

EDUCACIÓN

SECRETARÍA DE EDUCACIÓN PÚBLICA



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

Research in Computing Science

Vol. 155 No. 9
September 2026

Research in Computing Science

Series Editorial Board

Editors-in-Chief:

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

Associate Editors:

Jesús Angulo, *Ecole des Mines de Paris, France*
Jihad El-Sana, *Ben-Gurion Univ. of the Negev, Israel*
Alexander Gelbukh, *CIC-IPN, Mexico*
Ioannis Kakadiaris, *University of Houston, USA*
Petros Maragos, *Nat. Tech. Univ. of Athens, Greece*
Julian Padget, *University of Bath, UK*
Mateo Valero, *UPC, Barcelona, Spain*
Olga Kolesnikova, *CIC-IPN, Mexico*
Rafael Guzmán, *Univ. of Guanajuato, Mexico*
Juan Manuel Torres Moreno, *U. of Avignon, France*
Miguel González-Mendoza, *ITESM, Mexico*

Editorial Coordination:

Alejandra Ramos Porras

Research in Computing Science, Año 25, Volumen 155, No. 9, septiembre de 2026, 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. 04-2026-043011360400-102. ISSN 3122-4725, 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 septiembre de 2026.

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

Queda estrictamente prohibida la reproducción total o parcial de los contenidos e imágenes de la publicación sin previa autorización del Instituto Politécnico Nacional.

Research in Computing Science, year 25, Volume 155, No. 9, September 2026, 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.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without prior permission of Centre for Computing Research of the IPN.

Advances in Pattern Recognition

**J. Arturo Olvera-López
J. Francisco Martínez-Trinidad
Osslan O. Vergara-Villegas (eds.)**



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



Instituto Politécnico Nacional, Centro de Investigación en Computación
México 2026

ISSN: 3122-4725

Copyright © Instituto Politécnico Nacional 2026

Former 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, Gustavo A. Madero, Ciudad de México, México

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

<http://www.ipn.mx>

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

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

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

Indexed in LATINDEX, DBLP and Periodica

Electronic edition

Table of Contents

	Page
Shoplifting Prediction Using Computer Vision: Spatio-Temporal Video Abstraction via Probabilistic Graphs	5
<i>L. Enrique Morales-Márquez, J. Arturo Olvera-López, Ivan Olmos-Pineda</i>	
Explainable Robustness of Multimodal Spam Detection under Adversarial Text Attacks.....	17
<i>Samantha Acosta-Ruiz, Mireya Tovar-Vidal, José A. Reyes-Ortiz</i>	
Audio Noise Reduction Based on Spectral Attenuation	31
<i>Abril Donahi Castro-Sánchez, Gengis Cruz-Coronel</i>	
Machine Translation for Northeastern Puebla Sierra Nahuatl: Evaluating Preprocessing Scenarios and Model Convergence	45
<i>Javier Tolentino-Ramírez</i>	
Pill Counting Using Watershed Based Segmentation Methods	59
<i>Luis F. De la Rosa-Pineda, Victor M. Leyva-Osorio</i>	
Two Hands Static Mexican Sign Language Dataset for Object Detection	75
<i>Víctor Martínez-Sánchez, Ivan Villalon-Turrubiates, Francisco Cervantes</i>	
The HAJ: Initial Steps of a Novel Cryptographic Algorithm	85
<i>Daniel Alarcón-Narváez, Fausto A. Jacques-García, L. A. Lizama-Pérez, S. L. Canchola-Magdalenó, R. C. Chaparro</i>	

Shoplifting Prediction Using Computer Vision: Spatio-Temporal Video Abstraction via Probabilistic Graphs

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

Benemérita Universidad Autónoma de Puebla,
Facultad de Ciencias de la Computación,
Mexico

luise.morales@viep.com.mx,
{jose.olvera, ivan.olmos}@correo.buap.mx

Abstract. Anomaly detection in video surveillance, such as shoplifting, faces significant challenges due to high tensorial dimensionality, sensitivity to visual noise and highly computational costs. This paper proposes an approach that abstracts video sequences into directed probabilistic graphs. Through kinematic pose extraction and stochastic modeling of micro-states, the system translates complex spatio-temporal dynamics into topological signatures. We introduce a Structural Affinity metric D_t to evaluate network divergence in real-time. Experimental validation on a standardized dataset demonstrates that routine shopping behavior exhibits a dense, cyclic topology, whereas illicit actions break this pattern. Our approach ensures clear structural separability, providing a robust, interpretable, and computationally low-cost analytical foundation for automated security systems.

Keywords: Computer vision, probabilistic graphs, shoplifting detection, topological abstraction, spatio-temporal representation.

Predicción de hurtos en tiendas mediante visión artificial: abstracción espacio-temporal de vídeo a través de grafos probabilísticos

Resumen. La detección de anomalías en videovigilancia, como el robo en tiendas, presenta desafíos importantes debido a la elevada dimensionalidad tensorial, la sensibilidad al ruido visual y los altos costos computacionales. Este artículo propone un enfoque que abstrae secuencias de video en grafos probabilísticos dirigidos. Mediante la extracción cinemática de la postura y el modelado estocástico de microestados, el sistema traduce dinámicas espacio-temporales complejas en firmas topológicas. Introducimos una métrica de afinidad estructural, D_t , para evaluar la divergencia de la red en tiempo real. La validación experimental en un conjunto de datos estandarizado demuestra que el comportamiento habitual de compra exhibe una topología densa y cíclica, mientras que las acciones ilícitas rompen este patrón. Nuestro enfoque garantiza una clara separabilidad estructural, proporcionando una base analítica robusta,

interpretable y de bajo costo computacional para sistemas de seguridad automatizados.

Palabras clave: Visión por computadora, grafos probabilísticos, detección de robo en tiendas, abstracción topológica, representación espacio-temporal.

1 Introduction

The increased development and adoption of video surveillance systems have led to the generation of enormous volumes of visual data [1]. Real-time analysis of video information for anomaly detection is one of the most prominent tasks currently underway in the field of computer vision. Within this task, identifying criminal acts such as shoplifting, which involves subtle actions easily mistaken for common behavior, presents a complex problem. In contrast to anomalies such as assaults or traffic accidents, shoplifting typically involves joint interactions between the perpetrator and their surroundings, characterized by concealment or reconnaissance of the scene where the events take place [2], and which visually resemble the routine behavior of ordinary customers.

In recent years, researchers have approached video analysis by processing entire video sequences using tensors in multidimensional spaces, where spatial and temporal features are processed together, and by implementing complex deep learning architectures such as three-dimensional convolutional neural networks (3DCNNs). While these approaches demonstrate highly accurate detection rates, they also have significant limitations, such as considerably large computational costs, sensitivity to environmental variations like noise, blur, or changes in lighting, and a lack of explainability resulting from their black-box construction, which hinders the understanding and selection of the characteristics of an action that might trigger a burglary alert.

To address these weaknesses, this paper proposes an approach focused on the structural representation of the thief's behavior. Instead of delegating the extraction of temporal features to an architecture, an explicit model of the sequence of body movements is designed in a probabilistic directed graph.

This proposal argues that dynamic behaviors can be discretized into a space of finite semantic states or nodes, where the development of actions is modeled through transitions or edges. By translating the sequence of raw human poses into a graph topology, three contributions are achieved: 1) Drastic reduction of dimensionality by transforming thousands of similar human poses into a compact adjacency matrix. 2) Extraction of the topological signature of theft demonstrating that the anomalous behavior associated with shoplifting exhibits structures different from those related to normal activity. 3) Robust description of visual disturbances, establishing mathematical foundations for classification models to be developed in the future.

The remainder of this paper is organized as follows: Section 2 reviews the state of the art in graph-based video representation and anomalies. Section 3 details the mathematical formulation of graph representation, followed by Section 4, which presents experimental results. Finally, Section 5 sets out the conclusions and guidelines for future work.

2 Related Work

The computational representation of video events has been approached from multiple perspectives, ranging from extracting local frame features in the spatial domain to modeling global relationships by considering the contextual and temporal information of a series of frames.

For years, action recognition relied heavily on manually extracted features. Wang et al. [3] developed a dense path-based approach that tracked key points in video using optical flow. Their method extracted local descriptors, such as oriented gradient histograms (HOGs), optical flow histograms (HOFs), and motion boundary histograms (MBHs). While still an effective method for capturing motion patterns in short actions, it lacks a global description of the classes. Consequently, it cannot capture the semantics of long-term actions or the dependencies needed to identify prolonged and complex events such as shoplifting.

More recently, graph-based representation has become increasingly common for capturing spatiotemporal dependencies, popularized primarily by the rise of deep learning. In [4], Yan et al. introduced Spacetime Graph Convolutional Networks (ST-GCNs). Their method constructs graphs from human skeletons extracted using pose estimation, where joints serve as nodes, and identical joints in consecutive frames form temporal edges. Although ST-GCNs efficiently learn spatial and temporal patterns, they treat the graph simply as a data structure for deep feature extraction. Similarly, Wang and Gupta [5] proposed representing videos as Spacetime Region Graphs by extracting bounding boxes from objects and computed features with I3D (Inflated 3D ConvNet). They connect objects and actors as nodes based on appearance similarity and spatial proximity to capture interactions through Graph Convolutional Networks (GCNs).

While highly accurate, both approaches [4, 5] rely on massive and computationally expensive deep architectures that operate on continuous streams of pixels or coordinates, effectively obscuring the meaning of behavior within black boxes.

The community has also focused on capturing explicit relationships and mitigating noise in surveillance videos. Chen et al. [6] addressed anomaly detection using Spatiotemporal Attention Graph Networks (STGAs). By extracting skeletons and representing them as graphs, attention scores are calculated to adaptively assign weights to each joint, determining actions and mitigating background disturbances, thus allowing for the explicit modeling of relationships between objects and the scene.

Zhang et al. [7] proposed a Cognitively Guided Video Anomaly Detection (CG-VAD) framework that integrates scene knowledge graphs with Convolutional Neural Networks (CNNs) to evaluate object interactions. However, despite their effectiveness in filtering noise, these methods still rely on complex attention and continuous processing mechanisms, requiring significant computational resources that limit their scalability for real-time commercial surveillance with multiple data streams.

Despite the advances made in graph theory in recent years, the current trend focuses on generating graphs directly from deep neural networks or using them as intermediate structures for graph convolutions. This approach seeks to maximize accuracy at the expense of transparency and interpretability.

This work departs from this trend. It does not aim to use graphs as hidden intermediate layers within a deep architecture, nor to process continuous convolutional

features. Instead, it translates movement data into discrete microstates to construct a probabilistic transition graph. This graph acts as a direct and interpretable representation of the events occurring at the scene, enabling the analysis and understanding of the thief's behavior through topological divergence.

3 Methodology

The core of the approach proposed in this article consists of transforming a video sequence into a directed probabilistic graph. Since visual features can be sensitive to noise, human behavior is abstracted into the structural domain, following a stochastic model.

3.1 Dataset Characteristics

The primary attributes of the MNNIT Shoplifting Dataset [2] are detailed in Table 1. The dataset consists of short video sequences captured in a controlled laboratory environment simulating a retail store.

Unlike standard deep learning architectures that require a strict train and test data split to optimize weights, our proposed graph-based framework focuses on extracting a mathematical topological representation. The sequences were manually tagged with 28 semantic microstates. This evaluation ensures that the statistical distribution and the optimal separability threshold are representative of all available behavioral variations without the bias of a restricted testing partition.

3.2 Extraction and Preprocessing

Given a video sequence, the initial phase focuses on isolating the thief's dynamics through a structured preprocessing pipeline, as illustrated in Figure 1.

To ensure robustness before graph construction, the raw data undergoes the following stages:

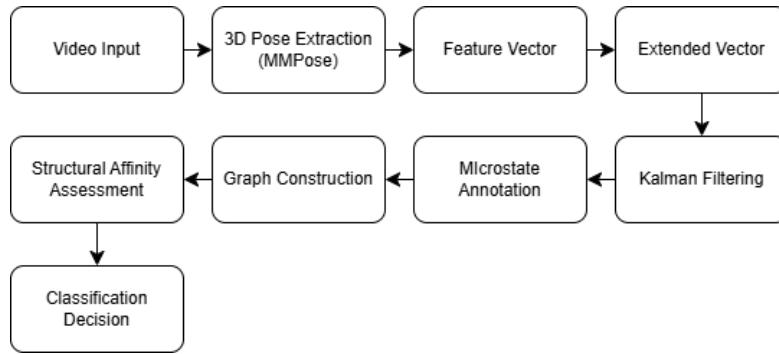
3D Pose Extraction: Starting with the video input, skeletons are extracted using the MMPose estimation model [8] to determine the three-dimensional spatial coordinates of the human body's joints.

Kinematic Feature Vector Construction: From these coordinates, a 15-dimensional vector containing selected biomechanical angles is calculated. Specifically, these 15 features were deterministically selected to focus on upper-body kinematics (e.g., shoulder flexion, elbow extension, wrist rotation, and torso inclination). This criteria intentionally isolates the manipulative actions essential to shoplifting and concealment, effectively filtering out irrelevant ambulatory noise from the lower limbs. To fully capture the temporal dynamics of the actions, this vector is expanded to 45 dimensions by computing and appending the angular velocity and angular acceleration for each angle at every frame.

Noise Reduction and Smoothing: Since raw video data and pose estimations are highly susceptible to visual noise, lighting variations, or occlusion, a Kalman filter [9]

Table 1. Key characteristics of the MNNIT Shoplifting Dataset.

Attribute	Specification
Recording conditions	Controlled indoor setup, 32MPX camera
Resolution	620x480 pixels, (5 videos with 1920x1080 pixels)
Frame Rate	30 frames per second
Total Sequences	183 clips, around 300 frames (10 seconds) per clip
Class distribution	93 Shoplifting sequences, 90 Normal behavior sequences
Action diversity	Normal: walking, inspecting items. Shoplifting: concealment in clothes/bags

**Fig. 1.** Proposed general methodology pipeline detailing the transformation of raw video input into topological signatures for anomaly classification.

is applied to the 45-dimensional sequences to obtain clean and smooth kinematic trajectories prior to their discretization.

To discretize the continuous space of body movements, the graph nodes V are defined by 28 semantic microstates corresponding to movements performed by the person e.g., 'approaching shelf', 'standing', 'reaching forward', 'inspecting item', 'retracting arm', 'concealing in pocket', etc.

This taxonomy was derived empirically from observing the dataset to capture the sequential intent behind the motion. Ground Truth is implemented through supervised annotation in which the training videos were analyzed using a sampling interval of every 5 frames, as recommended for motion capture [10]. At each selected time point, a label is assigned to the action performed or the state v_k corresponding to the present action. This systematic sampling technique reduces the redundancy inherent in a 30fps video and ensures that each node v_k represents a specific action intention and not a fleeting movement.

3.3 Graph Construction

The jump between actions that determine criminal activity is modeled as graph transitions. The adjacency matrix is constructed by estimating the conditional

probability of transitioning from state v_i to state v_j . Based on manually determined sequences, the transition probability is estimated following the Maximum Likelihood Estimation (MLE) principle for Markov chains [11] following Equation 1:

$$P(v_j|v_i) = \frac{C(v_i \rightarrow v_j)}{\sum_{k \in V} C(v_i \rightarrow v_k)}, \quad (1)$$

where $C(v_i \rightarrow v_j)$ is the number of transitions in the adjacency matrix constructed from the video set from state v_i to state v_j , and v is a state from graph. However, for sparse graphs, unannotated or very infrequent transitions can generate probability of zero, due to the computational nature of the calculations. To mitigate this condition, Laplace Smoothing [12] is applied, transforming the matrix into a logarithmic space by introducing a regularization term $\epsilon = 1 \times 10^{-20}$. This step prevents the arithmetic underflow that can occur in long strings of probabilities close to zero, as can be seen in Equation 2:

$$\log P(v_j|v_i) = \log P(v_j|v_i) + \epsilon. \quad (2)$$

3.4 Temporal Modulation

A traditional Markov chain is characterized by memory loss; that is, the probability of leaving a state does not depend on the time spent on it. However, human behavior can be influenced by the length of time spent performing an action. To model this characteristic, the graph explicitly penalizes remaining in the same state [13].

It is defined d as the duration the offender remains in the current state v_i , and a time modulation function is introduced. The logarithmic probability of remaining in the same state is dynamically penalized at each instant in time based on a decay rate λ according to Equation 3:

$$\log P'(v_j|v_i) = \log P(v_j|v_i) - \lambda(d + 1). \quad (3)$$

This formula emulates exponential decay and is based on the modeling of state duration and decay distributions [14]. The value λ forces the graph to diminish the importance of the current action after a reasonable time, allowing the system to progress through the action graph. By explicitly penalizing state retention based on duration, the proposed framework transitions from a strict Markovian assumption into a Semi-Markovian space. This mathematical modulation acts as a temporal memory mechanism, satisfying the long-range dependencies required by complex human behaviors without the prohibitive computational overhead of deep recurrent architectures.

3.5 Structural Affinity

To classify the topology observed during the playback of a video sequence, the system simultaneously maintains two matrices P_{normal} and $P_{shoplifting}$. When a transition occurs ($v_i \rightarrow v_j$), the jump divergence is calculated using an approximation of the Likelihood Ratio test [15] adapted to measure a Structural Affinity that allows the use



Fig. 2. (a) Frames corresponding to normal behavior. (b) Frames showing shoplifting. Images from MNNIT Shoplifting Dataset [2].

of both graphs, normal and shoplifting. The level of theft alarm is calculated based on Equation 4:

$$A_{trans} = \left(\frac{P_{shoplifting}(v_j|v_i)}{P_{shoplifting}(v_j|v_i) + P_{normal}(v_j|v_i) + \epsilon} \right) \times 100. \quad (4)$$

Since video sequences can contain noise caused by atypical micro-movements lasting only an instant, evaluating a single frame can produce unstable results. Therefore, instantaneous affinity should be combined with a global metric D_t using a Weighted Moving Average (EWMA) [15], see Equation 5:

$$D_t = \alpha D_{t-1} + (1 - \alpha) A_{trans}, \quad (5)$$

where $\alpha \in [0,1]$ behaves as a historical weighting or inertia factor inspired by time series analysis for quality control [16].

4 Results

To validate the topological mapping capability of the behavioral patterns, the entire dataset was processed, considering all 93 shoplifting sequences and the 90 normal behavior sequences. The main objective of these experiments was to evaluate the effectiveness of data representation and to depict the structural separability between the two classes from a mathematical and graphical perspective before processing the data with any type of classifier.

The central proposition of this paper establishes that licit and illicit behaviors possess divergent topological signatures. To verify this, graph metrics were extracted for each video in the dataset. In the Scatter Plot of Figure 3 shows the clustering coefficients against the graph density of each video evaluated. The x -axis correspond to the clustering coefficient, which quantifies the formation of cyclical and return sequences, where values tending towards zero reveal a certain acyclic directionality. Simultaneously, the y -axis represents the network density, which measures the

proportion of transitions executed, that is, the complexity of the chain of actions and gives indications of a clear or random intention.

Normal sequences tend to be located in the middle and upper regions of both axes. A high clustering coefficient indicates a greater prominence of cyclical behaviors, such as picking up objects and observing them out of curiosity rather than performing a specific action. Higher density also reflects action sequences consistent with less restricted activity. A person who does not engage in shoplifting tends to perform closed cycles, such as picking up a product, examining it, and returning it to repeat the process. Therefore, the network of actions in normal behaviors tends to be more convoluted, as the chain of movements is often more erratic and random.

Shoplifting sequences maintain lower values in both density and clustering coefficient, indicating predominantly linear rather than cyclical behaviors, and suggesting that thefts are represented by less complex behaviors. The low clustering coefficient prevalent in shoplifting examples suggests patterns that move primarily in one direction with more sporadic returns or cycles compared to normal behaviors. Furthermore, the lower density compared to sequences without shoplifting is consistent with premeditated actions and a clearer objective, ignoring other possible actions.

Although the decision boundary is not obvious, the point clouds of both classes are positioned in particular regions, indicating that this kinematic discretization is capable of effectively isolating anomalies.

To quantify the system's response to theft, the maximum structural affinity, or theft probability, reached by each video was evaluated. The maximum values are summarized in Figure 4. The results suggest that, statistically, both types of behavior are completely separable.

Statistical analysis of the D_t distribution shows that for normal behavior, the median divergence is 44%, with a range from 41% to 48%. The low variance and compact dispersion indicate that most legitimate sequences remain similar and without significant deviations. It is important to mention that the absolute maximum is 59%, an observation that helps establish a divergence decision threshold, as will be discussed in the following paragraph.

Conversely, the shoplifting distribution exhibits a median that spikes to 87%, with a range between 82% and 93%. While the dispersion is greater, a consequence of the variations in object concealment techniques, the concentration of most instances in a block of no more than 10% indicates that the method captures the majority of illicit action sequences. Even the outliers present at 61% divergence allow for setting a strict threshold of 60% optimally to separate the two classes, as the total number of normal sequences remains below the threshold, indicating a low rate of topological false positives. Therefore, ambiguous movements do not appear to have a significant impact on the model. Furthermore, the distribution of shoplifting videos shows that most exceed 80% divergence, unlike normal videos, which remain below 50%.

As a particular case to illustrate the behavior of structural divergence in real time, Figure 5 shows the evolution of this metric over time for all videos in dataset.

Figure 5 shows the temporal evolution of structural divergence through transitions in the video sequences. It should be noted that, to standardize sizes, all sequences in each class are adapted to the largest video size, maintaining the divergence value over extended time intervals. When analyzing the behavior in the normal class, values below

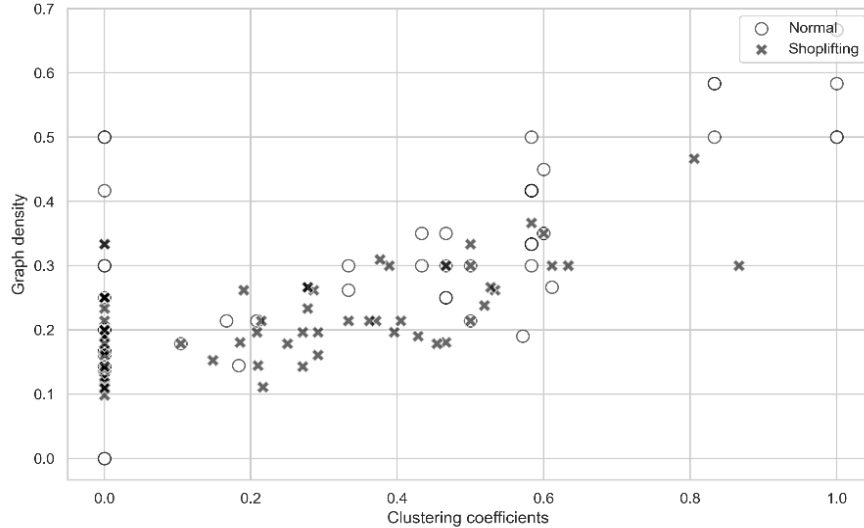


Fig. 3. Scatter Plot of separability of classes considering clustering coefficients and graph density for all dataset.

the threshold are observed at all times. While the maximum ranges are variable, the flatness is stable, remaining at the bottom of the graph.

In contrast, the trajectories associated with shoplifting tend to increase abruptly, and once they exceed 60% divergence, the risk rarely falls below this threshold again. The fact that videos of this class cluster above videos of innocent activity demonstrate the model's ability to profile anomalous behaviors associated with theft with a low false positive rate.

Experiments on the dataset show that criminal actions possess a distinct and isolable structural signature characterized by low density and a low clustering coefficient, behavior contrary to normal sequences.

It has also been found that formulating a structural divergence metric that takes past actions into account allows for reliably stable reaction to and recognition of anomalies in the face of micro-movements in noise. Finally, graph representation drastically reduces data complexity by using adjacency matrices, enabling the analysis of multiple surveillance streams with minimal computational resources.

The temporal decay (λ) and EWMA inertia (α) parameters were determined through a sensitivity analysis exploring the range [0.1, 0.9] for both variables. For the decay parameter, values above 0.7 caused the system to retain memory of previous micro-states for too long, delaying the transition into anomaly detection.

Conversely, values below 0.3 forced the model to forget past actions too quickly, making it overly sensitive to noise. The balance was found at $\lambda = 0.5$, which aligns with the natural duration of human reaching and concealing motions.

Similarly, for the inertia factor, an $\alpha < 0.5$ reduced the historical weighting, acting almost like an instantaneous derivative that triggered false positives due to minor pose estimation inaccuracies.

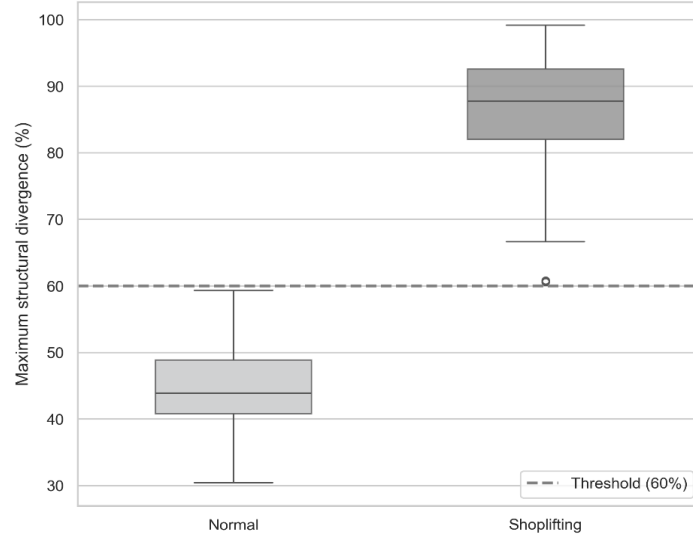


Fig. 4. Box Plot of distribution of the maximum peak of structural divergence D_t reached by each evaluated video.

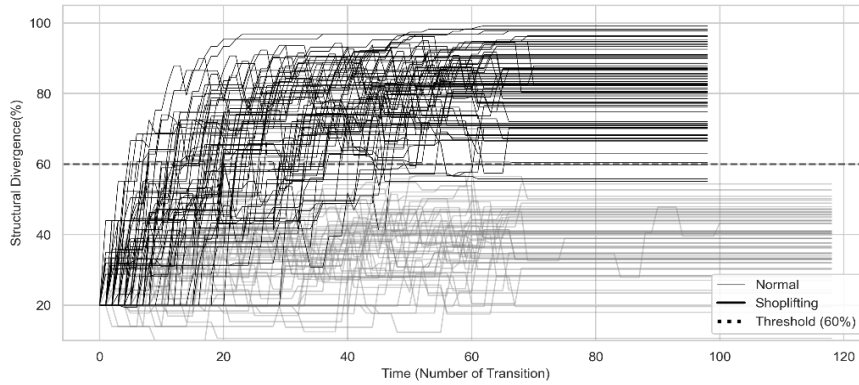


Fig. 5. Temporal evolution of the structural divergence D_t evaluated on the complete dataset.

Values of $\alpha > 0.8$ heavily suppressed sudden topological divergences, creating false negatives by ignoring legitimate theft actions. Establishing $\alpha = 0.7$ successfully smoothed tracking jitter while maintaining a swift reaction to actual illicit structural signatures.

5 Conclusions

This article presents a framework for modeling complex human behavior in video sequences using topological abstraction. By discretizing motion data into probabilistic

graphs, the proposed methodology captures the spatiotemporal dynamics of an event without the computational exhaustiveness associated with continuous frame-by-frame pixel analysis.

Translating sequences of human postures into microstates achieves dimensionality reduction by using lightweight adjacency matrices, preserving the behavioral intention of the shopper or thief. Experimental results show that shoplifting actions possess a distinctive and isolable structural signature.

This work shown that anomalous events tend to deviate from the cyclical and exploratory nature of normal shopping behavior. Instead of forming dense networks with high clustering coefficients, shoplifting sequences choose directed and acyclic paths characterized by low graph density and a terminal focus on concealment. Furthermore, the implementation of the structural affinity metric D_t proved effective in reliably tracking and isolating anomalies, even considering the variability and noise of benign micromovements.

Finally, this approach indicates that structural divergence works as a reliable and practical indicator of shoplifting intent. The observed separability between legitimate and illicit trajectories can yield a low false positive rate, which is an essential requirement for real-world security implementations.

Future research will focus on transitioning this framework from a descriptive anomaly detection tool to a fully predictive environment. The proposed path involves using these graph-based features as direct inputs for advanced machine learning architectures.

By combining this structural representation with continuous estimators of human posture, the final objective is to establish a real-time anticipated probability of shoplifting, enabling intervention strategies in busy retail environments.

References

1. Ramachandra, B., Jones, M., Vatsavai, R.R.: A Survey of Single-Scene Video Anomaly Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(5), pp. 1 (2020). Doi: 10.1109/TPAMI.2020.3040591.
2. Ansari, M.A.: Shoplifting Dataset - CV Laboratory MNNIT Allahabad (2023). Doi: 10.17632/r3yjf35hxr.1.
3. Wang, H., Kläser, A., Schmid, C., Liu, C.: Dense Trajectories and Motion Boundary Descriptors for Action Recognition. *International Journal of Computer Vision*, 103(1), pp. 60–79 (2013). Doi: 10.1007/s11263-012-0594-8.
4. Yan, S., Xiong, Y., Lin, D.: Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1) (2018). Doi: 10.1609/aaai.v32i1.12328.
5. Wang, X., Gupta, A.: Videos as Space-Time Region Graphs. *Lecture Notes in Computer Science*, pp. 413–431 (2018). Doi: 10.1007/978-3-030-01228-1_25.
6. Chen, H., Mei, X., Ma, Z., Wu, X., Wei, Y.: Spatial–Temporal Graph Attention Network for Video Anomaly Detection. *Image And Vision Computing*, 131, pp. 104–629 (2023). Doi: 10.1016/j.imavis.2023.104629.
7. Zhang, M., Wang, J., Qi, Q., Zhuang, Z., Sun, H., Liao, J.: Cognition Guided Video Anomaly Detection Framework for Surveillance Services. *IEEE Transactions on Services Computing*, 17(5), pp. 2109–2123 (2024). Doi: 10.1109/tsc.2024.3407588.
8. Welcome To MMPose's Documentation: <https://mmpose.readthedocs.io/en/latest/>.

9. Welch, G., Bishop, G.: An Introduction to the Kalman Filter. Technical Report TR 95-041. Department of Computer Science, University of North Carolina at Chapel Hill (1995)
10. Vidanapathirana, M., Sudasingha, I., Vidanapathirana, J., Kanchana, P., Perera, I.: Tracking and Frame-Rate Enhancement for Real-Time 2D Human Pose Estimation. *The Visual Computer*, 36(7), pp. 1501–1519 (2019). Doi: 10.1007/s00371-019-01757-9.
11. Bishop, C.M.: *Pattern Recognition and Machine Learning*. Springer Verlag (2006)
12. Jurafsky, D., Martin, J.H.: *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition* (2000)
13. Yu, S.: Hidden Semi-Markov Models. *Artificial Intelligence*, 174(2), pp. 215–243 (2009). Doi: 10.1016/j.artint.2009.11.011.
14. Rabiner, L.: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, 77(2), pp. 257–286 (1989). Doi: 10.1109/5.18626.
15. Chandola, V., Banerjee, A., Kumar, V.: Anomaly Detection. *ACM Computing Surveys*, 41(3), pp. 1–58 (2009). Doi: 10.1145/1541880.1541882.
16. Hunter, J.S.: The Exponentially Weighted Moving Average. *Journal of Quality Technology*, 18(4), pp. 203–210 (1986). Doi: 10.1080/00224065.1986.11979014.

Explainable Robustness of Multimodal Spam Detection under Adversarial Text Attacks

Samantha Acosta-Ruiz¹, Mireya Tovar-Vidal¹, José A. Reyes-Ortiz²

¹ Benemérita Universidad Autónoma de Puebla,
Facultad de Ciencias de la Computación, Puebla,
Mexico

² Universidad Autónoma Metropolitana,
División de Ciencias Básicas e Ingeniería, Azcapotzalco, Mexico City,
Mexico

jaro@azc.uam.mx, ar224570157@alm.buap.mx, mireya.tovar@correo.buap.mx

Abstract. Multimodal spam detection systems achieve high performance under clean conditions by combining textual and visual information; however, their robustness under adversarial scenarios remains insufficiently explored. While textual adversarial attacks have been widely studied in text-only classification, it is still unclear whether visual information helps preserve spam detection performance when the textual branch is intentionally perturbed. Results indicate that PWWS-based adversarial attacks significantly reduce spam recall in text-only classification from 0.9820 to 0.8619, thereby increasing critical false negatives. In contrast, multimodal models preserve substantially higher recall under the same attack conditions, reaching 0.9708, which indicates that visual information acts as a robustness support mechanism. Grad-CAM and OCR overlap analysis further support these findings by showing stronger visual alignment in rescue cases and weaker correspondence when both modalities fail. The results suggest that multimodal fusion provides not only better classification performance under clean conditions, but also greater resilience against adversarial manipulation, underscoring the need for adversarial robustness evaluation in practical multimodal spam detection systems.

Keywords: Multimodal spam detection, adversarial text attacks, robustness evaluation, multimodal fusion, explainable AI.

Robustez explicable de la detección multimodal de spam frente a ataques adversarios de texto

Resumen. Los sistemas multimodales de detección de spam logran un alto rendimiento en condiciones normales al combinar información textual y visual; sin embargo, su robustez en escenarios adversarios no ha sido suficientemente explorada. Si bien los ataques adversarios

textuales han sido ampliamente estudiados en la clasificación basada únicamente en texto, aún no está claro si la información visual ayuda a mantener el rendimiento de la detección de spam cuando la rama textual es perturbada intencionalmente. Los resultados indican que los ataques adversarios basados en PWWS reducen significativamente la tasa de recuperación (*recall*) de spam en la clasificación de solo texto, pasando de 0,9820 a 0,8619, lo que aumenta el número de falsos negativos críticos. En cambio, los modelos multimodales mantienen una tasa de recuperación sustancialmente mayor bajo las mismas condiciones de ataque, alcanzando 0,9708; esto demuestra que la información visual actúa como un mecanismo de apoyo a la robustez. El análisis de superposición mediante Grad-CAM y OCR respalda estos hallazgos al mostrar una alineación visual más fuerte en los casos recuperados con éxito y una correspondencia más débil cuando ambas modalidades fallan. Los resultados sugieren que la fusión multimodal no solo proporciona un mejor rendimiento de clasificación en condiciones normales, sino también una mayor resiliencia frente a la manipulación adversaria, lo que subraya la necesidad de evaluar la robustez adversaria en sistemas prácticos de detección multimodal de spam.

Palabras clave: Detección multimodal de spam, ataques adversarios de texto, evaluación de robustez, fusión multimodal, IA explicable.

1 Introduction

The growing reliance on email communication in personal, academic and organizational settings has also increased the exposure to cybersecurity threats. Spam emails remain one of the most common attack vectors for phishing campaigns, malware distribution, fraudulent promotions and social engineering attempts, making reliable spam detection an essential task in modern cybersecurity [3]. To mitigate these risks, machine learning and deep learning models have been widely adopted for automatic spam filtering, often achieving high performance under clean and controlled conditions. In particular, recent comparative studies have shown that transformer-based architectures, such as DistilBERT, provide highly competitive results for malware and spam email identification, highlighting the effectiveness of NLP-based approaches for email security tasks [4].

However, strong performance under normal conditions does not guarantee robustness against adversarial manipulation. Attackers can introduce lexical, syntactic or semantic perturbations that preserve the malicious intent of an email while altering its representation enough to fool the classifier. Even small modifications are usually enough to reduce the detection performance, especially in security-sensitive applications, such as spam filtering, where misclassifying malicious messages as legitimate can allow phishing attempts, malware delivery and other harmful threats to bypass protection mechanisms [10].

In real-world scenarios, spam emails are rarely limited to plain text. Many malicious campaigns rely on embedded images, banners, logos, visual promotions

and structured layouts that complement or even replace textual content to evade traditional text-based filters. In these cases, the malicious signal is frequently distributed across both textual and visual modalities, which makes text-only analysis insufficient for reliable detection. This limitation has driven the development of multimodal spam detection systems that combine textual and visual information, allowing models to capture complementary evidence and improve classification performance [11].

Nevertheless, most adversarial robustness studies still focus on text-only classification or report multimodal results using aggregate metrics such as accuracy and F_1 -Score. These metrics provide limited insight into whether visual information can compensate for adversarial perturbations introduced into textual content and under which conditions multimodal rescue or failure occurs. Understanding these rescue and failure cases is particularly important for determining whether multimodal architectures offer genuine robustness rather than performance gains only under clean conditions.

This work explores the explainable robustness of multimodal spam detection against adversarial text attacks. We investigate whether combining textual and visual information improves robustness when adversarial perturbations are introduced into the textual branch, and whether visual information helps preserve correct spam classification under attack conditions. Through Grad-CAM, OCR overlap analysis, and rescue-case evaluation, we provide a more interpretable framework for understanding adversarial robustness beyond conventional aggregate performance metrics, enabling a more rigorous interpretation of when visual evidence compensates for textual degradation and when both modalities fail.

2 Related Work

The robustness of spam detection systems has become an increasingly important research topic due to the growing sophistication of adversarial manipulation in email communication. Although modern classifiers achieve strong performance under standard evaluation settings, their reliability can be significantly reduced when attackers introduce carefully crafted perturbations designed to evade detection. In natural language processing, this problem is particularly critical because small lexical or contextual modifications are often sufficient to alter model predictions without changing the semantic intent of the original message.

Early adversarial text attacks demonstrated that carefully crafted perturbations can significantly degrade classifier performance while preserving the original semantic intent. TextBugger [6] introduced character- and word-level perturbations capable of deceiving deep text classifiers under both white-box and black-box settings, highlighting the vulnerability of NLP models to subtle textual modifications. Building upon this idea, PWWS [7] showed that identifying influential words through probability-weighted saliency and replacing them with semantically similar alternatives can effectively fool classifiers while maintaining readability.

The effectiveness of saliency-guided adversarial generation was further demonstrated by TextFooler [5], which uses word-importance ranking and synonym substitution to reduce classification performance while preserving semantic consistency substantially. More recently, TextJosher [9] extended these ideas through a transfer-based black-box attack capable of generating adversarial examples that remain effective across different models without requiring direct access to the target classifier.

While adversarial attacks have been studied extensively in text-only classification, real spam campaigns often combine textual and visual content to evade traditional filters. Images, banners, logos and structured document layouts often contain relevant malicious information that cannot be captured through textual analysis alone. This limitation has motivated the development of multimodal spam detection systems that integrate both modalities to improve classification performance.

Architectures such as MMPC-RF [2] and MMTD [11] show that feature-level fusion between textual and visual representations consistently outperforms unimodal approaches, particularly when malicious intent is distributed across both modalities.

Similarly, recent studies using DistilBERT combined with structural and multimodal features report stronger classification performance and greater robustness in spam detection tasks [1]. However, these studies primarily evaluate classification effectiveness under clean conditions and provide limited evidence regarding adversarial robustness.

In parallel, explainability methods have become increasingly relevant for understanding model behavior under adversarial conditions. Techniques such as Grad-CAM [8] make it possible to visualize the image regions that contribute most to the final prediction, helping determine whether the model relies on semantically meaningful evidence or on irrelevant visual artifacts. This capability is particularly important in multimodal security applications because apparent robustness may result from spurious correlations rather than genuine complementary information between modalities. However, explainability is rarely incorporated into adversarial evaluations of multimodal spam detection systems.

Despite advances in adversarial NLP, multimodal spam detection, and explainable AI, their intersection remains insufficiently explored. Existing studies either focus on adversarial attacks in text-only settings or evaluate multimodal spam detection under clean conditions, primarily emphasizing performance improvements achieved through fusion strategies. Consequently, there is limited understanding of how textual adversarial perturbations affect multimodal robustness and whether visual information can compensate for the resulting degradation in spam detection performance.

Furthermore, the contribution of the visual modality as a robustness mechanism under adversarial conditions remains unclear. As a result, it is still uncertain whether multimodal architectures provide genuine robustness against adversarial perturbations or merely achieve improved predictive performance under standard evaluation settings.

3 Methodology

The experiments were carried out using the MMTD dataset [11], a multilingual and multimodal spam detection corpus containing textual content, document images and binary labels for spam and ham classification. The original dataset consists of 31,212 samples organized into three main attributes: textual message (*texts*), associated image path (*pics*) and class label (*labels*), where 1 represents spam and 0 represents ham. Since this work focuses on adversarial robustness in the textual branch, only English samples were selected to ensure linguistic consistency during the adversarial generation. Language identification was performed through the textual modality and the entries without textual content (*no_text*) were eliminated, resulting in 7,712 samples in English.

To improve data quality, text-length distribution was analyzed using boxplot-based quartile filtering to identify and remove atypically short and excessively long messages. This process reduced noise caused by abnormal samples and produced a final textual subset of 7,082 instances while preserving the original class imbalance between spam and ham messages. All retained samples were also verified to maintain valid correspondence with their associated document images, ensuring consistency for multimodal training and evaluation. A standard text normalization process was applied using the *textacy* preprocessing pipeline, including lowercase conversion, URL removal, punctuation normalization, elimination of non-semantic artifacts, and whitespace normalization. This step preserved the semantic content of each message while reducing irrelevant noise for both classification and adversarial evaluation. Since adversarial perturbations are introduced later at the lexical level, preprocessing focused on preserving meaningful textual structure rather than orthographic artifacts.

The dataset was divided using a stratified 70/15/15 strategy for training, validation, and final testing, preserving the original class distribution. The training and validation partitions were used during a 3-fold stratified cross-validation procedure to obtain robust performance estimates and standard deviation analysis, while the final test partition was reserved exclusively for the robustness evaluation. DistilBERT was selected as the text-only baseline due to its strong performance and lower computational cost compared to larger transformer architectures. The maximum sequence length was set to 128 tokens, as the text-length distribution indicated that most messages could be fully represented within this range while maintaining computational efficiency.

For the multimodal model, DistilBERT and ResNet50 were employed as textual and visual encoders, respectively. The textual representation was obtained from the first-token embedding of the final hidden layer of DistilBERT (*last_hidden_state[:,0,:]*), which provides a contextual representation of the complete message. For the visual branch, a ResNet50 model pretrained on ImageNet was used, replacing its original fully connected classification layer with an identity mapping. Consequently, the visual representation corresponded to the 2048-dimensional feature vector produced after the global average pooling stage of the network. To facilitate multimodal fusion, both textual and visual representations were projected into 256-dimensional vectors and concatenated

into a unified 512-dimensional representation, which was subsequently processed through fully connected layers for the final spam classification.

The adopted early-fusion strategy enables the model to jointly learn complementary information from both modalities and evaluate whether visual evidence can compensate for the adversarial degradation introduced into the textual branch.

To evaluate the adversarial robustness, PWWS [7] was applied to the test set as a textual attack method. PWWS generates adversarial examples by identifying the most influential words for classification and replacing them with semantically similar alternatives while preserving readability and malicious intent. Since the attack targets only the textual modality, two adversarial evaluation configurations were considered: text-only attack and multimodal adversarial setting. In the first case, DistilBERT was evaluated directly on the perturbed text. In the second case, the adversarial text replaced the original textual branch, while the visual modality remained unchanged, allowing an analysis of whether visual information helps preserve spam detection performance under attack conditions.

This design intentionally isolates the effect of textual adversarial perturbations, allowing the contribution of visual information to robustness preservation to be evaluated independently from image-based attacks. Consequently, the experimental setup focuses on determining whether visual evidence can compensate for textual degradation rather than evaluating joint multimodal attack strategies.

Because adversarial generation may fail or produce duplicate outputs, a re-alignment stage was required for multimodal evaluation after PWWS generation. While the original multimodal baseline was constructed using the complete aligned text-image test set, adversarial evaluation preserved only samples containing valid adversarial text, document images, and consistent label assignments. After removing duplicates and invalid matches, the final aligned multimodal subset contained 1058 instances composed of 718 spam samples and 340 ham samples. Among them, 171 corresponded to successful adversarial attacks, while 887 retained the status of failed attacks.

Consequently, the clean multimodal subset reported during adversarial evaluation does not correspond to the complete multimodal baseline, but rather to the clean evaluation restricted to the same aligned samples used in the adversarial setting. Therefore, all clean-versus-adversarial comparisons were conducted on the same aligned subset, ensuring a fair and consistent evaluation protocol.

Performance was evaluated using Accuracy, Precision, Recall for the spam class, F_1 -Score, and confusion matrices under both clean and adversarial conditions. Special attention was paid to recall preservation and false-negative behavior since misclassifying spam as legitimate messages represents the highest security risk in practical deployment. In addition, statistical significance testing was performed to determine whether the observed differences between clean and adversarial conditions were attributable to adversarial perturbations rather than random variation.

Table 1. Baseline performance under clean conditions for multi-modal, text-only and image-only spam detection models.

Model	Accuracy	Precision	Recall	Spam F ₁ -score
Text-only (DistilBERT)	0.9793	0.9875	0.9820	0.9848
Image-only (ResNet50)	0.9566	0.9827	0.9528	0.9675
Multimodal (Text + Image)	0.9929	0.9927	0.9969	0.9948

Finally, an explainability analysis was incorporated to investigate rescue and failure behavior under adversarial conditions. Grad-CAM was used to visualize the image regions that most contributed to spam predictions, while OCR overlap was quantified using the Intersection over Union (IoU) between OCR-extracted text regions and Grad-CAM activation areas. This analysis provided both qualitative and quantitative evidence regarding whether visual attention remained aligned with semantically relevant spam content and whether the visual modality contributed to preserving correct classifications under textual adversarial perturbations.

4 Results

This section presents the experimental evaluation of multimodal spam detection under textual adversarial attacks. The analysis focuses on whether visual information helps preserve spam detection performance when the textual branch is intentionally manipulated, with special attention to recall preservation for the spam class, since false negatives represent the highest security risk in practical spam filtering systems. In addition to aggregate metrics, Grad-CAM and OCR overlap are used to analyze rescue and failure cases under adversarial conditions.

Following the experimental protocol described in the methodology section, model performance was evaluated using Accuracy, Precision, Recall for the spam class, and F₁-score. A 3-fold stratified cross-validation procedure was applied over the development split to ensure reliable performance estimation, while final results were obtained using the independent test set.

Table 1 summarizes the performance of the model in clean conditions for text-only, image-only and multimodal configurations. DistilBERT achieves strong results in text-only classification, while ResNet50 also shows competitive performance using only document images. However, the multimodal model achieves the best overall performance, reaching the highest Accuracy, Precision, Recall, and F₁-score. In particular, the recall of spam increases from 0.9820 in the text-only configuration to 0.9969 in the multimodal model, suggesting that the textual and visual modalities provide complementary information for spam detection. These results establish a solid baseline and suggest that multimodal fusion improves detection reliability before introducing adversarial perturbations. This baseline serves as a reference point to quantify the impact of adversarial perturbations and assess whether the observed differences are still statistically significant under attack conditions.

Table 2. Performance degradation in the text-only model under PWWS adversarial attack.

Setting	Accuracy	Precision	Recall Spam	F ₁ -score
Text-only clean	0.9793	0.9875	0.9820	0.9848
Text-only attacked (PWWS)	0.8383	0.8966	0.8619	0.8789

Table 3. Multimodal performance under clean and adversarial textual conditions evaluated on the same test partition.

Setting	Accuracy	Recall Spam	F ₁ -score
Multimodal clean	0.9339	0.9654	0.9519
Multimodal adversarial	0.8222	0.9434	0.8780

To evaluate whether this robustness remains under adversarial manipulation, PWWS was first applied to the text-only setting using DistilBERT. Since the attack targets only the textual modality, this experiment allows for the quantification of the impact of lexical perturbations on spam detection performance before analyzing the multimodal setting.

Table 2 shows a substantial degradation in performance under attack, with Accuracy decreasing from 0.9793 to 0.8383 and Recall for the spam class dropping from 0.9820 to 0.8619. This corresponds to an absolute decrease of 14.1 percentage points in Accuracy and 12.0 percentage points in spam Recall. This reduction indicates that lexical perturbations are sufficient to alter model predictions while preserving semantic meaning, increasing false negatives and exposing the vulnerability of text-only spam detection systems under adversarial conditions. These findings suggest that relying solely on textual information makes the classifier highly sensitive to adversarial manipulation.

After establishing the vulnerability of the text-only model, the next step was to analyze whether multimodal fusion helps preserve performance when the textual branch is attacked. To ensure a fair comparison between clean and adversarial conditions, PWWS perturbations were generated directly from the same multimodal test partition. Consequently, each adversarial sample preserved its original image, label, and sample identifier, allowing clean and adversarial evaluations to be performed on exactly the same set of 1,406 instances. Therefore, the multimodal results reported in Table 3 correspond to paired evaluations in which only the textual modality was modified while the visual information remained unchanged. This setup enables a direct assessment of the extent to which visual information contributes to maintaining spam detection performance under adversarial textual perturbations.

Table 3 presents multimodal performance under textual adversarial attack. Although the overall Accuracy decreases from 0.9339 to 0.8222 and Recall for the spam class decreases from 0.9654 to 0.9434, this corresponds to an absolute reduction of 11.2 percentage points in Accuracy but only 2.2 percentage points in spam Recall. These results suggest that the visual modality helps preserve spam detection sensitivity even when the textual branch is compromised. Unlike the text-only setting, where adversarial perturbations produce substantially larger

Table 4. Comparison between text-only and multimodal models under adversarial conditions.

Model	Accuracy	Recall Spam	F ₁ -score
Text-only attacked	0.8383	0.8619	0.8789
Multimodal adversarial	0.8222	0.9434	0.8780

Table 5. Statistical significance analysis of spam recall degradation under adversarial conditions.

Model	Recall Clean	Recall Attack	Z-score	P-value
Text-only	0.9820	0.8619	7.34	2.13×10^{-13}
Multimodal	0.9654	0.9434	2.30	0.0212

degradation, multimodal fusion provides complementary evidence that partially compensates for lexical manipulation. To determine whether these differences were statistically meaningful rather than the result of random variation, a statistical significance analysis was subsequently performed.

To directly compare both attacked settings, Table 4 contrasts the performance of the text-only attacked model and the multimodal adversarial model. This comparison is particularly important because it allows us to determine whether visual information functions only as complementary support under clean conditions or as a genuine robustness mechanism under adversarial attacks.

As shown in Table 4, while the text-only attacked model achieves a spam recall of 0.8619, the multimodal adversarial model preserves a substantially higher recall of 0.9434 under the same attack conditions. This corresponds to an absolute improvement of 8.2 percentage points in spam recall. Although the multimodal model exhibits slightly lower Accuracy, it maintains considerably stronger spam detection capability, as reflected by its higher recall and lower number of false negatives. Specifically, the number of spam emails incorrectly classified as legitimate messages decreases from 92 in the text-only setting to 54 in the multimodal configuration. These results suggest that visual information provides complementary evidence that helps compensate for the degradation introduced in the textual modality.

Although the previous results indicate that multimodal fusion preserves substantially higher spam recall under adversarial conditions, it is important to determine whether the observed differences are statistically significant rather than the result of random variation. Therefore, a two-proportion Z-test was conducted using spam recall under clean and adversarial conditions for both the text-only and multimodal models.

Table 5 shows that the degradation produced by PWWS is statistically significant for both architectures ($p < 0.05$). However, the impact is considerably smaller in the multimodal setting. While the text-only classifier experiences a reduction of 12.0 percentage points in spam recall, the multimodal model shows a decrease of only 2.2 percentage points. Despite the adversarial perturbations, the multimodal architecture preserves a substantially higher spam detection capability. These findings provide statistical evidence that multimodal fusion

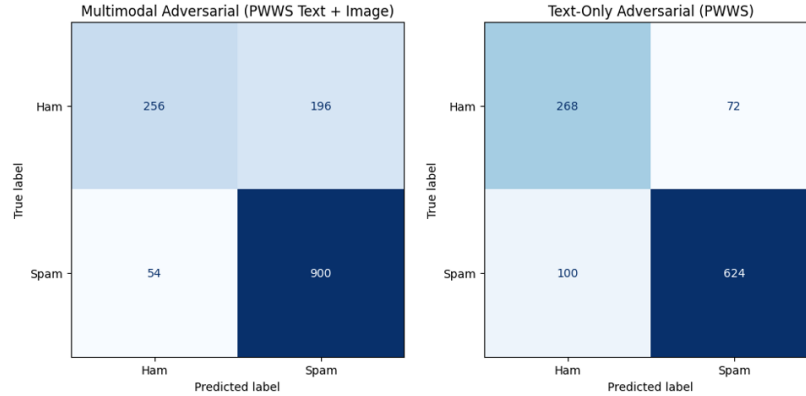


Fig. 1. Confusion matrices for the text-only and multimodal models under PWWS adversarial conditions.

reduces the impact of textual adversarial attacks and support the hypothesis that visual information contributes to maintaining spam detection performance when the textual modality is compromised.

While the statistical analysis quantifies the significance of the observed degradation, it does not reveal how the errors are distributed across classes. To further examine the practical impact of adversarial perturbations, Figure 1 presents the confusion matrices obtained for the text-only and multimodal models under adversarial conditions. It should be noted that the confusion matrices correspond to the evaluation settings used for each experiment and therefore contain different class distributions. Consequently, absolute counts should not be directly compared across models. Instead, the analysis focuses on the relative distribution of classification errors, particularly false negatives, since these determine the spam recall and represent the most critical failure in spam filtering systems.

As shown in Figure 1, the text-only model incorrectly classifies 100 spam emails as legitimate messages, whereas the multimodal model reduces this number to 54. Although the absolute counts originate from evaluation sets with different class distributions, the reduction in false negatives is consistent with the higher spam recall achieved by the multimodal architecture. From a cybersecurity perspective, this behavior is particularly important because false negatives allow malicious emails to bypass the filtering system. Overall, the confusion matrix analysis confirms that multimodal fusion better preserves spam detection performance under adversarial conditions.

Although the above analyses demonstrate that multimodal fusion improves robustness against textual adversarial attacks, they do not explain how the visual modality contributes to the final decision process.

To gain further insight into this behavior, an explainability analysis based on the superposition of Grad-CAM and OCR was performed. Specifically, the Intersection over Union (IoU) metric was used to quantify the overlap between

Table 6. OCR-GradCAM overlap measured using IoU under clean and adversarial conditions.

Setting	Mean IoU	Std IoU	Median IoU	Valid Samples
Multimodal Clean	0.0744	0.0556	0.0705	1406
Multimodal Adversarial	0.0686	0.0545	0.0631	1406

Table 7. Rescue and failure case analysis under adversarial conditions.

Case Type	Count	Percentage (%)	Mean IoU	Std IoU
Both Correct	1027	73.04	0.0708	0.0546
Both Incorrect	113	8.04	0.0635	0.0578
Multimodal Failure	137	9.74	0.0577	0.0539
Rescue Case	129	9.17	0.0669	0.0502

Grad-CAM activation regions and text regions detected by OCR under clean and adversarial conditions.

Table 6 presents the overlap between Grad-CAM activation regions and OCR-detected text areas for clean and adversarial inputs. Although PWWS alters the textual content, the decrease in overlap remains relatively small, with the mean IoU dropping from 0.0744 to 0.0686 and the median IoU from 0.0705 to 0.0631. These findings suggest that adversarial perturbations primarily affect the textual branch, while the visual attention patterns remain comparatively stable. Consequently, the visual modality appears to retain access to relevant document information that may help mitigate the impact of textual manipulation.

While IoU provides a global measure of attention stability, it does not reveal how the interaction between textual and visual information influences individual prediction outcomes. Therefore, a rescue and failure analysis was performed to identify situations in which multimodal fusion successfully compensated for errors introduced by adversarial perturbations.

Table 7 presents the distribution of prediction outcomes under adversarial conditions. Most samples (73.04%) were correctly classified by both models, indicating that many adversarial perturbations were insufficient to alter the final prediction. More importantly, 129 samples (9.17%) correspond to rescue cases, where the text-only model produced an incorrect prediction while the multimodal model remained correct. These cases provide direct evidence that visual information can compensate for degradation introduced in the textual modality. In contrast, multimodal failures account for 9.74% of the samples, indicating that visual information does not always improve prediction outcomes. Nevertheless, the presence of a substantial number of rescue cases supports the higher spam recall observed in the multimodal setting and reinforces the hypothesis that visual information contributes positively to robustness under adversarial conditions.

An analysis of the IoU values across case categories reveals that multimodal failures exhibit the lowest average overlap (0.0577), whereas correctly classified samples show higher overlap values (0.0708). Rescue cases present an intermediate overlap value (0.0669). Although these observations do not establish a causal

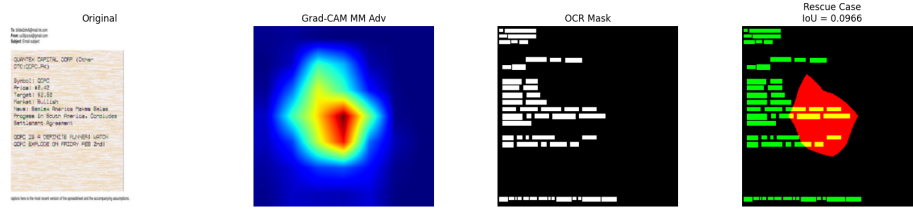


Fig. 2. Grad-CAM and OCR overlap analysis for a rescue case under textual adversarial attack.



Fig. 3. Grad-CAM and OCR overlap analysis for a both-fail case under textual adversarial attack.

relationship, they suggest that stronger alignment between Grad-CAM activations and OCR-detected regions may be associated with more reliable multimodal decisions. To complement this quantitative analysis, representative Grad-CAM visualizations were inspected for rescue and failure cases, providing a qualitative perspective on the model's behavior under adversarial perturbations.

Figure 2 presents a rescue case where the textual branch is successfully attacked, but the multimodal model preserves correct spam classification by maintaining attention over promotional content, suspicious layouts, and highlighted textual areas. The stronger OCR overlap ($\text{IoU} = 0.0966$) indicates better alignment between visual attention and relevant document regions, allowing the visual modality to compensate for textual degradation.

In contrast, Figure 3 illustrates a both-fail case, where neither modality provides sufficient evidence for correct classification. In this example, Grad-CAM exhibits weaker correspondence with relevant document regions, and the lower OCR overlap ($\text{IoU} = 0.0239$) reflects reduced alignment between visual attention and the information contained in the email. Although absolute IoU values are relatively low, this behavior is expected because Grad-CAM provides coarse localization rather than precise segmentation masks. Therefore, IoU should be interpreted as a comparative measure of attention alignment rather than an exact localization metric. The contrast between rescue and failure cases nevertheless helps explain why multimodal systems preserve higher spam recall under adversarial conditions while still exhibiting occasional failures.

Overall, the results consistently indicate that multimodal spam detection provides greater robustness against textual adversarial attacks than text-only classification. While PWWS significantly degrades the performance of the textual

branch, the integration of visual information helps preserve spam recall and reduces critical false negatives under attack. The statistical analysis, confusion matrix evaluation, rescue-case analysis, and Grad-CAM visualizations collectively support the conclusion that visual information contributes to maintaining classification performance when the textual modality is compromised. These findings suggest that multimodal architectures not only improve performance under clean conditions but also provide a more resilient framework for practical spam detection when adversaries attempt to manipulate the textual modality.

5 Conclusions

This work investigated the robustness of multimodal spam detection systems against textual adversarial attacks by comparing text-only and multimodal architectures within a unified experimental framework. Unlike conventional evaluations performed exclusively under clean conditions, this study examined whether visual information can help preserve spam detection performance when the textual modality is intentionally manipulated through PWWS adversarial perturbations.

The experimental results demonstrate that textual adversarial attacks substantially degrade the performance of text-only classification models. In contrast, the multimodal architecture exhibited a considerably smaller reduction in spam recall and a lower number of false negatives, preserving its ability to correctly identify malicious emails under attack. Statistical significance analysis confirmed that both architectures were affected by adversarial perturbations; however, the impact was notably smaller in the multimodal setting. This behavior was further supported by the confusion matrix analysis, which showed a consistent reduction in critical classification errors when visual information was incorporated.

Additional insight was obtained through the explainability analysis. The OCR-GradCAM overlap remained relatively stable under adversarial conditions, while the rescue-case analysis revealed multiple instances in which the multimodal model maintained correct predictions despite failures in the textual branch. Furthermore, representative Grad-CAM visualizations showed that successful predictions tended to preserve attention over visually relevant spam-related regions, whereas failure cases exhibited weaker correspondence between attention maps and document content.

Taken together, these findings indicate that visual information contributes not only to improving classification performance but also to preserving detection capability when the textual modality is compromised. Consequently, evaluating spam detection systems solely under clean conditions may overestimate their practical robustness. The results support multimodal architectures as a more resilient alternative for real-world spam filtering scenarios in which adversaries actively manipulate textual content.

Future work will investigate more challenging multimodal adversarial scenarios, including joint text-image attacks designed to simultaneously

manipulate both modalities. Such experiments would help determine whether the robustness observed in this study remains when visual information is also targeted. Another promising direction involves causal and counterfactual analyses aimed at understanding which visual characteristics are responsible for the rescue behavior observed in multimodal predictions. Additionally, future research will explore multimodal defense strategies, including adversarially robust training and cross-modal consistency mechanisms, to further improve the security and reliability of practical spam detection systems.

References

1. Asliyukse, H., Tonkal, O., Kocaoglu, R.: A comparative evaluation of a multimodal approach for spam email classification using distilbert and structural features. *Electronics* 14(19), 3855 (2025)
2. Hnini, G., Riffi, J., Mahraz, M.A., Yahyaouy, A., Tairi, H.: Mmpc-rf: a deep multimodal feature-level fusion architecture for hybrid spam e-mail detection. *Applied Sciences* 11(24), 11968 (2021)
3. 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)
4. Jáñez-Martino, F., Fidalgo, E., Alaiz-Rodríguez, R., Carofilis, A., Martínez-Mendoza, A.: Comparative analysis of natural language processing models for malware spam email identification. In: *Proceedings of the First International Conference on Natural Language Processing and Artificial Intelligence for Cyber Security*. pp. 59–63 (2024)
5. Jin, D., Jin, Z., Zhou, J.T., Szolovits, P.: Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In: *Proceedings of the AAAI conference on artificial intelligence*. vol. 34, pp. 8018–8025 (2020)
6. Li, J., Ji, S., Du, T., Li, B., Wang, T.: Textbugger: Generating adversarial text against real-world applications. *arXiv preprint arXiv:1812.05271* (2018)
7. Ren, S., Deng, Y., He, K., Che, W.: Generating natural language adversarial examples through probability weighted word saliency. In: *Proceedings of the 57th annual meeting of the association for computational linguistics*. pp. 1085–1097 (2019)
8. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: visual explanations from deep networks via gradient-based localization. *International journal of computer vision* 128(2), 336–359 (2020)
9. Wang, P., Zhang, S., Xu, Y., He, X., Xu, W.: Textjoshier: A transfer-based black-box attack method against text classifiers. *Information Sciences* p. 122888 (2025)
10. Wu, B., Zhu, Z., Liu, L., Liu, Q., He, Z., Lyu, S.: Attacks in adversarial machine learning: A systematic survey from the lifecycle perspective. *International Journal of Computer Vision* 134(5), 197 (2026)
11. Zhang, Z., Deng, Z., Zhang, W., Bu, L.: Mmtd: A multilingual and multimodal spam detection model combining text and document images. *Applied Sciences* 13(21), 11783 (2023)

Audio Noise Reduction Based on Spectral Attenuation

Abril Donahi Castro-Sánchez, Gengis Cruz-Coronel

Benemérita Universidad Autónoma de Puebla,
Facultad de Ciencias de la Computación,
Mexico

{cs226570036,cc226570037}@alm.buap.mx

Abstract. This paper presents an experimental characterization of an audio denoising method based on the Short-Time Fourier Transform (STFT) and spectral attenuation. The study analyzes how processing parameters, noise-profile estimation, input signal-to-noise ratio (SNR), and noise characteristics influence the performance of the method. A controlled synthetic signal was first used to study the influence of the attenuation factor, spectral floor, STFT window length, and frame overlap. The analysis identified a favorable attenuation region around $\alpha = 1.7$, whereas low values of the spectral-floor parameter showed comparatively small influence on SNR improvement. The selected configuration was subsequently evaluated for input SNR levels from 0 to 15 dB using both known and estimated noise profiles. A complementary experiment with a speech signal contaminated by white, cafeteria, and traffic noise was also performed. With an estimated noise profile and an input SNR of 10 dB, the mean SNR improvements were 8.15 ± 0.18 dB for white noise, 4.12 ± 1.27 dB for cafeteria noise, and 8.66 ± 3.04 dB for traffic noise. The results show that the performance of spectral attenuation is influenced by both parameter selection and noise characteristics.

Keywords: Audio denoising, short-time Fourier transform, spectral attenuation, noise estimation, signal-to-noise ratio.

Reducción de ruido de audio basada en atenuación espectral

Resumen. Este artículo presenta una caracterización experimental de un método de reducción de ruido en audio basado en la Transformada de Fourier de Tiempo Reducido (STFT) y la atenuación espectral. El estudio analiza cómo los parámetros de procesamiento, la estimación del perfil de ruido, la relación señal-ruido (SNR) de entrada y las características del ruido influyen en el rendimiento del método. Inicialmente se utilizó una señal sintética controlada para estudiar la influencia del factor de atenuación, el nivel de suelo espectral (*spectral floor*), la longitud de la ventana de la STFT y el solapamiento de tramas. El análisis

identificó una región de atenuación favorable en torno a $\alpha = 1,7$, mientras que los valores bajos del parámetro de suelo espectral mostraron una influencia comparativamente menor en la mejora de la SNR. Posteriormente, se evaluó la configuración seleccionada para niveles de SNR de entrada de 0 a 15 dB, utilizando tanto perfiles de ruido conocidos como estimados. También se llevó a cabo un experimento complementario con una señal de voz contaminada por ruido blanco, ruido de cafetería y ruido de tráfico. Con un perfil de ruido estimado y una SNR de entrada de 10 dB, las mejoras medias de la SNR fueron de $8,15 \pm 0,18$ dB para ruido blanco, $4,12 \pm 1,27$ dB para ruido de cafetería y $8,66 \pm 3,04$ dB para ruido de tráfico. Los resultados demuestran que el rendimiento de la atenuación espectral está influido tanto por la selección de parámetros como por las características del ruido.

Palabras clave: Reducción de ruido en audio, transformada de Fourier de tiempo reducido, atenuación espectral, estimación de ruido, relación señal-ruido.

1 Introduction

Audio signals acquired in practical environments are frequently affected by unwanted acoustic components originating from recording conditions, transmission systems, surrounding sources, or sensor noise. Such contamination can degrade perceived quality and may also affect subsequent signal analysis when relevant information must be extracted from temporal or spectral representations. Consequently, noise reduction is an important preprocessing task in audio signal processing [5, 8].

Audio signals commonly exhibit time-varying spectral characteristics, making a purely temporal or global frequency-domain representation insufficient for describing their local behavior. The Short-Time Fourier Transform (STFT) addresses this limitation by analyzing short signal segments and providing a time-frequency representation of their spectral content [1, 7]. The resulting representation can be used not only for visualization through spectrograms, but also for modifying spectral components associated with unwanted noise.

Among classical approaches to noise reduction, spectral subtraction and related attenuation methods are particularly attractive because of their computational simplicity and interpretability. These methods estimate the spectral characteristics of the noise and reduce their contribution to the magnitude spectrum of the contaminated signal, while the phase of the observed signal can be preserved for reconstruction [3]. Despite this relatively simple formulation, their practical behavior depends on several design choices, including the quality of the noise estimate, the amount of spectral attenuation, the spectral floor, and the parameters used to construct the STFT.

This dependence creates an important experimental problem. Reporting a single set of processing parameters and a single noise condition does not indicate whether the observed improvement is robust to parameter changes, different

contamination levels, or changes in the temporal and spectral characteristics of the noise. In particular, a configuration that performs well for stationary white noise may behave differently for environmental noise whose spectrum changes over time or overlaps with components of the signal of interest.

Therefore, the objective of this work is to experimentally characterize an STFT-based spectral attenuation method under controlled and progressively more realistic conditions. The study investigates two main questions: how sensitive is the achieved noise reduction to the principal processing parameters, and how does its performance change with the contamination level, noise-profile acquisition strategy, and type of acoustic noise.

The experimental study is organized in three stages. First, a controlled synthetic signal is used to analyze the influence of the attenuation factor α , the spectral-floor parameter β , STFT window length, frame overlap, and the interaction between the attenuation parameters. Second, the selected configuration is evaluated at input SNR levels of 0, 5, 10, and 15 dB using multiple generated noise signals and comparing known and estimated noise profiles. Finally, a recorded speech signal is contaminated at a controlled input SNR with white, cafeteria, and traffic noise to evaluate the method under different acoustic characteristics. The main contribution of this work is a reproducible experimental analysis aimed at identifying the sensitivity of the main processing parameters, evaluating the effect of noise-profile estimation, and characterizing the performance of the method under different contamination levels and noise conditions.

The remainder of this paper is organized as follows. Section 2 reviews related work on time–frequency audio analysis and spectral noise reduction. Section 3 describes the implemented denoising methodology. Section 4 presents the experimental setup and evaluation protocol. Section 5 reports and discusses the experimental results and their validity. Finally, Section 6 presents the conclusions and future research directions.

2 Related Work

Noise reduction based on spectral processing has been studied since the earliest work on speech enhancement in the frequency domain. One of the classical methods is spectral subtraction, proposed by Boll [3], in which the noise spectrum is estimated and its contribution to the spectrum of the noisy signal is reduced.

The signal is then reconstructed using the modified magnitude and the phase information from the observed signal. Although this procedure can reduce broadband noise with a relatively simple implementation, it can also produce residual components and perceptible distortions when the noise estimate or the degree of attenuation is inadequate.

To address this limitation, Berouti et al. [2] studied a variant of spectral subtraction in which the noise is overestimated and a minimum level is imposed on the spectral components after subtraction. These two elements correspond to the parameters denoted by α and β in this work. The parameter α controls the

amount of noise subtracted from the observed spectrum, whereas β establishes a spectral floor that prevents the magnitude from decreasing without bound. This formulation makes it possible to adjust the trade-off between residual noise reduction and alteration of the signal of interest.

In addition to the attenuation parameters, estimation of the noise profile is an important part of spectral processing. Martin [6] proposed a method for estimating the power spectral density of nonstationary noise by tracking spectral minima, without relying directly on a voice activity detector. Cohen [4] subsequently introduced an estimation procedure based on minima-controlled recursive averaging, designed to track changes in the noise under nonstationary conditions and at low signal-to-noise ratios. These studies show that the quality of the noise estimate directly affects the performance of the subsequent noise reduction stages.

The time–frequency representation also requires several analysis choices. The STFT requires the selection of a window function, its length, and the hop size between frames [1, 7]. These parameters determine the trade-off between temporal and frequency resolution and may therefore affect the representation used during noise estimation and attenuation.

In view of these previous studies, this work focuses on the experimental characterization of an STFT-based spectral attenuation scheme. The analysis examines the influence of the main processing parameters, the input SNR, the method used to obtain the noise profile, and the characteristics of the additive noise. Controlled conditions with a synthetic signal are considered first, followed by a speech signal corrupted by white and environmental noise.

3 Methodology

The evaluated method follows a sequence of processing steps in the time–frequency domain. The input signal is preprocessed to remove its DC component and maintain a common scale among the clean reference, the noisy signal, and the noise. The Short-Time Fourier Transform (STFT) is then computed, a spectral noise profile is obtained, and the magnitude of the noisy signal is modified through spectral attenuation. The phase of the observed signal is retained, and the processed signal is recovered using the inverse STFT. Finally, the result is compared with the clean signal using the signal-to-noise ratio (SNR). Figure 1 summarizes the main processing stages.

3.1 Preprocessing

Before the spectral analysis, two preprocessing operations were applied: DC component removal and normalization using a common scale. The DC component is removed by centering each signal around zero:

$$x_{\text{dc}}[n] = x[n] - \mu_x, \quad (1)$$

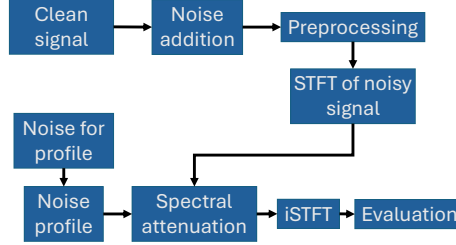


Fig. 1. General workflow of the evaluated audio denoising procedure.

where μ_x is the mean value of the signal:

$$\mu_x = \frac{1}{N} \sum_{n=0}^{N-1} x[n]. \quad (2)$$

The clean signal, the noisy signal, and the noise were then divided by the same scaling factor, computed as the maximum absolute value of the noisy signal after removing its DC component:

$$x_{\text{proc}}[n] = \frac{x_{\text{dc}}[n]}{\max(|x_{\text{mix,dc}}[n]|)}. \quad (3)$$

Using a common scaling factor preserves the relative amplitudes of the signals within the same experimental condition [5].

3.2 Time–Frequency Representation Using the STFT

The preprocessed signal was represented in the time–frequency domain using the short-time Fourier transform. For a discrete-time signal $x[n]$, the STFT can be expressed as:

$$X(k, m) = \sum_{n=0}^{N_w-1} x[n + mH] w[n] e^{-j \frac{2\pi kn}{N_w}}. \quad (4)$$

where $w[n]$ is the window function, N_w is its length, H is the hop size between frames, k is the frequency index, and m is the time index [1, 7, 8].

A Hann window was used in the implementation. The window length and frame overlap were later examined as part of the sensitivity analysis because both parameters affect the temporal and frequency resolution of the representation.

3.3 Noise Profile Estimation

Two approaches were considered for obtaining the spectral noise profile. The first, referred to as the known profile, directly uses the noise employed to corrupt the

signal. The profile is calculated by averaging the magnitude of its STFT across the time frames:

$$N_p(k) = \frac{1}{M} \sum_{m=0}^{M-1} |N(k, m)|. \quad (5)$$

where M is the number of frames available from the noise used in the evaluation.

The second condition uses a profile estimated from a 0.30 s noise-only segment that is not included in the 2 s interval used to evaluate the noisy signal. The profile is calculated in the same manner:

$$\hat{N} * p(k) = \frac{1}{M_0} \sum_{*m=0}^{M_0-1} |N_{\text{est}}(k, m)|. \quad (6)$$

where M_0 is the number of frames obtained from the segment used for estimation.

This separation allows a condition in which the noise used during evaluation is known to be compared with another in which only an independent sample from the same acoustic environment is available.

3.4 Spectral Attenuation

Noise reduction was performed by modifying the magnitude of the STFT of the noisy signal. Let $|X(k, m)|$ denote its magnitude and $N_p(k)$ the noise profile used for the corresponding condition. The processed magnitude is obtained as:

$$|\hat{S}(k, m)| = \max(|X(k, m)| - \alpha N_p(k), \beta N_p(k)). \quad (7)$$

The parameter α controls the amount of the noise profile subtracted from the observed magnitude. Larger values produce stronger attenuation, but they can also remove components of the signal of interest. The parameter β sets a spectral floor that limits how far each component can be reduced [3, 2].

After modifying the magnitude, the phase of the noisy signal is retained to form a complex representation:

$$\hat{S}(k, m) = |\hat{S}(k, m)| e^{j \angle X(k, m)}. \quad (8)$$

3.5 Signal Reconstruction

The processed signal was recovered in the time domain using the inverse short-time Fourier transform (iSTFT). To allow a direct comparison among the signals, the reconstructed output was adjusted to match the length of the clean reference signal.

This procedure retains the frame-based analysis structure used in the STFT and produces a time-domain signal from the modified spectrum [1].

3.6 Evaluation Metric

Quantitative performance was evaluated using the signal-to-noise ratio (SNR). For a clean reference signal $x_{\text{ref}}[n]$ and a signal under evaluation $x_{\text{test}}[n]$, the SNR was calculated as:

$$\text{SNR} = 10 \log_{10} \left(\frac{\sum_{n=0}^{N-1} x_{\text{ref}}^2[n]}{\sum_{n=0}^{N-1} (x_{\text{ref}}[n] - x_{\text{test}}[n])^2} \right). \quad (9)$$

In addition to the output SNR, the improvement produced by the processing was calculated as:

$$\Delta \text{SNR} = \text{SNR} * \text{out} - \text{SNR} * \text{in}. \quad (10)$$

Thus, the comparison considers both the final SNR and the level of contamination present before processing.

The implementation was developed in Python within a Jupyter Notebook environment. General-purpose libraries were used for numerical operations, STFT and inverse-STFT computation, audio loading and resampling, and visualization. The specific noise-reduction logic, including preprocessing, noise-profile construction, spectral attenuation, signal reconstruction, and SNR-based evaluation, was explicitly implemented for this study. No noise-reduction function or package was used to directly perform the complete denoising procedure.

4 Experimental Setup

The evaluation consisted of three experiments. The first examined the sensitivity of the method to changes in the main processing parameters. The second analyzed the performance of the selected configuration at different contamination levels and compared the use of a known noise profile with an estimated profile. Finally, the third experiment used a speech signal corrupted by different types of noise to evaluate the performance of the method under a wider range of acoustic conditions.

4.1 Synthetic Signal and Controlled Noise Generation

A synthetic signal composed of three sinusoidal components was used in the first two experiments:

$$x_{\text{clean}}(t) = 0.6 \sin(2\pi 440t) + 0.3 \sin(2\pi 880t) + 0.2 \sin(2\pi 1760t). \quad (11)$$

The sampling rate was $f_s = 16000$ Hz, and each evaluation signal had a duration of 2 s, corresponding to 32000 samples.

To prevent the noise level from depending on an arbitrary amplitude value, the generated white noise was scaled to a predefined input SNR. For a clean signal with power P_x and a noise signal with power P_n , the scaling factor was calculated as:

$$g = \sqrt{\frac{P_x}{P_n \cdot 10^{\text{SNR}_{\text{obj}}/10}}}. \quad (12)$$

The noisy signal was then obtained as:

$$x_{\text{mix}}[n] = x_{\text{clean}}[n] + g, n[n]. \quad (13)$$

This procedure allowed the experimental conditions to be compared using known contamination levels.

4.2 Parameter Sensitivity Analysis

The first experiment examined the influence of the main parameters of the method before selecting a configuration for the subsequent tests. All evaluations in this stage used an input SNR of 10 dB and the known noise profile so that the comparison focused only on the processing parameters.

The attenuation factor α was first evaluated over a broad range from 0.2 to 3.0, in increments of 0.2. After identifying the best-performing region, a finer evaluation was conducted between 1.5 and 2.1, in increments of 0.05.

The spectral-floor parameter β was then studied using values between 0 and 0.2. The interaction between values of α near the best-performing region and low values of β was also evaluated to determine whether the effects of the two parameters were independent or whether performance was particularly sensitive to a specific combination.

Finally, STFT window lengths of 256, 512, 1024, 2048, and 4096 samples were analyzed, together with overlap values of 25%, 50%, and 75%. A Hann window was used in all cases.

4.3 Evaluation at Different SNR Levels

After the sensitivity analysis, a common configuration was selected for the remaining tests: $\alpha = 1.7$, $\beta = 0.02$, a Hann window of 1024 samples, and an overlap of 512 samples, equivalent to 50%.

Using this configuration, input SNR levels of 0, 5, 10, and 15 dB were evaluated. Ten trials were conducted at each level, each using a white-noise signal generated with a random seed.

Each trial was evaluated under two conditions. In the first, a known profile was obtained directly from the same noise used to corrupt the signal. In the second, an independent 0.30 s segment was generated with the same statistical characteristics and scaling level, and this segment was used to estimate the noise profile.

The output SNR and the improvement ΔSNR were calculated for each condition. The results of the ten trials were summarized using the mean and sample standard deviation.

4.4 Evaluation Using a Speech Signal and Different Types of Noise

To complement the evaluation with a real signal, a speech recording originally sampled at 48 kHz was used. The signal was resampled to 16 kHz, and a 2 s segment containing continuous speech was selected.

The same signal was corrupted by three types of noise: artificially generated white noise, cafeteria noise, and traffic noise. The two environmental noises were obtained from the DEMAND database (Diverse Environments Multichannel Acoustic Noise Database), which contains environmental noise recordings from different indoor and outdoor settings [9].

An input SNR of 10 dB was used in all three cases, and the previously selected parameters were retained. For white noise, ten trials were conducted using signals generated with different random seeds. For cafeteria and traffic noise, ten nonoverlapping time segments from each recording were evaluated.

For each environmental noise segment, 0.30 s was used to estimate the profile, and the following 2 s was used to corrupt the speech signal. Thus, the segment used to construct the profile did not overlap with the interval used to calculate the SNR.

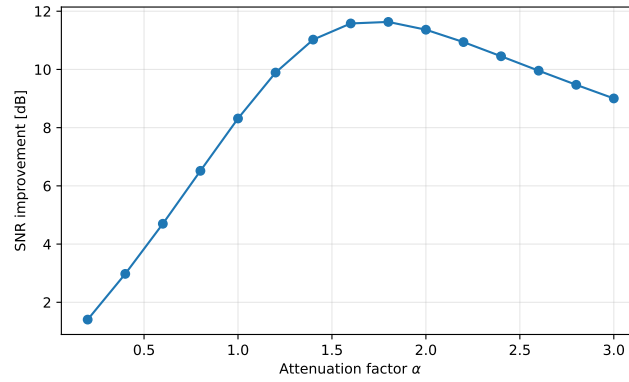


Fig. 2. SNR improvement as a function of the attenuation factor α .

5 Results and Discussion

The results are presented in the same order as the three experiments described in Section 4. The sensitivity of the attenuation and STFT parameters is examined first. The performance of the selected configuration is then studied at different noise levels and using different approaches to obtain the noise profile. Finally, the same configuration is evaluated using a speech signal and different types of noise.

5.1 Parameter Sensitivity

The first experiment showed that the parameters examined do not affect the performance of the method equally. The most pronounced variation occurred when the attenuation factor α was changed. At an input SNR of 10 dB, the improvement

increased steadily from low values of α and reached a best-performing region around 1.6–1.8. Beyond this region, further increases led to a gradual decrease in the output SNR.

In the initial evaluation, $\alpha = 1.0$ produced an improvement of 8.31 dB, whereas values between 1.6 and 1.8 yielded improvements above 11.5 dB. The finer evaluation conducted between 1.5 and 2.1 produced the highest improvement at $\alpha = 1.70$, with a value of 11.66 dB. The trend obtained over the complete evaluation of this parameter is shown in Figure 2.

After identifying the best-performing region for α , the value was fixed at $\alpha = 1.70$ to examine the influence of the spectral floor β . Unlike α , this parameter showed lower sensitivity within the range of small values. For $\alpha = 1.70$, the improvement was 11.69 dB with $\beta = 0$, 11.68 dB with $\beta = 0.01$, and 11.66 dB with $\beta = 0.02$. The difference between $\beta = 0$ and $\beta = 0.02$ was only 0.034 dB. As β increased beyond this range, the improvement gradually decreased.

Although the two parameters were initially analyzed separately, their combined behavior was also evaluated to verify that the selection of α did not depend on a single value of β . For all the low values of β tested, the best value of α within the analyzed range remained 1.70. Therefore, the selection of α was not tied to an isolated combination with a particular value of β .

After examining the spectral attenuation parameters, the influence of the STFT representation was studied. Window length also produced marked changes. For the synthetic signal used in the experiment, the improvement increased from 7.13 dB with a 256-sample window to 14.33 dB with a 4096-sample window. This behavior is consistent with the characteristics of the test signal, which consists of stationary sinusoidal components for which a longer window provides a more detailed frequency representation.

However, the 4096-sample window was not automatically selected as the best option for the subsequent tests. At a sampling rate of 16 kHz, it corresponds to a duration of 256 ms, which considerably reduces temporal resolution and may be unsuitable for time-varying signals such as speech. For this reason, an intermediate window length of 1024 samples, equivalent to 64 ms, was retained for the subsequent evaluations.

After selecting an intermediate window length of 1024 samples, this value was held constant while the effect of frame overlap was examined. Overlap values of 25%, 50%, and 75% produced improvements of 9.30, 11.66, and 11.97 dB, respectively. Although 75% yielded the highest value, the difference relative to 50% was approximately 0.31 dB. A 50% overlap was therefore selected, reducing the number of frames without a substantial loss in performance.

Based on these results, the configuration used in the subsequent experiments was $\alpha = 1.70$, $\beta = 0.02$, a 1024-sample Hann window, and an overlap of 512 samples. This selection is not regarded as a generally optimal configuration, but rather as a working configuration obtained under the controlled conditions of the first experiment.

Table 1. SNR improvement at different noise levels.

Input SNR [dB]	Known profile [dB]	Estimated profile [dB]
0	12.51 ± 0.24	11.65 ± 0.23
5	12.04 ± 0.24	11.28 ± 0.23
10	11.51 ± 0.23	10.87 ± 0.22
15	11.00 ± 0.22	10.46 ± 0.21

Table 2. SNR improvement for speech with different types of noise.

Noise type	Known profile [dB]	Estimated profile [dB]
White	8.28 ± 0.10	8.15 ± 0.18
Cafeteria	4.95 ± 0.83	4.12 ± 1.27
Traffic	8.93 ± 2.70	8.66 ± 3.04

5.2 Effect of Noise Level and Noise Profile

Once the processing configuration had been fixed, the effect of the input SNR was examined. Table 1 shows the mean improvement obtained over the ten trials conducted at each noise level.

Both types of profile produced a positive improvement at every SNR level tested. The improvement was greater when the initial noise level was higher and gradually decreased as the input SNR increased. For the known profile, the improvement decreased from 12.51 dB at an initial SNR of 0 dB to 11.00 dB at an initial SNR of 15 dB.

The estimated profile produced slightly lower values than the known profile under all conditions. The mean difference between them was 0.86 dB at an input SNR of 0 dB and decreased to 0.54 dB at an input SNR of 15 dB. The standard deviations remained close to 0.2 dB, indicating consistent behavior across the different white-noise signals generated.

5.3 Evaluation Using Speech and Different Types of Noise

The results show that the type of noise affects both the mean improvement and the variability of the method. White noise produced a high and stable improvement, with a standard deviation below 0.2 dB for the estimated profile. Cafeteria noise produced the lowest mean improvement, at 4.12 ± 1.27 dB, whereas traffic noise produced the highest mean improvement, at 8.66 ± 3.04 dB, but also showed the greatest variation across segments. Table 2 summarizes the SNR improvement obtained for the different types of noise.

Therefore, the higher mean observed for traffic noise does not imply that this condition is always more favorable. The standard deviation shows that the result depends strongly on the time segment analyzed. In contrast, white noise produced much more uniform behavior.

The lower performance obtained with cafeteria noise may be related to its more complex acoustic content, which includes speech components and

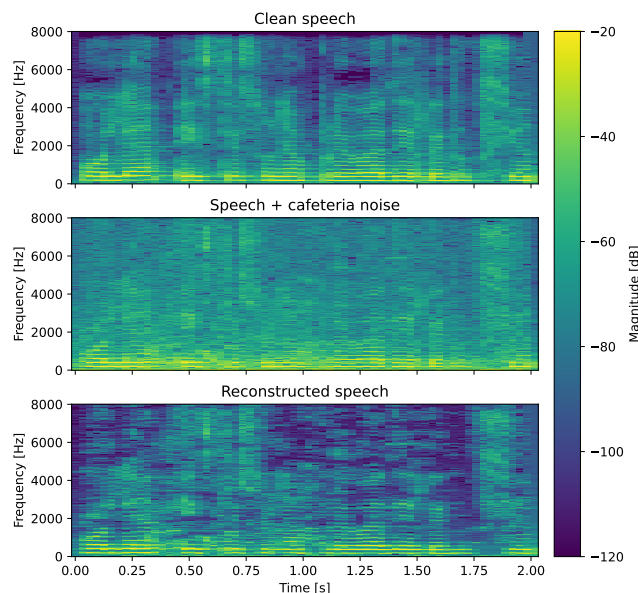


Fig. 3. Spectrogram comparison for clean speech, speech contaminated with cafeteria noise, and reconstructed speech using the estimated noise profile.

other sources that may overlap spectrally with the signal of interest. During listening, a clear reduction in the overall environmental noise was observed, although some background speech components remained after processing. This observation was used only as qualitative support and does not constitute a formal perceptual evaluation. Figure 3 shows an example for cafeteria noise. The comparison illustrates the spectral distribution of the clean speech, the increase in background content after contamination, and the reduction obtained after applying attenuation using the estimated profile.

5.4 Experimental Validity and Scope of the Results

The experimental design sought to maintain comparable conditions across the trials. In the synthetic-signal experiments, the same clean signal was used, and the noise power was explicitly adjusted to obtain the input SNR defined for each case. Therefore, the changes observed during the sensitivity analysis can be attributed mainly to changes in the processing parameters rather than to differences in the initial noise level.

After the sensitivity analysis, the same processing configuration was retained for the remaining tests. This decision allowed the method to be evaluated at different noise levels and with different types of noise without tuning the parameters to favor any particular condition. In addition, ten signals generated with different random seeds were used in the white-noise tests, whereas ten

nonoverlapping time segments from each recording were analyzed for the environmental noises. The results were reported using the mean and standard deviation to account for variation across the trials rather than a single case.

For the environmental noises, the different segments provide information about temporal variation within the same recording. However, these segments were taken from only one cafeteria recording and one traffic recording. The results therefore represent the acoustic conditions examined rather than all possible environments within these categories. Similarly, the speech evaluation was conducted using a single speaker and a two-second segment.

The configuration used in the speech and environmental-noise tests was initially selected using a synthetic signal and white noise. Retaining this configuration without further adjustment made it possible to examine its behavior outside the condition used for its selection, although this does not imply that the selected values are optimal for every signal or acoustic environment. In particular, the favorable behavior observed with longer windows during the sensitivity analysis is related to the stationary nature of the synthetic signal.

6 Conclusions

This work presented an experimental evaluation of an STFT-based spectral attenuation scheme for noise reduction. The study examined the effects of the main processing parameters, the method used to obtain the noise profile, the initial noise level, and the characteristics of the noise. Controlled experiments using a synthetic signal were combined with a subsequent evaluation using a speech signal.

The sensitivity analysis showed that the parameters examined did not affect the results in the same way. The attenuation factor α had a clear effect on SNR improvement, with a favorable range around 1.6–1.8 and the highest tested improvement obtained at $\alpha = 1.70$. In contrast, β produced only small variations within the range of low values. Window length also had a marked effect on performance, although longer windows benefited from the stationary nature of the synthetic signal. An intermediate configuration was therefore selected for the subsequent tests instead of simply using the value that produced the highest SNR under the initial condition.

The evaluation at different noise levels showed positive improvements at all the SNR levels examined. A profile estimated from an independent noise segment produced slightly lower results than the known profile but remained consistent across the trials. This indicates that the method retains its noise-reduction capability even when the noise used to corrupt the signal is not directly available for constructing the spectral profile.

The speech test showed that performance also depends on the characteristics of the noise. White noise produced a high improvement with little variation across trials, whereas traffic noise achieved a similar or higher mean improvement but with considerably greater dispersion. Cafeteria noise produced the lowest

improvement, illustrating the difficulty faced by the method when the noise contains acoustic components that overlap with the speech signal.

Overall, these results characterize the behavior of the method under different processing and noise conditions. The main contribution of the study is the systematic evaluation of these factors through controlled conditions, repeated trials, and subsequent validation using speech and environmental noises.

Future work could extend the evaluation to a larger number of speakers and independent environmental recordings and include complementary metrics related to perceptual quality and speech intelligibility. Automatic adaptation of the attenuation parameters to the characteristics of the observed noise could also be investigated.

References

1. Allen, J.B., Rabiner, L.R.: A unified approach to short-time Fourier analysis and synthesis. *Proceedings of the IEEE* 65(11), 1558–1564 (1977)
2. Berouti, M., Schwartz, R., Makhoul, J.: Enhancement of speech corrupted by acoustic noise. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing*. pp. 208–211 (1979)
3. Boll, S.F.: Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 27(2), 113–120 (1979)
4. Cohen, I.: Noise spectrum estimation in adverse environments: Improved minima controlled recursive averaging. *IEEE Transactions on Speech and Audio Processing* 11(5), 466–475 (2003)
5. Lerch, A.: *An Introduction to Audio Content Analysis: Applications in Signal Processing and Music Informatics*. Wiley-IEEE Press (2012)
6. Martin, R.: Noise power spectral density estimation based on optimal smoothing and minimum statistics. *IEEE Transactions on Speech and Audio Processing* 9(5), 504–512 (2001)
7. Müller, M.: *Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications*. Springer (2015)
8. Oppenheim, A.V., Schaffer, R.W.: *Discrete-Time Signal Processing*. Pearson, 3rd edn. (2010)
9. Thiemann, J., Ito, N., Vincent, E.: The diverse environments multi-channel acoustic noise database (DEMAND): A database of multichannel environmental noise recordings. *Proceedings of Meetings on Acoustics* 19, 035081 (2013)

Machine Translation for Northeastern Puebla Sierra Nahuatl: Evaluating Preprocessing Scenarios and Model Convergence

Javier Tolentino-Ramírez

Benemérita Universidad Autónoma de Puebla,
Facultad de Ciencias de la Computación, Puebla,
Mexico

tr226570041@alm.buap.mx

Abstract. This study investigates the impact of text preprocessing and orthographic representation on the neural machine translation of the Northeastern Puebla Sierra Nahuatl variant, an agglutinating language. Using the SLR92 corpus extracted from ELAN files, we evaluate three preprocessing scenarios targeting the vowel-length colon marker (":"): Keeping, Removing, and Substituting. Fine-tuning the open-source NLLB-200-3.3B model across all three scenarios reveals consistent training dynamics across three phases: an explosive early learning phase (Epochs 1–5), a high-accuracy convergence phase reaching peak validation BLEU scores of up to 26.64 (Keeping), 25.42 (Substituting), and 25.26 (Removing), and a stable performance plateau. Final evaluation on a locked test set of 3,865 sentences demonstrates solid structural consistency, with Substituting achieving the top BLEU score of 18.59 (WER 67.38%) and Keeping yielding the lowest Word Error Rate at 66.85% (BLEU 18.44).

Keywords: Neural machine translation, Nahuatl language, morphological segmentation, granularity density.

Traducción automática para el náhuatl de la Sierra Nororiental de Puebla: evaluación de escenarios de preprocesamiento y convergencia del modelo

Resumen. Este estudio investiga el impacto del preprocesamiento de texto y la representación ortográfica en la traducción automática neuronal de la variante del náhuatl de la Sierra Norte de Puebla, una lengua aglutinante. Utilizando el corpus SLR92 extraído de archivos ELAN, evaluamos tres escenarios de preprocesamiento centrados en el marcador de dos puntos (":") que indica la longitud vocálica: mantenerlo, eliminarlo y sustituirlo. El ajuste fino del modelo de código abierto NLLB-200-3.3B en los tres escenarios revela dinámicas de entrenamiento

consistentes a lo largo de tres fases: una fase inicial de aprendizaje explosivo (épocas 1–5), una fase de convergencia de alta precisión —alcanzando puntuaciones máximas de BLEU en validación de hasta 26,64 (mantener), 25,42 (sustituir) y 25,26 (eliminar)— y una meseta de rendimiento estable. La evaluación final en un conjunto de prueba fijo de 3.865 oraciones demuestra una sólida consistencia estructural; la opción de sustitución logra la puntuación BLEU más alta (18.59; WER 67.38%), mientras que la opción de mantener el marcador arroja la tasa de error por palabra (WER) más baja (66.85%; BLEU 18.44).

Palabras clave: Traducción automática neuronal, lengua náhuatl, segmentación morfológica, densidad de granularidad.

1 Introduction

The analysis of indigenous languages is a fundamental pillar for the development of machine translation systems that not only preserve these languages but also guarantee universal access to information. This is critical in areas such as the justice systems of indigenous communities, where accurate linguistic understanding is indispensable to guarantee due process and prevent unjust sentences resulting from misinterpretation.

In Mexico, 364 linguistic variants are spoken, grouped into 68 linguistic groups and 11 language families. In the case of the Nahuatl language, there are at least 14 variants [5].

The original representation of the Nahuatl language is considered a logosyllabic hieroglyphic system [3]. Its use was highly functional for recording names and administrative data; although it was not used extensively for complex sentences like the Mayan script, it possessed the capacity to do so.

Nahuatl is an agglutinating language, which is relevant to understanding its syntax and the construction of lexical meanings. In agglutinating languages, words are formed by joining multiple morphemes to a root, each of which generally maintains a distinct and recognizable meaning or grammatical function [4].

To identify the baseline characteristics of a language, like Nahuatl, it is necessary to perform a preliminary statistical analysis first. This initial step gives important insights into its vocabulary structure, word boundaries, and text density. Understanding these baseline details ensures that subsequent machine learning models can be tailored and fine-tuned with the best configurations to adapt precisely to the morphological complexities of the language.

To effectively exploit these statistical insights, state-of-the-art machine translation architectures capable of handling low-resource and morphologically rich languages are required. In this context, open-source multilingual pretrained systems provide a robust foundation for transfer learning. Among these, the No Language Left Behind (NLLB) model stands out as a powerful multilingual Transformer framework designed to scale translation across massively parallel language directions.

2 NLLB Model

No Language Left Behind (NLLB) is an open-source system designed to translate more than 200 languages based on three pillars [7]:

- Massive Data: It combines advanced web mining (using the LASER3 encoder) with FLORES-200, a high-quality, human-translated dataset used to evaluate thousands of translation directions.
- Mixture of Experts (MoE) Architecture: It utilizes a model that activates only the necessary parts of the neural network for each specific task. This allows handling hundreds of languages efficiently without the information of one interfering negatively with another.
- Training and Safety: It applies progressive learning strategies and toxicity filters to ensure that translations are not only accurate but also safe.

The NLLB-200 model includes the Northern Puebla Nahuatl variant within its training, although it does not specify whether it is the eastern or northeastern version. Retraining (or fine-tuning) the model with the corpus of the northeastern variant will allow observing the model's capabilities in translating this variant into Spanish.

The model used in the present research was NLLB-200 Dense 3.3B, which is a multilingual encoder-decoder Transformer based on the M2M-100 architecture. It comprises 24 encoder and 24 decoder layers, each with 16 attention heads, a hidden dimension of 2,048, and feed-forward networks of dimension 8,192. The model uses a 256,206-token SentencePiece vocabulary with language identifiers and supports sequences of up to 1,024 positional embeddings (trained on inputs of up to 512 tokens). Regularisation is achieved with 0.1 dropout on both hidden states and attention weights, while ReLU is used as the activation function. Unlike the NLLB-200 MoE model, its computational cost scales directly with its parameter count because every layer is executed for each inference.

3 Results in Other Translation Tasks

There are several case studies on the application of machine translation for indigenous languages of Mexico, a summary of these studies is listed in Table 1.

4 Methodology

In this study, the methodology is structured into three main stages. First, the dataset was extracted and cleaned from the raw EUDICO Linguistic Annotator (ELAN) files. Second, the corpus is analyzed using four Natural Language Processing (NLP) techniques: Tokenization, Bag of Words (BoW), Bigrams, and Granularity Density under three scenarios. Third, the scenarios are compared after a separated fine-tuning of the multilingual NLLB model for each one.

Table 1. Comparative analysis of machine translation models for Mexican indigenous languages.

Ref.	Direction	Model	Corpus Size	Configuration	BLEU
[10]	Mazatec → Spanish	mBART50	~9,000 sentences	Epochs: 3, batch	10.45
	Spanish → Mazatec			size: 4, beams: 5,	21.20
	Mixtec → Spanish		~13,000 sentences	learning rate:	15.66
	Spanish → Mixtec			5e-2	16.03
[12]	Spanish → Ayuuk	Transformer	~6,000 sentences	Layers: 6, embedding: 256, batch: 128, epochs: 250	5.56
	Ayuuk → Spanish				3.52
[2]	Spanish → Nahuatl	Transformer	~35,000 sentences	Epochs: 25 to 40, Optimizer: AdamW, Learning rate: 9.12e-04	66.45
[1]	Spanish → Mixtec	Transformer	~1,000 sentences	Layers: 6, heads: 4, embedding: 256, batch size: 128, epochs: 250	15.97
[6]	Purépecha → Spanish	Transformer	~10,000 sentences	Layers: 64–256, epochs: 50–100	15.85

4.1 Data Extraction and Cleaning from ELAN Files

The primary linguistic data consists of annotated ELAN files under the identifier SLR92 [9]. An automated extraction pipeline using `pympi` was developed to align Nahuatl tiers with their corresponding Spanish translations based on temporal overlap.

To ensure data purity, a multi-stage regular expression cleaning process was implemented:

- Text encapsulated within metadata tags (`<sci>`, `<i>`, `<b>`) was identified as non-linguistic noise and removed entirely.
- All characters were normalized using the Unicode Normalization Form C (NFC) to ensure consistent encoding of Nahuatl-specific diacritics and the Spanish *ñ*.

4.2 Corpus Analysis and Metrics

After the pre-processing and text normalization pipeline, the cleaned corpus of the Northeastern Puebla Sierra Nahuatl variant was analyzed. To identify the baseline characteristics of Nahuatl writing, the following techniques were applied:

- Tokenization: This process consists of dividing the content into basic components called tokens (such as words, numbers, and punctuation marks) based on their structure. For this study the colon symbol (":") was explicitly preserved for subsequent comparative evaluation.
- Bag of Words (BoW): This is a classical and compact text representation that aggregates the set of terms appearing within a corpus. Despite being a significant simplification, this method is not free from the complexities of natural language due to the ambiguity of terms out of context, which manifests through synonymy and polysemy.
- Bigrams: The *One Sense per Collocation* principle establishes that the meaning of a word remains constant each time it appears within the same collocation or bigram. This property is fundamental for Word Sense Disambiguation (WSD) tools, which utilize classifiers based on decision lists that implicitly assume this rule.
- Granularity Density: Granularity density was analyzed to understand how character-level markers change the vocabulary structure. Specifically, this analysis measures how the text density, unique vocabulary size, and lexical ambiguity change across different scenarios.

In Nahuatl, a colon (:) typically denotes vowel length, an essential feature that can completely alter a word's meaning. To determine how representing this feature impacts machine translation, it was conducted a statistical comparison across three dataset scenarios: keeping the original colons, removing them entirely, and substituting them with whitespace.

To guarantee the statistical validity of the evaluation, the master dataset $\mathcal{D}$ was partitioned into three disjoint sets using a seeded pseudo-random distribution:

$$\mathcal{D} = \mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{val}} \cup \mathcal{D}_{\text{test}}, \quad (1)$$

where:

- $\mathcal{D}_{\text{train}}$ (70%): Used for weight optimization via backpropagation.
- $\mathcal{D}_{\text{val}}$ (15%): Used for evaluating the accuracy of translation during training.
- $\mathcal{D}_{\text{test}}$ (15%): A "locked" set used exclusively for the final reporting of evaluation metrics, ensuring no data leakage during the training stage.

The Bilingual Evaluation Understudy (BLEU) metric compares tokenized n -grams of generated text against reference human translations. The final output is scaled to a percentage between 0 and 100, where 0 indicates low quality or no overlap with the reference, and close to 100 represents an almost perfect match with the human translation. This value is obtained multiplying a Brevity Penalty (BP) by the geometric mean of n -gram precisions [8]:

$$\text{BLEU} = 100 \cdot \text{BP} \cdot \exp \left(\sum_{n=1}^4 0.25 \cdot \log p_n \right). \quad (2)$$

The Brevity Penalty (BP) penalizes translations that are shorter than the reference texts to prevent high precision through truncation:

$$\text{BP} = \begin{cases} 1 & \text{if } c > r, \\ e^{(1-r/c)} & \text{if } c \leq r, \end{cases} \quad (3)$$

where:

- $c = \sum_{i=1}^M |c_i|$: Total length of generated translations across M sentences after standardized tokenization.
- $r = \sum_{i=1}^M r_i$: Effective reference length across M sentences, where r_i is the reference length closest to candidate length $|c_i|$.

The modified n -gram precision p_n calculates the proportion of matched n -grams across the entire dataset. A clip is applied to avoid overcounting:

$$p_n = \frac{\sum_{i=1}^M \sum_{g \in C_i} \text{count_clip}(g)}{\sum_{i=1}^M \sum_{g \in C_i} \text{count}(g)}, \quad (4)$$

where:

- $\text{count_clip}(g)$: Count of n -gram g in candidate sentence C_i , capped at the maximum frequency of g in any single matching reference sentence.
- $N = 4$: Evaluates unigrams up to 4-grams ($n \in \{1, 2, 3, 4\}$).
- $w_n = 0.25$: Uniform weight assigned to each n -gram order ($w_1 = w_2 = w_3 = w_4 = \frac{1}{4}$).
- Smoothing: Exponential decay smoothing is applied when $p_n = 0$ to avoid taking $\log(0)$.

Word Error Rate (WER) is also a widely used measure for evaluating the accuracy of speech recognition and machine translation systems. Its values generally range from 0 to 1, where a score of 0 represents a perfect match between the reference and the predicted text, while a score of 1 (or higher) indicates that the two texts are entirely dissimilar, sharing little or no overlap, it is calculated as:

$$\text{WER} = \frac{S + D + I}{N}, \quad (5)$$

where, S , D , and I represent substitutions, deletions, and insertions, while N is the total number of words in the reference.

4.3 Model Configuration

The model was configured using the default parameters of NLLB-200-3.3B except for the number of epochs, the evaluation strategy, generation parameters, memory optimization of the available hardware, Learning Rate (LR) and Weight Decay (WD).

The parameters LR and WD were adjusted following the recommendations for adafactor optimizer[11], a summary of model configuration is shown in Table 2.

Table 2. Training hyperparameters and configurations for `Seq2SeqTrainingArguments`.

Hyperparameter	Value	Technical Description
Epoch Control		
<code>num_train_epochs</code>	25	Total number of complete passes over the training corpus.
<code>eval_strategy</code>	"epoch"	Performs model evaluation at the end of each epoch.
Generation Parameters (Metrics)		
<code>predict_with_generate</code>	True	Forces autoregressive decoding to evaluate with BLEU/WER.
Memory Optimization (A100 GPU)		
<code>bf16</code>	True	Enables 16-bit brain floating point precision.
<code>optim</code>	"adafactor"	Optimizer designed for massive models with lower memory footprint.
<code>dataloader_num_workers</code>	4	Parallel threads for data loading and pre-processing.
Anti-Overfitting Strategies		
<code>learning_rate</code>	1e-5 (0.00001)	Initial learning rate for the optimizer.
<code>weight_decay</code>	1e-3 (0.001)	L2 regularization penalty applied to weights.

5 Results

Bag of Words technique was conducted to get the 20 top words for each scenario and 20 top bigrams as well. It was observed that the "Keeping" and "Removing" scenarios share the exact same word boundaries and token distributions because stripping out the vowel-length marker (":") is a simple character-level string replacement that reduces orthographic complexity. In the "Keeping" scenario, the vocabulary is dominated by words like *wa:n* and *yo:n* peaking at around 14,500 occurrences, followed closely by *se:* and the letter *n*, while the top bigram pairs are *yehwa yo:n* (~1,800 occurrences), *wa:n yo:n*, and *a:mo a:mo*.

The "Removing" scenario preserves this entire distribution shape and order, simply normalizing the tokens to *wan* and *yon* without changing any absolute frequency counts. On the other hand, the "Substituting" scenario introduces a radical change in tokenization by implementing morphological segmentation. This aggressive mechanism causes the single-letter token *n* to explode to over 72,000 occurrences, becoming a massive statistical outlier because it acts as a highly frequent relational particle, determiner, or clitic detached from its host words, while other fragments like *wa*, *a*, *yo*, and *ke* take over the top spots. Consequently, the bigram distribution directly reflects this split, causing the

original single words to reappear as the top bigrams *wa n* ($\sim 15,000$) and *yo n* ($\sim 14,000$) matching their original unigram frequencies.

From Figure 1 it is observed that the transition from "Keep" to "Remove" represents an orthographic normalization layer where stripping the vowel-length marker (":") collapses variants like *wa:n* and *wan* into a single token, successfully reducing unique vocabulary sparsity while keeping the total token count exactly identical at 347,109.

Conversely, moving from "Remove" to "Substitute" introduces aggressive morphological segmentation by breaking compound words or attached particles into independent units, this structural change causes the unique vocabulary to plummet to its lowest point (15,679) while the total token count explodes to over half a million, ultimately reducing the dictionary size the model needs to learn at the cost of significantly increasing the overall sentence sequence lengths across the corpus.

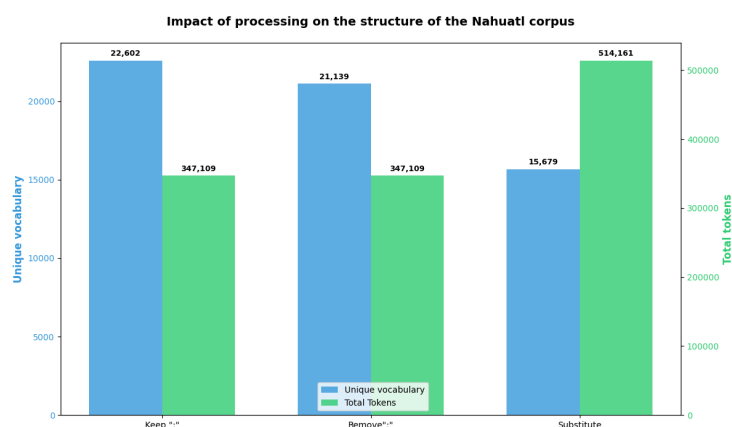


Fig. 1. Impact of each scenery in total number of tokens.

The information given in Figure 2 confirms the invariance of removing colon for context analysis, both scenarios lie at around 5 to 7 tokens per sentence. Conversely, the "Substituting" scenario is significantly flattened and shifted to the right, where the peak density broadens to around 8 to 12 tokens per sentence while the right tail stretches well past 30 to 40 tokens as a direct structural consequence of the morphological splitting. For modern NLP frameworks like Transformers, this Substituting scenario significantly increases the maximum and average sequence lengths and lowers the information density per token compared to the other scenarios.

In order to evaluate the performance impact of these structural differences, all three preprocessing configurations (Keeping, Removing, and Substituting) were implemented and evaluated during NLLB model training. Applying the NLLB

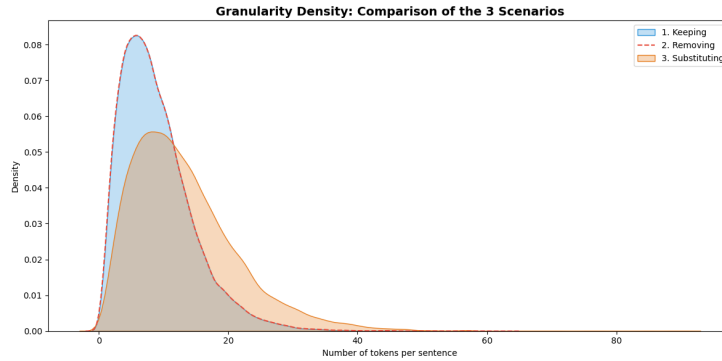


Fig. 2. Granularity density of tokens per sentence for each scenery.

model across all three scenarios yields the BLEU and WER evaluation scores shown in Figure 3 and Figure 4, respectively.

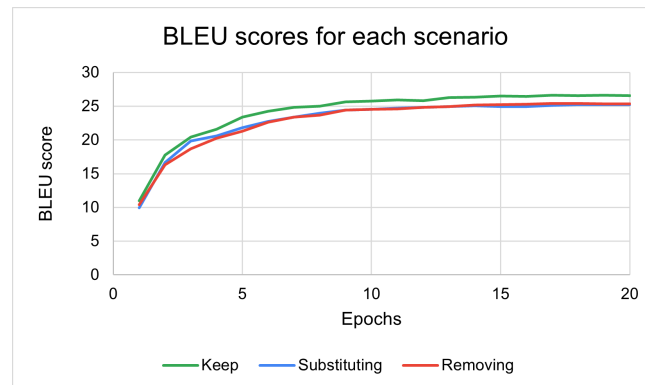


Fig. 3. BLEU scores for each epoch in validation set.

By analyzing the behavior of BLEU scores across epochs for all three scenarios, three distinct phases are clearly identified:

- Explosive Learning Phase (Epochs 1 to 5): Models across all three scenarios rapidly assimilate foundational linguistic structures. BLEU scores increase sharply from initial values of 10.99 (Keeping), 9.96 (Removing), and 10.37 (Substituting) at Epoch 1 to 23.37, 21.83, and 21.30, respectively, by Epoch 5.
- Convergence Phase (Epochs 6 to 15): Performance stabilizes into a high-accuracy region. The "Keeping" scenario converge to BLEU scores near to 26 at Epoch 15, while the "Removing" and "Substituting" scenarios converge to BLEU scores near to 25.

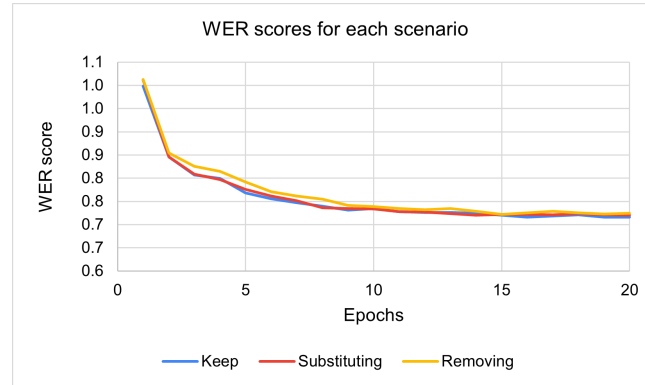


Fig. 4. WER scores for each epoch in validation set.

- Plateau Phase (Epochs 16 to 20): Rather than experiencing severe performance degradation, BLEU scores plateau across all setups, maintaining stable scores above 25.2 for Removing and Substituting, and above 26.4 for "Keeping", demonstrating consistent generalization without sharp overfitting through 20 epochs.

The WER metric measures the word error rate by accounting for substitutions, deletions, and insertions required to align predicted sequences with the reference text. Across all three scenarios, initial error rates at Epoch 1 are high due to complete undertraining—starting at 99.82% (0.9982) for "Keeping", 101.20% (1.0120) for "Removing", and 101.30% (1.0130) for "Substituting". Following this, a rapid downward trajectory emerges as WER drops significantly by Epoch 2 (84.58% for Keeping, 84.62% for Removing, and 85.43% for Substituting).

Error rates continue to decline in parallel with BLEU growth before reaching their respective minimums: 71.60% at Epoch 20 for Keeping, 72.01% at Epoch 20 for "Removing", and 72.24% at Epoch 15 for Substituting, after which scores stabilize within tight plateaus.

The final evaluation on the locked test set, comprising 3,865 sentences, demonstrated competitive performance across all three tokenization strategies. The "Substituting" scenario achieved the highest test BLEU score of 18.59 (with a WER of 67.38%), closely followed by "Removing" (BLEU 18.49, WER 66.89%) and "Keeping" (BLEU 18.44, WER 66.85%). Compared to validation training curves—where validation BLEU peaked between 25.26 and 26.64 and WER stabilized near 71.60%–72.24%—the test evaluation reveals a slight generalization gap in translation precision (BLEU) on completely unseen data, alongside improved word-level accuracy (WER drops by approximately 5 percentage points across all scenarios on the test set).

During training, "Keeping" reached the highest validation BLEU score (26.64 vs. 25.42 for "Substituting"), this shows that holding whole words works well when the validation set shares the exact same words as the training data. However,

on the final unseen test set, "Substituting" achieved a slightly better BLEU score, this confirms that breaking words into smaller pieces helps the model handle new, unknown words (out-of-vocabulary) much better when training data is limited.

Even though breaking up words made sentences much longer—increasing total tokens by over 44% and doubling average sentence lengths in Substituting—the final test scores across all three setups were very similar (BLEU 18.44–18.59; WER 66.85%–67.38%). This shows that the large pretrained NLLB model is strong enough to handle big changes in text formatting without losing translation quality, and that simple text cleanup mitigates sparsity without making sentences too long or complicated for the fine tuning process.

6 Conclusions

In this study, we investigated the impact of character-level text preprocessing and orthographic representation on the neural machine translation of the Northeastern Puebla Sierra Nahuatl variant, a low-resource agglutinating language. By conducting a comprehensive statistical evaluation across three distinct tokenization setups (Keeping, Removing, and Substituting), we demonstrated how character-level markers directly alter vocabulary granularity and corpus density. While the "Keeping" and "Removing" configurations preserved identical word boundaries, the "Substituting" scenario implemented an aggressive morphological segmentation. This structural alteration collapsed the unique vocabulary size while causing the total token count to increase, thereby altering sequence lengths and context distribution during transformer training.

Fine-tuning the open-source NLLB-200 model across all three configurations revealed consistent training dynamics characterized by rapid early learning (Epochs 1–5) followed by convergence in later epochs (Epochs 15–20), where validation BLEU scores reached 26.64 for Keeping, 25.42 for Substituting, and 25.26 for Removing.

The final evaluation on the locked test set of 3,865 sentences demonstrated overall structural consistency across preprocessing methods. The "Substituting" approach achieved the top BLEU score of 18.59 on test data, whereas the "Keeping" configuration yielded the lowest Word Error Rate at 66.85%. Although a modest generalization gap manifests in BLEU when comparing test results to validation peaks, word-level error rates showed improved alignment on test data. Ultimately, these findings underscore that comparative orthographic evaluation combined with advanced state-of-the-art multilingual architectures enhances translation performance in endangered agglutinating languages.

For future work, it is highly recommended to explore more specialized text processing techniques designed for translation tasks of agglutinated languages, like Nahuatl, which could help manage complex word formations more effectively. Additionally, future studies should revise other promising transformer topologies and architectures to see how they perform on indigenous translation tasks. When comparing different frameworks, it is crucial to check carefully if other models separate a complete, isolated test set after training to completely avoid

data leakage and ensure a fair evaluation. Finally, an extensive fine-tuning of hyperparameters (such as experimenting with different learning rates, batch choices, and optimization steps) is strongly recommended to find the absolute best configuration for translating these indigenous variants.

Acknowledgment. The author thankfully acknowledge the computer resources, technical expertise and support provided by the National Supercomputing Laboratory of Southeastern Mexico and the Ministry of Science, Humanities, Technology and Innovation.

References

1. Bautista Morales, R., Martínez Ramírez, Y., Rocha Peña, L.E., Montes Santiago, R.E.: Arquitectura de un traductor automático para el idioma mixteco: un enfoque específico para lenguas indígenas con escasos recursos lingüísticos. *Revista de Investigación en Tecnologías de la Información* 12(28), 71–81 (Nov 2024)
2. Bello García, S.K., Sánchez Lucero, E., Bonilla Huerta, E., Hernández Hernández, J.C., Ramírez Cruz, J.F., Pedroza Méndez, B.E.: Nahuatl neural machine translation using attention based architectures: A comparative analysis for RNNs and transformers as a mobile application service. In: *Advances in Soft Computing*, pp. 120–139. *Lecture Notes in Computer Science*, Springer International Publishing, Cham (2021)
3. Cossich Vielman, M.V.: El sistema de escritura jeroglífico náhuatl: Análisis epigráfico de los onomásticos de cinco documentos del siglo XVI de Tepetlaóztoc. Tesis de maestría, Universidad Nacional Autónoma de México (UNAM), Ciudad de México, México (2014)
4. Davletshin, A.: Descripción funcional de la escritura jeroglífica náhuatl. *Estudios de Cultura Náhuatl* 62, 43–93 (2021)
5. Embriz Osorio, A., Zamora Alarcón, Ó. (eds.): *Lenguas Indígenas Nacionales en Riesgo de Desaparición*. Instituto Nacional de Lenguas Indígenas (INALI), México, D.F. (2012)
6. González Servín, C., Maldonado Sifuentes, C.E., Kolesnicova, O., Sidorov, G.: Evaluating the impact of domain adaptation on transformer-based models for low-resource Purépecha-Spanish translation. *Int. J. Comb. Optim. Probl. Inform.* 17(2), 27–37 (Feb 2026)
7. NLLB Team, Costa-jussà, M.R., Cross, J., Çelebi, O., Elbayad, M., Heafield, K., Heffernan, K., Kalbassi, E., Lam, J., Licht, D., Maillard, J., Sun, A., Wang, S., Wenzek, G., Youngblood, A., Akula, B., Barrault, L., Gonzalez, G.M., Hansanti, P., Hoffman, J., Jarrett, S., Sadagopan, K.R., Rowe, D., Spruit, S., Tran, C., Andrews, P., Ayan, N.F., Bhosale, S., Edunov, S., Fan, A., Gao, C., Goswami, V., Guzmán, F., Koehn, P., Mourachko, A., Ropers, C., Saleem, S., Schwenk, H., Wang, J.: No language left behind: Scaling human-centered machine translation (Jul 2022)
8. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU. In: *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02*. Association for Computational Linguistics, Morristown, NJ, USA (2001)
9. Shi, J., Amith, J.D., Chang, X., Dalmia, S., Yan, B., Watanabe, S.: Highland puebla nahuatl speech translation corpus for endangered language documentation. In: *Proceedings of the First Workshop on Natural Language Processing for Indigenous Languages of the Americas*. pp. 53–63 (2021)

10. Tonja, A.L., Maldonado-sifuentes, C., Mendoza Castillo, D.A., Kolesnikova, O., Castro-Sánchez, N., Sidorov, G., Gelbukh, A.: Parallel corpus for indigenous language translation: Spanish-mazatec and Spanish-Mixtec. In: Proceedings of the Workshop on Natural Language Processing for Indigenous Languages of the Americas (AmericasNLP). Association for Computational Linguistics, Stroudsburg, PA, USA (2023)
11. Wang, Y., Xu, C., Sun, Q., Hu, H., Tao, C., Geng, X., Jiang, D.: PromDA: Prompt-based data augmentation for low-resource NLU tasks. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, Stroudsburg, PA, USA (2022)
12. Zacarías Márquez, D., Meza Ruiz, I.V.: Ayuuk-Spanish neural machine translator. In: Proceedings of the First Workshop on Natural Language Processing for Indigenous Languages of the Americas. Association for Computational Linguistics, Stroudsburg, PA, USA (2021)

Pill Counting Using Watershed-based Segmentation Methods

Luis F. De-la-Rosa-Pineda, Victor M. Leyva-Osorio

Benemérita Universidad Autónoma de Puebla,
Facultad de Ciencias de la Computación,
México

{rp226570038, lo226570039, cs226570036, cc226570037}@alm.buap.mx

Abstract. This paper presents a methodology for automatic pill counting using digital image processing techniques. The proposed method starts with the acquisition of a digital image, which is converted to grayscale and smoothed using a Gaussian filter to reduce noise. The image is then binarized using Otsu's method, followed by morphological operations to remove small objects and generate markers. A distance transform and the Watershed algorithm are used to separate pills that are in contact. Finally, the segmented regions are identified and counted using connected components. The methodology was evaluated under controlled conditions using images of groups of pills placed in different positions, including cases where some pills were touching. The method successfully identified and counted 32 pills in the first test image and 17 pills in the second. These results show that the combination of thresholding, morphological operations, and Watershed segmentation can be used for automatic pill counting under the evaluated conditions.

Keywords: Digital image processing, pill counting, Otsu method, image segmentation, Watershed segmentation.

Conteo de pastillas basado en métodos de segmentación de Watershed

Resumen. Este artículo presenta una metodología para el conteo automático de pastillas mediante técnicas de procesamiento digital de imágenes. El método propuesto comienza con la adquisición de una imagen digital, la cual se convierte a escala de grises y se suaviza utilizando un filtro gaussiano para reducir el ruido. A continuación, la imagen se binariza mediante el método de Otsu, seguido de operaciones morfológicas para eliminar objetos pequeños y generar marcadores. Se emplean la transformada de distancia y el algoritmo de Watershed para separar las pastillas que se encuentran en contacto. Finalmente, las regiones segmentadas se identifican y cuentan utilizando componentes conexos. La metodología se evaluó en condiciones controladas utilizando imágenes de grupos de pastillas dispuestas en diferentes posiciones,

incluyendo casos en los que algunas pastillas se tocaban entre sí. El método identificó y contó con éxito 32 pastillas en la primera imagen de prueba y 17 en la segunda. Estos resultados demuestran que la combinación de umbralización, operaciones morfológicas y segmentación Watershed puede utilizarse para el conteo automático de pastillas bajo las condiciones evaluadas.

Palabras clave: Procesamiento digital de imágenes, conteo de pastillas, método de Otsu, segmentación de imágenes, segmentación Watershed.

1 Introduction

Digital image processing is an area focused on the analysis and manipulation of digital images using computational methods. These techniques make it possible to extract useful information from images and have become an important tool in different areas of science and engineering [3].

One of the applications of digital image processing is the automatic detection and counting of objects. In this type of application, an image can be processed to separate the objects of interest from the background and then identify each object individually. However, the counting process becomes more difficult when two or more objects are touching, since after binarization they can appear as a single connected region. This makes it necessary to use additional segmentation techniques before performing the final count [7].

For pill counting, digital image processing provides an alternative to manual counting by allowing pills to be detected from a captured image and counted from the regions obtained after segmentation. Previous works have shown that operations such as grayscale conversion, filtering, thresholding and mathematical morphology can be used for this purpose [5].

The objective of this work is to develop and evaluate a methodology for automatic pill counting using digital image processing. The main problem addressed is the separation of pills that are in contact, since they can be detected as a single object during conventional connected component analysis. The proposed methodology combines grayscale conversion, Gaussian filtering, Otsu thresholding, morphological operations, distance transform and marker based Watershed segmentation. Finally, connected component labeling is used to determine the number of pills present in the processed image.

2 Related Work

Different image processing techniques have been used for automatic object counting and for the separation of objects that are in contact. These methods usually combine image preprocessing, segmentation and connected component analysis according to the characteristics of the objects and the conditions in which the images are acquired. Phromlikhit et al. [5] proposed a tablet counting

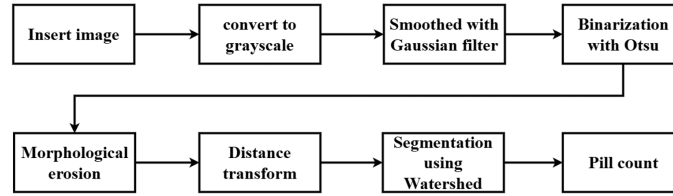


Fig. 1. Block diagram of the proposed methodology.

machine based on image processing. Their method uses images acquired with a digital camera and applies grayscale conversion, noise filtering, thresholding, erosion and dilation. After processing the image, connected component labeling and centroid information are used to determine the number of tablets. This work shows that relatively simple image processing operations can be used for automatic tablet counting. However, the separation of pills that remain connected after thresholding requires additional consideration.

For objects that are touching, Watershed segmentation provides an alternative for separating connected regions. Vincent and Soille [8] presented an efficient Watershed algorithm for digital grayscale images based on an immersion process. In this approach, the image is interpreted as a topographic surface and the resulting watershed lines define the boundaries between regions.

The use of Watershed for object counting has also been studied in applications where several objects can form connected groups. Tan et al. [7] developed a method for the segmentation and counting of touching rice grains. Their approach combines Watershed segmentation with corner detection and a neural network to reduce over segmentation and under segmentation problems. Their results also show that the way in which objects touch and their shape can affect the segmentation process. Based on these works, the methodology presented in this paper uses a combination of thresholding, morphological processing and marker based Watershed segmentation. Unlike approaches based only on connected components, the use of markers and the distance transform is intended to separate pills that remain connected after binarization before performing the final count.

3 Methodology

The methodology used for pill counting is shown in Figure 1. The process consists of a sequence of image processing operations, starting with image acquisition and ending with the identification and counting of the segmented objects. MATLAB was used to implement and evaluate the proposed methodology.

3.1 Image Acquisition

The first step consists of acquiring the image containing the pills to be counted. The images are captured in RGB format, where each pixel is represented by three components corresponding to the red (R), green (G), and blue (B) channels.

After loading the image in MATLAB, it is represented as a numerical matrix that can be processed by the following stages of the methodology. Image acquisition is an important step because factors such as lighting, camera resolution, shadows, and reflections can affect the segmentation process.

3.2 Grayscale Conversion

Before applying the segmentation methods, the RGB image is converted to grayscale. This reduces the image from three color channels to a single intensity channel, since the proposed method uses intensity information rather than color information [3].

The grayscale intensity Y is calculated from the RGB components using the following expression [3]:

$$Y = 0.2989R + 0.5870G + 0.1140B, \quad (1)$$

where R , G , and B represent the intensity values of the red, green, and blue channels, respectively.

3.3 Gaussian Smoothing

After the grayscale conversion, a Gaussian filter is applied to reduce small intensity variations and noise that could affect the subsequent segmentation stages. Gaussian smoothing is performed by convolving the image with a kernel whose values are determined by a Gaussian function [3].

The two-dimensional Gaussian function can be expressed as:

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right), \quad (2)$$

where σ controls the amount of smoothing applied to the image. A larger value of σ produces stronger smoothing, while a smaller value preserves more local variations.

For the selected configuration of the proposed method, a value of $\sigma = 2$ was used. This value provided enough smoothing to reduce the small amount of noise present in the test images without excessively blurring the boundaries of the pills.

3.4 Otsu Image Binarization

After smoothing the grayscale image, the next step is to separate the pills from the background. For this purpose, the image is converted into a binary representation, where each pixel is assigned to either the background or the objects of interest.

The binarization process is performed using a threshold T . For an input image $I(x, y)$, the resulting binary image $g(x, y)$ can be defined as:

$$g(x, y) = \begin{cases} 1, & \text{if } I(x, y) \geq T, \\ 0, & \text{if } I(x, y) < T. \end{cases} \quad (3)$$

Instead of using a fixed threshold, the value of T is automatically determined using Otsu's method. This method selects a threshold from the intensity histogram by maximizing the variance between two classes of pixels [4].

For an image of size $M \times N$ with L intensity levels $[0, 1, \dots, L-1]$, let n_i represent the number of pixels with intensity i . The probability of occurrence of each intensity level is defined as:

$$p_i = \frac{n_i}{MN}. \quad (4)$$

For a possible threshold k , the pixels are divided into two classes: C_1 , containing intensity levels from 0 to k , and C_2 , containing intensity levels from $k+1$ to $L-1$.

The probabilities associated with these two classes are:

$$P_1(k) = \sum_{i=0}^k p_i \quad (5)$$

and

$$P_2(k) = \sum_{i=k+1}^{L-1} p_i = 1 - P_1(k). \quad (6)$$

The mean intensity of each class is then calculated as:

$$\mu_1(k) = \frac{1}{P_1(k)} \sum_{i=0}^k ip_i, \quad (7)$$

$$\mu_2(k) = \frac{1}{P_2(k)} \sum_{i=k+1}^{L-1} ip_i. \quad (8)$$

The global mean intensity of the image is given by:

$$\mu_G = \sum_{i=0}^{L-1} ip_i. \quad (9)$$

Using these values, the between class variance for each possible threshold k is calculated as:

$$\sigma_B^2(k) = P_1(k) [\mu_1(k) - \mu_G]^2 + P_2(k) [\mu_2(k) - \mu_G]^2. \quad (10)$$

Finally, Otsu's method selects the threshold K that maximizes the between class variance:

$$K = \arg \max_{0 \leq k \leq L-1} \{\sigma_B^2(k)\}. \quad (11)$$

In this way, the threshold is obtained directly from the intensity distribution of each image instead of using a fixed value for all test images.

3.5 Small Objects Removal

After binarization, small regions or artifacts that do not correspond to pills may remain in the binary image. To remove these regions, a cleaning process based on the area of the connected components is applied.

First, the connected components in the binary image are identified and their areas are calculated. Components with an area smaller than a predefined threshold of 1000 pixels are removed. This step reduces the presence of small regions that could affect the subsequent segmentation and counting process.

3.6 Marker Generation Through Morphological Erosion

After removing small objects, morphological erosion is applied to the binary image. Erosion is one of the fundamental operations of mathematical morphology and reduces the boundaries of foreground regions according to the shape and size of a structuring element [6].

For a binary image represented by the set A and a structuring element B , erosion can be defined as [6]:

$$A \ominus B = \{z \mid B_z \subseteq A\}, \quad (12)$$

where B_z represents the structuring element B translated to position z . Therefore, a point remains in the resulting set only when the structuring element fits completely inside the object at that position.

In the proposed methodology, a disk shaped structuring element is used to reduce the regions corresponding to the pills. This operation generates smaller regions located approximately at the center of each object. These regions are used as internal markers for the subsequent Watershed segmentation process.

For the selected configuration, a disk with a radius of 72 pixels was used. The effect of changing this parameter is evaluated in the experimental results, since the size of the structuring element influences the separation and preservation of the markers.

3.7 Distance Transform

The distance transform is used to calculate the distance from each foreground pixel to the nearest background pixel in a binary image. As a result, pixels located near the center of an object have larger distance values, while pixels close to its boundary have smaller values [2].

Using the Euclidean distance, the distance transform can be expressed as:

$$D(x, y) = \min_{(x', y') \in B} \sqrt{(x - x')^2 + (y - y')^2}, \quad (13)$$

where $D(x, y)$ represents the distance from the foreground pixel (x, y) to the nearest background pixel, and B is the set of pixels belonging to the image background.

For the proposed methodology, the negative distance transform is used:

$$D_n(x, y) = -D(x, y). \quad (14)$$

By inverting the distance values, the central regions of the pills become local minima. This representation is used in the following Watershed segmentation stage to help separate objects that are in contact.

3.8 Segmentation Using Watershed

The Watershed algorithm is a segmentation method based on the topographic interpretation of an image. In this representation, low intensity values can be interpreted as valleys or basins, while higher values represent ridges. The segmentation process can then be described as a flooding process that starts from the local minima of the image [1, 8].

For the proposed methodology, the negative distance transform obtained in the previous step is used as the input for the Watershed algorithm. Since the central regions of the pills correspond to local minima in this representation, they can be used to guide the segmentation of objects that are in contact.

Let $M_1, M_2, \dots, M_R$ represent the set of regional minima present in an image $g(x, y)$. For a flooding level n , the set of points below this level can be expressed as [8]:

$$T_n = (x, y) \mid g(x, y) < n. \quad (15)$$

For each regional minimum M_i , the portion of its catchment basin that has been flooded at level n is defined as:

$$C_n(M_i) = C(M_i) \cap T_n, \quad (16)$$

where $C(M_i)$ represents the catchment basin associated with the minimum M_i .

As the flooding level increases, different catchment basins begin to approach each other. When regions associated with different minima would merge, a separation line is created between them. The collection of these lines forms the Watershed boundaries that divide the image into different regions [8].

In this work, a marker based Watershed approach is used. The regions previously obtained through morphological erosion are used as internal markers for the pills. These markers guide the segmentation process and help reduce the generation of unnecessary regions caused by small local variations in the image.

3.9 Final Cleaning and Object Counting

After the Watershed segmentation, a final cleaning process is applied to remove small regions that do not correspond to valid pills. These regions may appear due to small segmentation errors or fragments generated during the separation of objects.

The cleaning process is based on the area of the segmented regions. Connected components with an area smaller than 15000 pixels are removed from the binary

image. This threshold was selected according to the approximate size of the pills in the test images.

Once the final binary image is obtained, the remaining connected components are identified using 8-connectivity. For a pixel $p = (x_p, y_p)$, its 8-neighborhood can be defined as [3]:

$$N_8(p) = (x, y) \mid \max(|x - x_p|, |y - y_p|) = 1. \quad (17)$$

Using connected component labeling, each independent region in the binary image is assigned a different label. The total number of detected pills is then obtained from the number of labeled connected components. If k represents the total number of valid labeled regions, the final count is given by $N = k$.

Therefore, after the cleaning and segmentation stages, each remaining connected component is considered as one detected pill.

4 Results and Experimentation

This section presents the results obtained with the proposed methodology and evaluates the effect of some of the parameters used during image processing. In addition to the final pill count, the intermediate results are analyzed to identify the conditions in which the segmentation performs correctly and the cases where errors may occur.

4.1 Experimental Setup

The proposed methodology was implemented in MATLAB and evaluated using images containing pills with different shapes and arrangements. The test images include circular pills and elongated capsules, as well as cases where multiple objects are in contact.

The selected configuration used a Gaussian filter with $\sigma = 2$. After binarization using Otsu's method, connected components smaller than 1000 pixels were removed. A disk-shaped structuring element with a radius of 72 pixels was then used to generate the internal markers for the Watershed segmentation. After segmentation, regions with an area smaller than 15000 pixels were removed before performing the final connected-component labeling. The parameters of the selected configuration are summarized in Table 1.

These values were selected from the evaluated parameter ranges and were used for the final pill counting experiments. Some of the parameters were subsequently varied to evaluate their influence on the segmentation and final counting results.

4.2 Image Processing Results

Two images were initially used to evaluate the proposed methodology. The first image contains circular pills, while the second contains elongated capsules. In both cases, some objects are in contact, allowing the segmentation process to be evaluated with different pill shapes and arrangements.

Table 1. Selected parameters used in the proposed methodology.

Parameter	Selected value
Gaussian smoothing (σ)	2
Initial area threshold	1000 pixels
Structuring element	Disk
Structuring element radius	72 pixels
Final area threshold	15000 pixels
Connectivity	8

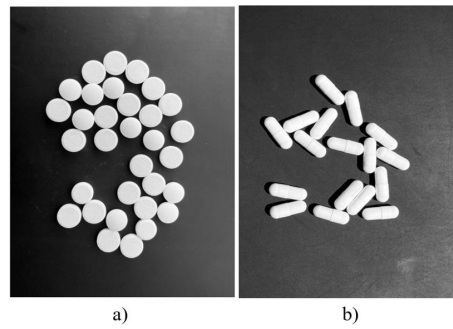


Fig. 2. Preprocessed images after grayscale conversion and Gaussian smoothing: (a) circular pills and (b) elongated capsules.

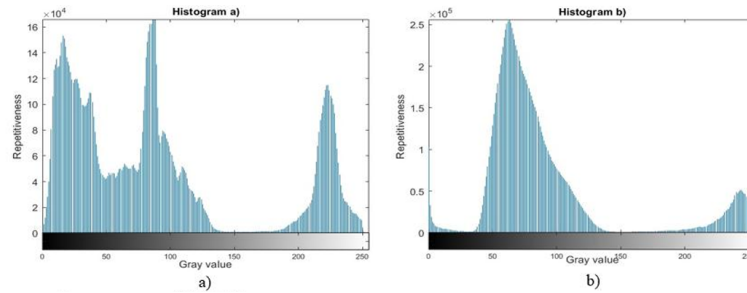


Fig. 3. Intensity histograms of the test images.

The images were first converted to grayscale, removing the color information and keeping the intensity information used in the following processing stages. A Gaussian filter with $\sigma = 2$ was then applied to reduce small intensity variations and noise. The resulting preprocessed images are shown in Figure 2.

The intensity histograms of the test images are shown in Figure 3. Differences can be observed between the gray-level distributions of both images. These variations can be produced by factors such as lighting, reflections, and the conditions in which each image was captured.

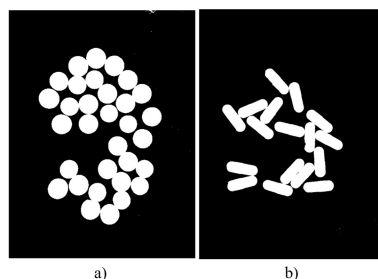


Fig. 4. Binarized images using Otsu's method.

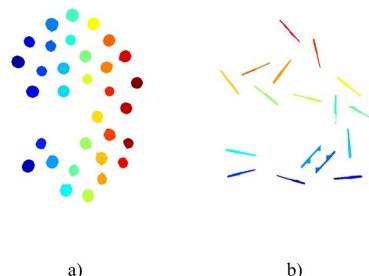


Fig. 5. Internal markers obtained through morphological erosion.

Because the intensity distribution is not the same for both images, Otsu's method was used to calculate the binarization threshold independently for each case. The resulting binary images are shown in Figure 4, where the pills are separated from the background.

After removing small regions, morphological erosion was applied using a disk-shaped structuring element with a radius of 72 pixels. The resulting internal markers are shown in Figure 5. These markers represent the regions used to identify the individual pills during the Watershed segmentation.

The distance transform was then calculated from the binary images. The distance values increase toward the central regions of the objects, and the negative of this transformation was used for the subsequent Watershed segmentation.

Figure 6 shows the result obtained after applying the Watershed algorithm. The generated boundaries allow several regions that were previously in contact to be separated.

Finally, the segmented images were cleaned by removing regions below the final area threshold. The resulting images are shown in Figure 7. After connected-component labeling, 32 regions were detected in the first image and 17 regions in the second image.

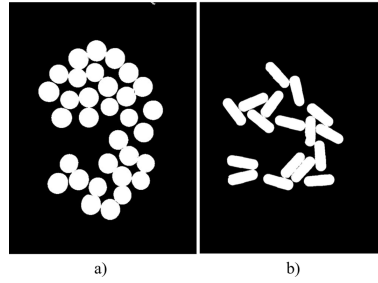


Fig. 6. Watershed segmentation of the test images.

Table 2. Effect of the structuring element radius on the final pill count.

Radius [px]	Image A	Image B
36	22	16
48	27	16
60	32	16
72	32	17
84	32	20

Table 3. Effect of Gaussian smoothing on the final pill count.

σ	Image A	Image B
1	32	17
2	32	17
3	32	16
4	32	16

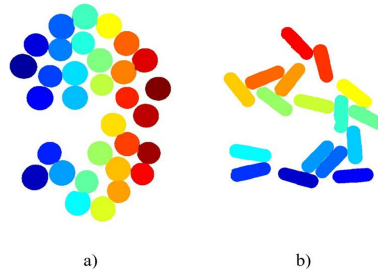


Fig. 7. Final segmented images used for pill counting.

4.3 Parameter Evaluation

To evaluate the influence of the parameters used in the proposed methodology, three parameters were varied independently: the radius of the structuring element used for marker generation, the value of σ used for Gaussian smoothing, and the final area threshold. During each experiment, only one parameter was modified while the remaining parameters were kept at their initial values.

Table 4. Effect of the final area threshold on the pill count.

Area threshold [px]	Image A	Image B
1000	33	17
5000	32	17
10000	32	17
15000	32	17
20000	32	17
25000	32	17

The first experiment evaluated the radius of the disk-shaped structuring element. The results are shown in Table 2.

The results show that the structuring element radius has an important effect on the final segmentation. Small radius values produced an incorrect number of detected regions, especially in Image A. Increasing the radius improved the result until a radius of 72 pixels, where 32 and 17 pills were detected in Images A and B, respectively. However, increasing the radius to 84 pixels caused Image B to produce 20 regions. This shows that increasing the radius does not always improve the segmentation and that its value must be selected according to the size and shape of the objects.

The second experiment evaluated the effect of Gaussian smoothing. The results are presented in Table 3.

Image A maintained a count of 32 pills for all the evaluated values of σ . Image B produced the expected count of 17 pills with $\sigma = 1$ and $\sigma = 2$, while increasing the value to 3 or 4 resulted in 16 detected regions. This indicates that, for the evaluated images, the elongated capsules in Image B were more sensitive to stronger smoothing.

Finally, the final area threshold was varied between 1000 and 25000 pixels. The results are shown in Table 4.

Compared with the previous parameters, the final area threshold showed less influence on the counting results within the evaluated range. A threshold of 1000 pixels produced one additional region in Image A.

From 5000 to 25000 pixels, both images maintained counts of 32 and 17 pills. The selected value of 15000 pixels was therefore kept for the final configuration.

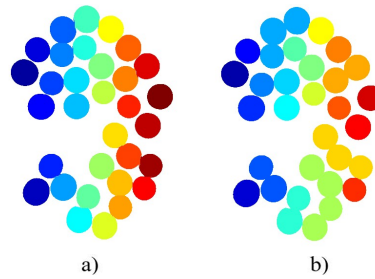
4.4 Pill Counting Results

After evaluating the different parameter values, the selected configuration was used for the final counting process. This configuration uses $\sigma = 2$, a structuring element radius of 72 pixels, an initial area threshold of 1000 pixels, and a final area threshold of 15000 pixels. The final counting results are summarized in Table 5.

The selected configuration produced 32 detected pills in Image A and 17 in Image B. These results are consistent with the final segmented images shown previously. The parameter evaluation also showed that the correct separation of

Table 5. Final pill counting results using the selected configuration.

Test image	Detected pills
Image A	32
Image B	17

**Fig. 8.** Effect of the structuring element radius on Image A: (a) $r = 36$, resulting in under segmentation and 22 detected regions; (b) $r = 72$, resulting in 32 detected regions.

the objects depends mainly on the generation of appropriate internal markers, particularly on the radius used during morphological erosion.

4.5 Analysis and Validity of Results

The experiments show that the proposed methodology can correctly separate and count the pills in the two test images when appropriate parameter values are used. With the selected configuration, 32 pills were detected in Image A and 17 capsules in Image B.

The parameter evaluation showed that the radius of the structuring element had the greatest influence on the final count among the parameters evaluated. Figure 8 shows the effect of using different radius values in Image A. With $r = 36$, some pills are not separated correctly, resulting in 22 detected regions. With the selected radius of $r = 72$, the pills are separated individually and 32 regions are obtained.

A different effect can be observed in Image B, as shown in Figure 9. With the selected radius of $r = 72$, 17 regions are obtained.

When the radius is increased to 84 pixels, some capsules are divided into more than one region, increasing the final count to 20. In this case, the larger radius produces over segmentation.

These results indicate that the generation of internal markers is an important stage of the methodology. If the selected radius is not appropriate for the objects in the image, the number of regions obtained after Watershed segmentation can change and produce an incorrect final count. A small radius produced under segmentation in Image A, while a larger radius produced over segmentation in Image B.

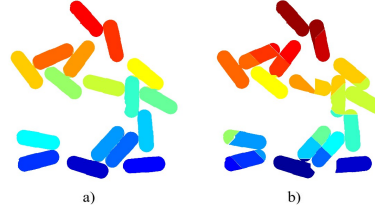


Fig. 9. Effect of the structuring element radius on Image B: (a) $r = 72$, resulting in 17 detected regions; (b) $r = 84$, producing over segmentation and 20 detected regions.

The experiments with Gaussian smoothing showed a smaller effect on the final count. Image A maintained 32 detected regions for all the evaluated values of σ . Image B maintained 17 regions with $\sigma = 1$ and $\sigma = 2$, but the count decreased to 16 when σ was increased to 3 and 4. This indicates that the result for Image B was more sensitive to stronger smoothing within the evaluated range.

The final area threshold showed the smallest influence among the three evaluated parameters. A threshold of 1000 pixels resulted in 33 detected regions in Image A, while Image B maintained a count of 17. For threshold values between 5000 and 25000 pixels, both images maintained counts of 32 and 17 regions, respectively.

Although the selected configuration produced the expected results for both test images, the experiments also show some limitations of the methodology. The same parameter values may not provide the same segmentation result for objects with different shapes and sizes. In particular, the generation of internal markers depends on the radius of the structuring element, which can lead to under-segmentation or over-segmentation when an inappropriate value is used.

Therefore, the results obtained in this work validate the methodology for the evaluated images and parameter ranges. Additional images should be captured under different conditions, considering variations in pill arrangement, size, and shape, as well as changes in lighting and background, to evaluate the performance of the method under different scenarios.

5 Conclusions

In this work, a methodology for automatic pill counting using digital image processing was presented. The method combines grayscale conversion, Gaussian smoothing, Otsu thresholding, morphological operations, distance transform, and marker based Watershed segmentation. With the selected configuration, the method detected 32 pills in Image A and 17 capsules in Image B.

Although the methodology is based on classical image processing tools and does not use artificial intelligence, their combination made it possible to successfully count the pills in the images evaluated in this work. In particular, the use of morphological operations, distance transform, and Watershed segmentation allowed pills that were in contact to be separated before the final counting stage.

The experiments also showed that the selection of the processing parameters can affect the segmentation result.

One limitation of the proposed methodology occurs when pills overlap. Since the method works with two dimensional images, an overlapping pill can partially hide another object, making their separation more difficult with the techniques used in this work. Therefore, the current methodology is mainly intended for cases where the pills are visible but may be in contact with each other.

Future work could include images captured under different conditions, with variations in pill arrangement, size, shape, lighting, and background. The evaluation of methods for handling partially overlapping pills and automatically adjusting the processing parameters could also be considered.

References

1. Beucher, S., Lantuéjoul, C.: Use of watersheds in contour detection. In: Proceedings of the International Workshop on Image Processing, Real-Time Edge and Motion Detection/Estimation. Rennes, France (1979)
2. Borgefors, G.: Distance transformations in digital images. *Computer Vision, Graphics, and Image Processing* 34(3), 344–371 (1986)
3. Gonzalez, R.C., Woods, R.E.: *Digital Image Processing*. Pearson, 4 edn. (2018)
4. Otsu, N.: A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics* 9(1), 62–66 (1979)
5. Phromlikhit, C., Cheevasuvit, F., Yimman, S.: Tablet counting machine base on image processing. In: 2012 Biomedical Engineering International Conference (BMEiCON). pp. 400–403 (2012)
6. Soille, P.: *Morphological Image Analysis: Principles and Applications*. Springer, Berlin, Heidelberg, 2 edn. (2003)
7. Tan, S., Ma, X., Mai, Z., Qi, L., Wang, Y.: Segmentation and counting algorithm for touching hybrid rice grains. *Computers and Electronics in Agriculture* 162, 493–504 (2019)
8. Vincent, L., Soille, P.: Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13(6), 583–598 (1991)

Two Hands Static Mexican Sign Language Dataset for Object Detection

Víctor Martínez-Sánchez, Ivan Villalon-Turrubiates,
Francisco Cervantes

Instituto Tecnológico y de Estudios Superiores de Occidente (ITESO),
Tlaquepaque, Jalisco
Mexico

{ng683728, villalon, fcervantes}@iteso.mx

Abstract. Currently, there is a trend to use machine learning and deep learning techniques for image classification and object detection into computer-vision systems. Creating and training such models require a considerable amount of images. Gathering hand-gesture data for sign language is not an easy task for researchers because unlike popular belief, sign language is not universal. This is partly due to there are no standard signs for all languages. Hence, these datasets must be created according to a specific language. In this paper we propose a Mexican Sign Language (MSL) dataset that includes 21 static hand-gestures using both hands, each representing a letter in the Spanish alphabet. The dataset consists of 10,000 high resolution colored images in JPG format captured at different environments. The dataset is evaluated in a deep learning model with an overall accuracy of 99.46%.

Keywords: Mexican sign language, dataset, hand-gestures, computer vision.

Conjunto de datos de lengua de señas mexicana estática con dos manos para la detección de objetos

Resumen. Actualmente, existe una tendencia a utilizar técnicas de aprendizaje automático y aprendizaje profundo para la clasificación de imágenes y la detección de objetos en sistemas de visión por computadora. La creación y el entrenamiento de dichos modelos requieren una cantidad considerable de imágenes. La recopilación de datos sobre gestos manuales para la lengua de señas no es una tarea sencilla para los investigadores, ya que, contrariamente a la creencia popular, la lengua de señas no es universal. Esto se debe, en parte, a que no existen señas estandarizadas para todos los idiomas; por consiguiente, estos conjuntos de datos deben crearse en función de un idioma específico. En este artículo, proponemos un conjunto de datos de Lengua de Señas Mexicana (LSM) que incluye 21 gestos manuales estáticos realizados con ambas manos, cada uno de los cuales representa una letra del alfabeto español. El conjunto de datos consta de 10.000 imágenes en

color de alta resolución en formato JPG, capturadas en diversos entornos. Dicho conjunto de datos se evalúa mediante un modelo de aprendizaje profundo, alcanzando una precisión global del 99,46 %.

Palabras clave: Lengua de señas mexicana, conjunto de datos, gestos manuales, visión por computadora.

1 Introduction

According to the World Health Organization (WHO) in 2018 [1], over 6.1% of the world's population, around 466 million people, has disabling hearing loss; 432 million adults (93%) and 34 million children (7%). Approximately one third of people over 65 years of age are affected by disabling hearing loss. It's estimated that by 2050 over 900 million people, one in every ten people, will have disabling hearing loss. Projections were calculated based on the WHO 2008 prevalence estimates.

Most people with disabling hearing loss live in low- and middle-income countries. México is one of the countries included in this groups. According to the 2018 Encuesta Nacional de Dinámica Demográfica (ENADID) results [2], In México 6% (7.1 million) have disabilities where 33% (2.4) million have disabling hearing loss.

Mexico is a developing country. Special education and inclusion in Mexico has been increasing over the last years. It started with an inclusive education movement based on the special education system. In 1971 the educational reform proposed to create the general direction of special education (Dirección General de Educación Especial, DGEE) [3]. This entity was responsible for implementing an educational subsystem to support people with disabilities and outstanding capacities. In 1996, a national project for educational integration in the country was proposed [3].

The results were not encouraging. The study revealed deficiencies in the inclusion methods. Mexico is still so far from having a success inclusion and reasons can be categorized in two main points: teacher training on inclusion techniques and the tools used in the process. If we focus on inclusion tools, we notice that enough solutions do not exist which allow communication with deaf people. Current tools tested in Mexico consist of electronic devices based on instrumented gloves with flexion sensors [4].

However, in the end, they are tools that involve a cost for a society that, in most cases, does not have sufficient resources to implement these solutions. On the other hand, society is moving towards new technologies. The main attributes are: practicality, simplicity and low cost.

So we understand there exists a need to design devices to translate sign language into a regional language. This article is intended to be the basis for training deep learning models in the future. As we know, one of the requirements to train models is to have a reliable dataset based on different people and images in multiple environments. Images shown in this article were not captured in a controlled environment. This improves assertiveness in recognizing gestures by getting closer to reality.

In the following sections we will delve into the state of the art and compare the related works through a table. Later, we will describe in detail our work in the dataset section, the main challenges and advantages against other datasets. We will detail the

Table 1. Sign language datasets used in recent articles.

Name	Dataset Type	Language	Details
LSA64 [8]	video	Argentinian	64 signs
ASLLVD [9]	video	American	3000 signs
DSGS [12]	video	Swiss German	100 lexical items
ISL-HS [13]	video	Irish	23 hand-shapes
MU HandImages [10]	images	American	36 gestures
BdSL [11]	images	Bangladesh	36 characters
ArSL2018 [14]	images	Arabic	32 signs
THISL [16]	images	Indian	26 gestures
MSL [15]	video	Spanish	249 words

necessary steps followed in evaluating the dataset in a deep learning model. Finally, we will conclude the paper by talking about future work.

2 Related Work

Sign language image recognition begins with one of the early pioneers Jason Isaacs who worked on pattern classification for American Sign Language (ASL) using the wavelet transform [5]. Sign language is a non-verbal language used by deaf people to communicate with others which involves challenges and difficulties for researchers [6]. In addition, people usually use more than one sign to represent the same word. This causes conflicts and little progress in standardization, even within the same region. Despite the government's effort to include deaf people in society, unfortunately very few people can communicate in sign language. Solutions such as the use of interpreters is limited to the high cost that makes difficult to implement it in a country like Mexico.

Commonly deaf people have trouble reading and writing. Therefore, inclusion towards the digital age is not easy for them. An example of the above is multimedia content. It is easy to notice the increase in the number of videos.

However, the content is focused on hearing people. To improve communication between deaf people and hearing people, researchers are needed to develop sign recognition systems. The goal is generating mobile devices accessible to all types of people. In Mexico there is no comprehensive solution. Some of the current work is limited to applications that detect, for example, junction points on the hands [7]. Hence, we are still far from a robust comprehensive sign language recognition system.

A previous step to the development of these applications is the generation of a dataset. Machine learning needs to be powered by thousands of images. In addition to the volume in the number of images, signs involve variability in captures such as lighting conditions, posture, participants, environment, angles, and rotation [8]. Additionally we need to append an extra variable, the language. Table 1 provides an overview of the main latest works.



Fig. 1. Images of MSL static dataset for Spanish alphabet.

These works can be classified into two categories: video and image datasets. Video datasets generally are focused on detecting words rather than static signs. In most cases RGB conventional cameras are used as the capture source [8-12,14]. Other solutions use depth cameras like those of Kinect [12]. In most cases, images datasets contribute as an initial effort for video datasets. Then, the first step is to create a static dataset. Our initial effort is to replicate the work proposed in [16] where they create a two hand indian sign language (THISL) dataset.

3 Dataset

Currently datasets generated by researchers around the worlds are focus on the ASL for image. There are very few public datasets that use the MSL dictionary as a reference. We generated a dataset for object detection which contains 10,000 images based on 21 static gestures of the MSL alphabet. The images have a dimension of 1920x1080 pixels. They are 24-bit deep images stored in JPG format with an approximate weight of 270KB each. Only 21 out of 27 signs (letters) were included as part of the MSL dataset. It is because some letters require hand movement and face pose to generate the sign.



Fig. 2. Letter A from the MSL dataset performed by a girl.

MSL dataset was generated using both hands as depicts in Figure 1. In the first collage image, the fingers are positioned on the palm and only the thumb is open pointing to one side. This represents the letter A in the MSL. Similarly, each of the subsequent images represents a specific sign of the alphabet. These images were taken on multiple locations and different environments.

The image capture process was implemented through cycles. Each cycle consisted of capturing twenty-five images with the right hand and twenty-five images with the left hand from a person, making a single sign, in a previously selected environment. 10 environments in total. Each of them with distinct attributes, highlighting the following: natural versus artificial light, vegetation versus furniture, paintings and home decoration, background with stone wall versus background with a plain wall in a single color and two colors, windows with bars, light exposure tiers, and objects on the wrist. Inclination and rotation was forced in both hands to ensure the sign is recognized independently of the angle of the hand. The idea was to include the natural hand movement. Six people in total participated remotely building this dataset. Images were stored into a cloud-based repository where later the images were downloaded for training.

The twenty one images displayed in Figure 1 are a sample of the full MSL dataset. They are cropped to display only the hand-gesture. However, the original images cover more space than just the one used by the hands, as shown in figure 2.

We ran into difficulties when we generated the MSL dataset. At the beginning, it was not possible to know precisely the moment when an image was captured.

That generated displaced, blurred or poorly formed images as well we faced multiple misunderstandings capturing gestures because people were mistaken for the sign. Also, after some time capturing gestures, we lost the image count.

Hence, additionally we implemented a custom solution to capture images through a solution based on Python programming language. It's a simple algorithm that triggers image captures using configurable parameters. e.g. capture time slot, total required captures and a box to display the current gesture and hand side. Each captured image was stored into a custom folder using the gesture name as reference.

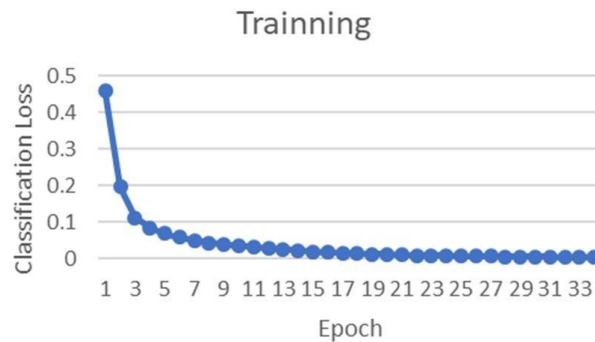


Fig. 3. Classification loss versus 33 epochs implemented with the resnet152 as backbone in the training over the MSL dataset.

The dataset is focused on object detection. Images themselves contain a hand-gesture from the Mexican alphabet. However, it is necessary to know the location of the gesture within the image in the training stage. In our case, each image is linked to an annotation file saved as XML that includes bounding box detailed information. The selected format for annotation files is Pascal VOC. Unlike the COCO annotation format [17], It generates an annotation file per image which makes it easier to grow and maintain the dataset. Additionally, the image label corresponding to represented hand-gesture is also included in the file. The annotations were made using the LabelImg [18] tool which is a graphical solution created in Python and the open-source widget toolkit for creating graphical user interfaces QT [19].

4 Experiments and Results

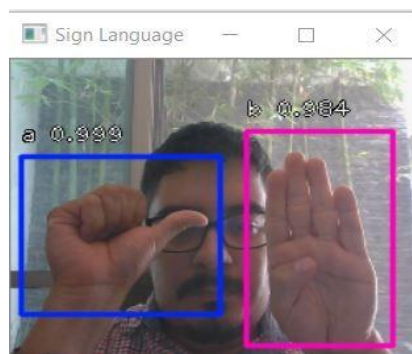
A model was trained using RetinaNet, presented by Facebook AI Research (FAIR) [20]. RetinaNet is one of the most recent deep learning models. It allows object detection in real time. In the same way as other models such as YOLOV3 or Mask RCNN, RetinaNet exhibits a robust ability to identify objects in a single stage. Moreover, RetinaNet shows a significant accuracy rate along with higher detection rate [21].

It was implemented on a virtual machine hosted in the cloud. It contains eight Tesla V100 cards destined to the high-performance computing (HPC) field. Each of these graphic processing units has 5,120 Compute Unified Device Architecture (CUDA) cores capable of developing a Power of 7.5 Floating point operations per second (TFLOP) in double-precision floating-point format (FP64). The full capacity of the virtual machine is 768GB of RAM.

The training consisted on 33 iterations of approximately 43 minutes each. To train the model, the MSL dataset was divided into two parts; the first with 9,000 images for training and the second with 1,000 images used for validation. The training was performed using a deep residual neural called resnet152 [23] that is based on convolutional neural networks (CNN) with 71,511,192 configurable parameters in total. CNN are especially useful for locating patterns in images in order to recognize

Table 2. First epoch results using distinct backbones to train a deep learning model using the MSL dataset.

Backbone	Color/Gray	Epoch Time	Loss	Classification Loss
resnet50	color	25:34	1.6276	0.4773
resnet50	gray	25:33	1.6257	0.4684
resnet152	color	42:39	1.5623	0.4585
vgg16	color	29:39	3.2998	2.1012
densenet201	color	1:37:40	2.1113	0.5945

**Fig. 4.** Custom implemented application to predict hand-gesture using a webcam.

objects, hand-gestures in our case. In the end, the accuracy achieved after 33 epochs was 99.46% as depicts in Figure 3.

At the beginning of the training, we tried out this MSL dataset with distinct backbones. Only one epoch was executed due to time limitation with the server. This is because, training these kind of models like RetinaNet demand a large amount of resources that a local machine does not have. So we decided to use a cloud solution with a dedicated machine containing the necessary resources. Unfortunately, these are high cost solutions. Hence, a single epoch was executed for each distinct backbone, except for the selected resnet152 backbone. In this case, the model was trained using 33 epochs.

During the training we worked with the RetinaNet implementation in TensorFlow 2.0 and Keras as described in [20]. This is a work-in-progress solution found on the GitHub site under the name tf-retinanet [22]. The input parameters used in the training are: the folder where the MSL dataset is located, the folder containing the annotation files, a file with the twenty one classes, and the name of the backbone. Also, within this library there is a Python method that we used to predict hand-gestures on an image. The method returns a tuple with three values which are: boxes, scores, and labels.

For a practical evaluation process of the MSL dataset, a Python application was developed which recognizes gestures through a web camera in real time using the OpenCV library. Regarding the response time on a local machine, for each prediction of a sign, it requires an average time of 200 milliseconds. The application is capable of

recognizing multiple gestures at the same time on the same frame and showing the name of the sign as well as the accuracy, as depicts in Figure 4.

The backbones we tested include: resnet50, resnet152, vgg16 and densenet201. They were selected based on the work in [24]. Additionally, we applied an extra step to test the model using gray-scale images. The one epoch training results are displayed in Table 2.

In such results, we can note the training with gray-scale images versus training with color images does not make much difference. The one epoch elapsed time and loss value are pretty similar in both scenarios. However, we can notice a difference when we use distinct backbones. We got the best loss value in the training using the resnet152 backbone. Although time increased significantly, we see better performance in loss value. Opposite to this result was the vgg16 backbone where the classification loss value was the highest. Therefore, it was ruled out to continue using it in training.

5 Conclusions

The desire to have an MSL dataset to train deep learning models led us to build ours. In this paper we evaluated the two-hand static MSL dataset specialized in object detection using RetinaNet model with distinct backbones. Despite the different positions and angles in the images, the accuracy of the model greater than 99%.

The relevance of generating an MSL dataset lies as part of a contribution to the scientific community of artificial intelligence, so that, they can take this presented dataset as reference, train object detection models and materialize its contribution through applications aimed at the benefit of the Mexican deaf community.

The MSL dataset is planned to be uploaded to a repository that researchers have access to. Likewise, it is intended to make the code of the applications and libraries used in this research available to the community.

It has been demonstrated that it is possible to use object detection models applied to the MSL with positive results. It is possible to predict hand-gestures through low-cost devices such as webcams. It was shown that the prediction of these gestures can be performed in real time on not controlled environments with distinct attributes. Finally, the MSL dataset is able to work with child or adult gestures, different skin color, the right or left hand, and accessories used on them.

6 Future Work

In this preliminary work, the MSL dataset contains only 21 of the 27 static MSL gestures. The reason is because to generate the other 6 MSL gestures (J, K, Ñ, Q, X, Z), hand movement is required. Current research work is planned to continue in future developments. One of the goals is to complement the MSL dataset to recognize dynamic gestures in real time. It is intended to continue working in not controlled environments. An initial idea is to use long short-term memory (LSTM) by adapting an artificial recurrent neural network (RNN) architecture to our model.

We aspire to integrate the entire project into low-cost devices such as mobile phones, tablets, laptops, or even Google Glass. The integration of these devices in Mexican society will benefit the inclusion of people with hearing loss.

References

1. Deafness and Hearing Loss: <https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss> (2018)
2. National Institute of Statistics and Geography (INEGI): Mexico National Survey of Demographic Dynamics. <https://en.www.inegi.org.mx/programas/enadid/2018/> (2018)
3. Romero, S., García, I., Chris, F., Lomeli, K.A.: Preparing Teachers for Inclusion in Mexico. How Effective is this Process?. *Journal of Education for Teaching*, 39(5), pp. 509–522 (2013). Doi: 10.1080/02607476.2013.836340.
4. García, F., Tinoco, L., González, H.G., García, M.J., Moreno, J.: Modelado e implementación de un sistema de enseñanza de lenguaje de señas mexicano. *Ciencia ergo-sum*, 14(2), pp. 185–190 (2007)
5. Isaacs, J., Foo, S.: Hand Pose Estimation for American Sign Language Recognition. In: Thirty-Sixth Southeastern Symposium on System Theory, pp. 132–136 (2004). Doi: 10.1109/SSST.2004.1295634.
6. Von, A.U., Zieren, J., Canzler, U., Bauer, B., Kraiss K.F.: Recent Developments in Visual Sign Language Recognition. *Univ Access Inf Soc*, 6, pp. 323–362 (2008). Doi: 10.1007/s10209-007-0104-x.
7. Martínez, M.E., Rojano, J.R., Benitez, E., Sánchez, H.E.: Data Acquisition Software for Sign Language Recognition. *Research in Computing Science*, pp. 205–2011 (2019). Doi: 10.13053/rcs-148-3-17.
8. Ronchetti, F., Quiroga, F., Estrebow, C., Lanzarini, L., Rosete, A.: LSA64: An Argentinian Sign Language Dataset. Instituto de Investigación en Informática LIDI (2016). Doi: 10.48550/arXiv.2310.17429.
9. Neidle, C., Thangali, A., Sclaroff, S.: Challenges in Development of the American Sign Language Lexicon Video Dataset (ASLLVD) Corpus. *IEEE Workshop on Computer Vision and Pattern Recognition for Human Communicative Behavior Analysis (CVPR4HB)*, pp. 1–8 (2008)
10. Barczak, A.L., Reyes, N.H., Abastillas, M., Piccio, A., Susnjak, T.: A New 2D Static Hand Gesture Colour Image Dataset for ASL Gestures. *Res. Lett. Inf. Math. Sci.*, 15, pp. 12–20 (2011)
11. Islam, M., Sultana, S.S., Jessan, N., Rabby, A.S., Hossain, S.: Ishara-Lipi: The First Complete Multipurpose Open Access Dataset of Isolated Characters for Bangla Sign Language. *International Conference on Bangla Speech and Language Processing (ICBSLP)*, pp. 1–4 (2018). Doi: 10.1109/ICBSLP.2018.8554466.
12. Ebling, S., Camgoz, N., Braem, P., Tissi, K., Sidler, S., Stoll, S., Hadfield, S., Haug, T., Bowden, R., Tornay, S., Razavi, M., Magimai, M.: SMILE Swiss German Sign Language Dataset (2018)
13. Oliveira, M., Chatbri, H., Ferstl, Y., Farouk, M., Little, S., O'Connor, N., Sutherland, A.: A Dataset for Irish Sign Language Recognition. *Proceedings of the Irish Machine Vision and Image Processing Conference (IMVIP), International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pp. 1–8 (2017)
14. Ghazanfar, L., Nazeeruddin M., Jaafar, R., Rawan, A.: ArASL: Arabic Alphabets Sign Language Dataset. *Data in Brief*, 23 (2019). Doi: 10.1016/j.dib.2019.103777.

15. Cervantes, J., García, F., Rodríguez, L., Rendón, A., Lopez, A.: Recognition of Mexican Sign Language from Frames in Video Sequences. Huang and K.-H. Jo: ICIC Part II, LNCS 9772, pp. 353–362 (2016)
16. Teja L.S., Jain L., Sharmay A.: Two Hand Indian Sign Language Dataset for Bechmarking Classification Models of Machine Learning (2019). Doi: 10.1109/ICICT46931.2019.8977713.
17. COCO and Pascal VOC Data Format for Object Detection: <https://towardsdatascience.com/coco-data-format-for-object-detection-a4c5eaf518c5> (2017)
18. Tzutalin: Label Img. Git Code. <https://github.com/tzutalin/labelImg> (2015)
19. QT Open-Source: Widget Toolkit for Creating Graphical User Interfaces as Well Ascross-Platform Applications. <https://www.qt.io/> (2018)
20. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal Loss for Dense Object Detection. arXiv preprint (2017). Doi: 10.48550/arXiv.1708.02002.
21. Pradipta, A., Imon A.: Performance Comparison of Different CNN Models for Indian Road Dataset. ICGSP 3rd International Conference on Graphics and Signal Processing (2019). Doi: 10.1145/3338472.3338480.
22. TF RetinaNet: GitHub. <https://github.com/fizyr/tf-retinanet> (2020)
23. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778 (2016). Doi: 10.1109/CVPR.2016.90.
24. Bressen, K., Adams, L., Erxleben, C., Hamm, B., Niehues, S., Vahldiek, J.: Comparing Different Deep Learning Architectures for Classification of Chest Radiographs. arXiv. (2020). Doi: 10.48550/arXiv.2002.08991.

The HAJ: Initial Steps of a Novel Cryptographic Algorithm

Daniel Alarcón-Narváez¹, Fausto A. Jacques-García², L.A. Lizama-Pérez^{2,*},
S.L. Canchola-Magdaleno¹, R.C. Chaparro¹

¹ Autonomous University of Queretaro, Computer Science, Querétaro,
Mexico

² Universidad Autónoma Metropolitana, Basic Sciences and Engineering,
Mexico

*l.lizama@correo.1er.uam.mx, daniel.alarcon@uaq.mx

Abstract. We present the Hill-Alarcón-Jacques (HAJ) algorithm, a symmetric cryptographic construction that combines the classical Hill cipher with the Gauss-Jacques method for modular matrix inversion. The principal novelty of this initial HAJ study is the integration of a matrix-based encryption/decryption process with a modular elimination procedure for obtaining the inverse matrices required during decryption. Particular attention was paid to the use of large matrix dimensions, as matrix size directly influenced the computational cost of the proposed approach. The construction incorporated multi-round encryption, block-based processing, dynamic subkey generation, and flexible matrix dimensions. The experimental evaluation considered both full-size matrices and block-based configurations, using national-anthem plaintexts and thirty runs per configuration across desktop, macOS, and MATLAB Online environments. The results showed comparatively stable memory consumption for a given configuration, whereas execution time varied according to plaintext length, matrix dimension, block size, and computational platform. These experiments provided an initial empirical characterization of HAJ and established a computational baseline for analyzing its scalability and resource requirements. The results represented the initial algorithmic and experimental stage of a broader investigation into matrix-based cryptographic primitives, while also identifying cryptanalytic and theoretical questions that remained to be addressed before stronger security claims could be established.

Keywords: Gauss-Jacques method, Hill cipher, modular inverse calculation, security, symmetric cryptography, matrix-based cryptography, lightweight cryptography.

El HAJ: Primeros pasos de un nuevo algoritmo criptográfico

Resumen. Presentamos el algoritmo Hill-Alarcón-Jacques (HAJ), una construcción criptográfica simétrica que combina el cifrado clásico de Hill

con el método de Gauss-Jacques para la inversión de matrices modulares. La principal novedad de este estudio inicial sobre HAJ radica en la integración de un proceso de cifrado y descifrado basado en matrices con un procedimiento de eliminación modular para obtener las matrices inversas requeridas durante el descifrado. Se prestó especial atención al uso de matrices de grandes dimensiones, dado que el tamaño de la matriz influía directamente en el coste computacional del enfoque propuesto. La construcción incorporó cifrado de múltiples rondas, procesamiento por bloques, generación dinámica de subclaves y dimensiones matriciales flexibles. La evaluación experimental consideró tanto matrices de tamaño completo como configuraciones basadas en bloques, utilizando textos en claro consistentes en himnos nacionales y realizando treinta ejecuciones por configuración en entornos de escritorio, macOS y MATLAB Online. Los resultados mostraron un consumo de memoria comparativamente estable para una configuración dada, mientras que el tiempo de ejecución varió en función de la longitud del texto en claro, la dimensión de la matriz, el tamaño del bloque y la plataforma computacional. Estos experimentos proporcionaron una caracterización empírica inicial de HAJ y establecieron una línea base computacional para analizar su escalabilidad y sus requisitos de recursos. Los resultados representaron la etapa algorítmica y experimental inicial de una investigación más amplia sobre primitivas criptográficas basadas en matrices, al tiempo que identificaron cuestiones criptoanalíticas y teóricas pendientes de abordar antes de poder establecer afirmaciones de seguridad más sólidas.

Palabras clave: Método de Gauss-Jacques, cifrado de Hill, cálculo de la inversa modular, seguridad, criptografía simétrica, criptografía basada en matrices, criptografía ligera.

1 Introduction

Cryptography is the study and practice of techniques for securing communication in the presence of unauthorized third parties. In modern cryptography, there are two main types of encryption: symmetric cryptography and asymmetric cryptography [1].

Symmetric cryptography, also known as secret-key cryptography, uses a shared key between the sender and receiver to encrypt and decrypt the message. The security of this method depends on the shared key being known only to the sender and receiver. Examples of symmetric cryptography algorithms are DES (Data Encryption Standard), AES (Advanced Encryption Standard), and the Hill-Cipher symmetric algorithm [2].

On the other hand, asymmetric cryptography, also known as public-key cryptography, uses a pair of keys: a public key and a private key. The public key is shared with anyone who wants to send a message to the owner of the private key. The owner uses their private key to decrypt the message. Examples of asymmetric cryptography algorithms are RSA (Rivest-Shamir-Adleman), Diffie-Hellman, and ECC (Elliptic Curve Cryptography) [2].

Both types of encryption have their advantages and disadvantages and are used in different situations. The choice of the appropriate encryption method depends on the nature of the information to be encrypted and the security needs of the system.

In recent years, the design of lightweight cryptographic algorithms has gained considerable attention, especially for Internet of Things (IoT) and resource-constrained environments. Comprehensive evaluations of symmetric ciphers for IoT show that algorithmic efficiency and robustness are equally critical, as trade-offs between security and performance determine practical deployment [3,4]. These works highlight the need for symmetric ciphers that combine mathematical rigor with computational feasibility.

One of the main challenges in Hill-type ciphers is their vulnerability to linear cryptanalysis and known-plaintext attacks. Several modern studies have focused on analyzing and breaking variants of the Hill cipher. For example, Wen et al. [5] demonstrated that Hill-based schemes combined with chaotic systems can still be attacked using statistical analysis. Similarly, Teseleanu [6] analyzed a Hill-plus-chaos variant and showed weaknesses when subjected to standard cryptanalytic techniques. These findings suggest that any novel Hill variant must be thoroughly evaluated against both classical and modern attacks.

At the same time, mathematical advances in modular inverse computation remain relevant for cryptographic constructions. Bufalo [7] provided a detailed analysis of modular inverse properties and highlighted their applications in cryptography. Recent developments also include efficient methods for inversion modulo powers of integers, such as those presented by Xu et al. [8], which illustrate the ongoing interest in optimizing matrix inversion techniques for cryptographic use cases.

Finally, new designs continue to explore robust Hill-based algorithms. Rrgh-out et al. [9] proposed an image encryption algorithm based on a novel Hill variant, showing the community's continuous exploration of this family of ciphers.

However, the existing literature indicates that security analysis—such as resistance to differential and linear attacks, as well as statistical tests of randomness—is often missing or insufficient, which motivates the present work. In this context, we introduce a new symmetric cryptographic algorithm called Hill-Alarcon-Jacques (HAJ), which, to the best of our knowledge, represents the first attempt to combine the Hill cipher with the Gauss-Jacques method for modular inverse calculation. This novel integration has not been explored in prior works, making HAJ a unique and previously unreported contribution to the field of symmetric cryptography.

The remainder of this paper is organized as follows. Section 2 describes the computational environment. Section 3 introduces the theoretical background, including the classical Hill cipher and the Gauss-Jacques modular inverse method. Section 4 details the methodology of the proposed Hill-Alarcón-Jacques (HAJ) algorithm. Section 5 presents the algorithm pseudocode. Section 6 reports both

performance results. Section 7 concludes the paper and discusses future research directions.

2 Materials

1. Equipment used:
 - Desktop HP computer with the following specifications. Processor: 11th Gen Intel(R) Core(TM) i3-1115G4 @ 3.00GHz 2.90 Ghz RAM: 8 GB Operating System: Windows 11 Home.
2. Equipment used:
 - Desktop Macbook Air computer with the following specifications: Processor: 11th Gen Intel(R) Core(TM) i3-1115G4 @ 3.00GHz 2.90 Ghz, RAM: 8 GB, Operating System: macOS Monterrey.
3. Software used:
 - MATLAB (Online Trial Version): The online version of MATLAB provided by MathWorks was utilized for performing numerical computations, implementing algorithms, and visualizing results. It was accessed through a compatible web browser, and the programming and data analysis capabilities of the MATLAB environment were leveraged.
 - MATLAB version R2021a (Free Desktop Version): In addition to the online version, the free desktop version of MATLAB was also employed. Specifically, version R2021a was installed on the local computer and used for code development, debugging, conducting experiments, and performing additional analysis.

3 Theoretical Framework

Our proposed algorithm follows the general principles of the Hill cipher. It involves dividing the plaintext into blocks, each of which is represented as a matrix. These matrices are then multiplied by a secret key matrix, resulting in the ciphertext.

The key innovation lies in the calculation of the matrix inverse using the Gauss-Jacques method, which offers a more efficient and modular approach compared to traditional methods. By utilizing modular inverse calculation, our algorithm ensures the encryption process remains secure even with a relatively small key size. All the theoretical information is found in the article Towards a Symmetric Crypto Algorithm: The HAJ [11]. The proposed HAJ algorithm, while focused on enhancing cryptographic robustness, also has the potential to be adapted for lightweight environments such as the Internet of Things (IoT), due to its modular structure and efficient matrix-based operations. In resource-constrained environments, encryption algorithms must ensure both computational efficiency and cryptographic strength. A recent approach proposes a lightweight symmetric cryptosystem based on AES and a chaotic S-box, designed for embedded platforms. Using techniques like the Lorenz system and Hilbert scan patterns, the system achieved validated cryptographic properties through NIST tests and demonstrated efficient performance on real-world sensors [12].

3.1 Hill Cipher

The Hill-Cipher algorithm is a linear substitution encryption algorithm that uses a matrix as a key to encrypt and decrypt the message. It was proposed by Lester S. Hill in 1929 [13], and is based on the key matrix, which is an invertible square matrix. The Hill Cipher algorithm is used in data security applications to protect information from malicious attacks. This algorithm is considered one of the first symmetric encryption algorithms and has been used in military and government applications for decades. However, its use has decreased due to the arrival of more secure and efficient algorithms [14].

In the Hill Cipher, the original message is divided into blocks of length n , where n is the size of the key matrix. Each block is multiplied by the key matrix, and the result is reduced modulo a prime number to obtain the encrypted message. The decryption process involves multiplying the encrypted message by the inverse matrix of the key and reducing modulo the prime number.

Below are some relevant experiments and references related to the Hill Cipher:

Experiment 1: Known-plaintext attack In this experiment, an attacker knows some blocks of plaintext and the corresponding blocks of ciphertext. Using this information, the attacker can calculate the key matrix by using linear algebra and decrypt the rest of the ciphertext message [16].

Experiment 2: Known-ciphertext attack In this experiment, an attacker knows some blocks of ciphertext and the corresponding blocks of plaintext.

Using this information, the attacker can calculate the inverse matrix of the key by using linear algebra and decrypt the rest of the ciphertext message [18].

Experiment 3: Frequency analysis In this experiment, an attacker can use frequency analysis to decrypt the ciphertext message. The attacker can use the frequency of letters in the ciphertext message to determine the most common letters and, from there, deduce the key and decrypt the entire message [16].

Experiment 4: Side-channel attack Side-channel leakages, such as electromagnetic emissions, can significantly reduce the security margin of symmetric encryption algorithms. Recent work proposed a key classification framework based on deep transfer learning, using a divide-and-conquer strategy over semi-byte structures to reduce key enumeration complexity. The model, composed of symmetric subnetworks and an end-to-end mapping design, achieved up to 91% attack accuracy while maintaining robustness under small-sample conditions [3].

Although the Hill Cipher is a solid encryption algorithm, it is not invulnerable to attacks. The experiments mentioned above demonstrate some of the vulnerabilities of the Hill Cipher and the importance of choosing an appropriate key matrix and ensuring that the prime number used for modulo reduction is large enough. Additionally, it is important to remember that encryption algorithms are one of the many tools used in data security, and data security depends on a combination of technical security measures and management procedures.

3.2 Gauss-Jacques Method for Modular Inverse Calculation

The Gauss-Jacques method plays a crucial role in our proposed symmetric cryptographic algorithm for modular inverse calculation. This method allows us to efficiently compute the modular inverse of variable-sized matrices, making it particularly suitable for cryptographic applications.

The Gauss-Jacques method is implemented as a separate function, taking two inputs: the matrix 'K' and the prime number 'm'. It ensures that 'K' is a square matrix, and if not, it raises an error. The function returns two outputs: the modular inverse matrix 'InvMod' and the identity matrix 'I'.

The algorithm proceeds as follows:

- The original matrix 'K' is augmented with the identity matrix to form an augmented matrix.
- For each row in 'K', a pivot element is identified. If the pivot is zero, a row interchange operation is performed to ensure a non-zero pivot.
- The extended Euclidean algorithm is used to calculate the modular inverse 'x' of the pivot element with respect to 'm'.
- The current row of 'K' is multiplied by 'x' modulo 'm' to obtain the updated row.
- For each row other than the current row, row operations are performed to eliminate the non-zero elements below and above the pivot element.
- Finally, the resulting matrix is divided into two parts: 'InvMod', which contains the modular inverse matrix, and 'I', which represents the identity matrix.

3.3 Extended Euclidean Algorithm

The extended Euclidean algorithm is a fundamental tool for finding the greatest common divisor (gcd) of two integer numbers in a Euclidean space.

It is used within the Gauss-Jacques method to calculate the modular inverse of the pivot element.

The extended Euclidean algorithm function takes two positive integers, 'a' and 'b', as inputs. It returns three integers, 'x', 'y', and 'd', such that 'd' represents the gcd of 'a' and 'b', and $ax + by = d$.

The algorithm proceeds as follows:

- If 'b' is equal to zero, the function sets 'x' to 1, 'y' to 0, and 'd' to 'a' before returning.
- Otherwise, the function recursively calls itself with 'b' and 'mod(a,b)' as the new inputs, storing the results in 'x1', 'y1', and 'd1'.
- 'x' is set to 'y1', 'y' is set to 'x1 - floor(a/b)*y1', and 'd' is set to 'd1'.

4 Methodology

The HAJ algorithm retains the Hill cipher's block-matrix multiplication as the core encryption operation. The novelty lies in the computation of the modular

inverse of each key matrix: instead of the standard adjugate/determinant approach, HAJ applies the Gauss-Jacques elimination method to obtain the inverse modulo m . During each round, a fresh key matrix is generated and its inverse computed by Gauss-Jacques, enabling efficient decryption and supporting the multi-round, subkey structure.

4.1 A Round of HAJ

The previous section describes the generalization for creating and understanding how the HAJ algorithm would work, i.e., the summation of the blocks and the 7 times it will be performed. Now, in the following paragraphs, we will describe how we intend to generate each block, the bits that will be used, the size of each block, and how the primary and secondary keys will be generated.

Obtaining the String Size There are different functions to know the size of a string, where any of them will be used to find that value. What is interesting in this section is that we seek for that value to be a multiple of three hundred. Therefore, we will use the modulo 300 of the string length to verify it, and if necessary, some characters will be added to the string to make it a multiple of three hundred.

Example code:

```
if s % 300 != 0
    a = 300 - s;
    for i = 1:a
        p(lp+i) = 'Q';
    end
end
```

Key Generation The primary key will be generated using the Gauss-Jacques method, where the size of the matrix will be determined by the size of the resulting block in the previous steps. However, for each block, a primary key will be generated, and for each round of the 7 rounds, secondary or sub-keys will be generated to improve security at each iteration.

For the secondary keys, we will use the left circular shift. A circular shift in this case to the left is the operation of rearranging the entries in a string, either by moving the final entry to the first position while shifting all other entries to the next position, or by performing the inverse operation. A circular shift is a special type of cyclic permutation, which in turn is a special type of permutation.

This operation is used in various fields and different algorithms. An example of this in cryptography is the Data Encryption Standard algorithm, where positions are moved from the order of the primary key with respect to a shift list determined beforehand.

4.2 Encryption and Decryption Process

To demonstrate the functionality of our proposed symmetric cryptographic algorithm, we implemented the encryption and decryption processes using the provided code. The algorithm utilizes the Hill cipher with modular inverse calculation using the Gauss-Jacques method.

For encryption, the algorithm requires the selection of a prime number, denoted as 'm', greater than the highest ASCII value intended for encryption. Additionally, the number of matrices, 'num_matrices', and the block size, 'block_size', need to be defined. The matrices are generated randomly, ensuring that the determinant is non-zero modulo 'm'. This approach aligns with recent advancements in symmetric key generation that integrate hybrid models based on Latin square matrices and subtractive random number generators to enhance key randomness, diffusion, and resistance to brute-force attacks [2].

The plaintext is obtained from the user and is processed in blocks. If the length of the plaintext is not a multiple of the block size, padding is applied to make it a multiple. The encryption process involves multiplying the block of plaintext with the corresponding key matrix, ensuring modulo 'm' arithmetic.

The decryption follows a similar process but uses the inverse of the key matrix calculated using the Gauss-Jacques method implemented in MATLAB [20]. The encrypted text is processed in blocks, and the decryption is performed by multiplying each encrypted text block with the inverse matrix modulo 'm'.

The resulting decrypted plaintext is displayed as the final output of the algorithm.

The algorithm 4.3 shows the pseudocode for a round of HAJ where the K is the same size as plaintext. For example, if the plaintext is about 170 characters, then the K is going to be 170x170 size and the algorithm 5 shows the pseudocode by blocks.

4.3 Pseudocode of the HAJ Round

In this section, we present a pseudocode of the HAJ Round algorithm.

Listing 1.1. Pseudocode for text encryption and decryption:

```
m = 257;
seeds = [36, 33, 88, 91, 1];

File = 'NationalAnthems/NameAnthem.txt';

fid = fopen(File, 'rb');
p = fread(fid, 'uint8');
fclose(fid);

lonp = length(p);

r = mod(lonp, 3);
```

```

if  $r \sim = 0$ 
    pad = 3 - r;
    p = [p ; zeros(pad,1)];
end

results = zeros(30,4);

for i = 1:length(seeds)

    seed = seeds(i);
    rng(seed);

    n = 3;
    k = randi([0 m-1], n, n);

    c = mod(k * reshape(p(1:3),3,1), m);

    InvMod = modular_inverse(k, m);
    p_decrypted = mod(InvMod * c, m);

end

writematrix(results, 'results.csv');

```

5 The HAJ: Pseudocode

In this section, we present pseudocode of the HAJ algorithm.

Listing 1.2. Pseudocode for text encryption and decryption with multiple rounds:

```

m = 257 # Modulus for encryption
seeds = [280, 494, 50, ..., 404, 181] # List of 30 seeds
File = 'NationalAnthems/NameAnthem.txt' # Path to the text file

# Read the text file
file = open(File, 'rb')
p = file.read()
file.close()

# Ensure the length of p is a multiple of 300
if len(p) % 300 != 0:
    # pad p with zeros

# Prepare the results array
results = [[0 for _ in range(4)] for _ in range(len(seeds))]

```

```
numbloque = 3
tamanho_bloque = len(p) // numbloque

for j in range(len(seeds)):
    # Initialize random seed
    seed = generate_random_seed()
    initialize_random_number_generator(seed)

    for ronda in range(num_rondas):
        for i in range(numbloque):
            k = generate_random_matrix(tamanho_bloque, tamanho_bloque)

            encryption_file = open('encryption_results.txt', 'w')
            decryption_file = open('decryption_results.txt', 'w')

            # Encrypt block by block, 7 times
            c = []
            for ronda in range(num_rondas):
                for i in range(0, len(p), tamanho_bloque):
                    block = get_current_block(p, i, tamanho_bloque)
                    encrypted_block = encrypt_block(block)
                    c.append(encrypted_block)
                write_encryption_results_to_file(encryption_file, c)

            # Decrypt block by block, 7 times
            p_decifrado = []
            for ronda in range(num_rondas):
                for i in range(0, len(c), tamanho_bloque):
                    block = get_current_block(c, i, tamanho_bloque)
                    decrypted_block = decrypt_block(block)
                    p_decifrado.append(decrypted_block)
                write_decryption_results_to_file(decryption_file, p_decifrado)

            close_files(encryption_file, decryption_file)

            # Measure memory usage
            measure_memory_usage()

            # Store the results
            store_results(results, j)

# Save results to a CSV file
save_results_to_csv(results).
```

6 Experimental Results

To evaluate the performance and security of our algorithm, we conducted a series of experiments using various plaintexts and key sizes. The results demonstrate that our algorithm achieves robust encryption while maintaining computational efficiency.

6.1 Phase 1

The experimentation is carried out with ten national anthems, which are listed below:

1. Australia,
2. Canada,
3. Great Britain,
4. India,
5. Ireland,
6. Jamaica,
7. New Zealand,
8. Singapore,
9. South Africa,
10. USA.

These anthems were used for the first part of the experimentation. In this specific chapter, the experimentation consists of a single round of the HAJ algorithm. That is, encryption and decryption are performed only once. The particularity is that a matrix of the size of the plaintext is generated.

The length of each anthem is as shown in table 1.

For example, the national anthem of the USA consists of 1705 characters with all the current verses and choruses, without being processed through the algorithm to make it a multiple.

Therefore, a random matrix K of 1705×1705 elements ranging from 0 to 256 is generated, which is multiplied by the plaintext to generate the ciphertext. For this particular example described here, it would be the anthem of the USA. The tables 2, 3, and 4 show the average, standard deviation, maximum, and minimum values of the execution time expressed in seconds, as well as the memory usage represented in Mb, from the thirty runs performed for each anthem where the key is the same size as the plaintext.

6.2 Phase 2 Results

The anthems used for this phase are the same that we used on subsection 6.1. The difference is that the length of each anthem changed because we made them multiple of 300 for experimental purposes. The length of each anthem are as show table 5:

Table 1. Length of each anthem

Anthem	Length
Australia	620
Canadá	316
Gran Bretaña	496
India	336
Irlanda	800
Jamaica	591
Nueva Zelanda	1190
Singapur	204
Sudáfrica	226
USA	1705

The tables 6, 7, and 8 show the execution for each anthem dividing the plaintext into blocks of 3 and creating 21 keys the size of each block, this is because there is one key for each block and each round. The tables show as well the average, standard deviation, maximum, and minimum values of the execution time and memory usage.

The tables 9, 10, and 11 show the execution for each anthem dividing the plaintext into blocks of 30 and creating 210 keys the size of each block, this is because there is one key for each block and each round. The tables show as well the average, standard deviation, maximum, and minimum values of the execution time and memory usage.

The tables 12, 13, and 14 show the execution for each anthem dividing the plaintext into blocks of 300 and creating 2100 keys the size of each block, this is because there is one key for each block and each round. The tables show as well the average, standard deviation, maximum, and minimum values of the execution time and memory usage.

6.3 Comparison between the Performance of the Devices

In general, we can see that our algorithm is very stable in memory use. Each device, where the algorithm was executed, had very similar memory usage consumption.

The difference was in the execution time, with the MacBook Air being the device with the worst performance. The Windows device and the online version had very similar execution times.

Figure 1 show the average execution time and memory usage of the runs for a single round of the HAJ. Where the key is the same size as the plaintext. In

Table 2. Mac Runs Summary

Anthem	Parameters	Execution Time (s)	Memory Usage (MB)
Australia	Average	13.3370	5.9330
	SD	2.2320	0.0070
	Max	21.8156	5.9398
	Min	11.7482	5.9011
Canada	Average	2.2672	1.6134
	SD	0.1914	0.0070
	Max	2.7470	1.6202
	Min	2.0245	1.5815
Great Britain	Average	7.9106	3.8298
	SD	1.6691	0.0070
	Max	15.6925	3.8366
	Min	6.6204	3.7979
India	Average	2.5626	1.7630
	SD	0.1623	0.0068
	Max	2.9756	1.7697
	Min	2.3115	1.7323
Ireland	Average	27.7280	9.8431
	SD	3.9408	0.0070
	Max	39.7620	9.8499
	Min	24.6128	9.8112
Jamaica	Average	11.6885	5.3762
	SD	1.3711	0.0068
	Max	15.1518	5.3830
	Min	10.1736	5.3455
New Zealand	Average	92.7189	21.7070
	SD	18.0516	0.0070
	Max	149.5322	21.7138
	Min	82.6005	21.6751
Singapore	Average	0.7071	0.6921
	SD	0.1679	0.0070
	Max	1.3809	0.6989
	Min	0.5553	0.6602
South Africa	Average	0.7658	0.8320
	SD	0.0520	0.0070
	Max	0.9386	0.8388
	Min	0.7127	0.8001
USA	Average	280.4113	44.5373
	SD	59.3591	0.0070
	Max	576.2381	44.5441
	Min	255.1275	44.5054

Table 3. Runs Summary

Anthem	Parameters	Execution Time (s)	Memory Usage (MB)
Australia	Average	8.2658	5.9330
	SD	0.1805	0.0070
	Max	8.5023	5.9398
	Min	7.9580	5.9011
Canada	Average	1.0878	1.5841
	SD	0.0289	0.0070
	Max	1.1596	1.5909
	Min	1.0596	1.5522
Great Britain	Average	3.9785	3.8298
	SD	0.0473	0.0070
	Max	4.1067	3.8366
	Min	3.9137	3.7979
India	Average	1.2708	1.7630
	SD	0.0165	0.0068
	Max	1.3227	1.7697
	Min	1.2483	1.7323
Ireland	Average	17.6643	9.8431
	SD	0.3715	0.0070
	Max	18.3467	9.8499
	Min	17.0091	9.8112
Jamaica	Average	7.0676	5.3762
	SD	0.1952	0.0068
	Max	7.5738	5.3830
	Min	6.7943	5.3455
New Zealand	Average	61.2749	21.7070
	SD	2.0358	0.0070
	Max	64.1252	21.7138
	Min	57.9498	21.6751
Singapore	Average	0.3318	0.6921
	SD	0.0617	0.0070
	Max	0.6527	0.6989
	Min	0.3075	0.6602
South Africa	Average	0.4134	0.8321
	SD	0.0107	0.0070
	Max	0.4581	0.8388
	Min	0.4010	0.8001
USA	Average	198.1801	44.5373
	SD	11.1756	0.0070
	Max	213.1193	44.5441
	Min	176.1961	44.5054

Table 4. Online Runs Summary

Anthem	Parameters	Execution Time (s)	Memory Usage (MB)
Australia	Average	5.4928	5.9330
	SD	0.0870	0.0070
	Max	5.7460	5.9398
	Min	5.4067	5.9011
Canada	Average	0.9267	1.6134
	SD	0.0317	0.0070
	Max	1.0079	1.6202
	Min	0.8720	1.5815
Great Britain	Average	3.0410	3.8298
	SD	0.0336	0.0070
	Max	3.1122	3.8366
	Min	2.9755	3.7979
India	Average	1.0303	1.7630
	SD	0.0250	0.0068
	Max	1.0771	1.7697
	Min	0.9749	1.7323
Ireland	Average	11.6492	9.8431
	SD	0.2980	0.0070
	Max	12.6112	9.8499
	Min	11.3622	9.8112
Jamaica	Average	5.0064	5.3762
	SD	0.0669	0.0068
	Max	5.2021	5.3830
	Min	4.8887	5.3455
New Zealand	Average	56.5294	29.3796
	SD	1.1452	0.0070
	Max	59.4291	29.3864
	Min	54.8559	29.3477
Singapore	Average	0.2881	0.6921
	SD	0.0098	0.0070
	Max	0.3312	0.6989
	Min	0.2803	0.6602
South Africa	Average	0.3697	0.8321
	SD	0.0113	0.0070
	Max	0.4188	0.8388
	Min	0.3588	0.8001
USA	Average	103.6944	44.5373
	SD	3.7643	0.0070
	Max	110.1514	44.5441
	Min	98.1536	44.5054

Table 5. Length of each anthem for blocks

Anthem	Length
Australia	900
Canadá	600
Gran Bretaña	600
India	600
Irlanda	900
Jamaica	600
Nueva Zelanda	1200
Singapur	300
Sudáfrica	300
USA	1800

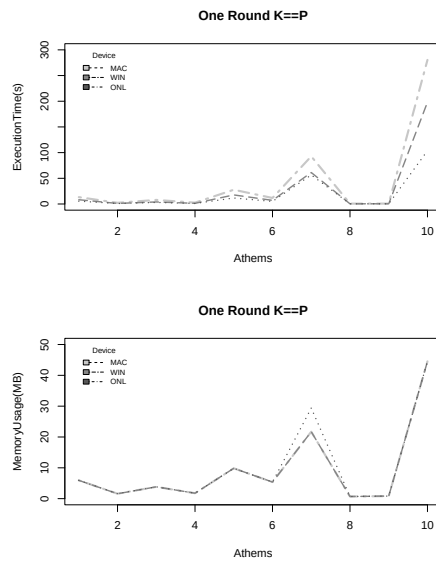


Fig. 1. Average accuracy across 30 runs.

Table 6. Runs per block (three) Mac.

Anthem	Parameters	Execution Time (s)	Memory Usage (MB)
Australia	Average	36.4588	15.8624
	SD	1.4329	0.0089
	Max	40.4950	15.8695
	Min	34.5879	15.8193
Canada	Average	11.5354	7.0801
	SD	0.5840	0.0089
	Max	13.0800	7.0873
	Min	10.9522	7.0371
Great Britain	Average	11.4738	7.0815
	SD	0.4413	0.0089
	Max	12.3882	7.0886
	Min	10.9528	7.0385
India	Average	11.5210	7.0803
	SD	0.5554	0.0089
	Max	13.4812	7.0874
	Min	10.9673	7.0372
Ireland	Average	36.5324	15.8637
	SD	1.2092	0.0089
	Max	39.1249	15.8709
	Min	34.5200	15.8207
Jamaica	Average	11.2850	7.0822
	SD	0.4914	0.0089
	Max	12.6359	7.0894
	Min	10.7407	7.0392
New Zealand	Average	83.7029	28.1561
	SD	3.1496	0.0089
	Max	93.8784	28.1633
	Min	79.3689	28.1131
Singapore	Average	1.5835	1.8089
	SD	0.1966	0.0089
	Max	2.2932	1.8160
	Min	1.4239	1.7658
South Africa	Average	1.6398	1.8091
	SD	0.0630	0.0089
	Max	1.8269	1.8162
	Min	1.4895	1.7660
USA	Average	277.0724	63.2675
	SD	17.8858	0.0089
	Max	301.9495	63.2746
	Min	246.4505	63.2244

the x-axis, we can see the numbers from 1 to 10. These numbers are the anthems as listed in table 6.1.

Figure 2 show the average execution time and memory usage of the runs by three blocks for the plaintext and running it seven times. Where the key is the

Table 7. Runs per block (three) Windows.

Anthem	Parameters	Execution Time (s)	Memory Usage (MB)
Australia	Average	22.2355	15.8624
	SD	0.0704	0.0089
	Max	22.3828	15.8695
	Min	22.0792	15.8193
Canada	Average	7.1483	7.0801
	SD	0.8943	0.0089
	Max	11.5577	7.0873
	Min	6.8720	7.0371
Great Britain	Average	30.0705	7.0815
	SD	124.5569	0.0089
	Max	689.5307	7.0886
	Min	6.9026	7.0385
India	Average	6.9861	7.0803
	SD	0.1767	0.0089
	Max	7.8852	7.0874
	Min	6.9125	7.0372
Ireland	Average	22.2087	15.8637
	SD	0.0743	0.0089
	Max	22.4107	15.8709
	Min	22.0785	15.8207
Jamaica	Average	6.9500	7.0822
	SD	0.0384	0.0089
	Max	7.0793	7.0894
	Min	6.9087	7.0392
New Zealand	Average	52.8028	28.1561
	SD	3.2683	0.0089
	Max	62.2822	28.1633
	Min	50.2036	28.1131
Singapore	Average	0.9744	1.8089
	SD	0.0135	0.0089
	Max	1.0212	1.8160
	Min	0.9550	1.7658
South Africa	Average	0.9827	1.8091
	SD	0.0106	0.0089
	Max	1.0160	1.8162
	Min	0.9656	1.7660
USA	Average	180.5225	63.2675
	SD	51.7192	0.0089
	Max	453.0388	63.2746
	Min	161.4610	63.2244

size of each block. In the x-axis, we can see the numbers from 1 to 10. These numbers are the anthems as listed in table 6.1.

Figure 3 show the average execution time and memory usage of the runs by thirty blocks for the plaintext and running it seven times. Where the key is the

Table 8. Runs per block (three) Online.

Anthem	Parameters	Execution Time (s)	Memory Usage (MB)
Australia	Average	16.1643	15.8624
	SD	1.0832	0.0089
	Max	19.7402	15.8695
	Min	15.4277	15.8193
Canada	Average	6.4902	7.0801
	SD	0.9334	0.0089
	Max	9.0094	7.0873
	Min	6.0004	7.0371
Great Britain	Average	6.4617	7.0815
	SD	0.7231	0.0089
	Max	8.1688	7.0886
	Min	5.9552	7.0385
India	Average	6.0322	7.0803
	SD	0.0324	0.0089
	Max	6.1227	7.0874
	Min	5.9866	7.0372
Ireland	Average	15.6482	15.8637
	SD	0.2025	0.0089
	Max	16.2677	15.8709
	Min	15.4033	15.8207
Jamaica	Average	6.6892	7.0822
	SD	0.9726	0.0089
	Max	8.6431	7.0894
	Min	5.9876	7.0392
New Zealand	Average	34.5667	28.1574
	SD	0.4773	0.0091
	Max	36.2345	28.1646
	Min	34.0755	28.1131
Singapore	Average	0.9882	1.8089
	SD	0.0109	0.0089
	Max	1.0342	1.8160
	Min	0.9801	1.7658
South Africa	Average	0.9895	1.8091
	SD	0.0083	0.0089
	Max	1.0206	1.8162
	Min	0.9815	1.7660
USA	Average	113.6309	63.2687
	SD	1.6772	0.0091
	Max	116.8265	63.2759
	Min	111.2799	63.2244

size of each block. In the x-axis, we can see the numbers from 1 to 10. These numbers are the anthems as listed in table 6.1.

Figure 4 show the average execution time and memory usage of the runs by three hundred blocks for the plaintext and running it seven times. Where the

Table 9. Runs per block (thirty) Mac.

Anthem	Parameters	Execution Time (s)	Memory Usage (MB)
Australia	Average	0.7155	1.5414
	SD	0.8525	0.0089
	Max	5.1868	1.5485
	Min	0.4533	1.4983
Canada	Average	0.4028	0.7251
	SD	0.9142	0.0089
	Max	5.2114	0.7322
	Min	0.1600	0.6820
Great Britain	Average	0.3778	0.7265
	SD	1.0161	0.0089
	Max	5.7564	0.7336
	Min	0.1704	0.6834
India	Average	0.3017	0.7252
	SD	0.4158	0.0089
	Max	2.4866	0.7324
	Min	0.1741	0.6822
Ireland	Average	0.7427	1.5427
	SD	0.9062	0.0089
	Max	5.4912	1.5499
	Min	0.4681	1.4997
Jamaica	Average	0.2054	0.7272
	SD	0.0472	0.0089
	Max	0.3668	0.7343
	Min	0.1560	0.6841
New Zealand	Average	1.2550	2.6832
	SD	0.3763	0.0089
	Max	3.2087	2.6903
	Min	1.1071	2.6401
Singapore	Average	0.2285	0.2338
	SD	0.7445	0.0089
	Max	4.1370	0.2409
	Min	0.0377	0.1907
South Africa	Average	0.0624	0.2339
	SD	0.0138	0.0089
	Max	0.0919	0.2411
	Min	0.0424	0.1909
USA	Average	5.1072	5.9324
	SD	1.6908	0.0089
	Max	9.7633	5.9396
	Min	3.1777	5.8894

Table 10. Runs per block (thirty) Windows.

Anthem	Parameters	Execution Time (s)	Memory Usage (MB)
Australia	Average	0.3077	1.5414
	SD	0.0204	0.0089
	Max	0.3641	1.5485
	Min	0.2825	1.4983
Canada	Average	0.1028	0.7251
	SD	0.0098	0.0089
	Max	0.1400	0.7322
	Min	0.0934	0.682
Great Britain	Average	0.1063	0.7265
	SD	0.0165	0.0089
	Max	0.1791	0.7336
	Min	0.0938	0.6834
India	Average	0.1048	0.7252
	SD	0.0157	0.0089
	Max	0.1687	0.7324
	Min	0.0929	0.6822
Ireland	Average	0.298	1.5427
	SD	0.0135	0.0089
	Max	0.3429	1.5499
	Min	0.2849	1.4997
Jamaica	Average	0.1042	0.7272
	SD	0.0115	0.0089
	Max	0.1449	0.7343
	Min	0.0926	0.6841
New Zealand	Average	0.6614	2.6832
	SD	0.0136	0.0089
	Max	0.716	2.6903
	Min	0.6445	2.6401
Singapore	Average	0.0287	0.2338
	SD	0.008	0.0089
	Max	0.0622	0.2409
	Min	0.02	0.1907
South Africa	Average	0.0277	0.2339
	SD	0.007	0.0089
	Max	0.052	0.2411
	Min	0.0184	0.1909
USA	Average	2.1344	5.9324
	SD	0.0177	0.0089
	Max	2.1894	5.9396
	Min	2.112	5.8894

key is the size of each block. In the x-axis, we can see the numbers from 1 to 10. These numbers are the anthems as listed in table 6.1.

Table 11. Runs per block (thirty) Online.

Anthem	Parameters	ExecutionTime(s)	MemoryUsage(MB)
Australia	Average	0.3089	1.5414
	SD	0.0091	0.0089
	Max	0.3470	1.5485
	Min	0.3034	1.4983
Canada	Average	0.1070	0.7251
	SD	0.0069	0.0089
	Max	0.1421	0.7322
	Min	0.1035	0.6820
Great Britain	Average	0.1089	0.7265
	SD	0.0093	0.0089
	Max	0.1499	0.7336
	Min	0.1043	0.6834
India	Average	0.1091	0.7252
	SD	0.0124	0.0089
	Max	0.1695	0.7324
	Min	0.1036	0.6822
Ireland	Average	0.3073	1.5427
	SD	0.0084	0.0089
	Max	0.3432	1.5499
	Min	0.3026	1.4997
Jamaica	Average	0.1083	0.7272
	SD	0.0080	0.0089
	Max	0.1460	0.7343
	Min	0.1042	0.6841
New Zealand	Average	0.6830	2.6832
	SD	0.0124	0.0089
	Max	0.7256	2.6903
	Min	0.6724	2.6401
Singapore	Average	0.0289	0.2338
	SD	0.0067	0.0089
	Max	0.0602	0.2409
	Min	0.0254	0.1907
South Africa	Average	0.0283	0.2339
	SD	0.0084	0.0089
	Max	0.0590	0.2411
	Min	0.0246	0.1909
USA	Average	2.1718	5.9324
	SD	0.0105	0.0089
	Max	2.2157	5.9396
	Min	2.1613	5.8894

7 Conclusions

The HAJ algorithm constituted an initial contribution to symmetric cryptography based on matrix operations by combining the classical Hill cipher with

Table 12. Runs per block (three hundred) for Mac.

Anthem	Parameters	Execution Time (s)	Memory Usage (MB)
Australia	Average	0.0819	0.4172
	SD	0.0291	0.0089
	Max	0.1932	0.4243
	Min	0.0573	0.3742
Canada	Average	0.0771	0.3295
	SD	0.0241	0.0089
	Max	0.1360	0.3367
	Min	0.0533	0.2865
Great Britain	Average	0.0762	0.3309
	SD	0.0347	0.0089
	Max	0.1894	0.3381
	Min	0.0441	0.2879
India	Average	0.0645	0.3297
	SD	0.0230	0.0089
	Max	0.1288	0.3368
	Min	0.0470	0.2866
Ireland	Average	0.0780	0.4186
	SD	0.0301	0.0089
	Max	0.1845	0.4257
	Min	0.0524	0.3755
Jamaica	Average	0.0733	0.3316
	SD	0.0232	0.0089
	Max	0.1178	0.3388
	Min	0.0444	0.2886
New Zealand	Average	0.1009	0.5390
	SD	0.0396	0.0088
	Max	0.2070	0.5461
	Min	0.0766	0.4959
Singapore	Average	0.0654	0.2754
	SD	0.0210	0.0089
	Max	0.1293	0.2826
	Min	0.0424	0.2324
South Africa	Average	0.0604	0.2756
	SD	0.0253	0.0089
	Max	0.1490	0.2827
	Min	0.0389	0.2325
USA	Average	0.1395	0.8740
	SD	0.0418	0.0088
	Max	0.2806	0.8811
	Min	0.1132	0.8309

Table 13. Runs per block (three hundred) for Windows.

Anthem	Parameters	ExecutionTime(s)	MemoryUsage(MB)
Australia	Average	0.0384	0.4172
	SD	0.0095	0.0089
	Max	0.0588	0.4243
	Min	0.0271	0.3742
Canada	Average	0.0356	0.3295
	SD	0.0083	0.0089
	Max	0.0670	0.3367
	Min	0.0254	0.2865
Gran Bretaña	Average	0.0348	0.3309
	SD	0.0104	0.0089
	Max	0.0673	0.3381
	Min	0.0232	0.2879
India	Average	0.0338	0.3297
	SD	0.0153	0.0089
	Max	0.1033	0.3368
	Min	0.0235	0.2866
Ireland	Average	0.0374	0.4186
	SD	0.0108	0.0089
	Max	0.0711	0.4257
	Min	0.0280	0.3755
Jamaica	Average	0.0322	0.3316
	SD	0.0061	0.0089
	Max	0.0462	0.3388
	Min	0.0238	0.2886
New Zealand	Average	0.0457	0.5390
	SD	0.0111	0.0088
	Max	0.0790	0.5461
	Min	0.0335	0.4959
Singapore	Average	0.0285	0.2754
	SD	0.0065	0.0089
	Max	0.0550	0.2826
	Min	0.0210	0.2324
South Africa	Average	0.0251	0.2756
	SD	0.0059	0.0089
	Max	0.0414	0.2827
	Min	0.0170	0.2325
USA	Average	0.0785	0.8740
	SD	0.0285	0.0088
	Max	0.2104	0.8811
	Min	0.0610	0.8309

the Gauss-Jacques method for modular matrix inversion. The principal novelty of this initial work was the integration of these two mathematical procedures into a single encryption/decryption framework, together with a multi-round and block-based implementation.

Table 14. Runs per block (three hundred) Online.

Anthem	Parameters	ExecutionTime(s)	MemoryUsage(MB)
Australia	Average	0.0355	0.4172
	SD	0.0125	0.0089
	Max	0.1008	0.4243
	Min	0.0312	0.3742
Canada	Average	0.0308	0.3295
	SD	0.0147	0.0089
	Max	0.1064	0.3367
	Min	0.0261	0.2865
Great Britain	Average	0.0297	0.3309
	SD	0.0092	0.0089
	Max	0.0661	0.3381
	Min	0.0252	0.2879
India	Average	0.0283	0.3297
	SD	0.0079	0.0089
	Max	0.0697	0.3368
	Min	0.0258	0.2866
Ireland	Average	0.0337	0.4186
	SD	0.0058	0.0089
	Max	0.0615	0.4257
	Min	0.0303	0.3755
Jamaica	Average	0.0293	0.3316
	SD	0.0079	0.0089
	Max	0.0677	0.3388
	Min	0.0258	0.2886
New Zealand	Average	0.0426	0.5390
	SD	0.0106	0.0088
	Max	0.0782	0.5461
	Min	0.0378	0.4959
Singapore	Average	0.0243	0.2754
	SD	0.0034	0.0089
	Max	0.0416	0.2826
	Min	0.0223	0.2324
South Africa	Average	0.0246	0.2756
	SD	0.0063	0.0089
	Max	0.0579	0.2827
	Min	0.0224	0.2325
USA	Average	0.0705	0.8740
	SD	0.0095	0.0088
	Max	0.1158	0.8811
	Min	0.0637	0.8309

The results obtained from thirty-run experiments across different plaintext sizes and computational environments provided an initial characterization of the algorithm's computational behavior.

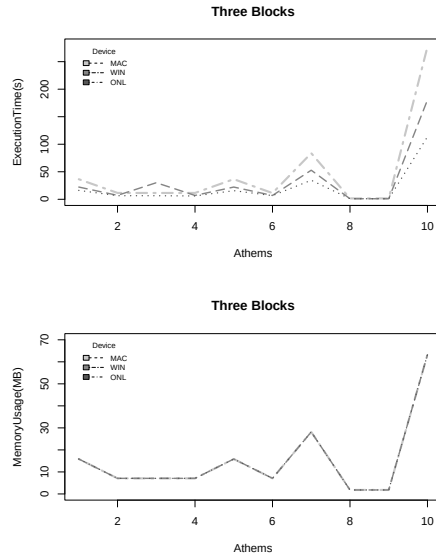


Fig. 2. Average accuracy across 30 runs.

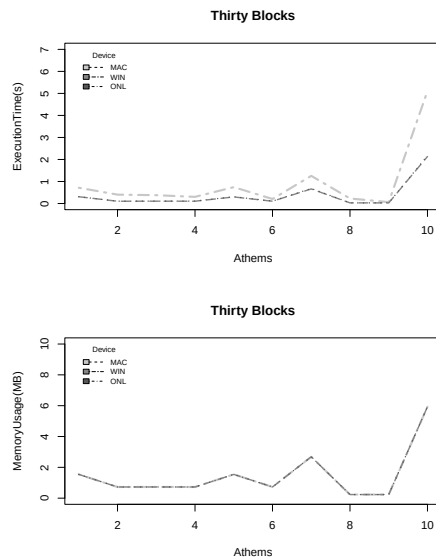


Fig. 3. Average accuracy across 30 runs.

The experimental results showed that memory consumption remained comparatively stable for a given configuration, whereas execution time was strongly

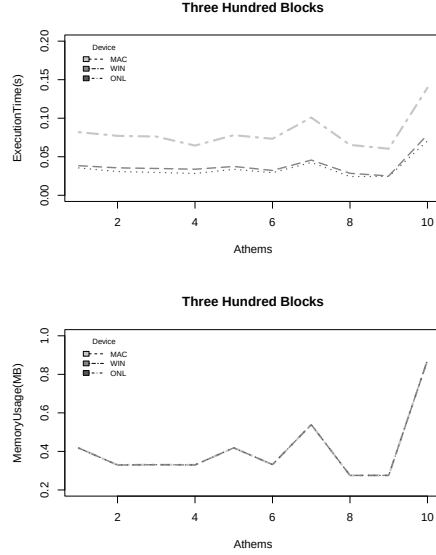


Fig. 4. Average accuracy across 30 runs.

influenced by plaintext length, matrix dimension, number of blocks, and computational platform.

An important observation was that the implementation was also able to operate with a matrix dimension of 1705×1705 in the experiments, demonstrating that the proposed construction could handle substantially large matrices within the evaluated computational environments.

This result provided a useful indication of the computational scalability of the approach, although matrix size alone could not be interpreted as evidence of cryptographic security, entropy, or randomness.

The experiments also showed the effect of changing the block configuration on the computational behavior of the algorithm. These observations provided practical evidence regarding the performance characteristics of HAJ and established an initial computational baseline for its further development. The results were interpreted as performance measurements rather than as a formal proof of cryptographic security.

The combination of the Gauss-Jacques method and the extended Euclidean algorithm provided a systematic procedure for computing the modular inverses required during decryption. This was particularly relevant because modular inversion represented an essential operation within the proposed matrix-based construction.

At the same time, the study primarily addressed algorithmic design and computational evaluation. A comprehensive security assessment remained necessary, including statistical randomness testing, differential and linear cryptanalysis,

known-plaintext and chosen-plaintext attacks, side-channel considerations, and a systematic analysis of key generation and management.

An important outcome of this research was that the matrix-based framework introduced by HAJ opened a broader line of investigation into the algebraic structure of cryptographic matrices. Thus, the contribution of HAJ was both concrete and foundational: it proposed a specific symmetric construction, evaluated its computational behavior across different configurations, demonstrated its operation with substantially large matrix dimensions, and motivated deeper investigation into the algebraic properties of matrix-based cryptographic primitives.

Future work should focus on formal cryptanalysis, optimization of matrix generation and inversion, larger and systematically selected parameter sets, standardized statistical testing, resistance to established attack models, implementation-level side-channel analysis, and the study of structured non-diagonalizable matrix constructions under classical and quantum attack models.

These investigations were necessary to determine whether the algebraic structures explored in this research could ultimately lead to cryptographic constructions with formally established security properties.

References

1. Stallings, W.: *Cryptography and Network Security: Principles and Practice*. Pearson Education (2017)
2. Paar, C., Pelzl, J.: *Understanding Cryptography: A Textbook for Students and Practitioners*. Springer Science & Business Media (2010)
3. Radhakrishnan, I., et al.: Efficiency and Security Evaluation of Lightweight Cryptographic Algorithms for Resource-Constrained IoT Devices. *Sensors*, 24(12), pp. 4008 (2024)
4. Sarker, K.U., et al.: A Systematic Review on Lightweight Security Algorithms for Resource-Constrained Systems. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications* (2025). Doi: 10.1007/s43926-025-00150-4.
5. Wen, H., Lin, Y., Yang, L., Chen, R.: Cryptanalysis of an Image Encryption Scheme Using Variant Hill Cipher and Chaos. *Expert Systems with Applications*, 250, 123748 (2024). Doi:10.1016/j.eswa.2024.123748.
6. Teseleanu, G.: Security Analysis of an Image Encryption Scheme Based on a New Secure Variant of Hill Cipher and 1D Chaotic Maps. In: *Proceedings of the 10th International Conference on Information Systems Security and Privacy (ICISSP 2024)*, SCITEPRESS, pp. 745–749 (2024). Doi:10.5220/0012356600003648.
7. Bufalo, M.: Some Properties of the Computation of the Modular Inverse with Applications in Cryptography. *Mathematics*, 11(4), pp. 70, (2023). Doi: 10.3390/math11040070.
8. Xu, G., Tian, Y., Yang, B.: On the Inversion Modulo a Power of an Integer. *arXiv Preprint*, arXiv:2506.02491, <https://arxiv.org/abs/2506.02491> (2025)
9. Rrghout, H., Kattass, M., Qobbi, Y., Benazzi, N., Jarjar, A.: Robust Image Encryption Algorithm Using a New Variant of Hill Cipher. In: *Proceedings of the 6th International Conference on NISS, ACM* (2023). Doi: 10.1145/3607720.3607750.

10. Alarcon, A.: Towards a Symmetric Crypto Algorithm: The HAJ. ITNG 2021 18th International Conference on Information Technology-New Generations (2021)
11. Alshammari, B.M., Guesmi, R., Guesmi, T., Alsaif, H., Alzamil, A.: Implementing a symmetric lightweight cryptosystem in highly constrained IoT devices by using a chaotic S-box. *Symmetry*, 131, p. 129, 2021.
12. Hill, L.S.: Cryptography in an Algebraic Alphabet. *The American Mathematical Monthly*, 36, pp. 306–312 (1929)
13. Schneier, B.: *Applied Cryptography: Protocols, Algorithms, and Source Code* in C. John Wiley & Sons (1996)
14. Ali, N.H.M., Hoobi, M.M., Saffo, D.F.: Development of Robust and Efficient Symmetric Random Keys Model based on the Latin Square Matrix. *Mesopotamian Journal of Cybersecurity*, 4(3), pp. 203–215 (2024)
15. Singh, S., Srivastava, S.: Cryptanalysis of Hill cipher using known plaintext attack. *International Journal of Scientific and Research Publications*, 4, pp. 1–4 (2014)
16. Cui, X., Zhang, H., Fang, X., Wang, Y., Wang, D., Fan, F., Shu, L.: A secret key classification framework of symmetric encryption algorithm based on deep transfer learning. *Applied Sciences*, 13(21), pp. 12025 (2023)
17. Agrawal, M., Gupta, D.: Cryptanalysis of Hill Cipher with Known Ciphertext Attack. *International Journal of Computer Applications*, 72, pp. 39–44 (2013)
18. Singh, S., Srivastava, S.: Cryptanalysis of Hill Cipher using Frequency Attack. *International Journal of Computer Applications*, 86, pp. 1–5 (2014)
19. Cantón, D.: Gauss-Jacques Method. <https://github.com/dCantonE/gauss-jacques> (2023)

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



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



Centro de Investigación
en Computación