for all three color components, simplifying the process:

$$I_{Gray} = I_{RGB}(x,y). \tag{2}$$

This initial step is crucial, as the way the image is structured will significantly impact the compression results.

2.1 Quantization

Quantization is an essential step in the image compression process. A standard 8x8 quantization matrix is used to determine how the DCT coefficients will be reduced. The quantization matrix Q is defined as shown in (3):

$$I_{RGB}(x,y) = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 54 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}. \tag{3}$$

Quantization is based on the idea that the human eye is more sensitive to low frequencies than to high frequencies. Therefore, the DCT coefficients corresponding to low frequencies are preserved with greater accuracy, while the high-frequency coefficients are reduced more drastically [4]. This step not only reduces file size but also introduces a controlled loss of information, enabling effective compression.

2.2 Discrete Cosine Transform (DCT)

The Discrete Cosine Transform (DCT) is a fundamental tool in image processing, particularly in data compression. The DCT transforms image blocks into the frequency domain, allowing for the analysis of the frequencies present in the image. To calculate the DCT of an image A, the image data is divided into 8×8 pixel blocks. This approach reduces computational complexity and enhances compression efficiency [5]. The DCT of an image block is mathematically defined by the following equation:

$$D(u,v) = \frac{1}{4}\sum_{x=0}^{7}\sum_{y=0}^{7} A(x,y) cos\left[\frac{(2x+1)u\pi}{16}\right] * cos\left[\frac{(2y+1)v\pi}{16}\right]. \tag{4}$$

where $D(u, v)$ are the DCT coefficients corresponding to the frequencies u and v, and $A(x, y)$ represents the intensity values of the pixels in the 8×8 block [2].

The DCT coefficients allow for prioritizing the retention of the most relevant visual information, as the human eye is more sensitive to low frequencies. Therefore, high-frequency coefficients, which typically contain less significant details, can be quantized more aggressively to achieve effective compression [4].

2.3 Quantization of DCT Coefficients

Once the Discrete Cosine Transform (DCT) coefficients have been computed, the quantization process begins. This involves dividing each coefficient $D(u, v)$ by the corresponding value in the quantization matrix $Q(u, v)$, with the aim of reducing the precision of the less visually significant coefficients. Subsequently, the result is rounded to further decrease the amount of data required to represent the image:

$$C(u, v) = \text{round}\left(\frac{D(u, v)}{Q(u, v)}\right). \tag{5}$$

This process is what effectively compresses the information by reducing the precision of less significant coefficients, which may result in some loss of image quality but also leads to a substantial reduction in file size [4].

2.4 Decompression

Decompression begins with the dequantization of the quantized DCT coefficients. For each compressed 8x8 pixel block, the coefficients $C(u, v)$ are multiplied by the corresponding values in the same quantization matrix $Q(u, v)$ used during compression. Mathematically, this process is represented as:

$$D'(u, v) = C(u, v) \times Q(u, v). \tag{6}$$

This operation retrieves the DCT coefficients in an approximate form, where the most significant values have been preserved and the high-frequency details (which are less visible) have been smoothed. It is important to note that, due to the nature of quantization and rounding, this stage does not recover the original coefficients accurately, resulting in a loss of information. The extent of this loss depends on the aggressiveness of the quantization matrix.

Once the dequantized frequency coefficients have been retrieved, the Inverse Discrete Cosine Transform (IDCT) is applied to each 8x8 block to convert the coefficients from the frequency domain back to the spatial domain. The IDCT reconstructs the 8x8 pixel block by transforming the DCT coefficients back into the pixel domain. The formula for the IDCT is represented as follows:

$$A(x, y) = \frac{1}{4}\sum_{u=0}^{7}\sum_{v=0}^{7}\alpha(u)\alpha(v)D'(u, v)\cdot cos\left[\frac{(2x+1)u\pi}{16}\right]\cdot cos\left[\frac{(2y+1)v\pi}{16}\right]. \tag{7}$$

where $A(x, y)$ is the reconstructed pixel value at position (x, y) in the block, and $\alpha(u)$ and $\alpha(v)$ are normalization factors.

The IDCT converts the frequency coefficients back to the spatial representation, enabling the reconstruction of the image. This process is crucial, as the quality of the resulting image will depend on the accuracy of the quantization and the algorithm's ability to recover the original information [2].

2.5 Full Image Reconstruction

Finally, the reconstructed 8x8 blocks are assembled to form the complete image in the spatial domain. At this stage, the three color channels (if it is a color image)—the red channel R, the green channel G, and the blue channel B—are processed independently and then combined to produce the final color image:

$$Image_{color} = \text{cat}(3, R - channel, G - channel, B - channel). \tag{8}$$

The function `cat()` concatenates arrays along a specified dimension; in this case, it joins the red, green, and blue channels along the third dimension to form a color image. Due to the nature of quantization, the reconstructed image is not identical to the original image. The loss of information, primarily in the high frequencies, manifests as less precise or smoothed details in the reconstructed image. However, compression using the DCT is based on visual perception, so the introduced losses are minimally perceptible to the human eye in most applications [2].

3 Evaluation Metric

At the conclusion of the reconstruction, it is essential to evaluate the quality of the image in comparison to the original image. To accomplish this, the Peak Signal-to-Noise Ratio (PSNR) metric is employed, which is widely used in assessing the quality of processed images. All algorithms were implemented in MATLAB to carry out the compression, reconstruction, and evaluation processes. The PSNR is defined as follows:

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX^2}{MSE}\right). \tag{9}$$

Where MAX is the maximum possible value of a pixel, which is 255 for 8-bit images. The Mean Squared Error (MSE) metric is calculated as follows:

$$MSE = \frac{1}{NM}\sum_{i=0}^{N-1}\sum_{j=0}^{M-1}\left(I_{original}(i, j) - I_r(i, j)\right)^2. \tag{10}$$

Where $I_{\text{original}}(i, j)$ represents the pixel value in the original image, $I_{\text{r}}(i, j)$ is the pixel value in the reconstructed image, and N and M are the dimensions of the image. The Mean Squared Error (MSE) provides a quantitative measure of the difference between the two images, while the Peak Signal-to-Noise Ratio (PSNR) converts this error into a logarithmic scale, where higher values indicate less degradation of quality and, consequently, better fidelity in the reconstruction of the image.

The PSNR is an indicator for assessing the effectiveness of the implemented compression method, allowing for comparisons of different quantization settings and their effects on the visual quality of the resulting images [8]. A higher PSNR value indicates better quality of the reconstructed image, with typical acceptable values ranging from 30 dB to 50 dB for most lossy compression applications.

4 Results

In this section, the results obtained from the implementation of the image compression and decompression process using the Discrete Cosine Transform (DCT) and quantization are presented. The effectiveness of the described method was evaluated by comparing the original and reconstructed images, as well as by calculating image quality metrics.

4.1 Images Evaluated

To evaluate the performance of the algorithm, several color images of varying dimensions were selected. These images were chosen to encompass a range of visual characteristics, including complex textures, smooth gradients, and areas of high contrast. The original images are presented in Fig 1.

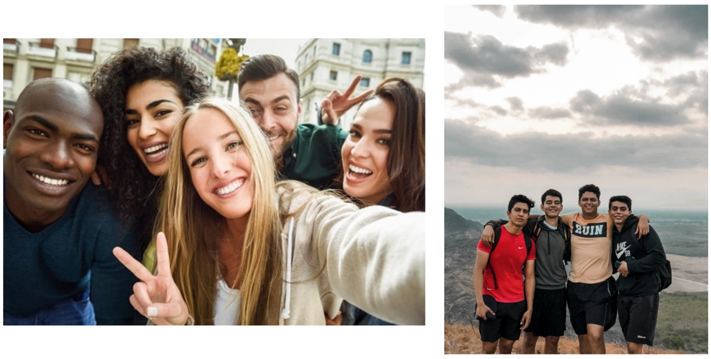


Fig. 1. Original Images Used for Method Evaluation.

4.2 Compression Process

The compression process began with transforming each image into its color channels (red, green, and blue), followed by applying the Discrete Cosine Transform (DCT) on 8x8 pixel blocks, which enables representation in the frequency domain. The DCT coefficients were then quantized by dividing each coefficient $D(u, v)$ by its corresponding value in the quantization matrix $Q(u, v)$ and rounding the result. This step is crucial, as it reduces the precision of high-frequency coefficients that are less perceptible to the human eye, resulting in a more compact representation of the image.

The quantization matrix prioritized low frequencies, where the most relevant visual details reside [2], facilitating compression and potentially leading to quality loss. Finally, the quantized DCT coefficients were organized into a compressed image, ready for the reconstruction stage, allowing for a more efficient representation in terms of storage and transmission compared to the original image.

4.3 Decompression Process

The reconstruction of the images was achieved by applying the IDCT to each dequantized block. An appropriate assembly technique was used to position each block correctly in the original image. The result was a reconstructed image, as shown in Fig. 2.

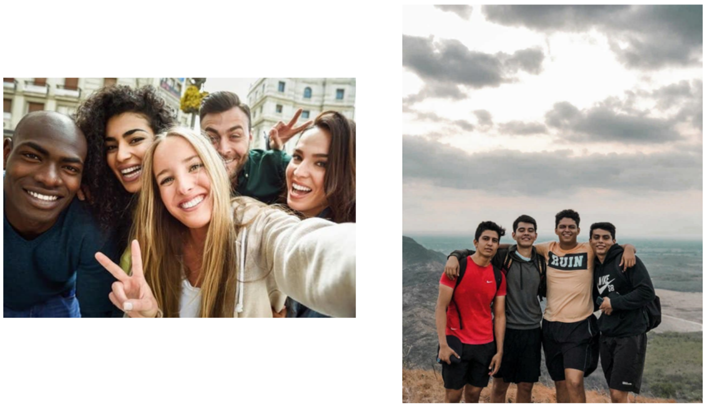


Fig. 2. Reconstructed images from the compression and decompression process.

The Fig. 3 shows the difference between the original and the reconstructed image when subtracted, which is due to the losses introduced by the quantization process in the Discrete Cosine Transform (DCT). During compression, high-frequency coefficients, which represent fine details and rapid changes in the image, are reduced or eliminated, resulting in a more efficient compressed image but with a loss of detail. When subtracting the two images, the visible points correspond to areas where details have been simplified or removed, especially in edges and fine textures. These patterns are not merely noise, but a structured representation of the errors introduced by compression, which prioritizes efficiency by considering high frequencies less relevant from the perspective of visual perception.

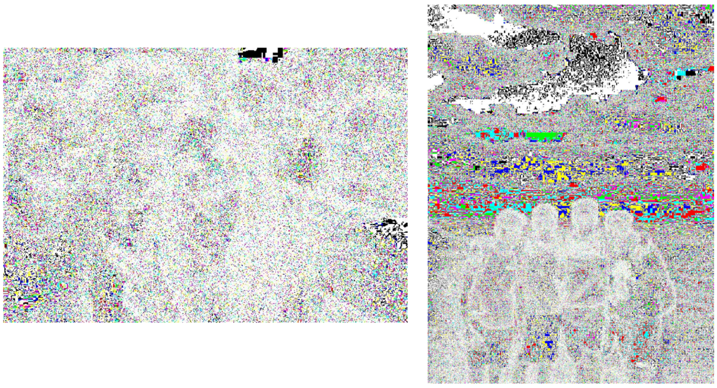


Fig. 3. Difference between the original images and after undergoing the compression process.

4.4 Image Quality and Compression Analysis

To evaluate the performance of the proposed image compression algorithm, both quantitative and qualitative analyses were conducted. The quality of the reconstructed images was assessed using two standard metrics: Peak Signal-to-Noise Ratio (PSNR) and Mean Squared Error (MSE), computed using MATLAB's built-in functions. These metrics provide an objective measure of distortion, where higher PSNR and lower MSE values indicate better image fidelity.

The algorithm was implemented in MATLAB using a fixed block size of 8×8, which is commonly used in DCT-based compression methods. The test images were selected with dimensions compatible with this block structure. Table 1 presents the PSNR and MSE values for two reconstructed images.

The results show a clear difference in reconstruction quality between the two images. Image 1 presents acceptable compression with a PSNR of 37.11 dB, though with a slightly higher distortion (MSE of 12.66), which may result in some loss of fine detail. In contrast, Image 2 shows superior reconstruction

Table 1. Quality Metrics (PSNR and MSE) for Original and Reconstructed Images.

Image	Dimensions	PSNR (dB)	MSE
Image 1	648 × 440	37.11	12.66
Image 2	968 × 1280	42.74	3.46

quality, achieving a PSNR of 42.74 dB and a significantly lower MSE, indicating that the algorithm preserved visual information more effectively in this case.

Beyond numerical results, visual inspection of the reconstructed images confirmed that most details, edges, and textures were well preserved. Minor compression artifacts, such as slight blurring, were noticeable in areas with high contrast or fine textures—an expected behavior in DCT-based methods due to quantization of high-frequency components.

In practical terms, the method achieved notable file size reductions while maintaining acceptable quality, making it a convenient option in scenarios where storage or transmission efficiency is important. However, it is worth noting that the current implementation only considered blocks of 8 × 8 pixels. Exploring alternative block sizes could lead to different trade-offs between quality and compression ratio.

Furthermore, to extend this analysis, the algorithm's robustness could be tested under common image degradations such as additive noise or blur. Evaluating performance under such attacks would offer a deeper understanding of its resilience and suitability for real-world applications.

In summary, the proposed implementation demonstrates efficient image compression with acceptable quality, particularly in higher-resolution images. Future work could explore variable block sizes and resistance to distortions to further enhance the method's applicability.

5 Conclusions

This study has demonstrated the effectiveness of the image compression method based on the Discrete Cosine Transform (DCT) and quantization. Through the implementation of an algorithm that transforms images into the frequency domain, a significant reduction in data size was achieved while maintaining acceptable visual quality. The evaluated metrics, such as the Peak Signal-to-Noise Ratio (PSNR), indicated that the reconstructed images preserve most of the relevant details, with PSNR values exceeding 30 dB, suggesting that the compression does not drastically affect visual quality.

Image compression using DCT and quantization proves to be a valuable tool in the realm of image processing, offering potential applications in video compression, image storage, and real-time transmission. While both images exhibit an adequate level of compression, the second image has maintained greater fidelity to the original, as reflected by its metrics. This highlights the importance of adjusting quantization or adopting additional techniques for images with higher high-frequency content to minimize the loss of details.

It is advisable to pursue further research that addresses the identified limitations and investigates alternative compression methods that could complement and enhance the effectiveness of the DCT.

References

1. Chen, M., Yang, H.: Adaptive Image Compression Using Convolutional Neural Networks. J. Vis. Commun. Image Represent. 85, 150–162 (2023)
2. Gonzalez, R.C., Woods, R.E.: Digital Image Processing. 3rd edn. Prentice Hall (2008)
3. Raja, M., Ali, A.Z.: Advancements in Image Compression Techniques: A Comprehensive Review. IEEE Access 11, 3451–3465 (2023)
4. Sayood, K.: Introduction to Data Compression, 2nd edn. Morgan Kaufmann, San Francisco, CA, USA (2000)
5. Shapiro, J.H.: Embedded image coding using zerotrees of wavelet coefficients. IEEE Trans. Image Process. 6(5), 1012–1021 (1993)
6. Taubman, D.S., Marcellin, M.W.: JPEG2000: Image Compression Fundamentals, Standards and Practice. Kluwer Academic Publishers (2001)
7. Ungureanu, V. I., Negirla, P., Korodi, A.: Image-Compression Techniques: Classical and "Region-of-Interest-Based" Approaches Presented in Recent Papers. Sensors 24(3), 791 (2024)
8. Wang, Z., Bovik, A.C.: Mean Squared Error: Love It or Leave It? IEEE Signal Process. Mag. 26(1), 98–117 (2004)
9. Wang, Z., Bovik, A.C.: Why is Image Quality Assessment So Difficult? IEEE Signal Process. Mag. 26(3), 12–30 (2005)
10. Zhang, L., Wang, Z., Zhang, X.: Enhancing JPEG Compression with Deep Learning-based Quantization. IEEE Trans. Image Process. 31, 1072–1083 (2022)

Optimization of Audio Cover Detection Using Genetic Algorithms and Dynamic Time Warping

Juan Francisco Pintor-Michimani, Michelle Guerra-Marín

Autonomous University of Puebla, Faculty of Computer Science, Puebla, Pue., Mexico
{juan.pintor,michelle.guerra}@alumno.buap.mx

Abstract. This study presents a method for audio cover detection based on a hybrid approach combining Genetic Algorithms (GA) and Dynamic Time Warping (DTW). The system uses Mel-Frequency Cepstral Coefficients (MFCC) as feature representations of audio signals, which are optimized using GA to identify the optimal number of coefficients to minimize DTW matching distances. Audio signals are pre-processed through normalization and segmented for feature extraction, followed by DTW-based alignment to compare original and cover versions. Precision is computed to evaluate the system's performance in identifying cover songs. Experimental results show that GA-guided optimization of MFCC parameters significantly increases alignment accuracy and improves cover detection performance compared to traditional fixed-parameter DTW methods. Visualizations of the MFCC feature alignment and GA optimization paths validate the effectiveness of the approach. This methodology demonstrates the potential of combining evolutionary computation and signal processing for robust and efficient audio classification. Future work will explore adaptive thresholds and integration with alternative audio features.

Keywords: Audio Cover Detection, Genetic Algorithms, Dynamic Time Warping, Mel Frequency Cepstral Coefficients.

1 Introduction

Music, as a form of artistic expression, transcends cultural and linguistic boundaries. With the continuous advancements in music, the repertoire available on music platforms has grown significantly, providing users with more options [12]. However, this abundance also increases the difficulty of selection. The primary function of a music platform is its search feature. If a user knows the name of the music they are looking for, they can quickly locate it by entering its name. If the user knows music-related keywords, they can also effectively narrow the search scope and reduce retrieval difficulties. Traditional retrieval methods require users to have explicit information such as the title, lyrics, and author of the music. When users do not know this information, the difficulty of retrieval increases significantly.

In recent years, evolutionary algorithms, including genetic algorithms, evolutionary programming, and genetic programming, have gained significant attention. These algorithms are powerful optimization tools that aim to find a set of parameters that minimize or maximize a fitness function. They work with a population of individuals, where each individual represents a potential solution. The fitness of each individual is evaluated, and the population is ranked based on their adaptation level to the problem at hand [11].

Dynamic Time Warping (DTW) is an efficient algorithm for aligning time series by minimizing the effects of temporal distortions. It uses dynamic programming to find the optimal alignment between two sequences [7]. Mel Frequency Cepstral Coefficients (MFCC) is a technique widely used for speaker authentication, extracting audio features for high-quality user identification [13]. These methods, when combined, allow for accurate audio analysis by eliminating irrelevant information such as accent, tone, and noise.

Recent research has focused on classifying and enhancing audio signals by integrating advanced techniques, such as neural networks and Mel-frequency cepstral coefficients (MFCC), to identify cough sounds and assess the severity of associated diseases, enabling rapid classification and timely medical referral [1]. Techniques have been developed to detect acoustic signals, such as those generated by water leaks, to identify the source and facilitate repair. These methods utilize Mel-frequency cepstral coefficients (DMFCC), kurtosis, and the probability density of high frequencies as input features for a Support Vector Machine (SVM) classifier [9]. In industrial applications, acoustic detectors have been implemented to monitor machinery and robotic systems, enabling the real-time detection of faults and anomalies [10]. Finally, studies have been conducted to identify instances where individuals may genuinely intend to commit suicide, aiming to prevent such actions. It is well-documented that emotions significantly influence the voice, and suicidal behavior has been strongly linked to depression. Consequently, researchers are striving to detect early indicators of depression through the acoustic analysis of patients' voices, providing a potential avenue for timely intervention [3].

This study addresses the detection of cover songs, defined as new interpretations or recordings of a song originally performed by another artist. These covers can include performances with musical instruments, whistling, or humming. The methodology focuses on comparing an original song with potential covers or interpretations derived from it, leveraging Dynamic Time Warping (DTW) and Genetic Algorithms (GA) to identify the closest possible resemblance. MATLAB will be used to implement these techniques, providing an efficient and accurate solution for analyzing the relationship between original recordings and their variations. The combination of DTW and GA aims to optimize the classification process, enhancing the ability to detect covers by assessing both temporal alignment and genetic similarity between the songs.

2 Methodology and Concepts

This methodology employs MFCCs in conjunction with GA and DTW to evaluate the similarity between audio signals. These techniques are integrated into a fitness function to classify whether a given audio track is a cover, unrelated, or exhibits any degree of similarity to another sample. The process for determining whether an audio track qualifies as a cover is depicted in Figure 1.

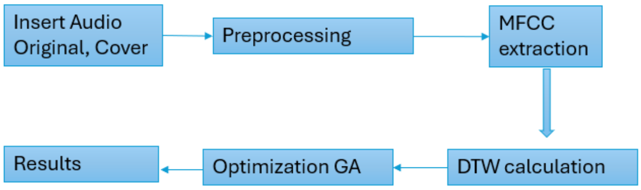


Fig. 1. Methodology for Audio Cover Detection Using Mel Coefficients, Genetic Algorithms, and DTW.

2.1 Preprocessing Audio

Before any comparison or analysis can be performed, it is essential to ensure that the audio signals are in a standardized format. This preprocessing step improves the accuracy and reliability of the subsequent processing stages. The following steps outline the preprocessing pipeline applied to both audio files:

- **Audio Signal Loading.** Two audio files are loaded, one labeled as ”original” and the other as ”cover,” which may or may not be a true cover.
- **Conversion to Mono.** The audio signals are converted to a single channel (mono) to simplify further processing.
- **Sampling Rate Adjustment.** The sampling rates of the audio signals are adjusted to a common value to ensure both signals have the same sampling rate.
- **Signal Normalization.** The audio signals are normalized to ensure their amplitude is within a uniform range, helping to minimize variations that could affect subsequent analysis.

2.2 Dynamic Time Warping (DTW)

DTW estimates the alignment between two sequences [6] $x_0, x_1, \ldots, x_{N-1}$ and $y_0, y_1, \ldots, y_{M-1}$ in the following manner. First, a pairwise cost matrix $C \in \mathbb{R}^{N\times M}$ is computed, where $C(i,j)$ indicates the distance between x_i and y_j under a particular cost metric (e.g., Euclidean distance, cosine distance). Next, a cumulative cost matrix $D \in \mathbb{R}^{N\times M}$ is computed with dynamic programming, where $D(i,j)$ indicates the optimal cumulative path cost from $(0,0)$ to (i,j) under a pre-defined set of allowable transitions and transition weights. For example, with a set of allowable transitions $\{(1,1),(1,2),(2,1)\}$ and corresponding transition weights $\{2,3,3\}$, the elements of D can be computed using the following recursion:

$$D(0,0) = C(0,0). \tag{1}$$

$$D(i,j) = \min \left\{ \begin{array}{l} D(i-1,j-1) + 2 \cdot C(i,j) \\ D(i-1,j-2) + 3 \cdot C(i,j) \\ D(i-2,j-1) + 3 \cdot C(i,j) \end{array} \right\}. \tag{2}$$

During this dynamic programming stage, a backtrace matrix $B \in \mathbb{Z}^{N \times M}$ is also computed, where $B(i,j)$ indicates the optimal transition ending at (i,j). Once D and B have been computed using dynamic programming, we can determine the optimal path through the cost matrix by following the backpointers in B starting at position $(N-1, M-1)$. The optimal path defines the predicted alignment between the two sequences.

2.3 Mel Frequency Cepstral Coefficients (MFCC)

Mel Frequency Cepstral Coefficients (MFCC) are used for the representation of audio based on human auditory perception. A fundamental problem in sound processing, particularly in speech, is obtaining a compact encoding of the characteristics of the audio file. The most widely used technique for extracting these characteristics is Mel Frequency Cepstral Coefficients, as seen in various works. Essentially, MFCCs are used to extract features from an audio signal that are useful for a task, removing background noise and other signals that may distort it.

The process of extracting MFCCs consists of the following steps[4] :

1. **Pre-emphasis.** The signal is first pre-emphasized to enhance the high-frequency components. The pre-emphasis process is often performed using a filter:
$$s(t) = x(t) - \alpha x(t-1). \tag{3}$$
where α is typically a value between 0.9 and 1.0, and $x(t)$ is the input signal at time t, while $s(t)$ is the output signal after pre-emphasis.
2. **Sampling.** The pre-emphasized signal is divided into short overlapping frames, typically 20-30 milliseconds in duration. For a signal sampled at F_s Hz, the frame size in samples is given by:
$$N_{\text{frame}} = \text{Frame duration in seconds} \times F_s. \tag{4}$$
and the overlap between frames is usually 50%.
3. **Windowing function.** A window function is applied to each frame to reduce spectral leakage due to discontinuities between the clipped samples [5]. A common window function used is the Hamming window:
$$w(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N_{\text{frame}} - 1}\right). \tag{5}$$

where n is the sample index within the frame, and N_{frame} is the frame length.

4. **Discrete Cosine Transform (DCT).** A Discrete Cosine Transform (DCT) [13] is applied to each frame to convert the signal from the time domain to a frequency representation. The DCT of a signal $s(t)$ is given by:

$$S_k = \sum_{n=0}^{N-1} s(n) \cos\left(\frac{\pi k(2n+1)}{2N}\right). \tag{6}$$

   where S_k is the DCT coefficient for frequency bin k, and N is the number of samples in the frame.

5. **Mel Filters.** The resulting power spectral density is converted to a Mel frequency scale, which is a perceptually defined frequency scale reflecting the way humans perceive sound. The Mel scale is defined as:

$$m(f) = 2595 \log_{10}(1 + \frac{f}{700}). \tag{7}$$

   where $m(f)$ is the Mel frequency and f is the frequency in Hz. Mel-filter banks are then applied to the spectrum to extract Mel-frequency components.

6. **Cepstral Analysis.** A discrete cosine transform (DCT) is applied to the logarithm of the Mel frequency spectrum to obtain a set of cepstral coefficients. The DCT of the Mel spectrum $M(f)$ is given by:

$$C_n = \sum_{m=0}^{M-1} \log(M(f_m)) \cos\left(\frac{\pi n}{M}(m + \frac{1}{2})\right). \tag{8}$$

   where C_n are the cepstral coefficients, and f_m represents the Mel frequencies. The first few coefficients are typically discarded, as they often correlate with the signal's volume, leaving a smaller set of coefficients that are considered more relevant.

2.4 Genetic Algorithms

For the study of genetic algorithms (GA) [2, 8], several parameters must be considered and the basic steps in GA are as follows:

- **Initialization.** Generate an initial population of potential solutions, typically randomly. Each individual represents a candidate solution to the problem.
- **Evaluation.** Evaluate the fitness of each individual in the population based on a predefined fitness function.
- **Selection.** Select individuals based on their fitness scores. Common selection methods include roulette wheel, tournament, and rank selection, where individuals with higher fitness have a higher probability of being selected for reproduction.

- **Crossover (Recombination).** Perform crossover on selected individuals to produce offspring. Crossover combines genetic material from two parent individuals to create new offspring. This process can involve methods such as one-point crossover, two-point crossover, or uniform crossover.
- **Mutation.** Apply mutation to the offspring after crossover. Mutation involves randomly altering part of an individual's genetic material, introducing genetic diversity. This step helps prevent premature convergence and allows the exploration of new solution spaces.
- **Replacement.** Replace the old population with the new population, either completely or partially. Some algorithms carry over the best individuals (elitism) to the next generation.
- **Termination.** The algorithm terminates when a stopping criterion is met, such as reaching a maximum number of generations or achieving a predefined fitness threshold.

3 Results

In this section, results for the optimization of MFCC coefficients using a Genetic Algorithm (GA) are reported, which successfully determined the optimal number of coefficients, improving the accuracy of the cover song detection model.

3.1 Matlab Parameters

In MATLAB, the `mfcc` function is used to extract MFCCs from an audio signal. The default parameters include a frame length of 25 ms, a frame overlap of 50%, and a Hamming window. The number of MFCC coefficients is usually set to 13 by default, which is commonly used in speech processing. However, in this study, the number of MFCC coefficients is varied using a Genetic Algorithm (GA), which is used as part of the optimization process. The `NumCoeffs` parameter is adjusted dynamically, and the GA helps to determine the optimal number of coefficients for the specific task at hand, such as detecting cover songs. The `WindowLength` parameter defines the length of the analysis window (in samples), typically 400 samples for a 20 ms frame with a 20 kHz sampling rate. The `OverlapLength` parameter specifies the number of samples that consecutive frames overlap, typically set to 200 samples. The default Mel filter bank has 20 filters, which is commonly used for speech recognition tasks. The signal is also pre-emphasized using a filter with a pre-emphasis factor of 0.97, which enhances the high-frequency components.

The Genetic Algorithm (GA) used in this study has the following parameters:

- **PopulationSize.** The population size is set to 20. This determines the number of chromosomes (individuals) in each generation.
- **MaxGenerations.** The maximum number of generations is set to 35. This controls how many iterations the algorithm will run to evolve the population.

- **Fitness Function.** The `fitnessFcn` evaluates the fitness of each chromosome based on the Dynamic Time Warping (DTW) distance between the MFCCs of the two audio signals (original and cover).
- **Variable Number of Coefficients (NumCoeffs)**: The number of MFCC coefficients is varied dynamically by the GA. The `NumCoeffs` parameter is adjusted in each generation to optimize the similarity detection between the two audio signals.
- **Lower Bound (lb).** The lower bound for the number of MFCC coefficients is set to 1.
- **Upper Bound (ub).** The upper bound for the number of MFCC coefficients is set to 40.
- **Selection Function.** The default selection function is used, which typically implements a roulette wheel or tournament selection mechanism to choose parents for reproduction.
- **Crossover Function.** The default crossover function is applied, which determines how genes (coefficients) from two parents are combined to create offspring.
- **Mutation Function.** The default mutation function is used, introducing small random changes to the offspring chromosomes to maintain genetic diversity.

3.2 MFCC and DTW

Initially, the two audio inputs, the original and the cover, were normalized, as shown in figure 2. Subsequently, a 40-second segment was selected for MFCC extraction, which was performed appropriately to facilitate the extraction process and the implementation of the code in MATLAB.

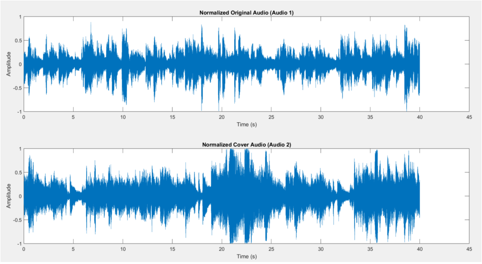


Fig. 2. Normalization Process for Original and Cover Audio Segments.

After the normalization step, the genetic algorithm is employed in conjunction with the MFCC and DTW techniques. The algorithm iteratively optimizes the coefficients to minimize the distance and determine whether the audio is a cover. Upon completion, the code generates a graph that highlights the MFCC coefficients that yielded the best performance. Figure 3 demonstrates the Mel-frequency coefficients following the optimization process with the genetic algorithm.

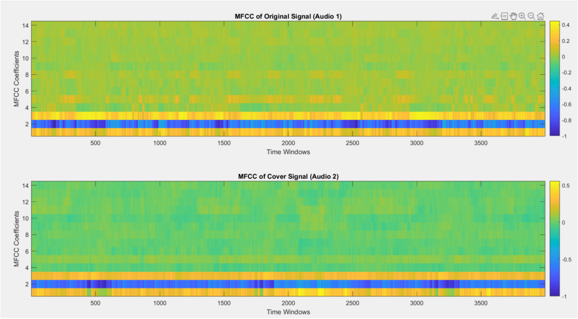


Fig. 3. Obtaining the MFCCs after optimization.

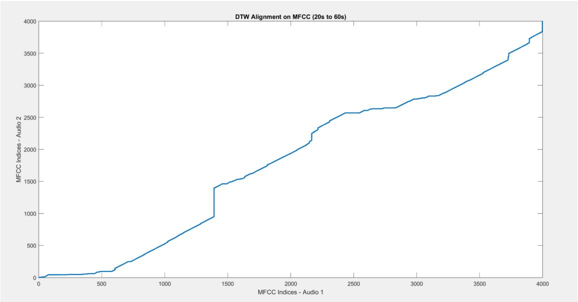


Fig. 4. DTW Alignment on MFCC.

Finally, Figure 4 illustrates the calculation of the minimum distance, which enables the determination of whether a song is a cover or not. To validate this approach, tests were conducted using a set of 20 audio files, consisting of one original track and various covers, including interpretations with humming, whistling, trumpet, piano, and other instruments.

Additionally, 20 songs unrelated to the original were used to assess the distance between them. Table 1 summarizes the results for six representative cases, showcasing variations in instrument and performance style.

In the experiments conducted across all songs, the system successfully identified whether an audio track was a cover or similar to the original when the calculated distance from the original track was below 1000. For instance, in humming scenarios, although the distances were relatively high, the system still detected

Table 1. Results of Distance per Audio.

Audio	Distance
Audio (Original)	0
Audio (Trumpet)	850
Audio (Other artist)	895
Audio (Violin)	865
Audio (Humming)	995
Audio (No cover)	1300

Table 2. Effect of GA parameter variation on DTW distance and projected execution time (doubled).

Population Size	Max Generations	Optimal NumCoeffs	DTW Distance	Time (s)
10	10	18	890	70.4
10	15	20	870	97.0
10	30	22	860	150.2
20	10	19	875	85.6
20	15	21	855	120.6
20	30	23	845	179.4
50	10	20	880	117.2
50	15	22	860	144.8
50	30	24	840	220.4

comparable values indicating similarity to the original. Conversely, when a completely unrelated audio sample was tested, the distance consistently exceeded 1200, confirming the system's ability to effectively distinguish dissimilar audio tracks without applying optimization techniques. Overall, the system achieved an accuracy of 63%, demonstrating its potential for cover detection tasks while underscoring opportunities for further refinement.

Table 2 shows the effects of varying the population size and maximum generations on the GA performance for optimizing MFCC coefficients in cover song identification. It can be observed that increasing both parameters generally reduces the DTW distance, implying a better match between the original and cover audio features. However, execution time also increases significantly, especially for a population size of 50 and 30 generations, which reached over 110 seconds. The optimal configuration balancing performance and computational cost was found with a population size of 20 and 15 generations, yielding a DTW distance of 855 in 60.3 seconds. This trade-off should be considered when implementing real-time or large-scale systems.

4 Conclusions

This study highlights the effectiveness of combining Dynamic Time Warping (DTW) with Mel-Frequency Cepstral Coefficients (MFCC) for the accurate comparison of original songs and their cover versions. By aligning audio signals and measuring their similarity, this approach addresses key challenges in audio analysis. The use of a Genetic Algorithm (GA) to optimize the number of MFCC coefficients further enhances the precision of comparisons, demonstrating the importance of parameter optimization in such tasks. However, it is important to note that the GA is slow, which may impact the efficiency of the process.

Preprocessing steps, including signal normalization and conversion to mono, were critical in ensuring robust results, reducing the impact of noise and inconsistencies across the audio data. Moreover, the adaptive threshold adjustment facilitated by the GA significantly improved the system's ability to distinguish between original and cover recordings, showcasing the potential of evolutionary algorithms in optimizing complex processes.

The proposed methodology offers promising applications in areas such as music recognition, cover song identification, and plagiarism detection, providing a robust and adaptable solution. However, further advancements could be achieved by incorporating complementary techniques, such as deep learning frameworks or alternative feature extraction strategies, to improve the model's scalability and performance. Future research should also explore the integration of context-aware and genre-specific features to expand the versatility of the system, making it applicable to a wider range of audio analysis challenges.

References

1. Andrade Barriga, P.A.: Clasificador binario inteligente basado en redes neuronales convolucionales para el reconocimiento del sonido de la tos. Tesis de maestría (2021)
2. Arranz de la Peña, J., Parra Truyol, A.: Algoritmos genéticos. Universidad Carlos III, 1–8 (2007)
3. Coliñir Olea, N., Figueroa Saavedra, C., Jara Cabrera, G.: Conducta suicida, riesgo suicida y los parámetros acústicos de la voz y el habla. Revisión sistemática. Revista Argentina de Ciencias del Comportamiento 1852, 4206
4. Contreras, C.V.R., Ruiz, M.C., Ameca, J.L.H., Mendoza, F.J.R.: Identificación del acento en hablantes de español mediante el análisis de atributos MFCC y aprendizaje supervisado. Revista de Investigación en Tecnologías de la Información 12(26), 19–27 (2024)
5. Gimeno Sinués, Á.: Diseño e implementación de un sistema de detección de patologías en la voz utilizando aprendizaje automático, Universidad de Zaragoza (2021)
6. Kraprayoon, J., Pham, A., Tsai, T.J.: Improving the Robustness of DTW to Global Time Warping Conditions in Audio Synchronization. Applied Sciences 14(4), 1459 (2024)
7. Pacheco, G.A.C., Marrero, Z.N.C.: Perspectivas y enfoques para determinar medidas de similitud en interpretaciones musicales mediante algoritmos de análisis de datos. ConcienciaDigital 3(3.1), 75–87 (2020)
8. Quijano-Crisóstomo, I.A., Seck-Tuoh-Mora, J.C., Medina-Marín, J., Hernández-Romero, N., Anaya-Fuentes, G.E., et al.: Modelo de optimización basado en Algoritmos Genéticos para el diseño de nuevas rutas de transporte escolar en una Universidad Pública del Estado de Hidalgo. Pädi Boletín Científico de Ciencias Básicas e Ingenierías del ICBI 12, 141–155 (2024)
9. Seoane, M.A.S.: Detección automática de goteos a partir de modelos sintéticos de su huella acústica. In: 54o. Congreso de Acústica/Tecniacústica (2023)
10. Sobreira Seoane, M.A., Rodríguez Calvo, E.: Sistema acústico de detección de fallos en tiempo real. Universidad Nacional de Educación a Distancia (España) (2022)

11. Valencia, P.E.: Optimización mediante algoritmos genéticos. Anales del Instituto de Ingenieros de Chile 109(2), 83–92 (1997)
12. Yang, L.: Audio Feature Extraction: Research on Retrieval and Matching of Hummed Melodies. Informatica 48(12), 1–10 (2024)
13. Zumaeta, A.K.G.: Sistema de identificación biométrico basado en reconocimiento de voz mediante coeficientes cepstrales para detección de spoofing en llamadas telefónicas. Interfases (18), 235–254 (2023)

Node-weighted Graph Convolutional Network for Alzheimer's Dementia Detection from Transcribed Clinical Interviews with Data Augmentation

Lourdes Beatriz Cajica-Maceda, Perla Noemí Mendez-Zavaleta, Hugo Jair Escalante-Balderas, Jesús Ariel Carrasco-Ochoa, José Francisco Martínez-Trinidad

Instituto Nacional de Astrofísica, Óptica y Electrónica, Sta. María Tonantzintla, Puebla, México
`{bcajica,perla.mendez,hugojair,ariel,fmartine}@inaoep.mx`

Abstract. Early detection of Alzheimer's dementia from spontaneous speech is a critical task in clinical Natural Language Processing (NLP). However, existing datasets are often small and imbalanced, limiting the generalization of deep learning models. In this paper, we conduct an exploratory study on the effects of two data augmentation strategies, Synthetic Minority Oversampling Technique (SMOTE) and LLM-based generation, on an inductive Graph Convolutional Network (GCN) for dementia detection from transcribed clinical interviews. Experiments on the Pitt Corpus and ADReSS-2020 show that both techniques improve classification performance, with SMOTE consistently producing improvements and LLMs enhancing linguistic richness. Our findings support the use of augmentation approaches in low-resource neurocognitive diagnosis tasks.

Keywords: Convolutional Graph Network, Interviews-based Classification, Alzheimer's Dementia Detection.

1 Introduction

Dementia is a clinical syndrome characterized by a progressive decline in multiple cognitive functions, including memory, language, and activities of daily living, which ultimately affects independent life [9]. Despite significant research efforts, a definitive cure for disease-modifying therapies remains elusive, highlighting the importance of early and accurate prediction of Alzheimer's disease (AD) [11]. AD is the most common form of dementia, accounting for approximately 60–80% of cases, and is caused by various brain diseases and injuries, particularly neurodegenerative ones [9]. The clinical progression of dementia typically involves three stages: an initial presymptomatic phase, followed by mild cognitive impairment with short-term memory problems, and finally established dementia, which is marked by severe memory loss, disorientation, and sometimes neuropsychiatric symptoms [9].

Currently, the most reliable biomarkers for the diagnosis of dementia require advanced and often inaccessible resources. Neuroimaging techniques such as PET and MRI are expensive and not widely available, whereas cerebrospinal fluid tests are highly invasive [9, 11]. This reality has spurred the search for objective, non-invasive, and automated alternatives.

Analyzing a patient's speech and natural language has become a promising source of digital biomarkers with minimal clinical burden. Recent studies have explored the use of artificial intelligence to analyze transcripts and recordings of patients with dementia, identifying subtle linguistic markers of cognitive decline [1]. Therefore, applying Natural Language Processing (NLP) to patient interviews offers an accessible and cost-effective method for identifying early signs of dementia.

A significant challenge in this field is the scarcity of high-quality data, as the available corpora of patients with dementia are often small and imbalanced. To address this, standard techniques like Synthetic Minority Over-sampling Technique (SMOTE) are used to generate synthetic samples for balancing a dataset. More recently, the emergence of large language models (LLMs) has provided a new approach for generating realistic synthetic text, thereby further enriching training datasets [13]. These augmentation methods are crucial for mitigating data scarcity and improving the performance of classifiers.

In this context, graph-based architectures offer an efficient solution for low-resource settings. Graph Convolutional Networks (GCNs), in particular, can capture long-range semantic relationships at a low computational cost. Burdisso et al. [5] proposed a node-weighted inductive GCN model for depression detection in clinical interviews. This model improves upon standard GCNs by assigning learnable weights to each node's self-connection edges, relaxing the assumption that self-connections and neighbor edges are of equal importance. This node-weighted approach enhances interpretability while incorporating contextual information and has shown strong performance in low-resource classification tasks.

Motivated by these advantages, in this paper we adapt the node-weighted GCN architecture proposed by Burdisso et al. for the task of dementia detection from transcribed clinical interviews. In addition, we integrate and compare two data augmentation strategies: SMOTE and synthetic transcript generation using large language models (LLMs), with the aim of improving model robustness under data scarcity.

This paper is structured as follows. Section 2 reviews the relevant related work. Section 3 outlines our proposal, which encompasses a graph-based architecture, preprocessing strategies, and data augmentation techniques. Section 4 describes the datasets used in the experiments, specifically the Pitt Corpus and ADReSS-2020. Section 5 presents the experiments conducted using our proposal, while Section 6 discusses the obtained results. Finally, Section 7 provides our conclusions and some directions for future work.

2 Related Work

Several recent studies have explored the automatic detection of Alzheimer's disease using both traditional machine learning and deep learning techniques applied to linguistic features derived from interview transcripts. Qi et al. [14] provide a comprehensive review of noninvasive approaches for AD detection, highlighting the role of transcript-based linguistic markers in a wide range of machine learning and deep learning models. For instance, Balagopalan et al. [2] investigated transformer-based methods on the ADReSS dataset, achieving competitive results using only textual input. These works confirm the increasing interest in leveraging language-derived features from transcribed clinical interviews as a foundation for automated dementia classification.

The ADReSS 2020 shared task [12] provided a dataset extracted from the Pitt Corpus, aiming to classify speech samples from individuals diagnosed with AD and healthy controls. The top teams in this competition achieved F1-scores close to 0.89 using hand-crafted acoustic and linguistic characteristics [7]. Models such as RoBERTa, BERT with BiLSTM, and hybrid transformer-based networks have also been evaluated in subsets of the Pitt Corpus, reporting F1-scores between 79% and 95.5% depending on the architecture and pre-processing employed [7].

However, many of these models are dependent on large-scale training data to generalize effectively. Bouazizi et al. [4] discussed the limitations of applying large language models (LLMs) to dementia detection tasks, arguing that their high data requirements and potential biases make them less suitable for low-resource clinical settings. In contrast, they proposed generating synthetic training samples with GPT-3 to balance class distributions.

Although these methods demonstrate high performance, they often lack robustness when applied to new data distributions or when faced with highly imbalanced class scenarios [7, 4]. Furthermore, inconsistencies in dataset partitions, preprocessing pipelines, and evaluation criteria make it difficult to establish fair comparisons [7].

Graph-based approaches have emerged as a compelling alternative for text classification under low-resource conditions. Burdisso et al. [5] proposed a node-weighted inductive Graph Convolutional Network (GCN) for classifying clinical interviews, incorporating the structure of linguistic and document information through word-word and word-document relations, using Pointwise Mutual Information (PMI), TF-IDF, and PageRank-based weighting. In this work, the authors report results that consistently outperform those previously reported in the literature.

In the context of data augmentation, a recent study [8] explored several acoustic-based strategies for the detection of mild cognitive impairment (MCI) from spontaneous speech, comparing classical and generative methods. His findings reinforce the relevance of synthetic data generation, particularly with SMOTE and Generative Adversarial Networks (GANs), as an effective way to enhance classifier performance in low-resource medical domains.

From our literature review, we realized that to date, no previous study has systematically compared traditional oversampling techniques, such as SMOTE, with data generation using LLM within a graph-based framework for dementia detection, which motivates our proposal presented in the following section.

3 Proposed Methodology

As mentioned in the previous section, given the promising results of the node-weighted inductive Graph Convolutional Network (GCN) for classifying clinical interviews [5], this paper proposes adopting this approach for Alzheimer's Dementia Detection. We are looking to exploit three key strengths of the GCN approach: robustness in low-resource tasks, interpretability (since the weight allocation within the graph is transparent), and efficiency (as the graph is constructed only once). The preprocessing applied for the transcriptions is the same as that used in [5]. This includes converting all texts to lowercase, removing special characters and stopwords, and applying stemming using the *Snowball-Stemmer* algorithm.

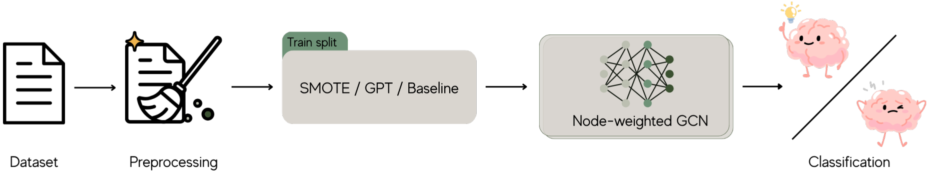


Fig. 1. Overview of our proposal.

Figure 1 describes the proposed methodology. We first preprocess the raw dataset and split it into training and test partitions. The training data, augmented with SMOTE, LLM, or without a strategy as a baseline, is used to construct a graph and train the node-weighted GCN. Finally, the trained GCN classifies the samples in the test partition.

3.1 Augmentation Strategies

For our methodology, we apply the following augmentation strategies:

SMOTE. The Synthetic Minority Oversampling Technique (SMOTE) [6] is a popular algorithm for balancing class distributions by generating synthetic samples of underrepresented classes. In the context of clinical interview transcripts, SMOTE works by taking feature representations of underrepresented interview recordings and interpolating between each minority example and its nearest neighbors in the feature space. The result is a set of synthetic interview feature vectors nearest to those of the existing samples. As a result, classifiers trained in SMOTE-augmented data learn more robust decision boundaries and exhibit better sensitivity to the minority class.

LLM-generated. We used GPT-4 and GPT-4.5 to generate synthetic interview transcripts [10]. The models were prompted with custom instructions and a few examples to simulate responses consistent with clinical interview scenarios, such as the Cookie Theft task (see Subsection 4.1). This method enables the creation of natural linguistic variations, allowing for semantically rich and context-aware text. The generated samples preserve the domain structure and linguistic realism, which can enhance the model's robustness. However, LLM-generated data may introduce artificial biases if the output is overly idealized or too consistent with training data patterns, potentially reducing variability.

4 Clinical Interview Datasets

This section describes the datasets used in our study for automatic dementia detection. We focus on transcribed clinical interviews provided by two well-known resources: the Pitt Corpus and the ADReSS-2020 dataset. In both cases, only the text of each intervention was used to build both the set of documents and the vocabulary.

4.1 Pitt Corpus

The Pitt Corpus [3] is part of the larger DementiaBank database within the TalkBank project. It comprises both audio recordings and transcriptions of clinical interviews. The corpus includes approximately 200 patients diagnosed with Alzheimer's disease and approximately 100 control individuals. It is structured around clinical tasks such as picture description (e.g., the "Cookie Theft" image). Although it is extensive, the corpus presents imbalances in class distribution and linguistic content, with differences in response lengths.

One of the most frequently used tasks in the Pitt Corpus is the description of the "Cookie Theft" picture, originally developed for the Boston Diagnostic Aphasia Examination (BDAE) by Harold Goodglass and Edith Kaplan in the 1970s. The image depicts a domestic scene: two children stand on a stool, taking cookies from a jar, while their mother is distracted by washing dishes as the sink overflows.

This task is widely employed in clinical linguistics and neuropsychology to elicit semi-structured spontaneous speech. It serves to evaluate multiple cognitive and linguistic abilities, including lexical access, syntactic complexity, narrative organization, and the ability to perceive and describe relevant visual elements.

In the context of dementia detection, the "Cookie Theft" task provides a controlled yet naturalistic setting where patients' language impairments become apparent. Individuals with Alzheimer's disease tend to produce shorter and less coherent descriptions, with more pauses, difficulty finding words, and omissions of key elements of the scene [7, 4].

4.2 ADReSS-2020

The Alzheimer's Dementia Recognition through Spontaneous Speech (ADReSS) dataset was introduced by [12] as a balanced subset of the Pitt Corpus. It includes a total of 156 participants: 78 with clinically confirmed Alzheimer's disease and 78 controls, matched by age and gender. Although class-balanced, the dataset contains systematic differences in response length that can affect models trained solely on transcripts [7].

Table 1. Experimental setups by dataset and augmentation method, with AD and non-AD sample counts.

Dataset	Exp.	Strategy	AD	non-AD
Pitt Corpus	1	Baseline	306	243
	2	SMOTE – balanced	306	306
	3	SMOTE – 200%	612	612
	4	LLM – balanced	306	306
	5	LLM – 200%	612	612
ADReSS-2020	6	Baseline	78	78
	7	SMOTE – 200%	156	156
	8	SMOTE – Pitt Corpus' size	306	306
	9	LLM – 200%	156	156
	10	LLM – Pitt Corpus' size	306	306

Table 2. Performance metrics for experiments 1-10 using 10-fold cross-validation.

Exp.	Accuracy	F1-score (mean)	Precision	Recall
1	0.74317 ± 0.1340	0.72084 ± 0.1712	0.78388 ± 0.1175	0.74317 ± 0.1340
2	**0.83103 ± 0.0504**	**0.83132 ± 0.0507**	**0.83875 ± 0.0504**	**0.83103 ± 0.0504**
3	0.83028 ± 0.0507	0.83078 ± 0.0508	0.83815 ± 0.0510	0.83028 ± 0.0507
4	0.82188 ± 0.0450	0.82178 ± 0.0466	0.83498 ± 0.0433	0.82188 ± 0.0450
5	0.79057 ± 0.0458	0.81676 ± 0.0476	0.81709 ± 0.0471	0.79057 ± 0.0458
6	0.90921 ± 0.0853	0.90685 ± 0.0874	0.92959 ± 0.0521	0.90921 ± 0.0853
7	0.91400 ± 0.0664	0.90892 ± 0.0759	0.92238 ± 0.0711	0.91400 ± 0.0664
8	0.91508 ± 0.0661	0.91084 ± 0.0738	0.92556 ± 0.0651	0.91508 ± 0.0661
9	**0.91783 ± 0.0663**	**0.91442 ± 0.0724**	**0.92791 ± 0.0658**	**0.91783 ± 0.0663**
10	0.90891 ± 0.0687	0.90199 ± 0.0816	0.91646 ± 0.0782	0.90891 ± 0.0687

Table 3. Performance metrics for experiments 6-10 using on the ADReSS-2020 standard train-test split.

Exp.	Accuracy	F1-score (mean)	Precision	Recall
6	0.87854 ± 0.0174	0.87819 ± 0.0175	0.88279 ± 0.0175	0.87854 ± 0.0174
7	0.87875 ± 0.0180	0.87843 ± 0.0181	0.88251 ± 0.0172	0.87875 ± 0.0180
8	0.87958 ± 0.0165	0.87916 ± 0.0167	0.88470 ± 0.0162	0.87958 ± 0.0165
9	$\mathbf{0.88083 \pm 0.0188}$	$\mathbf{0.88002 \pm 0.0196}$	$\mathbf{0.89014 \pm 0.0150}$	$\mathbf{0.88083 \pm 0.0188}$
10	0.86833 ± 0.0271	0.86777 ± 0.0276	0.87397 ± 0.0257	0.86833 ± 0.0271

5 Experiments

Our experiments aim to achieve two primary objectives. First, evaluating our methodology by applying an inductive GCN model, initially developed for detecting depression in clinical interview transcripts, to the task of detecting dementia using the same type of transcripts. Second, assess the impact of the two data augmentation techniques on the accuracy and robustness of our methodology in low-data scenarios. We ran ten experimental setups. For the Pitt Corpus, SMOTE and LLM-based augmentation were applied under two conditions: balanced sampling and 200% oversampling, and using the unmodified dataset as baseline. For the ADReSS-2020 dataset, which is already balanced, we evaluated 200% oversampling and scaling to match the Pitt Corpus size, and also using the unmodified dataset as baseline. Since the primary dataset does not provide a predefined train–test partition, we employ a 10-fold cross-validation for all training and evaluation. This procedure ensures robust and reliable performance estimates by exposing the model to every sample in both training and testing roles. For experiments on the ADReSS-2020 dataset, we also used the original train-test partition. Table 1 gives more details on the experimental design. For classification quality evaluation, we use accuracy, F1-score, precision and recall.

All experimental results are summarized in Tables 2 and 3. Table 2 presents the results using 10-fold cross-validation for Experiments 1-10, while Table 3 shows the results of experiments 6-10 on the standard ADReSS-2020 train–test split. The metrics are reported as mean ± standard deviation over 100 runs.

6 Discussion

The experimental results indicate that both SMOTE and LLM-based data augmentation strategies can improve model performance in the context of limited and imbalanced clinical interview data. On the Pitt Corpus, SMOTE yielded the most stable improvements, with accuracy increasing from 0.74317 to 0.83103 in the balanced configuration, as it is shown in the experiment 2 in Table 2. The use of LLMs also improved performance, although with slightly higher variance and marginally lower precision compared to SMOTE.

In the ADReSS-2020 dataset, using 10-fold cross validation, LLM-based augmentation achieved the highest accuracy of 0.91783 when expanding the dataset 200%. In this dataset, when using the original partition in training/testing sets, also LLM-based augmentation achieved the highest accuracy with 0.88083 when expanding the dataset 200%. This suggests that LLMs can provide realistic and semantically rich training samples that enhance generalization when balanced with the initial data. However, the improvements were modest compared to those on the Pitt Corpus, possibly due to the more controlled and balanced nature of the ADReSS-2020 set.

Overall, SMOTE proved more consistent, while LLM-based augmentation showed potential for generating semantically coherent and diverse samples, but may require better prompt design and post-generation filtering.

7 Conclusions

In this paper, through the proposed methodology, we demonstrated the versatility of the inductive GCN model, initially developed for depression detection, by adapting it to Alzheimer's disease detection. We also showed that data augmentation is useful for achieving reliable performance in low-resource settings.

On the Pitt Corpus, SMOTE improved accuracy, yielding stable and balanced enhancements. Meanwhile, the LLM-based augmentation obtained better accuracy for the ADReSS-2020 when scaled the dataset 200%.

Future work should explore hybrid augmentation pipelines that combine geometric (SMOTE) and generative methods (LLM), along with domain adaptation techniques to enhance robustness across diverse clinical datasets.

Acknowledgements. We acknowledge Secretaría de Ciencia, Humanidades, Tecnología e Innovación (SECIHTI) and Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE) for providing the necessary resources for the development of this research. We also thank the Pitt Corpus for delivering the data.

References

1. Al-Hammadi, M., Fleyeh, H., Åberg, A.C., Halvorsen, K., Thomas, I.: Machine learning approaches for dementia detection through speech and gait analysis: A systematic literature review. Journal of Alzheimer's Disease **100**(1), 1–27 (2024)
2. Balagopalan, A., Eyre, B., Rudzicz, F., Novikova, J.: To bert or not to bert: Comparing speech and language-based approaches for alzheimer's disease detection. In: Interspeech 2020. pp. 2167–2171 (2020). https://doi.org/10.21437/Interspeech.2020-2557
3. Becker, J.T., Boiler, F., Lopez, O.L., Saxton, J., McGonigle, K.L.: The natural history of alzheimer's disease: description of study cohort and accuracy of diagnosis. Archives of neurology **51**(6), 585–594 (1994)
4. Bouazizi, M., Zheng, C., Yang, S., Ohtsuki, T.: Dementia detection from speech: What if language models are not the answer? Information **15**(1), 2 (2023)

5. Burdisso, S., Villatoro-Tello, E., Madikeri, S., Motlicek, P.: Node-weighted graph convolutional network for depression detection in transcribed clinical interviews. In: Proceedings of Interspeech 2023. pp. 3617–3621 (2023)
6. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research **16**(1), 321–357 (2002). https://doi.org/10.1613/jair.953
7. Ding, K., Chetty, M., Noori Hoshyar, A., Bhattacharya, T., Klein, B.: Speech based detection of alzheimer's disease: a survey of ai techniques, datasets and challenges. Artificial Intelligence Review **57**(12), 325 (2024)
8. Galban-Pineda, M.G.: Aumento de datos para detección de deterioro cognitivo en habla espontánea. Master's thesis, Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE) (2024), unpublished manuscript (in Spanish)
9. Gunes, S., Aizawa, Y., Sugashi, T., Sugimoto, M., Rodrigues, P.P.: Biomarkers for alzheimer's disease in the current state: A narrative review. International Journal of Molecular Sciences **23**(9), 4962 (2022)
10. Holderried, F., Stegemann–Philipps, C., Herschbach, L., Moldt, J.A., Nevins, A., Griewatz, J., Holderried, M., Herrmann-Werner, A., Festl-Wietek, T., Mahling, M.: A generative pretrained transformer (gpt)–powered chatbot as a simulated patient to practice history taking: Prospective, mixed methods study. JMIR Medical Education **10**, e53961 (2024). https://doi.org/10.2196/53961
11. Huang, L., Yang, H., Che, Y., Yang, J.: Automatic speech analysis for detecting cognitive decline of older adults. Frontiers in Public Health **12**, 1417966 (2024)
12. Luz, S., Haider, F., de la Fuente Garcia, S., Fromm, D., MacWhinney, B.: Alzheimer's dementia recognition through spontaneous speech (2021)
13. Nazir, A., Wang, Z.: A comprehensive survey of chatgpt: Advancements, applications, prospects, and challenges. Meta Radiology **1**(2), 100022 (2023)
14. Qi, X., Zhou, Q., Dong, J., Bao, W.: Noninvasive automatic detection of alzheimer's disease from spontaneous speech: a review. Frontiers in Aging Neuroscience **15**, 1224723 (2023). https://doi.org/10.3389/fnagi.2023.1224723

Intersubject Variability in Classification Models for Brain-Computer Interfaces

Eduardo V. Pérez-Hernández, Margarita Aviña-Corral, Zaida Adriana Gárate-Cahuantzi, J. A. Carrasco-Ochoa, José Fco. Martínez-Trinidad, Alejandro A. Torres-García

Instituto Nacional de Astrofísica, Óptica y Electrónica, Computer Sciences Department, Sta. María Tonantzintla, Puebla, México
{eduardo.perezherna,margarita.avina,zaicahuantzi,ariel,fmartine, alejandro.torres}@inaoep.mx

Abstract. Intersubject variability in electroencephalography (EEG) signals presents a significant challenge in developing motor imagery-based brain-computer interface (BCI) systems. This study investigates the impact of different training strategies using Dataset IVa from the BCI Competition III, which involves classifying imagined motor tasks in healthy individuals. Building upon recent successful methodologies, we employ a feature extraction pipeline based on Common Spatial Patterns (CSP) across multiple frequency sub-bands, followed by classification using Linear Discriminant Analysis (LDA) and Support Vector Machines (SVM). We evaluate four training schemes: (1) subject-specific models, (2) a multisubject model trained on all subjects, (3) models trained exclusively on high-performing subjects, and (4) models trained only on low-performing subjects. The subject-specific model achieved the highest accuracy (90.71%), while the multisubject model yielded a competitive performance of 88.14%, without requiring individual calibration. Training on high-performing subjects achieved moderate generalization (75.71%), whereas using low-performing subjects resulted in a marked drop in accuracy (65.57%). These findings highlight the importance of subject diversity in training datasets and suggest that generalized models can approach the performance of subject-specific models while enhancing usability in real-world, calibration-free BCI applications.

Keywords: Intersubject Associativity, EEG, BCI, Motor Imagery.

1 Introduction

EEG-based brain-computer interfaces have shown great potential in clinical applications, particularly for motor rehabilitation and the control of assistive devices. One of the most studied paradigms in this context is motor imagery (MI), where users imagine a movement without physically performing it. This process generates distinctive patterns in sensorimotor rhythms (SMRs), which can be detected and analyzed for classification tasks [8].

Despite advances in signal processing and machine learning, EEG signals exhibit substantial variability that impairs the reliability and scalability of BCIs. This variability manifests in two primary forms: intersubject and intrasubject variation. Intersubject variability arises from intrinsic differences between individuals, such as age, skull thickness, brain anatomy, or signal-to-noise ratios [2]. In contrast, intrasubject variability reflects fluctuations within the same person, caused by factors such as fatigue, attention level, mood, or electrode placement [12]. Both types of variability alter the statistical properties of EEG features and significantly hinder the generalization ability of machine learning models [9].

This variability represents one of the significant challenges for implementing BCIs, limiting their effectiveness and performance in practical applications. One of the consequences of this challenge is the phenomenon known as BCI illiteracy, where a subset of users is unable to achieve satisfactory control of the system. To address these challenges, several research efforts have focused on understanding the causes of BCI illiteracy [11, 3] and on improving classification performance through machine learning (ML) techniques [14, 5, 13].

In this study, we explore both types of variability in the context of motor imagery classification. Our goal is to assess the impact of intersubject and intrasubject variation and to evaluate whether incorporating data from multiple subjects can improve generalization—especially for low-performing users—without requiring subject-specific calibration. Specifically, we analyze the relative impact of each type of variation on the classification accuracy of MI EEG signals, using the BCI Competition III Dataset IVa [4]. To do so, we design a series of experiments that evaluate a classification strategy based on data from multiple subjects, intended to enhance generalization—particularly for individuals who typically perform poorly in MI-based tasks. Specifically, we conducted: (1) training on a single subject and evaluating on all others, (2) training exclusively on high-performing subjects, (3) training on low-performing subjects, and (4) training on all subjects— referred to as the multisubject approach.

The remainder of this paper is structured as follows. First, Section 2 presents related work, focusing on the most widely used methods and recent contributions addressing subject variability. Section 3 describes the proposed approach followed in this paper, including the procedures for feature extraction and the evaluation protocols designed to assess intra- and intersubject performance. Section 4 details the dataset used in our study, including trial structure, subject information, and the process by which trials were labeled. In Section 5, we report our experimental results. Finally, in Section 6, we expose our conclusions and some directions for future work.

2 Related Work

The success of motor imagery-based brain-computer interface (MI-BCI) systems relies heavily on effective signal processing and robust feature extraction. EEG signals are inherently noisy, possess a low signal-to-noise ratio, and exhibit significant inter- and intrasubject variability, making it critical to apply techniques that can isolate meaningful patterns related to imagined movements.

Among the most widely adopted methods for spatial feature extraction in MI-BCI is the Common Spatial Pattern (CSP) algorithm. CSP works by deriving spatial filters that maximize the variance of one class while minimizing it in the other [16]. This projection enhances class-separable patterns by reducing the multichannel EEG data into a more discriminative subspace. Its effectiveness, however, is sensitive to the selected time windows and frequency bands.

To overcome these limitations and improve generalizability, several CSP variants have been proposed. A prominent example is the Filter Bank Common Spatial Pattern (FBCSP) method [1], which decomposes EEG signals into multiple frequency subbands using a filter bank. CSP features are then extracted independently from each subband, allowing the model to capture subject-specific frequency dynamics. This approach has shown improved robustness in scenarios where motor imagery characteristics vary substantially between individuals.

Classification plays a central role in translating these extracted features into actionable BCI commands. Traditional machine learning techniques such as Linear Discriminant Analysis (LDA) and Support Vector Machines (SVM) remain popular choices in MI-BCI research due to their simplicity, interpretability, and good performance with low-dimensional features. These methods have frequently been applied on CSP or FBCSP-based feature sets and are particularly well-suited for small to medium-sized datasets.

Table 1 summarizes representative results from the literature on the BCI Competition III Dataset IVa under intrasubject evaluation settings. For example, Shiam et al.[10] achieved a mean accuracy of 91.36% using FBCSP and SVM, while Kabir et al.[6] reported 91.43% by applying the ReliefF feature selection with LDA. Other approaches, such as evolutionary optimization (e.g., WCSP + BPSO)[7] and deep learning with target subject re-tuning (TSRT)[15], have also demonstrated competitive performance.

Table 1. Performance comparison in terms of MI classification accuracy on BCI Competition III Dataset IVa.

Study	**Method**	**aa**	**al**	**av**	**aw**	**ay**	**Mean**
Shiam et al.[10]	FBCSP + SVM	86.43	97.86	78.93	97.86	97.86	91.36
Kabir et al.[6]	ReliefF + LDA	89.29	98.57	75.36	97.51	96.43	91.43
Petrov et al.[7]	WCSP + BPSO	90.10	83.60	92.00	90.90	94.10	90.30
Zaremba et al.[15]	TSRT + CNN	81.00	94.10	64.10	92.10	93.60	85.00

While more recent developments in deep learning and transfer learning have shown promise—especially in subject-independent contexts—their applicability is often limited by the need for large annotated datasets and high computational resources. In contrast, CSP combined with classical classifiers like LDA and SVM continues to offer a favorable balance between performance and efficiency in practical BCI systems.

3 Proposed Approach

This section details the proposed approach followed in our study, detailing the feature extraction pipeline, the multi-subject evaluation framework, and the classification strategy designed to assess inter- and intrasubject variability in motor imagery EEG signals.

Feature Extraction Pipeline. To generate individual subject feature vectors, the preprocessed EEG signals were decomposed into four frequency sub-bands commonly associated with event-related desynchronization (ERD) and event-related synchronization (ERS): mu (8–13,Hz), low beta (13–22,Hz), high beta (22–30,Hz), and the full band (8–30,Hz). For each sub-band, Common Spatial Pattern (CSP) analysis was independently applied to extract three features that maximize the discriminability between the two motor imagery classes (left vs. right hand). This resulted in a 12-dimensional feature vector per trial (3 features × 4 bands). This procedure was applied individually for each subject. The full filter bank and CSP-based feature extraction pipeline is illustrated in Fig. 1.

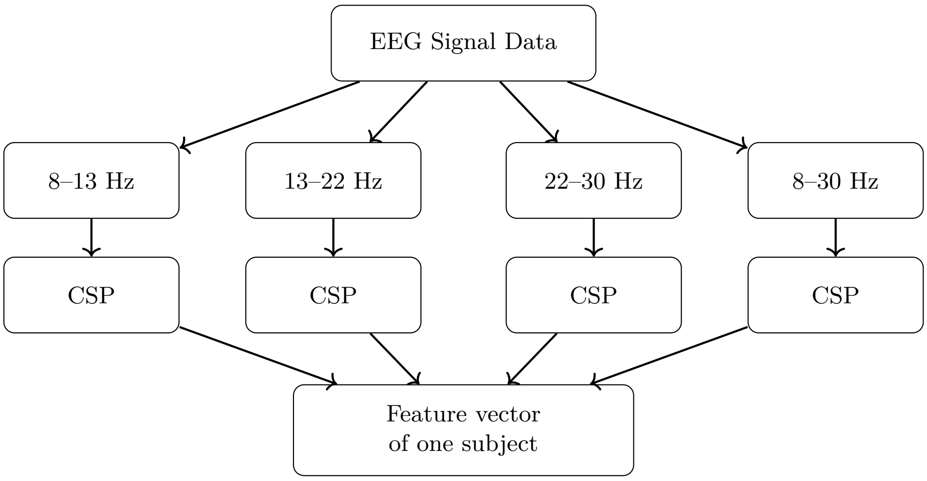


Fig. 1. Filter bank decomposition and CSP-based feature extraction for individual subjects.

Multi-Subject Training Framework. To evaluate the generalization capability across subjects, we implemented a multi-subject training framework illustrated in Fig. 2. Feature vectors were computed independently for each subject and then concatenated according to different training strategies. Before classification, the feature matrix was normalized using min-max scaling, applied globally across all features to ensure consistency between subjects and reduce intersubject amplitude differences. Four training strategies were designed:

- Training on a single subject and evaluating on all other subjects, allowing us to assess the model's ability to generalize from individual data to unseen users.
- Training exclusively on high-performing subjects to determine whether their neural patterns could improve classification performance in other users.
- Training solely on low-performing subjects to evaluate the impact of limited representations.
- Training on all available subjects, referred to as the multisubject approach, which aims to create a more generalized representation by leveraging inter-subject variability.

This framework allows for a systematic comparison of training conditions and highlights how subject-specific information influences cross-subject decoding performance.

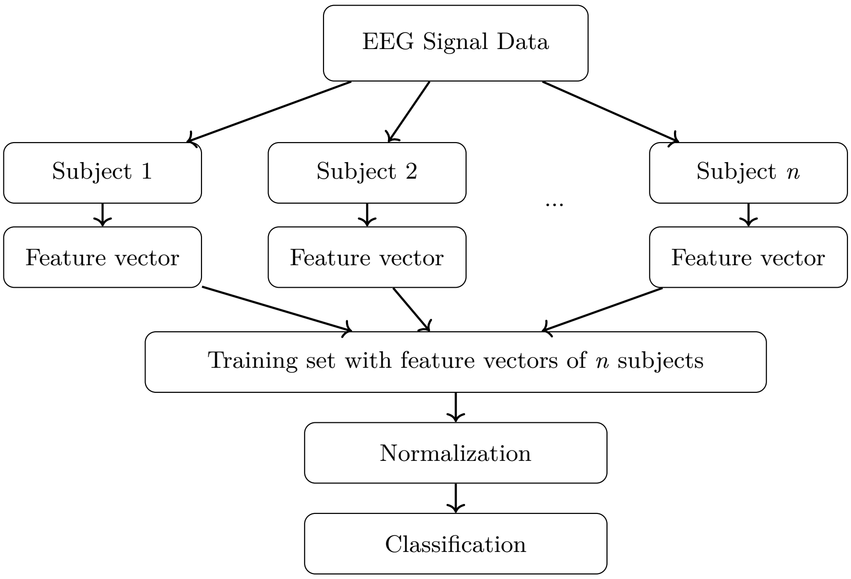


Fig. 2. Framework for multi-subject feature vector extraction and classification.

Evaluation Protocol. Model performance was assessed using stratified 5-fold cross-validation to ensure a balanced evaluation. For each fold, 80% of the available trials were used for training and the remaining 20% for testing. Stratification was performed independently for each subject to preserve the class distribution across folds, ensuring a proportional representation of left- and right-hand motor imagery trials. This protocol enables an evaluation of model generalization in the presence of intrasubject variability.

To enable a comparison of training strategies, we adopted an evaluation protocol across all experiments. Two classifiers were employed: Linear Discriminant Analysis (LDA) and Support Vector Machine (SVM). Their performance was averaged over the five folds to mitigate the impact of random sampling.

Given that the dataset was initially partitioned into individual subject training (80%) and testing (20%) subsets, the training portions from n subjects were concatenated to construct various multi-subject training schemes. Final evaluation was consistently conducted on the untouched 20% test sets, guaranteeing that no data leakage occurred during training.

4 Dataset Description

This study employs the IVa dataset from the BCI Competition III [4], which includes EEG recordings from five healthy subjects: aa, al, av, aw, and ay. The data was acquired using 118 electrodes positioned according to the international 10/20 system, with a sampling rate of 100 Hz.

The experimental paradigm involves three types of motor imagery tasks: left hand, right hand, and feet. For the purposes of this work, only trials corresponding to left- and right-hand motor imagery were selected. Each subject contributed a total of 280 trials, divided into a set of labeled training trials and a set of unlabeled testing trials, as summarized in Table 2. Notably, the number of training and testing trials varies across subjects.

Table 2. Trial distribution per subject of five healthy subjects: *aa, al, av, aw* and *ay.*

Subject	**Training**	**Testing**
	trials	**trials**
1 (aa)	168	112
2 (al)	224	56
3 (av)	84	196
4 (aw)	56	224
5 (ay)	28	252

To ensure consistency and class balance across evaluations, the dataset was restructured to contain 224 labeled trials for training and 56 trials for testing for each subject, representing an 80%/20% train-test split. The correct labels for the test trials are available within the dataset, allowing for comprehensive evaluation.

Each EEG signal was segmented into epochs aligned with the motor imagery period. A 2-second time window was extracted from each trial, capturing the time interval during which the subject actively performed the motor imagery task.

5 Experimental Results

This section presents the experimental evaluation of various training strategies designed to explore intersubject generalization in MI-based EEG classification. First, we establish a baseline by training and testing models on each single subject (intrasubject evaluation). Then, we assess the generalization capacity of models trained under four different intersubject conditions: (1) using a single subject, (2) training on high-performing subjects (*al*, *aw*), (3) training on low-performing subjects (*aa*, *av*), and (4) training on all subjects (multisubject approach). All evaluations were performed using LDA and SVM classifiers, and results are reported in terms of accuracy, F1-score, and AUROC.

5.1 Intrasubject Evaluation

As a baseline, we trained and evaluated separate models on each single subject. Table 3 summarizes the results obtained with both classifiers. Subjects *al*, *aw*, and *ay* achieved high classification accuracy values, indicating strong motor imagery signals and model separability. Conversely, subjects *aa* and *av* yielded comparatively lower performance, highlighting individual differences and motivating further exploration of intersubject training frameworks.

Table 3. Classification performance per subject (Intrasubject Evaluation).

Subject	**SVM**			**LDA**		
	Accuracy	**Precision**	**F1-score**	**Accuracy**	**Precision**	**F1-score**
aa	85.00	84.85	84.85	82.85	82.16	83.12
al	97.85	99.31	97.81	98.21	100.00	98.16
av	75.35	73.43	76.11	74.64	73.76	75.01
aw	96.42	95.17	96.49	97.14	96.57	97.17
ay	95.71	94.20	95.87	95.71	93.34	95.86

5.2 Training on a Single Subject

In this configuration, models were trained using data from a single subject and evaluated on all others. This setup allows us to assess how well a model trained on one individual's EEG patterns generalizes across different users. Table 4 presents the accuracy results for SVM and LDA classifiers, respectively. Diagonal elements represent intrasubject performance, while off-diagonal values reflect intersubject generalization.

Table 4. Intersubject evaluation accuracy: Training on single subjects using SVM and LDA classifiers.

	Train	**Test Subject**				
Classifier	**Subject**	**aa**	**al**	**av**	**aw**	**ay**
SVM	**aa**	**85.00**	68.21	60.35	47.85	50.00
	al	50.71	**97.85**	54.64	50.00	50.00
	av	51.42	54.64	**75.35**	50.00	50.00
	aw	73.21	92.50	58.21	**96.42**	59.28
	ay	58.21	77.85	53.21	45.35	**95.71**
LDA	**aa**	**82.85**	79.64	53.21	63.92	49.64
	al	66.42	**98.21**	55.35	61.42	50.00
	av	49.64	61.42	**74.64**	51.78	55.71
	aw	72.50	94.28	53.21	**97.14**	50.71
	ay	62.14	72.14	58.21	50.00	**95.71**

5.3 Training on High-Performing Subjects

To explore whether well-performing subjects can serve as effective training sources, we trained models using data from subjects *al* and *aw*, who previously showed good intrasubject results. This strategy assumes that clean and discriminative signals may provide strong generalization when applied to less reliable data. Table 5 reports classification metrics for this strategy.

Table 5. Performance with training on high-performing subjects (*al*, *aw*).

Subject	**SVM**			**LDA**		
	Accuracy	**Precision**	**F1-score**	**Accuracy**	**Precision**	**F1-score**
aa	76.07	77.63	75.38	73.57	69.35	76.40
al	97.50	99.31	97.44	97.85	100.00	97.79
av	57.85	57.27	60.34	56.42	59.31	44.58
aw	95.71	94.01	95.84	97.14	99.31	97.04
ay	51.42	50.72	67.30	50.00	50.00	66.66

5.4 Training on Low Performing Subject

To evaluate the opposite strategy, we trained models exclusively on low-performing subjects (*aa*, *av*). The aim is to test whether models trained on inconsistent data can generalize to better-quality signals. Table 6 summarizes these results.

Table 6. Performance with training on low-performing subjects (*aa*, *av*).

Subject	SVM			LDA		
	Accuracy	**Precision**	**F1-score**	**Accuracy**	**Precision**	**F1-score**
aa	84.64	84.42	82.85	82.50	82.62	82.60
al	62.85	57.55	72.83	65.35	60.62	74.75
av	77.85	75.53	78.65	79.64	81.64	78.65
aw	50.71	50.37	66.99	54.64	52.46	68.47
ay	51.78	50.98	67.51	61.07	58.78	57.85

5.5 Multisubject Approach

Finally, we trained models using the labeled training sets (80%) from all five subjects and evaluated them on the remaining 20% for each individual. This approach seeks to maximize training diversity and capture general patterns across individuals. Table 7 presents the results. Notably, the multisubject strategy achieved a mean accuracy of 88.14%, only 2.5% points below the intrasubject baseline, while eliminating the need for subject-specific calibration.

Table 7. Performance using the multisubject training strategy.

Subject	SVM			LDA		
	Accuracy	**Precision**	**F1-score**	**Accuracy**	**Precision**	**F1-score**
aa	84.64	91.20	83.37	80.00	81.70	79.92
al	97.50	99.31	97.44	95.00	98.62	94.76
av	73.57	75.70	71.87	70.00	79.35	63.25
aw	95.00	92.59	95.13	90.35	87.21	90.80
ay	90.00	89.12	90.03	79.64	73.26	82.27

5.6 Discussion

Table 8 presents a comparison of classification performance across different training strategies. The third column shows the accuracy obtained when training and testing a subject-specific model (intrasubject scenario). This strategy consistently achieved the highest average accuracy—90.06% with SVM and 89.71% with LDA—highlighting the benefits of subject-specific characteristics in the classification model training.

The Multisubject strategy, where the model is trained on the combined data from all available subjects and tested on a held-out individual, yielded slightly lower but competitive performance (88.14% for SVM and 82.99% for LDA). This strategy demonstrates strong generalization capabilities and serves as a viable solution when subject-specific data is unavailable.

The fifth column, the accuracy results for classification models trained with high-performing subjects (al, aw), are presented. Column six then shows the results from training with low-performing subjects (aa, av). Classification models trained on high-performing subjects achieved moderate average accuracy (75.71% with SVM and 74.99% with LDA), demonstrating reasonable generalization to other individuals with similar signal characteristics. Conversely, training with low-performing subjects resulted in significant performance degradation, particularly for LDA (down to 68.64%).

Table 8. Classification accuracy (%) for different training configurations using SVM and LDA.

Classifier	**Test Subject**	**Single Subject**	**Multisubject**	**Intrasubject (al, aw)**	**Intrasubject (aa, av)**
SVM	aa	85.00	84.64	76.07	84.64
	al	97.85	97.50	97.50	62.85
	av	75.35	73.57	57.85	77.85
	aw	96.42	95.00	95.71	50.71
	ay	95.71	90.00	51.42	51.78
	Avg.	**90.06**	**88.14**	**75.71**	**65.56**
LDA	aa	82.85	80.00	73.57	82.50
	al	98.21	95.00	97.85	65.35
	av	74.64	70.00	56.42	79.64
	aw	97.14	90.35	97.14	54.64
	ay	95.71	79.64	50.00	61.07
	Avg.	**89.71**	**82.99**	**74.99**	**68.64**

These results demonstrate that classification models trained on data from high-performing subjects achieve reasonable generalization to other individuals, particularly those with similar signal quality. However, performance degrades significantly when the training set is limited to low-performing subjects, highlighting the importance of subject diversity and signal quality in multisubject training scenarios. Notably, training with all subjects leads to a more balanced performance across individuals, approaching the accuracy of subject-specific models in some cases, while enabling broader generalization.

6 Conclusions

This work evaluated multiple training configurations for motor imagery classification under subject-independent conditions. The multisubject strategy achieved an average accuracy of 88.14%, only 2.5% points below the subject-specific (intrasubject) strategy models. This small gap is significant considering that the multisubject model operates without individual calibration, offering a more scalable and user-friendly alternative for real-world BCI systems.

Furthermore, the use of only three CSP components per sub-band preserved a low-dimensional feature space, which supports both computational efficiency and practical deployment, particularly in resource-constrained or real-time applications.

The results also show that classification models (LDA and SVM) trained exclusively on high-performing subjects retain strong performance for those same individuals but fail to generalize effectively to others. Likewise, classification models trained on low-performing subjects replicate their limited performance but show poor generalization to users with better-quality signals. These findings reinforce the idea that inter-subject variability—a significant challenge in EEG-based BCI—is best addressed through diversity in the training set.

For future work, we plan to investigate transfer learning techniques by integrating external EEG datasets that follow similar motor imagery paradigms. The goal is to exploit cross-dataset and cross-subject knowledge to enhance generalization further. By doing so, we aim to reduce calibration requirements for new users and improve performance in real-world applications.

Acknowledgments. The first three authors,wish to express their gratitude for the graduate scholarships granted by Secretaría de Ciencia, Humanidades, Tecnología e Innovación (SECIHTI), Mexico. These grants have enabled them to conduct the research presented in this work.

References

1. Ang, K.K., Chin, Z.Y., Zhang, H., Guan, C.: Filter bank common spatial pattern (fbcsp) in brain-computer interface. In: 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence). pp. 2390–2397 (2008). https://doi.org/10.1109/IJCNN.2008.4634130
2. Antonakakis, M., Schrader, S., Wollbrink, A., Oostenveld, R., Rampp, S., Haueisen, J., Wolters, C.H.: Inter-subject variability of skull conductivity and thickness in calibrated realistic head models. NeuroImage **223**, 117353 (2020). https://doi.org/10.1016/j.neuroimage.2020.117353
3. Becker, S., Dhindsa, K., Mousapour, L., Al Dabagh, Y.: Bci illiteracy: It's us, not them. optimizing bcis for individual brains. In: 2022 10th International Winter Conference on Brain-Computer Interface (BCI). pp. 1–3 (2022). https://doi.org/10.1109/BCI53720.2022.9735007
4. Blankertz, B., Muller, K.R., Krusienski, D.J., Schalk, G., Wolpaw, J.R., Schlogl, A., Pfurtscheller, G., Millan, J.R., Schroder, M., Birbaumer, N.: The bci competition iii: Validating alternative approaches to actual bci problems. IEEE transactions on neural systems and rehabilitation engineering **14**(2), 153–159 (2006)
5. Ghane, P., Zarnaghinaghsh, N., Braga-Neto, U.: Comparison of classification algorithms towards subject-specific and subject-independent bci. In: 2021 9th International Winter Conference on Brain-Computer Interface (BCI). pp. 1–6 (2021). https://doi.org/10.1109/BCI51272.2021.9385339
6. Kabir, M.H., Akhtar, N.I., Tasnim, N., Miah, A.S.M., Lee, H.S., Jang, S.W., Shin, J.: Exploring feature selection and classification techniques to improve the performance of an electroencephalography-based motor imagery brain–computer in-

terface system. Sensors **24**(15), 4989 (2024). https://doi.org/10.3390/s24154989, https://doi.org/10.3390/s24154989

7. Petrov, M., Atyabi, A.: Enhancing mi-bci classification with subject-specific spatial evolutionary optimization and transfer learning. In: 2024 IEEE International Conference on Systems, Man, and Cybernetics (SMC). pp. 4770–4777 (2024). https://doi.org/10.1109/SMC54092.2024.10830968
8. Saha, S., Ahmed, K.I.U., Mostafa, R., Hadjileontiadis, L., Khandoker, A.: Evidence of variabilities in eeg dynamics during motor imagery-based multiclass brain-computer interface. IEEE Transactions on Neural Systems and Rehabilitation Engineering **26**(2), 371–382 (2018). https://doi.org/10.1109/TNSRE.2017.2778178
9. Saha, S., Baumert, M.: Discrepancy between inter- and intra-subject variability in eeg-based motor imagery brain-computer interface. Frontiers in Neuroscience **14**, 1122661 (2020). https://doi.org/10.3389/fnins.2023.1122661
10. Shiam, A.A., Hassan, K.M., Islam, M.R., Almassri, A.M.M., Wagatsuma, H., Molla, M.K.I.: Motor imagery classification using effective channel selection of multichannel eeg. Brain Sciences **14**(5) (2024). https://doi.org/10.3390/brainsci14050462, https://www.mdpi.com/2076-3425/14/5/462
11. Shuqfa, Z., Lakas, A.: Towards user-centered design for motor imagery brain-computer interface: From mi-bci illiteracy to mi-bci unfamiliarity. In: 2024 IEEE/ACM International Conference on Big Data Computing, Applications and Technologies (BDCAT). pp. 270–275 (2024). https://doi.org/10.1109/BDCAT63179.2024.00050
12. Smith, M.E., Gevins, A.: Neurophysiologic monitoring of mental workload and fatigue during operation of a flight simulator. Proceedings of the Human Factors and Ergonomics Society Annual Meeting **49**(1), 117–121 (2005). https://doi.org/10.1177/154193120504900127
13. Tao, L., Cao, T., Wang, Q., Liu, D., Sun, J.: Distribution adaptation and classification framework based on multiple kernel learning for motor imagery bci illiteracy. Sensors **22**(17) (2022). https://doi.org/10.3390/s22176572, https://www.mdpi.com/1424-8220/22/17/6572
14. Wang, T., Du, S., Dong, E.: A novel method to reduce the motor imagery bci illiteracy. Medical & Biological Engineering & Computing **59**(10), 2205–2217 (2021). https://doi.org/10.1007/s11517-021-02449-0
15. Zaremba, T., Atyabi, A.: Cross-subject & cross-dataset subject transfer in motor imagery bci systems. In: 2022 International Joint Conference on Neural Networks (IJCNN). pp. 1–8 (2022). https://doi.org/10.1109/IJCNN55064.2022.9891904
16. Zhang, R., Xu, P., Liu, T., Zhang, Y., Guo, L., Li, P., Yao, D.: Local temporal correlation common spatial patterns for single trial eeg classification during motor imagery. Computational and Mathematical Methods in Medicine **2013**, 591216 (2013). https://doi.org/10.1155/2013/591216, https://doi.org/10.1155/2013/591216

Electronic edition
Available online: http://www.rcs.cic.ipn.mx