EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

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

Research in Computing Science

Vol. 154 No. 1
January 2025

Research in Computing Science

***RESEARCH IN COMPUTING SCIENCE**, Año 25, Volumen 154, No. 1, Enero de 2025, es una publicación mensual, editada por el Instituto Politécnico Nacional, a través del Centro de Investigación en Computación. Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, Ciudad de México, Tel. 57 29 60 00, ext. 56571. https://www.rcs.cic.ipn.mx. Editor responsable: Dr. Grigori Sidorov. Reserva de Derechos al Uso Exclusivo del Título No. 04-2019-082310242100-203. ISSN: en trámite, otorgado por el Instituto Nacional del 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 Othón de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738. Fecha de última modificación 10 de Enero de 2025.*

***RESEARCH IN COMPUTING SCIENCE**, Year 25, Volume 154, No. 1, January, 2025, is a monthly publication edited by the National Polytechnic Institute through the Center for Computing Research. Av. Juan de Dios Bátiz S/N, Esq. Miguel Othón de Mendizábal, Nueva Industrial Vallejo, C.P. 07738, Mexico City, Tel. 57 29 60 00, ext. 56571. https://www.rcs.cic.ipn.mx. Editor in charge: Dr. Grigori Sidorov. Reservation of Exclusive Use Rights of Title No. 04-2019-082310242100-203. ISSN: pending, granted by the National Copyright Institute. Responsible for the latest update of this issue: the Computer Research Center, Dr. Grigori Sidorov, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738. Last modified on January 10, 2025.*

Advances in Computing Science and Applications

Juan Carlos Chimal Eguía (ed.)

Instituto Politécnico Nacional, Centro de Investigación en Computación
México 2025

ISSN: in process

Formerly ISSNs: 1870-4069, 1665-9899

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

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

Indexed in LATINDEX, DBLP and Periodica

Electronic edition

ISSN 1870-4069

Table of Contents

Page

ISSN 1870-4069

SmallCovid-Net: COVID-19 Segmentation Using CT Scan Lung Images

Alvaro Salazar Urbina[1], Elías Ventura Molina[2], Sergio Flores Cortes[3], Antonio de Jesús Méndez Agüero[4], Daniel Jiménez Alcantar [5]

[1] Instituto Politécnico Nacional,
Centro de Investigación en Computación,
Mexico

[2] Instituto Politécnico Nacional,
Centro de Innovación y Desarrollo Tecnológico en Cómputo,
Mexico

[3] Instituto Politécnico Nacional,
Escuela Superior de Ingeniería Mecánica y Eléctrica Unidad Zacatenco,
Mexico

[4] Instituto Politécnico Nacional,
Dirección de Sistemas Informáticos,
Mexico

[5] Comisión Federal de Electricidad,
Departamento de Mantenimiento,
Mexico

{asalazaru, eventuram, amendeza}@ipn.mx,
sfloresc2400@alumno.ipn.mx, daniel.jimenez@cfe.mx

Abstract. Since the beginning of the COVID-19 pandemic declared in March 2020, the medical field has faced various challenges. If an injured region can be detected and segmented automatically, it would be a huge help for doctors to diagnoses the patient's infection. However, this is a difficult task because the virus can have different shapes and sizes, more- over it can also be located in any region of the lung. In order to diagnose a patient, lung Computer Tomography Scan (CT) images are required. Nevertheless, the manual review of CT images is a hard task because it requires medical specialist and is a slow process. To be able to segment automatically we proposed a new network named SmallCovid-Net, which is an improved version of the Segnet and Unet model. Because it only focuses on CT COVID-19 scans, the network requires few convolution and filter layers. As a consequence, it requires less training time and has obtained competitive results.

Keywords: Image segmentation, deep learning, COVID-19, computer tomography, Unet, mask R-CNN.

1 Introduction

The SARS-Cov2 or COVID-19 is an infectious and acute fatal disease that has had a devastating effect on the lives of people around the world. It was identified in the Chinese province of Wuhan in December 2019 and after a few months it spread throughout the world [4].

The COVID-19 infection begins in the mucous membranes of the throat and spreads to the lungs through the respiratory tract. Symptoms may include fever, dry cough, difficulty breathing, fatigue, loss of smell and taste, and dizziness. Which can appear from 2 to 14 days after being infected.

For the diagnosis of COVID-19, medical images from computed tomography (CT) and scanned x-ray (XR) have been used with good results [5]. However, in order to adequately interpret the images, it is necessary to have expert radiologist because the disease is very similar to other lung diseases. Furthermore, due to the complex nature of COVID-19 and the high degree of mortality, accurate and time-consuming diagnosis is necessary [13].

The COVID-19 virus primarily attacks the lungs, subsequently there is the possibility of developing infection and lung disease [11]. In order to make an adequate diagnosis, CT images are used. A characteristic that stands out in CT images of the COVID-19 virus is the appearance of ground glass opacity (GGO). However, there are more anomalies in the images that have a relationship with COVID-19, which are: consolidation and pleural effusion [17-19]. To decide whether a patient is healthy or sick, physical observation is used. Nevertheless, this is strenuous work and doctors with sufficient experience are rare. Therefore, a system is necessary that allows automatic classification and indicates the region of interest in the image [21].

Deep Learning (DL) is a tool regularly used in different areas of research such as: computer vision, speech recognition, image processing and natural language, among others. In models that use DL, feature extraction is automatic, so medical experts are not required to perform it. With a deep architecture and multiple processing units, this task can be achieved [9].

DL has been used to perform the classification of medical images with good results, using techniques that use convolutional neural network (CNN) models [10-18]. CNNs use a minimal process of convolution operations on each pixel of the images to extract the set of relevant features regardless of their position [6].

Also, DL has been applied in a wide range of the medical field, CNN has been used to determine if an x-ray contains a malignant tumor, to indicate risks of heart disease, among others [20]. For the segmentation task that indicates the region of interest, architectures such as Full Convolutional Networks (FCN) and U-Net have been used [15].

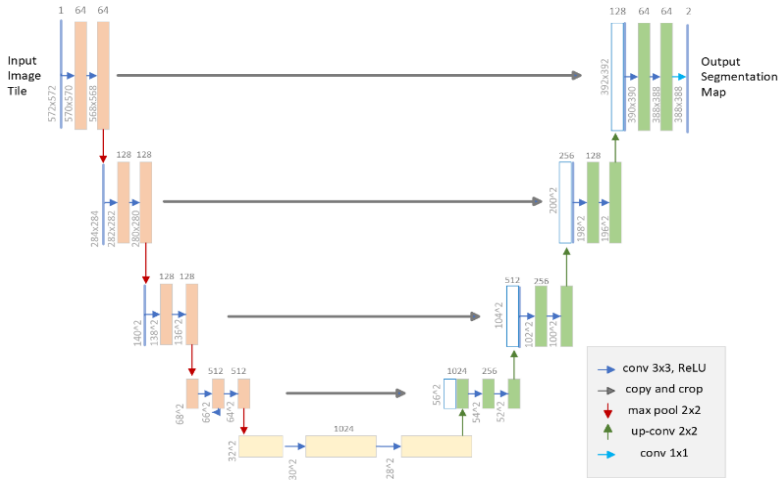

Fig. 1. The U-net model [12].

2 Related Work

2.1 U-Net

In the medical field, one of the most popular models for the segmentation task, which is based on an encoder-decoder architecture. It was proposed to have a tool that could use different types of images that would allow the doctor to have a better overview of the injures. It consists of two parts: a contractive path to capture context and an asymmetric expansion that enables precise locations [12].

The down sampling or contracting part has an architecture that resembles a fully convolutional network (FCN) that extracts features with 3X convolutions. The importance of this part is that it can extract the main features from the input image and the result is a feature vector.

On the other hand, the expansion part recovers the information from the first part by copying and trimming. For this phase the feature vector is constructed by convolutions and generates a segmentation output map.

In this architecture, an important part is the link operation between the two parts previously explained. Therefore, the model can generate an accurate segmentation mask (see Figure 1).

There are works based on this model that allow segmentation of medical images, but they require good quality images and the training time as well as its parameters are greater [16].

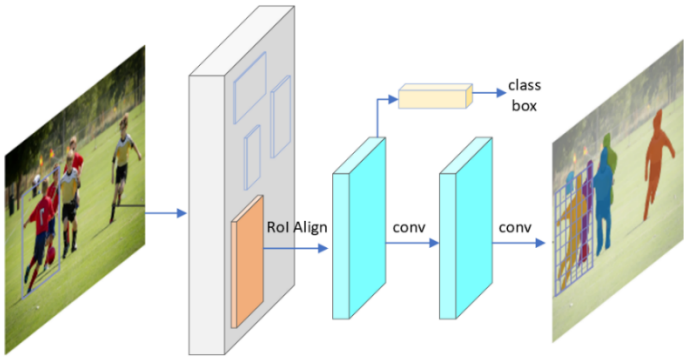

Fig. 2. Mask R-CNN used for detection and segmentation.

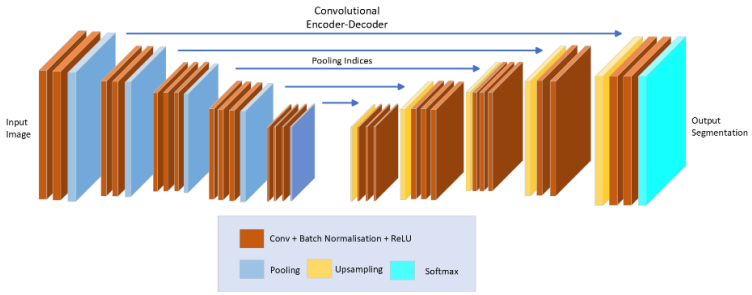

Fig. 3. The SegNet model [3].

2.2 Mask R-CNN

Instance segmentation is challenging due to the fact it requires the correct identification of all objects in the image as well as the precise segmentation of each instance. Therefore, this combines elements of computer vision task such as object detection (the objective is to individually classify objects and locate using bounding box) and semantic segmentation (the objective is to classify each pixel within a fixed set of categories) [12].

Badri Narayanan et al. proposed an encoder decoder architecture for image segmentation [3]. The model does not have fully connected layers, so it is completely convolutional. Its original objective was to do road segmentation. The network uses an unbalanced data set because the pixels of roads and buildings predominate. Segnet is composed of an encoder network, a corresponding decoder and a pixel-wise classification layer. The topology of the encoder network is identical to the 13 convolutional layers of the VGG16 network [8].

Mask R-CNN is a general framework for object instance segmentation. This approach identifies objects in an image while generating a segmentation mask for each instance. This architecture is simple to train and contains two main phases (Figure 2).

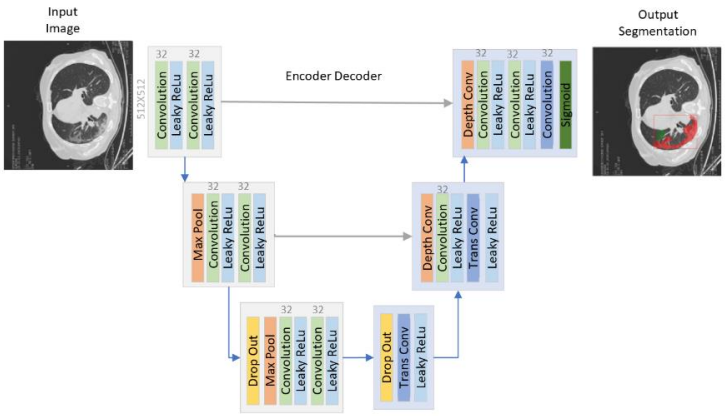

Fig. 4. SmallCovid-Net model.

The first one is the R-CNN architecture [14]; it has three elements: the backbone, Region Proposal Network (RPN) and object detection. The backbone is used for image feature extraction and map generation. Maps are used by RPN and integrate bounding boxes to achieve the detection task. Bounding boxes are classified as positive or foreground. The positives are used to create a Region of Interest (ROI) alignment. The second phase uses a new branch to perform the instance segmentation task [12, 2].

2.3 Segnet

Its main novelty is the way in which the decoder performs upsampling of a low-resolution feature map. The up-sampled maps are convolved with trained filters to produce dense feature maps.

Segnet is designed to be efficient in terms of memory and computing time during the inference phase because its objective lies in the segmentation of panorama images (Figure 3). It uses the stochastic gradient and a smaller number of train- able parameters compared to other architecture.

2.4 SmallCovid-Net

One of the most important features of COVID-19 is the detection of ground-glass opacities on a CT scan. GGO refers to an area of interest fading in the lung on a CT scan. Nevertheless, it is not possible to extract GGO using conventional CNN. In this architecture, the original image is the input and the training and learning process starts from a pixel-level feature. So, to highlight more areas of infections, we have used different filters.

The main objective of the proposed model is to perform the segmentation task of COVID-19 CT images to locate and mark the region containing the lesion. It uses the SegNet architecture as a base because it contains two phases: convolution and deconvolution. We have used the ReLU activation function in the convolution block

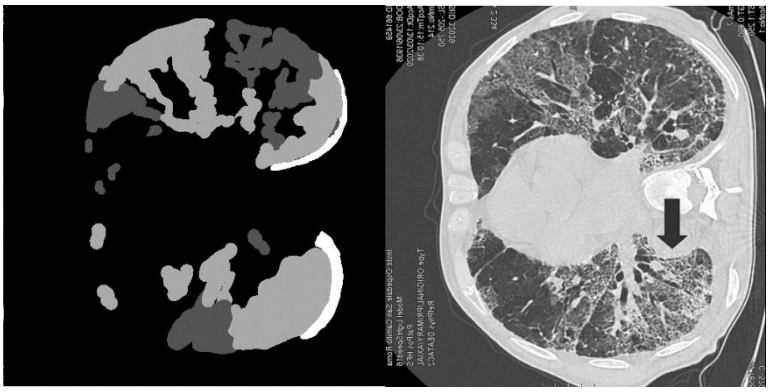

Fig. 5. CT scan image (right) and mask sample (left). In the right white is consolidation, dark gray is ground glass opacities and gray is pleural effusion.

because we only handle positive values, and it reduces the complexity of the model. Convolution blocks of 24 filter have been used for images of 256 X 256 pixels.

The fact that there are fewer layers means that there is an improvement in the training phase. On the other hand, the time for this phase has increased due to the needs of computing resources. Finally, the proposed model has obtained good results compared to other state of the art models.

3 Materials and Methods

The main source of materials for this work are the images from the Italian Society of Medicine which consists of scanned computed tomography images. The images have been segmented by radiological experts and for each image there is its segmentation mask counterpart. The format of the images is gray scale and their dimensions are 512 x 512 pixels [1]. The main source of materials for this work are the images from the Italian Society of Medicine which consists of scanned computed tomography images.

The images have been segmented by radiological experts and for each image there is its segmentation mask counterpart. The format of the images is gray scale and their dimensions are 512 x 512 pixels [1].

In this work the objective is to do a semantic segmentation although in the selected data set three types of lesions related to COVID-19 can be detected: consolidation, pleural effusion and ground glass opacities. The images belong to people who were infected in the early phases of the COVID-19 pandemic in European countries [7].

Of the 100 images that exist in the database, 72 were used for the training phase, 18 for the validation and 10 were reserved for the testing one. A process was carried out on the images to reduce them to 256 x 256 pixels.

The training phase consisted of a set of 20 iterations with 600 steps per iteration. The learning ratio was 0.0001 for the Gradient Descent optimizer.

The number of classes was two: the part of the image that contains the lesion and the part that is not a lesion. In this aspect, there is a considerable class imbalance because there are more pixels in healthy regions than in the infected ones. In this work the objective is to do a semantic segmentation although in the selected data set three types of lesions related to COVID-19 can be detected: consolidation, pleural effusion and ground glass opacities. The images belong to people who were infected in the early phases of the COVID-19 pandemic in European countries [7].

Of the 100 images that exist in the database, 72 were used for the training phase, 18 for the validation and 10 were reserved for the testing one. A process was carried out on the images to reduce them to 256 x 256 pixels.

The training phase consisted of a set of 20 iterations with 600 steps per iteration. The learning ratio was 0.0001 for the Gradient Descent optimizer.

The number of classes was two: the part of the image that contains the lesion and the part that is not a lesion. In this aspect, there is a considerable class imbalance because there are more pixels in healthy regions than in the infected ones.

3.1 Network Training

The training process was been done using open source tools such as Python 3.9 as a programming language and the hardware configured to execute the experiments was a personal computer with a processor Intel(R) Core(TM)©i9- 6700 CPU @ 4.20 Ghz with 16 cores and NVIDIA©GeForce GTX 1050 Ti, CUDA Toolkit 10.0 and CUDNN 7.4.1 were used to drop the time training.

Using a GPU reduce the training process and can complete simple task faster because it is possible to decompose complex task in simple ones and process them in parallel.

3.2 Performance Measures

To fully quantify the performance of our model, we have used performance mea- sures for the classification and segmentation task: precision, dice coefficient and recall. These measures are used in the medical field:

$$Precision = \frac{TP}{TP+FP}, \tag{1}$$

$$Dice = \frac{2|A \cap B|}{|A|+|B|}, \tag{2}$$

$$Recall = \frac{TP}{TP+FN}. \tag{3}$$

In the Equation 1 and 3 TP or true positive is the number of pixels labeled by the model as COVID-19 is correct. The FP or false positive is number of pixels labeled by the model as COVID-19 is wrong. The FN or false negative is the number of pixels labeled by the model as non-COVID-19 is wrong. The Equations 2 A refers to the predicted mask and B is the ground truth one.

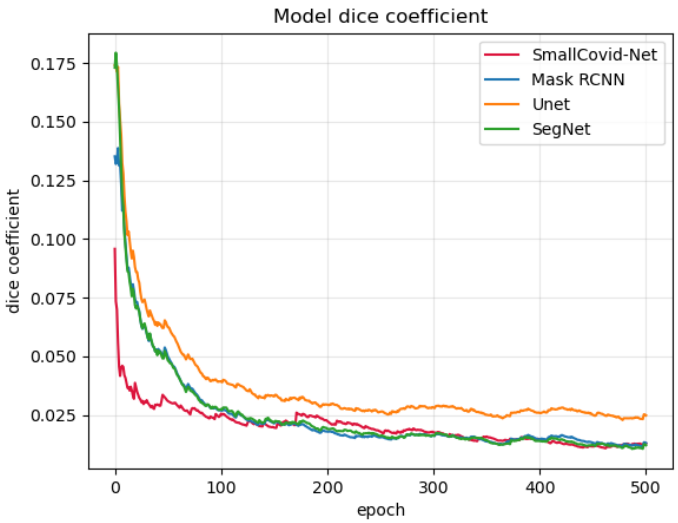

Fig. 6. Training loss.

Table 1. Performance metrics associated with different algorithms for the images in the testing dataset.

Method	Dice	Precision	Recall
Mask R-CNN	0.6801	0.6857	0.6333
Unet	**0.7202**	0.5190	**0.7667**
SegNet	0.7001	0.6667	0.7333
Proposed	0.7011	**0.7121**	0.7321

4 Results and Discussion

The proposed model uses a probability to determine whether each pixel is in a segment classified as COVID-19 or is a healthy region. In order to achieve the above, it is necessary to select a threshold that allows reliable results, which is the reason it has been decided to place the threshold at 0.8.

Due to the small size of the data set used for the segmentation task, it was decided to use the k-fold cross validation method. Of the 100 images, 10 have been selected randomly to carry out the testing phase. The remaining 90 images are used for training and validation phase.

Initially 90 images are divided into 5 equal sets (folds), 4 of these are used for the training phase and the remaining one is used for the validation one. During 5 times the training is carried out, in each one the validation set will be different. At the end of each

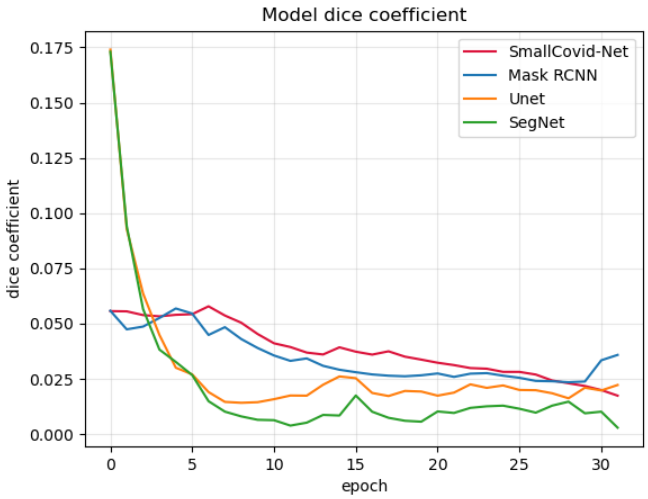

Fig.7. Validation loss.

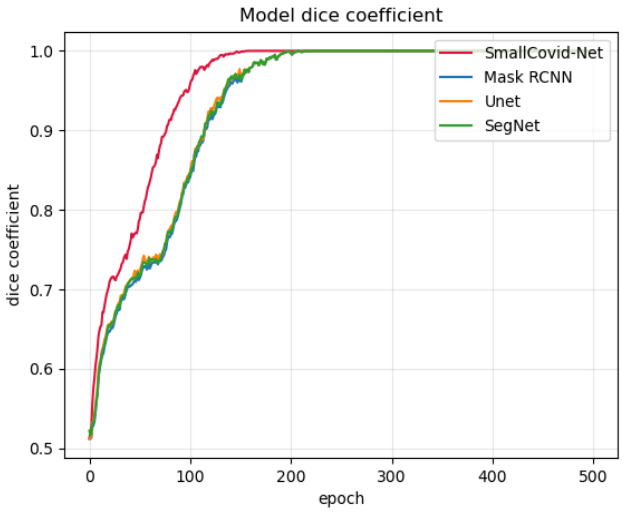

Fig. 8. Dice coefficient during the training phase.

training, the loss measure is calculated and with the average the model performance evaluation is obtained.

In these experiments we perform the segmentation task, but only the semantic one. All models are focused to identify where the injure is. In the Figures 6 and 7 we can see that all models have learned the way to identify the regions marked as COVID-19.

To show the results, the Table 1 has been created where the performance metrics of the models can be displayed. In order to evaluate the segmentation task, the Says coefficient has been used, which allows comparing the pixel segmentation generated

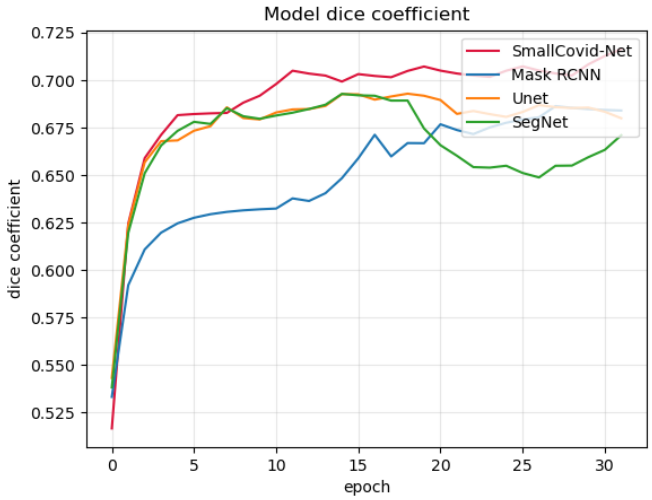

Fig. 9. Dice coefficient during the validation phase.

by the model against the ground truth. The Unet model obtained a better result for the dice and recall metrics, however our proposed model obtained the highest precision metric.

All models used were able to perform COVID-19 lesion classification from the image background, although they were unable to do instance segmentation because they could not determine the type of COVID-19 lesion they had segmented.

In Figures 8 and 9 we can identify that the best scores of the coefficient dice were obtained during the training phase.

5 Conclusions

In this article, we have proposed the SmallCovid-Net model for the segmentation of areas infected with the COVID-19 virus using computed tomography images.

We have made a comparison of our model with other models that perform image segmentation. As has been observed in this work, the segmentation of lesions caused by viruses is a very difficult task because there is no defined shape of lesions, and they can be located anywhere in the lungs. Due to the above, the results using the dice coefficient have yielded very low results, around 70%.

Considering that a perfect segmentation would yield a dice coefficient value of 100%, the model can be a support for medical personnel who want to detect lung injuries.

In the future, we want to apply this model to detect other types of lesions in other organs such as the brain. We also want to improve our model so that it can perform instance segmentation and not just semantic.

Acknowledgments. No funding was received to assist with the preparation of this manuscript.

References

1. Medical Segmentation.: COVID-19 CT segmentation dataset (2020) http://medicalsegmentation.com/covid19/
2. Abdulla, W.: Mask R-CNN for Object Detection and Instance Segmentation on Keras and Tensorflow. https://github.com/matterport/Mask_RCNN (2017)
3. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(12), pp. 2481–2495 (2017) doi: 10.1109/TPAMI.2016.2644615.
4. Chan, J.F.W., Yuan, S., Kok, K.H.: A Familial Cluster of Pneumonia Associated with the 2019 Novel Coronavirus Indicating Person-to-Person Transmission: A Study of a Family Cluster. The Lancet, 395(10223), pp. 514–523 (2020) doi: 10.1016/S0140-6736(20) 30154-9.
5. Chowdhury, M.E.H., Rahman, T., Khandakar, A.: Can AI Help in Screening Viral and COVID-19 Pneumonia?. IEEE Access 8, pp. 132665–132676 (2020) doi: 10.1109/ ACCESS.2020.3010287.
6. Gatys, L.A., Ecker, A.S., Bethge, M.: Texture and Art with Deep Neural Networks. Current Opinion in Neurobiology, 46, pp. 178–186 (2017) doi: 10.1016/j.conb.2017.08.019.
7. Hofmanninger, J., Prayer, F., Pan, J.: Automatic Lung Segmentation in Routine Imaging is Primarily a Data Diversity Problem, not a Methodology Problem. European Radiology Experimental, 4(1) (2020) doi: 10.1186/s41747-020-00173-2.
8. Jiang, Z.P., Liu, Y.Y., Shao, Z.E.: An Improved VGG16 Model for Pneumonia Image Classification. Applied Sciences, 11(23) (2021) doi: 10.3390/app112311185.
9. LeCun, Y., Bengio, Y., Hinton, G.: Deep Learning. Nature, 521(7553), pp. 436 (2015) doi: 10.1038/nature14539.
10. Litjens, G., Kooi, T., Bejnordi, B.E.: A Survey on Deep Learning in Medical Image Analysis. Medical Image Analysis, 42, pp. 60–88 (2017) doi: 10.1016/j.media.2017.07.005.
11. Liu, Z., Jin, C., Wu, C.C.: Association Between Initial Chest CT or Clinical Features and Clinical Course in Patients with Coronavirus Disease 2019 Pneumonia. Korean J Radiology, 21(6), pp. 736–745 (2020) doi: 10.3348/kjr.2020.0171.
12. Minaee, S., Boykov, Y., Porikli, F.: Image Segmentation Using Deep Learning: A Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(7), pp. 3523–3542 (2022) doi: 10.1109/TPAMI.2021.3059968.
13. Orioli, L., Hermans, M.P., Thissen, J.P.: COVID-19 in Diabetic Patients: Related Risks and Specifics of Management. Annales d'Endocrinologie, 81(2), pp. 101–109 (2020) doi: 10.1016/j.ando.2020.05.001.
14. Ren, S., He, K., Girshick, R.: Faster R-CNN: Towards Real-Time Object Detection with Region Prop Osal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(6), pp. 1137–1149 (2017) doi: 10.1109/TPAMI.2016.2577031.
15. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab, N., Hornegger, J., Wells, W.M.: Medical Image Computing and Computer-Assisted Intervention. MICCAI 2015. pp. 234–241. Springer International Publishing (2015)

16. Salazar-Urbina, A., Ventura-Molina, E.J., Yáñez-Márquez, C.: Minicovid-unet: CT-Scan Lung Images Segmentation for COVID-19 Identification. Computación y Sistemas, 28(1), pp. 75–84 (2024) doi: 10.13053/CyS-28-1-4697.
17. Shi, H., Han, X., Jiang, N.: Radiological Findings from 81 Patients with COVID-19 Pneumonia in Wuhan, China: A Descriptive Study. The Lancet Infectious Diseases, 20(4), pp. 425–434 (2020) doi: 10.1016/S1473-3099(20)30086-4.
18. Wang, L., Lin, Z.Q., Wong, A.: COVID-net: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest X-Ray Images. Scientific Reports, 10(1), pp. 19549 (2020) doi: 10.1038/s41598-020-76550-z.
19. Ye, Z., Zhang, Y., Wang, Y.: Chest CT Manifestations of New Coronavirus Disease 2019 (COVID-19): A Pictorial Review. European Radiology, 30(8), pp. 4381–4389 (2020) doi: 10.1007/s00330-020-06801-0.
20. Zhang, Y., Yukun, Y., Wang, D.: Clinical Application of Image Processing and Neural Network in Cytopathological Diagnosis of Lung Cancer. Chinese Journal of Thoracic and Cardiovascular Surgery, 6(04) (2003)
21. Zhang, Z., Ni, X., Huo, G.: Novel Coronavirus Pneumonia Detection and Segmentation Based on the Deep-Learning Method. Annals of Translational Medicine, 9(11) (2021) doi: 10.21037/atm-21-1156.

ISSN 1870-4069

Clasificación para frutos de fresa con CNN

César Augusto Pilón Alcalá, Juan Carlos Olguín Rojas

Universidad Autónoma Chapingo,
Departamento de Ingeniería Mecánica Agrícola,
México

jolguinr@chapingo.mx

Resumen. El uso de la inteligencia artificial (IA), en particular de las redes neuronales convolucionales (RNN), ha revolucionado la capacidad de clasificar objetos y productos con precisión, incluso para personas sin experiencia en el campo. En el caso específico de la clasificación de fresas *Fragaria* × *ananassa* en su estado óptimo, las RNN pueden analizar imágenes y discernir características que indican su frescura, madurez y calidad. Este estudio busca encontrar una arquitectura óptima para la clasificación de fresas basada en imágenes. Para lograr esto, se analizaron tres arquitecturas de redes neuronales convolucionales (LeNet5, VGG16 y VGG19). Se llevaron a cabo varios procedimientos, que consistieron en combinaciones de redes con sus respectivos hiperparámetros. Se utilizaron diferentes tasas de aprendizaje (en un rango de 1e 8 a 0.1), cuatro optimizadores distintos (Adagrad, Adam, RMSProp y SGD), y se emplearon técnicas de entrenamiento (extracción de características) para la clasificación del objeto de estudio (*Fragaria* × *ananassa*). Cada uno de estos procedimientos fue evaluado para determinar y comparar su rendimiento. La arquitectura que mejor desempeño obtuvo fue LeNet5, con el optimizador RMSProp y una tasa de aprendizaje de 0.001, logrando una precisión del 90.6%.

Palabras clave: Redes neuronales convolucionales, fragaria × ananassa, LeNet5, VGG16, VGG19.

Classification of Strawberry Fruits Using CNN

Abstract. The use of artificial intelligence (AI), particularly convolutional neural networks (CNNs), has revolutionized the ability to accurately classify objects and products, even for individuals without expertise in the field. In the specific case of classifying strawberries *Fragaria* × *ananassa* at their op-timal state, CNNs can analyze images and discern features indicating their freshness, ripeness, and quality. This study seeks to find an optimal architecture for strawberry classification based on images. To achieve this, three convolutional neural network architectures (LeNet5, VGG16, and VGG19) were analyzed. Various procedures were carried out, consisting of combinations of networks with their respective hyperparameters. Different learning rates were used (ranging from 1e-8 to 0.1), four distinct optimizers (Adagrad, Ad-am, RMSProp, and SGD), and

training techniques (feature extraction) were employed for the classification of the study subject (*Fragaria* × *ana-nassa*). Each of these procedures was evaluated to determine and compare their performance. The architecture that performed best was LeNet5, with the RMSProp optimizer and a learning rate of 0.001, achieving an accuracy of 90.6%.

Keywords: Convolutional neural networks, fragaria × ananassa, LeNet5, VGG16, VGG19.

1. Introducción

La fresa es una fruta de gran importancia y rentabilidad. Con su alto contenido de vitamina C y valor nutricional, es muy versátil en su uso, ya sea para el consumo directo, la decoración de postres o la elaboración de jugos, entre otros usos.

Sin embargo, su corta vida útil después de la cosecha y su susceptibilidad a lesiones cuando están muy maduras representan desafíos significativos para su almacenamiento y venta [18]. En el 2022 en México se ocupó el octavo lugar en producción de fresa con 568271.93 toneladas [7], y los estados con mayor producción son Michoacán con 354,047.99 t, Guanajuato con 96,119.00 t y Baja California con 93,179.00 t. [13].

Las fresas pertenecen a la familia de las rosáceas y pasan por el ciclo de floración, fructificación y maduración. No hay criterios oficiales establecidos para la clasificación de las diferentes etapas de madurez. El método tradicional para determinar las etapas de madurez de las fresas implica evaluar manualmente su apariencia, color, textura, sabor y firmeza. En consecuencia, resulta fundamental establecer un estándar para la clasificación de las etapas de madurez y desarrollar un método económico y eficaz para identificar con rapidez y precisión cada una de estas [20].

La duración de la etapa de consumo de las fresas después de ser cosechadas es breve, y demasiado maduras, tienden a dañarse, por eso es complicado almacenarlas y comercializarlas. Por lo tanto, es especialmente crucial seleccionar el momento adecuado para recolectarlas, considerando el periodo especifico de almacenamiento. Porque, para que la fresa casi madura pase a ser un fruto maduro requiere aproximadamente de 5 a 10 días [18].

La selección de los frutos de fresa en el estado óptimo de maduración garantiza un producto de excelentes condiciones y buena vida útil. Si se cosechan frutos fisiológicamente inmaduros no se desarrolla su color y sabor, convirtiéndose en un producto de calidad inferior. Si se cosecha un fruto maduro esté tendrá una vida comercial corta y se fermentará fácilmente dañando los frutos a su alrededor [4]. Una alternativa viable a este problema es la implementación de técnicas de visión computacional que no son invasivas.

Estudios previos como el de Yuanyuan Shao y su equipo en [12] utilizaron un sistema portátil de imágenes hiperespectrales para la adquisición de espectros de fresas en el campo y en interiores en 3 etapas de madurez: maduras, semimaduras y verdes. Por otro lado, Xiao-Qin Yue y su grupo de investigación [18] lograron reconocer mediante teléfonos inteligentes (smartphones) y con imágenes de fresas a longitudes de

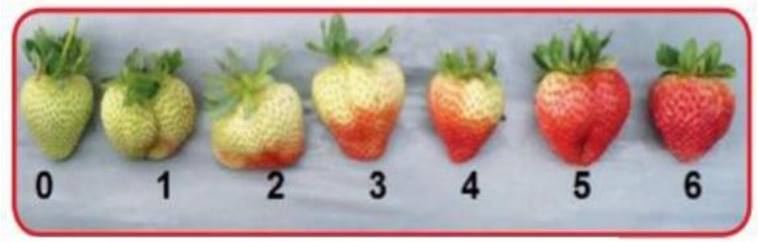

Fig. 1. Tabla de color de la fresa [4].

onda de 535 nm y 670 nm 3 etapas: madura, casi madura e inmadura y Xiaofen Du y sus colaboradores [6] clasificaron mediante una nariz electrónica (e-nose) compuesta por 18 sensores de gases de óxido metálico diferentes, para caracterizar los patrones volátiles de las fresas en 5 etapas de desarrollo: blanco, medio rojo, tres cuartos de rojo, completamente maduro y sobremaduro.

El procesamiento de imágenes utilizando métodos de aprendizaje profundo está siendo investigado para aplicaciones agrícolas debido al aumento de la velocidad de procesamiento y los avances en algoritmos informáticos [19]. Se han utilizado diversas arquitecturas de redes neuronales convolucionales, como YOLOv3 para clasificación y detección de las etapas de maduración de fresa [20], Mask R-CNN para detectar una máscara pixel a pixel de naranja [8] y magulladuras en imágenes de fresas capturadas por una cámara a color bajo luz incandescente y ultravioleta (UV) [19] y YOLOv5 para el reconocimiento en tiempo real del tallo/cáliz de las manzanas [16].

Una CNN (CNN del inglés Convolutional Neural Networks) es una red neuronal con varios tipos de capas especiales. Hoy en día este tipo de red está siendo muy usada por la industria para diversas tareas, especialmente de visión por computacional [15].

La falta de información relacionada con la clasificación de fresas con redes neuronales ha motivado la creación del presente trabajo, por lo que el objetivo de este escrito es encontrar una arquitectura de red neuronal convolucional para la clasificación de fresa en tres etapas de maduración (inmadura, madura y muy madura), para lograrlo, se considerarán algunas arquitecturas CNN que han reportado buenos resultados en diversas aplicaciones.

2. Materiales y métodos

2.1. Arquitecturas CNN utilizadas

Para lograr una clasificación de fresa en 3 etapas de madurez (inmadura, madura y muy madura) de manera no invasiva, se consideraron 3 arquitecturas; LeNet5, basada en la propuesta de Yann LeCun y colaboradores [11]; VGG16 y VGG19 de Karen Simonyan y Andrew Zisserman [14]. Ampliamente reconocidas y utilizadas en diversas áreas, como la robótica, seguridad, agricultura [17] y medicina [10] y en comparación con arquitecturas como Resnet 50, Resnet 101 o Inception v4, son arquitecturas que

Fig. 2. Ejemplo de fresa inmadura.

Fig. 3. Ejemplo de fresa madura.

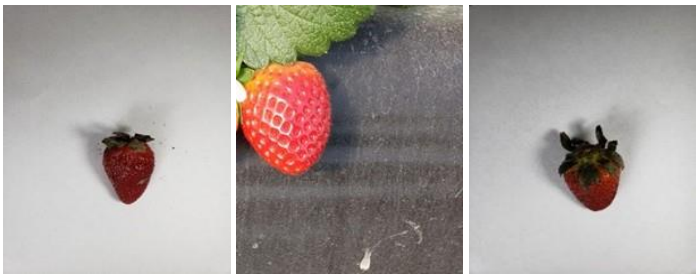

Fig. 4. Ejemplo de fresa muy madura.

emplean un menor número total de parámetros, lo que permite obtener un rendimiento aceptable en implementaciones de baja eficiencia computacional.

2.2. Conjunto de datos

Una de las actividades para realizar esta investigación, fue la creación de un conjunto de imágenes de fresa para identificar las 3 categorías a clasificar (inmadura, madura y muy madura). Para lograr esto, se realizó una búsqueda de imágenes de fresa en la web de diferentes tamaños de imagen y en 3 canales (RGB), las imágenes encontradas se recortaron a conveniencia y se categorizaron según la norma NMX-FF-062-SCFI-2002 [5] que establece que las fresas maduras deben tener una coloración roja que se extiende desde el ápice hasta la base del pedúnculo y cubrir al menos el 50% de la superficie y

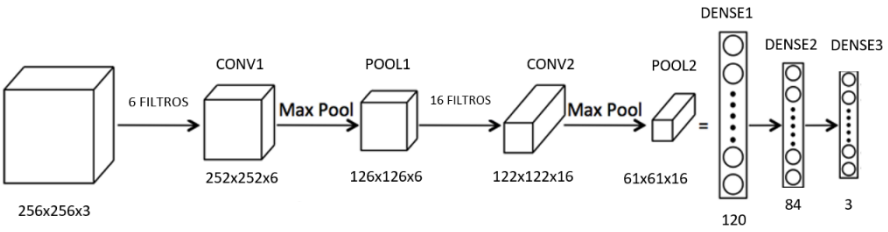

Fig. 5. Esquema ilustrativo de la arquitectura LeNet5.

el Instituto Tecnológico Superior de Coacolmán [4] (ver Fig. 1). Las fresas con características similares a 0, 1 y 2 se categorizaron como inmaduras; 3, 4 y 5 como maduras y 6 como muy maduras.

Se obtuvo un total de 1524 imágenes con diferentes dimensiones. Se generó un conjunto de datos de 710 imágenes para fresas en estado inmaduro, 453 para fresas maduras y 361 para fresas muy maduras. El 70% de cada conjunto de imágenes mencionado se empleó para el entrenamiento, el 15% para la validación y el 15% para las pruebas.

En las Fig. 2, 3 y 4 se muestra un ejemplo de las imágenes utilizadas en este estudio.

2.3. Arquitectura LeNet5

La arquitectura Lenet5 fue creada por Yann LeCun en 1998 y es ampliamente utilizada para el reconocimiento de dígitos escritos a mano (MNIST) [9], está compuesta por 2 capas convolucionales, 2 capas de agrupamiento máximo (max pooling) y 3 capas densas (dense). Para este trabajo se usó esta arquitectura y se propusieron 2 capas covolucionales de 6 y 16 filtros respectivamente con un tamaño de 5x5; 2 capas de agrupamiento máximo de 2x2 y 3 capas densas; en el clasificador se tiene una capa de 120 neuronas con función de activación relu; otra de 84 neuronas con función de activación relu y la última es de 3 neuronas con función de activación softmax (ver Fig. 5).

2.4. Arquitectura VGGNet

También, se utilizó la arquitectura VGG16 (ver Fig. 6) y VGG19 que originalmente fueron preentrenadas con Imagenet. Para cada una de estas, se empleó aprendizaje por transferencia, se usó la técnica de extracción de características (feature extraction) descongelando los pesos del bloque convolucional 5 de cada arquitectura y se reentrenaron 1 y 2 capas convolucionales, finalmente en el clasificador se tienen 2 capas densas de 512 neuronas con función de activación relu y una capa final de 3 neuronas con función de activación softmax.

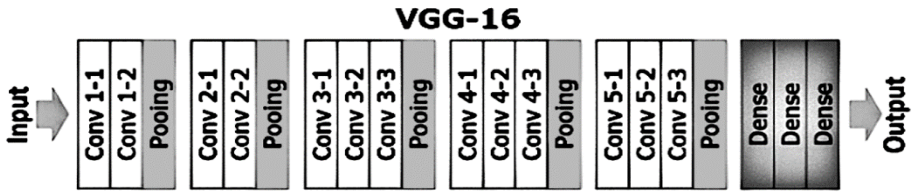

Fig. 6. Esquema ilustrativo de la arquitectura VGG16.

2.5. Diseño del experimento

Para los ensayos, se consideró la búsqueda por rejilla y se propusieron una combinación de hiperparámetros para medir el mejor rendimiento en clasificación, al final de cada entrenamiento, se seleccionaron los mejores rendimientos. Los factores involucrados en cada tratamiento fueron la arquitectura, el optimizador, la tasa de aprendizaje, y en el caso de las arquitecturas VGG, además, el número de capas finales a reentrenar.

Debido al amplio número de combinaciones en la búsqueda por rejilla, se optó por hacer un diseño de experimento en 2 etapas; en la primera se entrenaron las arquitecturas solo a 30 épocas, esto con el fin de obtener las mejores tasas de aprendizaje y reducir el espacio de búsqueda, posteriormente en la segunda etapa, se realizaron nuevamente los experimentos con las mejores tasas de aprendizaje, encontradas anteriormente, pero a 100 épocas. A continuación, se describirá con detalle cada una de las etapas.

2.6. Etapa 1. Búsqueda rápida de tasa de aprendizaje

El objetivo de esta búsqueda fue encontrar un intervalo de valores para la tasa de aprendizaje de los mejores rendimientos. Para ello se hicieron ensayos con valores diferentes de tasa de aprendizaje entre; 0.1 y 1e-8, se utilizaron 4 optimizadores diferentes; Adam, Adagrad, RMSProp y SGD para las arquitecturas Lenet5, VGG16 y VGG19; para las 2 últimas, se utilizó la técnica de extracción de características, la cual consistió en descongelar solo las capas convolucionales 1 y 2 y del bloque 5 (block5_conv1 y block5_conv2), y en otro ensayo se descongeló solo la capa 3 del bloque 5 (block_conv3) para cada arquitectura VGGNet. En esta etapa las mejores tasas de aprendizaje fueron 0.001, 0.0001, 0.00001, 0.000001 y 0.0000001.

2.7. Etapa 2. Búsqueda complete

El objetivo de esta fase fue encontrar la arquitectura, optimizador y tasa de aprendizaje que mejor se desempeñe para este problema específico. Las arquitecturas que se utilizaron fueron LeNet5, VGG16 y VGG19, los optimizadores fueron los 4 mencionados en la etapa 1, se usaron las 5 tasas de aprendizaje obtenidas en la etapa

Tabla 1. 10 mejores rendimientos de 100 tratamientos realizados de la etapa 2. Para obtener la exactitud se promedió la exactitud de entrenamiento, validación y prueba de cada tratamiento. Los tratamientos fueron realizados con 100 épocas.

Tratamiento	Arquitectura	Optimizador	Tasa de aprendizaje	Pesos iniciales	Exactitud
6	LeNet 5	Adam	0.001	Aleatorio	0.906
7	LeNet 5	Adam	0.0001	Aleatorio	0.908
11	LeNet 5	RMSProp	0.001	Aleatorio	0.906
16	LeNet 5	SGD	0.001	Aleatorio	0.907
26	VGG16 reentrenando 1 capa	Adam	0.001	ImageNet	0.904
27	VGG16 reentrenando 1 capa	Adam	0.0001	ImagNet	0.904
28	VGG16 reentrenando 1 capa	Adam	0.00001	ImageNet	0.909
46	VGG16 reentrenando 2 capa	Adam	0.001	ImageNet	0.906
48	VGG16 reentrenando 2 capa	Adam	0.00001	ImageNet	0.897
51	VGG16 reentrenando 2 capa	RMSProp	0.001	ImageNet	0.895

anterior y las técnicas de entrenamiento fueron 2 desde cero para la arquitectura LeNet5; mientras que, para VGG16 y VGG19 se optó por usar aprendizaje por transferencia con la técnica de extracción de características; un ensayo fue descongelando solo la capa convolucional 3 del bloque 5 (block_conv3), como se explicó en la etapa 1 y otro ensayo donde se descongelaron las capas 1 y 2 del bloque 5 (block5_conv1 y block5_conv2) para cada arquitectura VGGNet, haciendo un total del 100 ensayos.

Además, se realizó un reescalado de imágenes a (256, 256, 3) y se estableció un lote por época de 11 imágenes, para el entrenamiento se utilizó Keras, con Python 3.7 y una versión de Tensorflow de 2.1.0

2.8. Métricas de evaluación

Para valorar el rendimiento de cada tratamiento se emplearon medidas de precisión, exactitud, puntuación F1 y también se analizó la matriz de confusión, consultar [9].

3. Resultados

Los mejores 10 resultados experimentales de la etapa 2, se encuentran en la Tabla 1. Se puede observar que la exactitud se encuentra entre los intervalos de 0.895 a 0.908. y que en los tratamientos 7 y 28, con la arquitectura LeNet5 y VGG16 son los que tienen la mejor exactitud con 0.908 y 0.909 respectivamente.

Tabla 2. Comparación de métricas de desempeño de tratamientos 6, 7, 11, 16, 26, 27, 28 y 46. Se utilizaron 54 imágenes para 'Validación fresa muy madura', 68 para 'Validación fresa madura' y 106 para 'Validación fresa inmadura'.

LeNet5, tratamiento 6			
Categoría	**Precisión**	**Sensibilidad**	**Puntuación F1**
Validación fresa muy madura	0.95	0.7	0.81
Validación fresa madura	0.79	0.93	0.85
Validación fresa inmadura	0.97	0.99	0.98
Promedios	0.903	0.873	0.880
LeNet5, tratamiento 7			
Categoría	**Precisión**	**Sensibilidad**	**Puntuación F1**
Validación fresa muy madura	0.8	0.91	0.81
Validación fresa madura	0.86	0.81	0.83
Validación fresa inmadura	0.99	0.96	0.98
Promedios	0.883	0.893	0.873
LeNet5, tratamiento 11			
Categoría	**Precisión**	**Sensibilidad**	**Puntuación F1**
Validación fresa muy madura	0.94	0.81	0.87
Validación fresa madura	0.86	0.91	0.89
Validación fresa inmadura	0.97	1	0.99
Promedios	0.923	0.907	0.917
LeNet5, tratamiento 16			
Categoría	**Precisión**	**Sensibilidad**	**Puntuación F1**
Validación fresa muy madura	0.81	0.87	0.84
Validación fresa madura	0.89	0.82	0.85
Validación fresa inmadura	0.99	1	1
Promedios	0.897	0.897	0.897
VGG16 descongelando 1 capa, tratamiento 26			
Categoría	**Precisión**	**Sensibilidad**	**Puntuación F1**
Validación fresa muy madura	0.84	0.78	0.81
Validación fresa madura	0.76	0.88	0.82
Validación fresa inmadura	0.98	0.92	0.95
Promedios	0.860	0.860	0.860
VGG16 descongelando 1 capa, tratamiento 27			
Categoría	**Precisión**	**Sensibilidad**	**Puntuación F1**
Validación fresa muy madura	0.82	0.83	0.83
Validación fresa madura	0.78	0.88	0.83
Validación fresa inmadura	0.99	0.9	0.94
Promedios	0.863	0.870	0.867
VGG16 descongelando 1 capa, tratamiento 28			
Categoría	**Precisión**	**Sensibilidad**	**Puntuación F1**
Validación fresa muy madura	0.86	0.8	0.83
Validación fresa madura	0.79	0.85	0.82
Validación fresa inmadura	0.95	0.94	0.95
Promedios	0.867	0.863	0.867
VGG16 descongelando 2 capas, tratamiento 46			
Categoría	**Precisión**	**Sensibilidad**	**Puntuación F1**
Validación fresa muy madura	0.8	0.81	0.81
Validación fresa madura	0.78	0.82	0.8
Validación fresa inmadura	0.96	0.92	0.94
Promedios	0.847	0.850	0.850

En la Tabla 2 se reportan las comparaciones de las métricas de precisión, sensibilidad y F1 de las arquitecturas con mejor exactitud de la Tabla 1 (tratamientos 6, 7, 11, 16, 26, 27, 28 y 46).

Se determinó que el tratamiento 11 de la Tabla 2 con arquitectura LeNet5, con una tasa de aprendizaje de 0.001 y optimizador RMSProp fue el tratamiento con mejor

desempeño, con una exactitud de 0.906 y los mejores promedios en las métricas de desempeño; 0.923 en precisión, 0.907 en sensibilidad y 0.917 en puntuación F1.

En la evaluación de la matriz de confusión, se utilizaron 228 imágenes y el desempeño en validación se muestra en la Fig. 7 sin normalizar y en la Fig 8 normalizada.

4. Discusión

La arquitectura LeNet5 con una tasa de aprendizaje de 0.001 y un optimizador RMSProp obtuvo un 83% en la fresa muy madura, un 90% en la fresa madura y un 100% en la fresa inmadura en la matriz de confusión y una exactitud de 90.6%. Xiao Qin Yue y colaboradores [18], en su artículo, informaron que con teléfonos inteligentes (smartphones) tiene un 97% en fresa madura, un 87% en fresa casi madura y un 100% en fresa inmadura en su matriz de confusión de testeo, una exactitud de 94.44%, una precisión promedio de 94.85%, una sensibilidad promedio de 94.45% y una puntuación F1 promedio de 94.43%. Yuanyuan Shao y su equipo [12] reportaron que se obtuvo una exactitud en campo de hasta 96.7% usando imágenes hiperespectrales de fresa en 3 etapas de madures (madura, semi madura y no madura) y métodos como LS-SVM (Máquinas de Soporte Vectorial de Mínimos Cuadrados), concluyendo que la imagen hiperespectral puede utilizarse para la evaluación en tiempo real de la madurez de las fresas en el campo. Según Ji-Young Choi y su grupo de investigación [3], desarrollaron modelos de redes neuronales convolucionales (CNNs) para clasificar la calidad externa de fresas (frescas, magulladas o mohosas) utilizando 750 imágenes RGB. Se utilizaron 8 configuraciones para comparar las precisiones de validación según variables como la distribución de datos, el número de imágenes y las épocas de entrenamiento. Se observó un alto rendimiento de aprendizaje en poco tiempo cuando se utilizó el 90% de los datos de imagen para el entrenamiento y la validación. Las métricas de rendimiento fueron exactitud (AC) promedio de 97%, precisión (PR) promedio de 95.6%, especificidad (SP) promedio de 95.6%, sensibilidad (SE) promedio de 97.8% y puntuación F1 promedio de 95.6% del modelo 90% de entrenamiento y 10% de testeo. Además, las CNNs identificaron áreas dañadas en las fresas, mostrando su potencial para el monitoreo no destructivo en la industria alimentaria. Mientras que Liane Angelo Acero y colaboradores [1] emplearon imágenes clasificadas como deseables e indeseables, utilizando 350 imágenes de cada conjunto para el entrenamiento, 200 imágenes para la validación y 100 imágenes para la prueba. El modelo de CNN generado se simuló aplicando épocas (epoch) = 15 y un tamaño de lote (batch size) = 8. Obtuvieron una precisión de entrenamiento del 98.41%, una precisión de validación del 92.75%, y una precisión de prueba del 100%. Por ultimo Hossein Azizi y su equipo [2] utilizaron un conjunto de datos que contiene 800 imágenes de fresas en 4 clases (inmadura, semi-madura, madura y dañada), con 3 escenarios; el primero donde utilizan los datos originales; el segundo empleando técnicas fundamentales de aumento de datos (Fundamental Data Augmentation, FDA); y el tercero con la estrategia de aprendizaje para aumentar (Learning-to-Augment Strategy, LAS) y con modelos preentrenados (GoogleNet, ResNet18 y ShuffleNet) alcanzaron exactitudes del 96.88%, 97.50% y

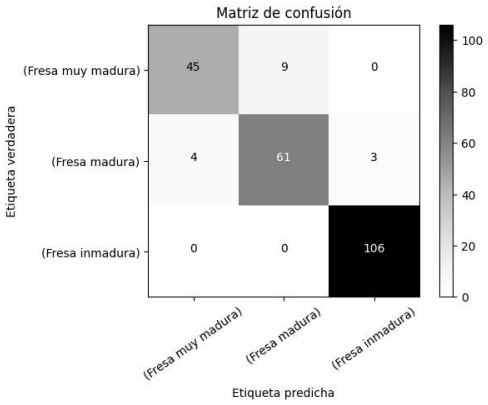

Fig. 7. Matriz de confusión normalizada de validación de LeNet5 con tratamiento 11.

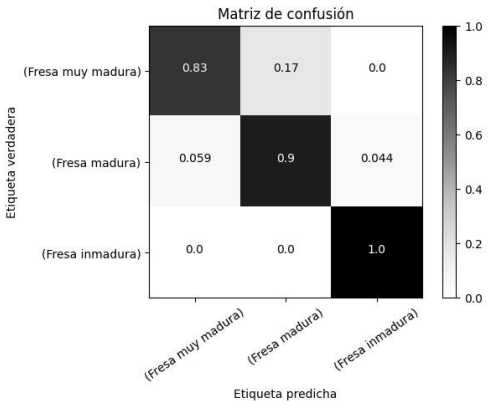

Fig. 8. Matriz de confusión normalizada de validación de LeNet5 con tratamiento 11.

98.85%, precisiones promedios del 96.93%, 97.73%, 98.81% sensibilidades promedios del 96.87%, 97.5% y 98.75% y especificidades promedios del 98.96%, 99.17% y 99.58% respectivamente con GoogleNet.

En cuanto a la escalabilidad, el método propuesto es una buena alternativa para ser implementado en aplicaciones más amplias de clasificación de madurez de frutas. El uso de las CNN permite procesar grandes volúmenes de datos, lo que es adecuado para sistemas de clasificación en tiempo real en entornos agrícolas. Este modelo mostró tener una alta eficiencia (exactitud de 90.6\%). Además, debido a su arquitectura flexible, el modelo puede ser adaptado fácilmente para clasificar diferentes tipos de frutas ajustando mínimamente los parámetros de entrenamiento, lo que sugiere su aplicabilidad en una variedad de contextos agrícolas. Sin embargo, se debe considerar probar este sistema en entornos agrícolas reales, ya que puede ayudar a identificar y corregir posibles deficiencias en el rendimiento del modelo en situaciones prácticas.

La implementación de sistemas de clasificación de madurez de frutas basados en CNN ofrece importantes beneficios para la industria agrícola y alimentaria.

Dicha clasificación optimiza las condiciones de almacenamiento, reduciendo la pérdida de calidad durante este proceso y prolongando la vida útil de las frutas. Esto también facilita una mejor planificación logística en la distribución, asegurando que las fresas lleguen frescas a los puntos de venta y mejorando la satisfacción del cliente mediante una calidad uniforme del producto.

Los datos generados en este estudio, proporciona información valiosa para la toma de decisiones a lo largo de toda la cadena de suministro. La tecnología basada en CNN también es altamente adaptable y escalable, permitiendo su aplicación en diversas áreas agrícolas y geográficas. En conjunto, estos avances tecnológicos pueden transformar significativamente la eficiencia y la calidad en la industria agrícola.

Finalmente, el despliegue de sistemas de inteligencia artificial (IA) en la agricultura y las cadenas de suministro de alimentos tiene importantes implicaciones éticas y sociales. Los beneficios incluyen una mayor eficiencia y productividad, mejor calidad y seguridad alimentaria, reducción de costos y datos más precisos para la toma de decisiones. Es importante abordar estos desafíos mediante políticas y regulaciones adecuadas, en donde se considere la capacitación para agricultores y mejoras continuas.

5. Conclusión

Se determinó que la red neuronal con arquitectura LeNet5, entrenada desde cero con una tasa de aprendizaje de 0.001 y el optimizador RMSProp, logra clasificar correctamente fresas inmaduras, maduras y muy maduras con una exactitud del 90.6%. Además, obtuvo los mejores promedios en las métricas de desempeño: 92.3% en precisión, 90.7% en sensibilidad y 91.7% en puntuación F1.

Aunque estos resultados son prometedores, existen varias oportunidades de mejora.

Una de las más importantes es aumentar el conjunto de datos de imágenes para el entrenamiento, validación y prueba, y asegurar un equilibrio en el número de imágenes para las diferentes clases de madurez de la fruta. También es necesario diversificar las condiciones de las imágenes de prueba, obteniéndolas en diferentes condiciones ambientales y considerando factores no observados previamente, como el desenfoque o la iluminación.

Referencias

1. Acero, L.A., Ong, J.D., Shi, C.J., Dadios E.P.: Strawberry Quality Classification Utilizing Convolutional Neural Network. In: IEEE 13th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM), pp. 1–4 (2021) doi: 10.1109/HNICEM54116.2021.9731924.
2. Azizi, H., Asli-Ardeh, E.A., Jahanbakhshi, A.: Vision-based Strawberry Classification Using Generalized and Robust Deep Networks. Journal of Agriculture and Food Research, 15(100931) (2024) doi: 10.1016/j.jafr.2023.100931.
3. Choi, J.Y., Seo, K., Cho, J.S.: Applying Convolutional Neural Networks to Assess the External Quality of Strawberries. Journal of Food Composition and Analysis, 102(104071) (2021) doi: 10.1016/j.jfca.2021.104071.

4. Instituto Tecnológico Superior de Coalcomán: Manual de Producción de fresa en Coalcomán Michoacán (2018)
5. Secretaría de Economía: NMX-FF-062-SCFI-2002. Productos alimenticios no industrializados para consumo humano - fruta fresca - fresa (Fragaria x ananassa, dutch) – especificaciones y método de prueba (2002) http://www.economia-nmx.gob.mx/normas/nmx/2002/nmx-ff-062-scfi-2002.pdf.
6. Du, X., Bai, J., Plotto, A.: Electronic Nose for Detecting Strawberry Fruit Maturity. Proc. Fla. State Hort. Soc., 123, pp. 259–263 (2010)
7. FAOSTAT: Base de datos estadísticos de la FAO (2022) http://www.fao.org/faostat/es/#data.
8. Ganesh, P., Volle, K., Burks, T.F.: Deep Orange: Mask R-CNN Based Orange Detection and Segmentation. IFAC-PapersOnline, 52(30), pp. 70–75 (2019) doi: 10.1016/j.ifacol.2019.12.499.
9. Géron, A.: Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow Concepts, Tools, and Tchniques to Build Intelligent Systems. O'Reilly (2019)
10. Islam, M.R., Matin, A.: Detection of COVID 19 from CT Image by the Novel LeNet-5 CNN Architecture. In: 23rd International Conference on Computer and Information Technology (ICCIT), pp. 1–5 (2020) doi: 10.1109/ICCIT51783.2020.9392723.
11. LeCun, Y., Bottou, L., Bengio, Y.: Gradient-based Learning Applied to Document Recognition. Proceedings of the IEEE, 86(11), pp. 2278–2324 (1998) doi: 10.1109/5.726791.
12. Shao, Y., Wang, Y., Xuan. G.: Assessment of Strawberry Ripeness Using Hyperspectral Imaging. Analytical Letters, 54(10), pp. 1547–1560 (2020) doi: 10.1080/00032719.2020.1812622.
13. SIACON: Sistema de información agroalimentaria de consulta. Modulo agrícola estatal del SIACON-NG. México: SIAP-SADER (2022) https://www.gob.mx/siap/documentos/siacon-ng-161430.
14. Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. ARXiv 1409.1556 (2014)
15. Torres, J.: Python Deep Learning: Introducción práctica con Keras y TensorFlow 2. Alpha Editorial (2020)
16. Wang, Z., Jin, L., Wang, S.: Apple Stem/Calyx Realtime Recognition Using YOLO-V5 Algorithm for Fruit Automatic Loading System. Postharvest Biology and Technology, 185, pp. 111808 (2022) doi: 10.1016/j.postharvbio.2021.111808.
17. Yang, H., Ni, J., Gao, J.: A Novel Method for Peanut Variety Identification and Classification by Improved VGG16. Scientific Reports, 11(1) pp. 15756 (2021)
18. Yue, X.Q., Shang, Z.Y., Yang, J.Y.: A Smart Data-Driven Rapid Method to Recognize the Strawberry Maturity. Information Processing in Agriculture, 7(4), pp. 575–584 (2020) doi: 10.1016/j.inpa.2019.10.005.
19. Zhou, X., Ampatzidis, Y., Lee, W.S.: Deep Learning-Based Postharvest Strawberry Bruise Detection Under UV and Incandescent Light. Computers and Electronics in Agriculture, 202, pp. 107389 (2022) doi: 10.1016/j.compag.2022.107389.
20. Zhou, X., Lee, X.S., Ampatzidis, Y.: Strawberry Maturity Classification from UAV and Near-Ground Imaging Using Deep Learning. Smart Agricultural Technology, 1 (2021) doi: 10.1016/j.atech.2021.100001.

ISSN 1870-4069

Adversarial Attacks in Word Processing: Impact on SPAM Detection Models

Samantha Acosta Ruiz

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

224570157@viep.com.mx

Abstract. Word processing in machine learning faces several challenges, such as language variability, semantic ambiguity, and the need to correctly interpret context. These challenges are intensified in spam detection, where attackers use sophisticated techniques to evade detection. This study examines the impact of poisoning and evasion attacks on machine learning models, highlighting that the former have a considerably greater effect. These attacks modify the training data, deteriorate the performance of the model and cause systematic errors in the predictions. The spam data class is especially affected, which compromises the model's ability to detect unwanted emails. For the generation of adversarial examples, a DistilBERT classifier was used. The models evaluated included AdaBoost, Support Vector Machine and RandomForest. To measure the impact of the attacks, metrics such as accuracy, recall, precision and F1-score were analyzed. The results showed that the most vulnerable model to these attacks was AdaBoost, which initially had an accuracy of 97.8% in classifying spam data. However, the poisoning attack turned out to be the most harmful, reducing this metric to 69%. This analysis highlights the importance of implementing robust defenses against poisoning attacks and developing advanced word-processing techniques to maintain the integrity and effectiveness of models in adverse environments.

Keywords: Adversarial machine learning, spam detection, generative adversarial example, distilBERT, text classification.

1 Introduction

Within the field of cybersecurity, it has been consistently recognized that human beings represent the most vulnerable link and the first line of vulnerability in any system. Although conventional threats can be assessed and quantified through penetration testing, social engineering (SE) poses more subtle and complex challenges. Social engineers use various tactics, such as phishing and adware, to manipulate users and obtain information voluntarily. In the context of social networks, social engineering takes on an appearance that resembles regular publications, albeit with a latent

malicious background. Intruders have the ability to impersonate legitimate entities, such as banking institutions in order to gain access to accounts or passwords by sending spam messages, these attacks usually start with meticulous and planned reconnaissance phases. There is little understanding in identifying SE attacks due to the subjectivity of social media posts [3]. In addition, it is challenging for a machine to recognize sentiments and read between the lines of social posts that may be a SE threat. Common network security appliances, such as firewalls, only detect illegitimate traffic at flow levels.

Meanwhile, the Intrusion Detection System (IDS)/Intrusion Prevention System (IPS) must be highly customized with intelligent rules to read application data [10]. In recent years, there has been much work focused on training deep learning models for threat intelligence using network properties, but SE attacks are mostly textual data that require integration with Natural Language Processing (NLP) [12].

In the fields of security and how to protect Artificial Intelligence (AI) models against similar threats, we have to talk about a particular area of Machine Learning (ML), in this case it is the Adversarial Machine Learning. When deploying on Machine Learning (ML)/Deep Learning (DL) model, it should be expected that the models do not present any variability, but this is not true in most cases. This is the central theme of Adversarial Machine Learning, which is a branch of ML that tries to find out what attacks a model can suffer in the presence of a malicious adversary and how to protect against them. There are three types of attacks [4]:

- **White box attacks:** The adversary has access to the architecture used by the model, training data, parameters and hyperparameters.
- **Black box attacks:** The adversary only has access to the inputs and outputs of the model.
- **Gray box attacks:** The attack is located at an intermediate point between the data types of previous attacks.

Although it is possible to carry out numerous attacks on machine learning models.

In Adversarial Machine Learning these attacks fall into four categories: poisoning, evasion, extraction and inversion [7]. These attacks exploit vulnerabilities inherent in ML models, leading to serious consequences in critical applications, such as security, health and finance.

Figure 1 shows a representation of where each attack would influence the lifecycle of the ML model, providing a clear view of the critical points of vulnerability. These attacks constitute significant threats to the integrity and security of Machine Learning Systems. The following are the types of attacks illustrated in Figure 1:

- **Poisoning Attacks:** Poisoning attacks are also known as causative attacks. The adversary tries to corrupt the training set with the aim that the learned model produces a misclassification that benefits him. This type of attack is carried out in the training phase, and it is very difficult to find the underlying vulnerability, since it can be transmitted to all models that use these data. This kind of attack can be carried out both in white box and black box, and compromises the availability of the models.

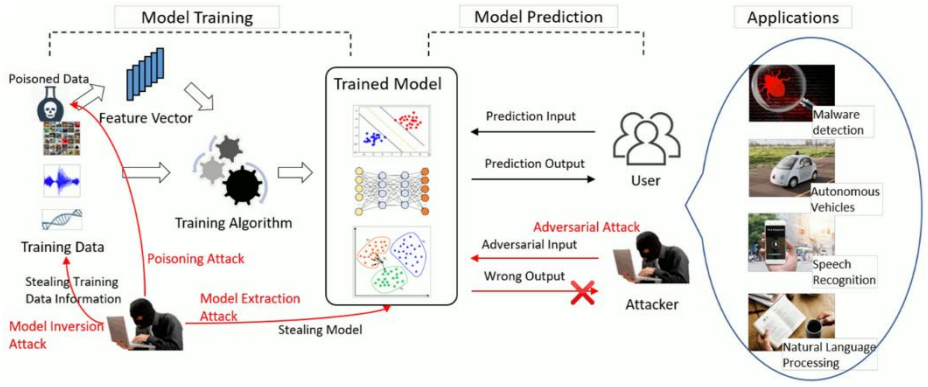

Fig. 1. Classification of adversarial machine attacks.

- **Evasion attacks:** Evasion attacks are known as exploratory attacks. The goal of the adversary is to inject a small amount of noise into the input so that a classifier predicts the output (or label) incorrectly. These noisy inputs, called adversary examples, can be created on different types of data, with the most widespread and well-known examples being images. In fact, adversarial examples of images are created so that they are imperceptible to the human eye.
- **Extraction attacks:** Model extraction attacks consist of an adversary trying to steal the parameters of a machine learning model. This type of attack allows to compromise the intellectual and confidential property of a model, and allows to carry out evasion and/or inversion attacks. This attack can be performed in both white box and black box.
- **Inversion attacks:** The model inversion attacks, consist in that an adversary trying to take advantage of the predictions of the model to compromise the user's privacy or infer whether or not certain data was used in the training set. It can be used in both white box and black box. This type of attack is especially relevant among models that have been trained with sensitive data such as clinical data, which require special protection.

1.1 Related Works

In text data, the discrete nature of the inputs makes the gradient-based attack on images no longer applicable and forces people to craft discrete perturbations on different granularities of text (character-level, word-level, sentence-level, etc.). In this section, we introduce the related work in attacking NLP architectures for different tasks [15].

The work [1] it is an analysis of the generation of abstracts of scientific articles exploring different types of architectures. For this work, 227 NLP articles applied to the tourism sector were collected and 3 types of initial representations were tested to generate the summaries. One based on the title of the article, another adding keywords

and finally another adding important research data. The results indicate that using a method with pre-training, such as GPT-3, can obtain good performance in the task.

The work [2] focuses on the analysis and classification of a set of texts labeled disaster and non-disaster, where those labeled as non-disaster include metaphorical context. The classification is focused on classical models, such as RandomForest, Support Vector Machine, Decision Tree and XGBOOST. This is achieved using SentenceBERT and n-grams as feature extractors, aiming to assess the significance of feature selection in identifying relationships between texts and the importance of specific words.

The work Hot-Flip [5] considers replacing a letter in a sentence in order to mislead a character-level text classifier (each letter is encoded in a vector). The attack algorithm manages to achieve this by finding the most influential letter replacement via gradient information. These adversarial perturbations can be noticed by human readers, but they do not change the content of the text as a whole, nor do they affect human judgments.

The work [11] considers manipulating the victim sentence at the word and phrase level. They try adding, removing or modifying the words and phrases in the sentences. In their approach, the first step is similar to Hot-Flip [5]. For each training sample, they find the most influential letters, called “hot characters”. Then, they label words that have more than 3 “hot characters” as “hot words”. “Hot words” compose “hot phrases”, which are the most influential phrases in sentences. Manipulating these phrases is likely to influence the model prediction, so these phrases compose a “vocabulary” to guide the attack. When an adversary is given a sentence, he can use this vocabulary to find the weakness of the sentence, add one hot phrase, remove a hot phrase in the given sentence, or insert a meaningful fact that is composed of hot phrases.

Deep Word Bug [6] and Text-Bugger [9] are black box attack methods for text classification. The basic idea of the former is to define a scoring strategy to identify the key tokens that will lead to a wrong prediction of the classifier if modified. Then they try four types of “imperceivable” modifications on such tokens: swap, substitution, deletion and insertion, to mislead the classifier. The latter follows the same idea, and improves it by introducing new scoring functions.

The works of Samanta and Mehta [13], Iyyer et al. [8] start by crafting adversarial sentences that grammarly correct and maintain the syntax structure of the original sentence. Samanta and Mehta [13] achieve this using synonyms to replace original words, or by adding some words that have different meanings in different contexts. On the other hand, Iyyer et al. [8] manage to fool the text classifier by paraphrasing the structure of sentences.

Therefore, this paper analyzes how word processing influences adversary attacks, using a SPAM database to create examples of adversaries. Subsequently, these examples will be tested with classifiers to perform a poisoning attack and determine if they detect them. The objective is to analyze word processing with a focus on creating adversarial examples for spam detection models, aiming to cause the most damage with the least amount of disturbance.

Table 1. Representation of a sample of the dataset.

Category	Message
Ham	So how many days since then?
Ham	Even u dont get in trouble while convincing..j…
Ham	So anyways, you can just go to your gym or wha…
Spam	You have been specially selected to receive a …
Spam	You will be receiving this week's Triple Echo…

Table 2. Representation of the processed text.

Category	Message
Ham	mani day sinc
Ham	even u dont get troubl convinc tel twice tel n…
Ham	anyway go gym whatev love smile hope ok good d…
Spam	special select receiv 3000 award call 08712402…
Spam	receiv week triple echo rington shortli enjoy

2 Methodology

In the development of this work, a Kaggle text database was downloaded for further analysis and testing [14]. The environment used was Google Colaboratory. To start the process, several libraries are used, including NLTK for handling and processing text files in Python, as well as Scipy, Scikit-Learn, Numpy and Matplotlib. Once these libraries are imported, the file with the text messages is loaded from the local environment. The file is analyzed to verify that the database contains texts in English and has no empty or null data; in this case, the data set was complete. The database consists of two columns: one for messages and one for tags. This particular database contains SPAM messages, which are divided into normal messages (Ham) and SPAM messages, with 4825 and 747 messages respectively, as shown in Table 1.

Then the following text preprocessing is performed: a function is applied to convert the text to lowercase. The text is divided into a word list. Special characters and empty words are removed. Subsequently, the words are reduced to their basic form (for example, "running" becomes "run") and punctuation marks are removed. In Table 2 you can see how the processed text from the dataset turned out.

Then, we define a function called make-adversarial that takes a sample of text and generates an adversarial version of it. An adversarial sample is a modified version of the original text that can change the classification of the machine learning model (for example, from "spam" to "ham").

The make-adversary function modifies the original text by replacing or deleting words to change their classification, thus generating an adversarial version of the text.

Based on the code suggested by [13], where three different types of modifications are proposed to alter a regular entry in an adversarial sample: (i) replacement, (ii) insertion and (iii) deletion of words in the text. Its objective is to change the class label of the sample by means of a minimum number of alterations. The pseudocode of the proposed method is given in Algorithm 1.

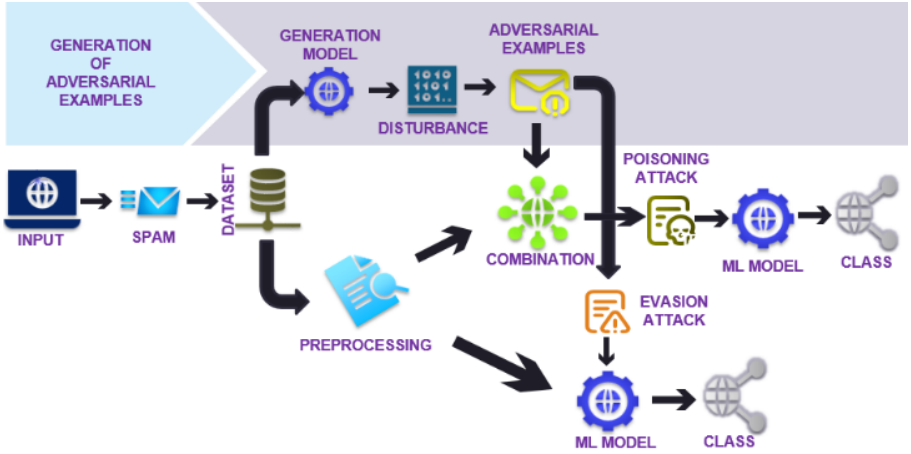

Fig. 2. Outline of the general process of the stages of elaboration of the project.

Algorithm 1. Convert the input text *s* into an adversarial sample

Require: Sample text –*s*, Classifier trained for sentiment analysis *F*.
1: Find class-label *y* of the sample *s: y* ← *F(s)*
2: Find contribution $C_F\ (w_i, y)$ of each word w_i towards determining the class-label of the sample *s* with the respect to the classifier *F*.
3: Rank words according to $C_F\ (\mathbf{w}, y) : \mathbf{w} \rightarrow \{w_1, w_2, \cdots, w_n\}$ where, $C_F\ (w_1, y) > C_F\ (w_2, y) > \cdots > C_F\ (w_n, y)$
4: **while** *y* does not change **do**
5: **if** w_i is an Adverb and $C_F\ (w_i, y)$ is considerably high **then**
6: Remove w_i from *s*
7: **else**
8: Consider a candidate pool $P = \{pk\} \forall k$ for w_i
9: $j \leftarrow$ argmin $C_F\ (p_k, y) \forall\ p_k \in P$
10: k
11: **if** w_i is Adjective and p_j is Adverb **then**
12: Add p_j before w_i in *s*
13: **else**
14: Replace w_i with p_j in *s*
15: **end if**
16: **end if**
17: $y \leftarrow F(s)$
18: $i \leftarrow i + 1$
19: **end while**

Following the process described in the previous pseudocode, a DistilBERT classifier was used to generate adversaries. This model had previously been refined with different datasets, different from those of the current project. To configure it, the HuggingFace Transformers library was used. DistiBERT has its own tokenizer, which preprocesses the texts to adapt them to the model. Since DistilBERT does not support texts larger than 512 tokens, a dictionary was established that instructs the tokenizer to trim texts

to 512 tokens and add padding if necessary, ensuring that all texts are of the same size. Subsequently, the SPAM dataset was uploaded.

The main purpose of the make-adversary function is to generate an adversary text from an original text string. To do this, before applying the algorithm that introduces the modifications to the text, Spacy is used to preprocess and tokenize the text, obtaining a list of Python objects with linguistic information for each token. The spacy-load command loads a language model included in the library, and the model returns a document object. The main idea for creating an adversary is to introduce changes that might confuse a classifier trained in sentiment analysis. DistilBERT is trained on an SPAM dataset. This classifier makes an initial prediction about the text and the result is saved in the sample-class variable.

Then, for each word, its contribution is calculated, which is obtained by deleting the word and using the score returned by the classifier as a contribution measure. The word and its contribution are stored in a dictionary in an orderly way. A high contribution value is obtained using percentiles of the calculated contributions, in this case equivalent to the 90th percentile. Each of the tokens of the text is traversed according to its contribution. It is checked if the token is an adverb with a high contribution and if so, the token is deleted. Otherwise, a second modification is carried out. As a second modification, the token is replaced by some synonym using NLTK.

The WordNet thesaurus is used to find synonyms for that word. If no synonyms are found, the algorithm proceeds with the next word. The best synonym will be stored in the variable P. If there is more than one possible synonym, they are all evaluated by replacing the word with each synonym and checking the classifier score. The synonym that causes the worst score is selected. Then, it is checked whether the original word is an adjective and the synonym is an adverb; if so, the synonym is inserted before the original word. Otherwise, the synonym replaces the original word.

Finally, after any modification, it is evaluated whether it causes a change in the classifier prediction. If so, the class and the adversary text are returned. If, after making all the modifications, the classifier still does not change the class of the original text, None is returned, indicating that the text cannot be corrupted. It is important to note that the function that creates the adversary dataset uses parallelism. The Python multiprocessing library is used, and with the reserved words ”with... as...” a managed context is set where the memory used by the pool variable is freed.

A number of 9 cores and a chunk size of 32 were used. The function ”Pool.imap-unordered” obtains the results in parallel within an iterator. This function automatically distributes the list of texts between the different threads for parallel processing. It is observed in Table 3, examples of the adversary dataset that was generated.

Table 3. Examples of Dataset with the adversary text.

Category 1	Message-G
Spam	urgent! call-option 09066350750 from your land…
Ham	live you uncommitted for soirée along June third?
Ham	Or just brawl that 6times
Ham	That make-up random watch my honest-to-god roo…
Ham	mail ME ir emasinglel singlecalciferol soon

Table 4. Results with the original data with the test set.

Model	Accuracy	Precision	Recall	F1-Score	Class
SVM	0.9865470	0.99	1.00	0.99	0
SVM	0.9865470	0.99	0.92	0.95	1
RandomForest	0.9793721	0.98	1.00	0.99	0
RandomForest	0.9793721	1.00	0.86	0.93	1
AdaBoost	0.9784753	0.98	0.99	0.99	0
AdaBoost	0.9784753	0.96	0.90	0.93	1

Figure 2 shows the general process followed in the methodology, as well as the procedure used to carry out the poisoning and evasion attacks. The data generated by the adversarial example generator was used to execute these attacks. To evaluate the behavior of the data, the SVM, Random Forest and Adaboost classifiers were used.

3 Results

In this section, it will be explained how the poisoning and evasion attacks were carried out with the aim of producing an incorrect classification (see Figure 2). To do this, the SVM, Random Forest and AdaBoost models were used, initially, to analyze the behavior of these models in the face of this type of disturbances and observe their performance. The SVM, Random Forest and Adaboost classifiers were chosen for their efficiency and particular characteristics: SVM excels at classifying high-dimensional data, Random Forest is robust against disturbances and handles complex data well, and Adaboost improves performance by combining several weak models. These models make it possible to compare how different classification approaches respond to adversary attacks.

The hyperparameters used were as follows: For SVM, a linear core with a value of C=1.0 was used. A number of estimators of 100 and a random state of 42 were used for Random Forest. For AdaBoost, it was configured with a number of estimators of 50 and a random state of 42. First of all, we will describe how the final dataset was formed, which contains five columns: the original message, the original tag, the preprocessing of the original message, the adversary message and its adversary tag.

It should be noted that no pre-processing was performed on the adversary message, as this could eliminate the noise that had been included in said message, therefore, this option was discarded. Classes are represented, 0 is not spam and 1 is spam. The data set was divided into 80% for training and 20% for testing, and a 42 seed was established

Table 5. Results with pre-processed data with the test set.

Model	Accuracy	Precision	Recall	F1-Score	Class
SVM	0.9874439	0.99	1.00	0.99	0
SVM	0.9874439	1.00	0.92	0.96	1
RandomForest	0.9802690	0.98	1.00	0.99	0
RandomForest	0.9802690	1.00	0.87	0.93	1
AdaBoost	0.9704035	0.97	0.99	0.98	0
AdaBoost	0.9704035	0.95	0.85	0.90	1

for random status. Tables 4 and 5 show the results of the training of the non-attack models.

From Tables 4 and 5, it is concluded that SVM maintains a very high performance in both data sets, with an overall accuracy close to 99%. Random Forest also shows a high performance, which improves slightly with the pre-processed data, suggesting that it is sensitive to pre-processing and improves its generalizability. On the other hand, AdaBoost shows a slight decrease in accuracy with pre-processed data, indicating that, although it is robust, it is more sensitive to variations in the data.

Text preprocessing seems to have a positive or neutral effect on the performance of SVM and Random Forest, while for AdaBoost, the performance decreases slightly with preprocessed data. This suggests that SVM and Random Forest are more resistant to changes in input data, while AdaBoost may be more sensitive to text preprocessing. The impact of text preprocessing varies between models: SVM shows high and consistent performance on both data sets, suggesting that preprocessing can be beneficial by maintaining accuracy and consistency.

Random Forest experiences a slight improvement with pre-processing, indicating better management of the text features after the initial processing. In contrast, AdaBoost exhibits a decrease in performance with pre-processed data, suggesting a possible loss of crucial information during the pre-processing process.

To analyze the poisoning attack, the variables of the original message with its preprocessing and the adversary message were concatenated into a single column. The same procedure was also performed for the labels. The results obtained are shown in Table 6. The results obtained by applying the poisoning attack to the dataset show a significant decrease in the performance of all models compared to the original or pre-processed data. The poisoning attack has significantly degraded the performance of all models, especially affecting the ability to correctly identify class 0 examples.

SVM and Random Forest show greater comparative resilience, maintaining a better performance in class 1, while AdaBoost is the most vulnerable to attack, with a more pronounced drop in its overall performance. These results underscore the importance of considering robustness against poisoning attacks when selecting or designing models for critical tasks.

Table 6. Results by applying the poisoning attack to the dataset.

Model	Accuracy	Precision	Recall	F1-Score	Class
SVM	0.7354260	0.76	0.41	0.54	0
SVM	0.7354260	0.73	0.92	0.82	1
RandomForest	0.7479820	0.81	0.41	0.55	0
RandomForest	0.7479820	0.73	0.94	0.83	1
AdaBoost	0.6905829	0.66	0.32	0.44	0
AdaBoost	0.6905829	0.70	0.90	0.79	1

Table 7. Results by applying the evasion attack to the dataset.

Model	Accuracy	Precision	Recall	F1-Score	Class
SVM	0.9883408	0.99	1.00	0.99	0
SVM	0.9883408	0.99	0.93	0.96	1
RandomForest	0.9748878	0.97	1.00	0.99	0
RandomForest	0.9748878	1.00	0.83	0.91	1
AdaBoost	0.9757847	0.98	0.99	0.99	0
AdaBoost	0.9757847	0.94	0.90	0.92	1

The analysis of the evasion attack consisted of introducing a small amount of noise at the entrance. In this case, the original labels were used to observe if there was any disturbance in the results, achieving an incorrect prediction as can be seen in Table 7.

The results obtained after applying the evasion attack to the dataset indicate that, despite the disturbances designed to evade detection, the models maintain a relatively high performance. All models retain high accuracy, which suggests remarkable robustness against this type of attack. However, a slight decrease in performance is observed for Class 1 in all models, particularly in Random Forest and AdaBoost. This suggests that the evasion attack has a more significant impact on the classification of certain examples, although not critically. In summary, the models are still effective against evasion attacks, but they present a slight vulnerability that could be exploited under certain conditions.

4 Conclusion

In this paper, the study has shown that, under normal conditions, the SVM, Random Forest and AdaBoost models exhibit outstanding performance in terms of accuracy, Recall and F1-Score. These models show high effectiveness in class detection without the influence of adversary attacks. SVM stands out for its consistency and accuracy close to 99%, while Random Forest and AdaBoost also present solid performance, although with some variations depending on data preprocessing. However, SVM and Random Forest show greater robustness and consistency against adversary attacks compared to AdaBoost.

All models suffer a decrease in performance when faced with poisoning and evasion attacks. The results suggest that SVM and Random Forest have a greater ability to maintain their accuracy and generalization in adverse conditions, while AdaBoost, although effective in normal scenarios, needs improvements to more effectively handle

adversary attacks. These findings highlight the importance of developing additional mitigation and robustness methods to face adversary attacks in classification systems, with the aim of improving the security and stability of the models in real environments.

In addition, poisoning attacks have a significantly greater impact on machine learning models compared to evasion attacks. Poisoning attacks directly alter the training data, degrading the overall performance of the model and leading to systematic errors in its predictions. These attacks manipulate training data, making the model less effective at identifying unwanted emails and negatively affecting its accuracy and detection ability.

In future work, the creation of a generative adversary text model of its own will be explored and more classification models will be analyzed to assess its vulnerability to these attacks.

Acknowledgments. The first author thanks the support provided by the CONAHCYT scholarship number 1106756.

References

1. Alcantara, T., Garcia, O., Calvo, H.: Analysis and Classification of Contextual Disaster Tweets. Research in Computing Science, 152(7), pp. 163–175 (2023)
2. Alvarez, M., Aranda, R., Diaz, A.: Automatic Generator of Scientific Summaries in Tourism Research. Research in Computing Science, 151(5), pp. 5–14 (2022)
3. Aun, Y., Gan, M.L., Wahab, N.: Social Engineering Attack Classifications on Social Media Using Deep Learning. Computers Materials Continua, 74(3), pp. 4917–4931 (2023) doi: 10.32604/cmc.2023.032373.
4. Calvo, J.: Can Deep Learning Be Fooled? (2022) https://www.europeanvalley.es/noticias/se-puede-enganar-al-deep-learning/.
5. Ebrahimi, J., Rao, A., Lowd, D.: Hotflip: White-Box Adversarial Examples for Text Classification. arXiv:1712.06751 (2017) doi: 10.48550/arXiv.1712.06751.
6. Gao, J., Lanchantin, J., Soffa, M.L.: Black-box Generation of Adversarial Text Sequences to Evade Deep Learning Classifiers. In: IEEE Security and Privacy Workshops (SPW), pp. 50–56 (2018) doi: 10.1109/SPW.2018.00016.
7. Gonzalez, S.: Adversarial Machine Learning: Introduction to Attacks on ML models (2022) https://www.welivesecurity.com/laes/2022/05/30/adversarial-machine-learning-introduccion -ataques-modelos-ml/.
8. Iyyer, M., Wieting, J., Gimpel, K.: Adversarial Example Generation with Syntactically Controlled Paraphrase Networks. arXiv:1804.06059 (2018) doi: 10.48550/arXiv.B1804.06059.
9. Li, J., Ji, S., Du, T.: Textbugger: Generating Adversarial Text Against Real-World Applications. arXiv:1812.05271 (2018) doi: 10.48550/arXiv.1812.05271.
10. Li, S., Yun, X., Hao, Z.: A Propagation Model for Social Engineering Botnets in Social Networks. In: 12th International Conference on Parallel and Distributed Computing, Applications and Technologies, pp. 423–426 (2011)
11. Liang, B., Li, H., Su, M.: Deep Text Classification Can Be Fooled. arXiv:1704.08006 (2017) doi: 10.48550/arXiv.1704.08006.

12. Lorenzen, C., Agrawal, R., King, J.: Determining Viability of Deep Learning on Cybersecurity Log Analytics. In: IEEE International Conference on Big Data (Big Data). pp. 4806–4811 (2018) doi: 10.1109/BigData.2018.8622165.
13. Samanta, S., Mehta, S.: Towards Crafting Text Adversarial Samples. arXiv:1707.02812 (2017) doi: 10.48550/arXiv.1707.02812.
14. KAAGLE: Spam Text Message Classification (2017) https://www.kaggle.com/datasets/team-ai/spam-text-message-classification.
15. Xu, H., Ma, Y., Liu, H.C.: Adversarial Attacks and Defenses in Images, Graphs and Text: A Review. International Journal of Automation and Computing 17, pp. 151–178 (2020) doi: 10.1007/s11633-019-1211-x.

ISSN 1870-4069

Web Tool to Support Electronic Democracy Processes Using Blockchain

Carlos Huerta García, Axel Ernesto Moreno Cervantes,
Nidia Asunción Cortez Duarte

Instituto Politécnico Nacional,
Escuela Superior de Cómputo,
Mexico

chuertag1600@alumno.ipn.mx, axelernesto@gmail.com,
ncortezd@ipn.mx

Abstract. The traditional voting process and contemporary electronic voting systems are insufficient for meeting the needs of the users for more reliable and secure democratic processes within collective governance structures. This paper presents the development of a web tool designed to support electronic democracy processes using Blockchain. The reliability and security of the democratic processes facilitated by the tool are increased through the joint support of the electronic democracy axes, the use of a smart contract deployed on a Blockchain network, and the employment of the HTTPS protocol. Authentication, confidentiality, integrity and non-repudiation information security services are provided throughout the implementation described in this paper, while ensuring that the security provisions do not have a significant impact on the performance of the web tool. The tool was developed using SCRUM, and its quality and functionality was verified through unit, integration, stability, security, and usability testing.

Keywords: Blockchain, web application, electronic democracy, SCRUM.

1 Introduction

1.1 ICT and Blockchain

As mentioned, Information and Communication Technologies (ICT) are tools through which communication and information consultation is possible in an increasingly instantaneous and sophisticated way [1]; in this sense, the Internet has become the most promising resource for that purpose, since its appearance in the late 1970s its evolution has been manifested in Web 1.0, Web 2.0 and Web 3.0 generations. According to [2] the World Wide Web (WWW) has become the broadest medium in which users can share, read and write information through devices connected to the Internet. In [2] they point out that in Web 1.0 people connected and gathered information from the network, in Web 2.0 people connected with each other, and Web 3.0 has been treated as the

knowledge or semantic web; although [3] points out different proposals for the latter: a virtual, semantic or decentralized web. On the Web 3.0 from the proposal of a decentralized web, networks with blockchain mechanisms or Blockchain arise.

The Blockchain is a single, consensual record of transactions distributed among several nodes of a network [4, 5]. Each block is chained to the next, by means of a cryptographic mechanism. In this way, each node stores the complete chain for verification as new blocks are validated. One of the uses of this technology may be found in smart contracts. Also, [4] indicates that these are agreements in which the automatic execution of instructions written in a programming language and executed within a Blockchain network. Blockchain networks mainly consist of asymmetric cryptography techniques for the use of digital signatures to guarantee the identity of users, hash functions to ensure data integrity, as well as transaction validation mechanism called consensus algorithm [6].

The social, economic and political spheres have been transformed by the introduction of ICT throughout its generations, as they modify the ways in which individuals perceive their environment and in which they perform their daily activities, as pointed out by [1] and [2]. It is stated that the Internet has been useful to generate new ways of relating through virtual spaces, or access to online services [1]. According to [1], the political uses of technology are more recurrent, whether in the areas of public management or decision making.

Thus, the concept of electronic democracy has been established as the use of ICT in democratic political processes, linked to the area of political decision-making to realize its key functions, such as the interest articulation, decision-making processes and information exchange between actors [1]. Three axes of electronic democracy have been proposed which constitute the stages of democracy: informative, deliberative and resolutive or participatory [1, 7]. It is argued that the informative area refers to the availability of information to Internet users (as well as its dissemination) to generate knowledge through the assimilation of such information that allows political decisions to be made. As for the deliberative scope, discussions, debates, consultations, agenda proposals and online videoconferences are generated. Finally, the resolutive or participatory axis refers to the participation of citizens in public decision-making through digital means (such as Internet voting), so that their demands are considered [1, 7].

1.2 Electronic Democracy Implementations

Regarding the political sphere, nowadays several countries have opted for a democratic form of government, in which power is exercised by the people through participatory legal mechanisms for political decision making [5]. Each country has implemented different models according to its needs and circumstances. Similarly, in the economic and social sphere, democratic governance is also employed. Two principles established in Article 6 of the General Law of Cooperative Societies are democratic administration and participation in cooperative integration [8]. Article 37 mentions the requirements that a call for an ordinary or extraordinary general assembly of members must meet,

emphasizing the availability of member participation to give validity to the agreements reached.

In [5], the authors point out that the traditional voting process that currently prevails in many Latin American countries (with differences between each country) generally consists of a series of steps that conclude in the quantification of votes to make a political decision. However, they all have the same objective: to ensure a transparent, secure and reliable process.

Furthermore, it has been shown that ICTs provide alternatives to the need for safer and more reliable electoral processes, giving rise to the use of electronic voting systems [5]. Electronic voting systems can be divided in two: electronic voting: consists of voting points controlled by operators, use of electronic devices and potential use of private networks; Internet voting: consists of the ability to vote from anywhere via Internet and distributed servers. Both provide different approaches to contribute towards the electoral process, however, they present different challenges.

On several occasions, these both present the following difficulties: vote counting and scrutiny processes entail high economic costs and require significant time; electoral frauds have been alleged in the different steps of the electoral process, which causes distrust among participants; manual processes entail risks of human error; a centralized electoral process also generates a lack of trust, since anyone with access to the system could alter the results of the process; systems that make use of private networks to exchange information are vulnerable to computer attacks, risking the integrity of the data [5]. Blockchain technology has recently been employed to address information integrity vulnerabilities and data decentralization.

Table 1 compares the tool described in this paper to relevant implementations. This evaluation assesses the provision of key security services and considers the extent to which each axis of electronic democracy, as well as the decentralization in data storage.

Modelo y sistema de votación electrónica aplicando la tecnología de cadena de bloques [5] operates as an exchange of votes between citizens and candidates. In this system, votes are interpreted as transactions, with each transaction recorded in the Blockchain network.

Sistema de voto electrónico basado en blockchain [9] operates as an exchange platform between citizens and candidates. Upon registering in the system, each voter is provided with a digital wallet, through which their single vote is subsequently transferred by the system. Thereafter, voters select a single candidate from a list displayed by the system, thereby transferring their vote to the candidate's digital wallet.

Sistema Electrónico por Internet (SEI) [10] employed by the Instituto Electoral de la Ciudad de México in 2023 and 2024 for democratic processes incorporates biometrics validation, authentication, confidentiality and non-repudiation services provision. The voting process is validated through the list of voters, a double voting prevention mechanism performed twice and the use of asymmetric cryptography signature algorithms.

Table 1. Electronic democracy implementations state of the art.

Implementation	Authentication, confidentiality, integrity, and non-repudiation provision	Informative axis of electronic democracy is considered	Deliberative axis of electronic democracy is considered	Resolutive axis of electronic democracy is considered	Data storage is decentralized
[5]	Yes	No	Yes	Yes	Yes
[9]	Yes	No	Yes	Yes	Yes
[10]	No	No	Yes	Yes	No
Developed tool	Yes	Yes	Yes	Yes	Yes

1.3 Web Tool to Support Electronic Democracy Processes Using Blockchain

This article discusses the development and implementation of a web tool designed to support electronic democracy processes in its three identified axes, which are not served by any other tools jointly (i.e., assisting each axis). So that it meets the requirements for designing valid participatory processes through the proposed technology [1] the dissemination of information and awareness, the mechanisms of consultation and deliberation and those concerning the decision-making process. The tool facilitates the registration and access to information resources, the use of consultations, voting, scheduling, and interactive video streaming with chat for deliberation, as well as the display and follow-up of deliberation outcomes. In addition, a deliberation mechanism which is not usually present on current implementations in this field was developed, such as the use of interactive video streaming featuring chat-based communication.

Meanwhile, the joint support of the three axes of electronic democracy with Blockchain technology and secure communications enables to address the needs of users in the aspects of security and reliability, which are not fully satisfied and give room for the possibility of harming third parties under poor implementations in the field of collective decision-making processes, held by any institution or collective. In order to ensure that the security provisions do not have a significant impact on the performance of the web tool, this approach will be tested by measuring the performance of the web application using the quantitative research methodology.

It has been identified that any democratic regime must have the following characteristics for the process of authentic voting [11]: free, periodic, competitive, clean and decisive. To verify its compliance, the developed tool offers the following security services: confidentiality, through the encryption of the information handled; anonymity, thanks to the intrinsic features frequently present in blockchain networks such as the use of pseudonyms; integrity, achieved from the creation and validation of the blocks in the blockchain network; non-repudiation, provided by the use of private keys and the policies for their use, once the blocks in the chain are approved they become immutable and irreversible [4].

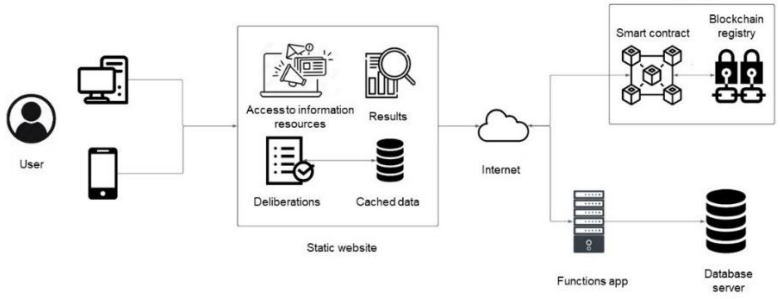

Fig. 1. Web tool architecture.

The remainder of this paper is organized as follows: Section 2 details development of the web tool. Implementation results are presented in Section 3. Conclusions and future work are discussed in Section 4.

2 Development

The web tool consists of a web application connected through the internet to the smart contract deployed on a blockchain network, to a database server for querying and storing the necessary data and to an external file server for querying the recordings of video transmissions and information resources, as shown in
Fig. **1**.

For the development of this web tool, the SCRUM methodology was chosen because of its work scheme in which work is defined, reviewed and advanced. This scheme allows to focus on the production of functional components and makes possible, by means of an incremental structure, to split the complexity of the project and to periodically satisfy determined objectives [12].

The work was organized in 7 work cycles of 3 weeks called Sprints, during which the team worked on incrementing the project. At the beginning of the development, the set of features that integrate the project were defined in 28 user stories, identifying their estimated time, priority, description and acceptance criteria. Along with the user stories, the Sprints were planned to develop the research, analysis and design, the functionality of the web tool with the database, the streaming video with chat, the methods of the smart contract, the designed user interfaces and the verified integration of components.

At the beginning of each Sprint, a planning meeting was held to define the tasks to be performed, and short meetings were held daily to synchronize activities. At the end of each Sprint, the completed work is reviewed according to the acceptance criteria for the increment and reflection is made on how to improve in the next cycle. This organization is managed through a prioritized list with all the user stories that make up the project and other similar lists with the user stories selected for each Sprint, as well as the increments produced at the end of the Sprint.

Also, within the team, roles associated with the methodology were assigned according to the experience with the methodology and the capabilities of each member. The team was composed of Carlos Huerta García, responsible for maximizing the value

of the project and managing the prioritized list of user stories; Axel Ernesto Moreno Cervantes, Nidia Asunción Cortez Duarte as facilitators of SCRUM practices and values, supporting the team to overcome any obstacles; Oliver Manuel Hernández Méndez, Rafael Hayyim Medina Sosa, Marco Antonio Ocaña Navarrete as cross-functional development team, in charge of delivering product increments at the end of each Sprint. Synthesizing the established user stories, the following actions were defined:

First, the registration process was established, where the user registers by providing his name, email and password. To gain access, users must verify their account with a one-time password that is sent to their e-mail address and reset their password through the same process. Due to the limited collection of data and the use to which it is put, it is guaranteed to satisfy the ARCO rights.

Subsequently, by accessing the tool with their email and password, a session is established which is verified at each further functionality and they have the option to create or join a collective while the collectives to which they belong or manage are displayed. To join a collective involves the use of a code displayed at the time of creation or management, while creation implies specifying the name, type and description of the collective, along with the attachment of one NEAR. Besides, when a collective is created, its administration is assigned to the user who creates it, while it is possible to assign more administrators with the email of a member of the collective who is not already an administrator.

Within the collective context, the registration of decision-making processes can be carried out, including information such as the name of the process, the expected resolution and a detailed description. Prior to initiating any deliberation, the process manager (the registrant) provides informational resources so that the collective members are fully informed before reaching a determination.

Deliberation mechanisms are a crucial part of the decision-making process. For its registration, a date is requested that meets a minimum 10-day deadline, in accordance with the General Law of Cooperative Societies. In addition, a title for the mechanism is required and, depending on the type of deliberation (whether it is a forum, voting or consultation), specific details must be provided, such as a description for the forum, the options for voting, and a set of questions and options for consultations. To participate in any mechanism, if it is the registered date participation will occur via postings, votes and video streaming. The prevention of double voting is achieved through the storage of a hash of each user who has participated. If the hash exists when the user attempts to vote for a second time, a warning is generated, indicating that the user may only cast a single vote.

Finally, at the conclusion of any deliberation, the corresponding link is entered to follow up on the resolution reached once the established date has elapsed. In the case of forums, a summary of the posts is added, thus completing the cycle of participation and follow-up on the platform.

The development of the source code was performed in a Git repository with the GitFlow framework. A team member was assigned responsible for each task and pull requests were incorporated upon completion, achieving an incremental integration of the project.

Upon tasks completion within a Sprint, a release branch was created consolidating the developed tasks with the main and development branch. In addition, Azure DevOps was implemented for continuous integration, BitBucket as a Git repository server, and Jira Software to manage issues associated with SCRUM methodology. All user stories were collected in Jira, along with the titles for each Sprint as epics and the tasks and subtasks defined related to the user stories, so that the status of each task or subtask was tracked on a Kanban board in the scope of a Sprint. At the end of each Sprint, its retrospective was posted on a Confluence page within the Jira project after analyzing the pace of work reflected in the Sprint Burndown reports generated by Jira.

Regarding the technologies employed, mongoDB was selected to store the information associated with the electronic democracy processes in a non-relational database. For the provision of the functionality, an Azure functions app with Node.js and TypeScript was determined. To consume these functionalities, a React with TypeScript static website was specified. Furthermore, the REST API approach was employed to ensure simplicity in the interactions between the functions application and the static website.

Concerning the Blockchain, it was defined to use NEAR which uses the Ed25519 signature to provide authentication, and the SHA-256 hash algorithm for integrity. It is significant to note that every communication between the various entities that comprise the web tool has been configured to be handled exclusively via the HTTPS protocol, in which the ECDHE-X25519, ECDSA-X25519, AES-256-GCM and SHA-384 algorithms usage are preferred to provide authentication, confidentiality and integrity in the exchange of information.

Two main modules were developed for the web application: a static website and a functions application. The website, built with React and managed with Yarn and Vite, integrates unit tests with Vitest. It is organized into mainly components and pages, each with an associated set of test cases for unit testing. The guidelines for design defined in Material Design 3 were followed to provide an interface that would be familiar to most users.

The implementation of features was structured around the functionalities and their associated integration test cases. For video streaming with chat, the SignalR real-time communication service was employed. In the continuous integration pipeline, it is tested, statically analyzed for code quality with SonarCloud and the site is packaged for deployment. The functions application, created with Azure Functions and Jest for testing, follows a similar process with a continuous integration pipeline of integration testing, static code quality analysis with SonarCloud and packaging. Each functionality is defined in a function triggered by HTTP requests, leveraging the elasticity of dynamic provisioning.

The smart contract is defined in a class with contract methods and function sets associated with users, collectives, decision making processes, deliberations, forums, meetings, consultations, voting and results so that the contract methods realize the functionalities of the corresponding sets. It is structured with methods for entities, managing state efficiently in associative arrays. The continuous integration pipeline used with the latter includes static analysis of code quality and contract construction, generating a compiled WebAssembly file for publication.

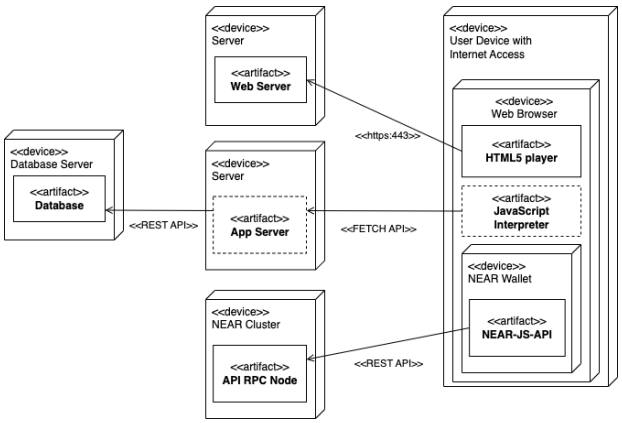

Fig. 2. Deployment Diagram.

The artifacts resulting from the integration pipelines were deployed using infrastructure as code, stored and used within the published packaging. The files that resulted from building the different projects that make up the development are unpacked and then utilized. The Azure for Students license was leveraged to deploy the dynamic website using Static Websites, configuring the redirection of paths to the input file. Also, the same license was used to deploy the functions application, configuring environment variables and resource sharing policies. In addition, an account was created on the blockchain network to deploy the smart contract, using near-cli and the compiled WebAssembly file obtained from building the source code.

Overall, each action is performed by the interaction between the static website and the user through a browser. From the website, it communicates with the functions application and the smart contract deployed on the Blockchain, so that the actions are reflected in the electronic democracy processes, displaying a feedback message of the action performed and the link to the details of the transaction recorded in the transaction browser of the blockchain network used. The distribution of each component can be illustrated as shown in the deployment diagram in Fig. 2.

3 Results

Referring to the tests performed, Table 2 shows the results obtained in the different types of tests. A total of 115 unit tests were executed with Vitest in which each branching point in the code units of the web site is tested without failures or errors in less than 27 seconds. For the acceptance tests, all the acceptance criteria established in each of the 28 user stories are met. An average response time of 43 milliseconds was recorded scaling from 20 to 100 users by increasing 5 users every minute for 20 minutes with Blazemeter, being always available. Also, using loader.io, a content display time of 1.7 seconds and an accessibility evaluation of 98/100 in PageSpeed Insights were obtained. 236 integration tests were performed with Jest in which each function

Table 2. Tests results obtained.

Tests type	Test cases	Time	Score
Unit tests	115	27 s	100
Integration tests	236	332.6 s	100
Acceptance tests	28	Not applicable	100
Performance tests	16	43 ms	Not applicable
Usability tests	8	1.7 s	98
Security tests	16	Not applicable	100

established in the web application was verified without failures or errors in less than 333 seconds. The confidentiality of information in communications between the web site, the function application and the smart contract was verified for each functional requirement through man-in-the-middle attacks using Wireshark to read network traffic and participate in the communication if possible. For all 16 functional requirements the confidentiality, integrity and authenticity of the information was rated as excellent.

As the version of the Azure functions application used at the time of tool deployment did not resolve the cold-start issue associated with serverless implementations, the performance tests revealed higher than expected average response times. However, the performance and usability tests did not reflect any significant impact on the performance of the developed tool, even with the security provisions in place.

Considering the acceptance tests results, it can be said that the three key axes of electronic democracy are fully met, contributing to the increased reliability of the democratic processes facilitated by the tool. Throughout the security tests, the provision of authentication, confidentiality and integrity services is verified. Considering the verified security services and the smart contract used in a public blockchain network for the registration of any action within a democratic process, the identified security vulnerabilities are addressed. Therefore, it can be said that the security of democratic processes carried out with the developed tool is increased. Finally, regarding the results of the unit and integration tests, it can be said that the functional application, static website and smart contract development have a remarkable quality.

4 Conclusions and Future Work

The web tool developed addresses the need to enhance the security and reliability in the decision-making processes in democratically driven collectives by utilizing a public blockchain network to record all actions within the electronic democracy processes, providing authentication, integrity and non-repudiation.

This facilitates the consultation of the actions undertaken throughout the democratic process. In addition to the use of HTTPS, the vulnerabilities present in the contemporary electronic democracy systems are mitigated. Moreover, it provides support for the three identified axes of electronic democracy. Firstly, the tool enables the provision and display of information resources in a manner that is consistent with the informative axis. Secondly, it streamlines participation through four distinct

deliberation mechanisms, namely voting, video streaming, consultations, and forums. This enables the completion of the deliberative axis and the fulfilment of the regulatory framework for cooperative societies in Mexico.

The capability to reach a resolution after collective participation and the registration of information related to the follow-up of the resolution reached fulfills the resolutive axis and concludes the electronic democracy process. The operation, stability, and usability of the tool were verified in the tests carried out.

It is proposed that a functionality be included for users to easily update their information, thereby providing flexibility and accuracy in profile management. Additionally, the implementation of an email notification system is proposed to inform collective members about new decision-making processes, information resources, updates on the status of scheduled deliberations and resolutions reached, with the aim of improving communication and participation of collective members.

In the context of the deliberation mechanisms present in the tool, it is proposed that the automatic generation of the summary of the publications in the forums be achieved with large language models based on artificial intelligence. Similarly, automatic generation of minutes in the video transmissions should be implemented using the same proposed technology. Another avenue for improvement involves the expansion of these deliberation mechanisms, with the objective of exploring and adding more tools to enrich the decision-making processes, thereby increasing the efficiency and inclusiveness of the platform.

Furthermore, the potential for decentralizing not only the validation of electronic democracy processes, but also the data associated with them, is also discussed. If a blockchain network managed by the collective is utilized, the integration of the logic of the functions application within the smart contract is contemplated. This is done to maximize decentralization, strengthen the autonomy of participants, and ensure the integrity of information in a distributed environment. Each member of the collective would possess a network node. Consequently, with the advancement of mobile computing power, it is now feasible to provide a tool to support the processes of electronic democracy that are conducted in large collectives, such as countries, with a blockchain network assisting the democratic exercise within a nation-state.

Acknowledgments. The authors would like to express their gratitude to the Instituto Politécnico Nacional (IPN) and Escuela Superior de Cómputo (ESCOM), as well as Oliver Manuel Hernández Méndez, Rafael Hayyim Medina Sosa, and Marco Antonio Ocaña Navarrete, for their invaluable contributions to the development and realization of this work.

References

1. Hernández, N.E.: El voto electrónico en la construcción de un modelo de democracia electrónica. Estudios políticos, 47, pp. 61–85 (2019) doi: 10.22201/fcpys.24484903e.2019.47.69500.

2. Nath, K., Dhar, S., Basishtha, S.: Web 1.0 to Web 3.0 - Evolution of the Web and Its Various Challenges. In: International Conference on Reliability Optimization and Information Technology (ICROIT), pp. 86-89 (2014) doi: 10.1109/ICROIT.2014.6798297.
3. Alabdulwahhab, F.A.: Web 3.0: The Decentralized Web Blockchain Networks and Protocol Innovation. In: 1st International Conference on Computer Applications & Information Security (ICCAIS), pp. 1–4 (2018) doi: 10.1109/CAIS.2018.8441990.
4. Tasende, I.: Blockchain y arbitraje: Un nuevo enfoque en la resolución de disputas. Especial énfasis en smart-contracts y criptodivisas. Revista de Derecho (Universidad Católica Dámaso A. Larrañaga), 22, pp. 138–159 (2020) doi: 10.22235/rd.vi22.2127.
5. Lucuy, G.A., Köller-Vargas, S.A., Galaburda, Y.: Modelo y sistema de votación electrónica aplicando la tecnología de cadena de bloques. Acta Nova, 9(2), pp. 236–256 (2019)
6. Olivares, J.C., Reyes-Archundia, E., Gutierrez, J.A.: Un sistema transactivo de energía ciberseguro usando cadenas de bloques de múltiples niveles. Computación y Sistemas, 27(3), pp. 851–867 (2023) doi: 10.13053/cys-27-3-4071.
7. Posada, L.J.: MIRA: Internet, participación y democracia: Las nuevas tecnologías y la reconexión con el ciudadano. Civilizar Ciencias Sociales y Humanas, 11(20), pp. 57–74 (2011) doi: 10.22518/16578953.24.
8. C.D.D.D.H.C.D. LA UNIÓN: Ley General de Sociedades Cooperativas. Diario Oficial de la Federación. Ciudad de México (2018)
9. Sánchez, S.A.: Sistema de voto electrónico basado en Blockchain. Pontificia Universidad Católica del Perú (2021)
10. Consejo General del Instituto Electoral de la Ciudad de México: Estudio de viabilidad técnica, operativa y financiera que presentan la Dirección Ejecutiva de Organización Electoral y Geoestadística y la Unidad Técnica de Servicios Informáticos para proponer el uso del Sistema Electrónico por Internet, como una modalidad adicional para recabar votos y opiniones en la Elección de Comisiones de Participación Comunitaria 2023 y en la Consulta de Presupuesto Participativo 2023 y 2024. Instituto Electoral de la Ciudad de México, Ciudad de México (2024)
11. Centro de Capacitación Judicial Electoral: Régimen democrático. Tribunal Electoral del Poder Judicial de la Federación (2010)
12. Scrum Guide: https://scrumguides.org/scrum-guide.html (2020)

ISSN 1870-4069

Evaluación comparativa de la representatividad de modelos RNCP en mastografías públicas del Hospital General de Ensenada

J.I. Ayala-Guebara[1,2], J.A. González-Fraga[1], J. Magaña-Magaña[2], E. Gutiérrez-López[1], G. J. Avilés-Rodríguez[3], L.M. Pellegrin-Zazueta[1]

[1] Universidad Autónoma de Baja California,
Facultad de Ciencias,
México

[2] Hospital General Ensenada,
México

[3] Universidad Autónoma de Baja California,
Escuela de Ciencias de la Salud,
México

{angel_fraga, luis.pellegrin}@uabc.edu.mx

Resumen. Las redes neuronales convolucionales profundas (RNCP) son ampliamente utilizadas en el aprendizaje supervisado y han demostrado su capacidad para aprender relaciones entrada-salida en grandes volúmenes de imágenes. En este artículo se presenta un estudio comparativo de la representatividad de modelos de RNCP entrenados con 2 conjuntos de datos distintos. El primer modelo se genera utilizando el dataset público CBIS-DDSM, el cual clasifica las imágenes de mastografía en categorías de benigno y maligno. El segundo modelo se entrena con la base de datos de mastografías del Hospital General de Ensenada, que clasifica las imágenes según la escala BI-RADS del 1 al 5. El tercero se entrena y evalúa con ambas bases de datos. Se generaron varios modelos utilizando las arquitecturas MobileNetV2, VGG16, Resnet, CNN, DenseNet e Inception. Tras su evaluación MobileNetV2 demostró el mejor rendimiento. Este estudio analiza y compara la exactitud (accuracy), precisión, sensibilidad y f1-score de los modelos para evaluar la representatividad y efectividad de cada conjunto de datos en la clasificación de imágenes de mastografía. Los hallazgos de esta investigación buscan proporcionar información valiosa para el desarrollo y mejora de herramientas de diagnóstico asistido por computadora, con el objetivo de optimizar el diagnóstico temprano y el tratamiento del cáncer de mama en diferentes entornos clínicos.

Palabras clave: Cáncer de mama, mastografía, redes neuronales, convolucionales profundas, aprendizaje supervisado, MobileNetV2, diagnóstico asistido por computadora.

Comparative Evaluation of the Representativeness of Deep Convolutional Neural Network Models in Public Mammograms from the General Hospital of Ensenada

Abstract. Deep convolutional neural networks (DCNNs) are widely used in supervised learning and have demonstrated their ability to learn input-output relationships in large volumes of images. This paper presents a comparative study of the representativeness of DCNN models trained on two different datasets. The first model is generated using the public CBIS-DDSM dataset, which classifies mammogram images into benign and malignant categories. The second model is trained on the mammogram database from the General Hospital of Ensenada, which classifies images according to the BI-RADS scale from 1 to 5. The third model is trained and evaluated with both datasets. Several models were generated using the MobileNetV2, VGG16, ResNet, CNN, DenseNet, and Inception architectures. After evaluation MobileNetV2 demonstrated the best performance. This study analyzes and compares the accuracy, precision, sensitivity, and F1-score of the models to assess the representativeness and effectiveness of each dataset in mammogram image classification. The findings of this research aim to provide valuable insights for the development and improvement of computer-aided diagnostic tools, with the goal of optimizing early diagnosis and treatment of breast cancer in diverse clinical settings.

Keywords: Breast cancer, mammography, deep convolutional neural networks, supervised learning, MobileNetV2, computer-aided diagnosis.

1. Introducción

El cáncer de mama actualmente es un problema de salud pública que requiere ser atendido de manera prioritaria. De acuerdo a la Organización Mundial de la Salud (OMS), el cáncer es una de las principales causas de muerte en el mundo, se reporta que 1 de cada 6 muertes se debe al cáncer [1]. De acuerdo con datos del Instituto Nacional de Estadística y Geografía (INEGI), en el año 2020 fallecieron un total de 97 323 personas por tumores malignos; 7 880 fueron por tumores malignos de mama, lo que equivale al 8 % de este total. Debido al cáncer de mama fallecieron 7 821 mujeres y 58 hombres [2]. Se ha identificado el cáncer de mama como el tipo de neoplasia maligna con mayor incidencia y mortalidad en las mujeres a nivel global. En México el cáncer de mama es el tipo que más afecta a las mujeres, y está catalogado como la principal causa de muerte por este padecimiento. En el último reporte global de la Agencia Internacional para la Investigación del Cáncer, se reportaron más de 27 000 nuevos casos y casi 7 000 decesos debidos a esta causa [3]. Debido a esto, el cáncer de mama es considerado como un problema de salud pública nacional, cuya consecuencia es un promedio de más de 18 muertes por día. La detección temprana del cáncer de mama es crucial para aumentar las tasas de supervivencia y mejorar los resultados del tratamiento. Los modelos de Red Neuronal Convolucional Profunda (RNCP) han demostrado un gran potencial en la identificación de anomalías en mastografías [4]. Sin embargo, la representatividad de estos modelos es un factor crítico que determina su

eficacia en diversas poblaciones. Este estudio se centra en evaluar la representatividad de los modelos de RNCP utilizando 2 bases de datos: CBIS-DDSM [5] que es una base de datos pública y HGE-DB la cual es una base de datos privada bajo el resguardo del Hospital General de Ensenada.

Este documento se organiza en cinco secciones principales: Introducción, Trabajo relacionado, Metodología, Resultados, Discusión y Conclusiones. Cada sección aborda aspectos clave del estudio, proporcionando un análisis detallado y conclusiones basadas en los hallazgos.

1.1. Objetivos del estudio

El objetivo general de este estudio es analizar y evaluar la capacidad de los modelos de redes neuronales convolucionales profundas (RNCP) para detectar de manera precisa y efectiva el cáncer de mama en imágenes de mamografías. El enfoque principal será examinar su desempeño en diferentes contextos clínicos y bases de datos, asegurando la representatividad y generalización de los resultados obtenidos a poblaciones diversas. Los objetivos específicos son:

- **Comparar el desempeño de los modelos de RNCP en diferentes conjuntos de datos**: Evaluar la exactitud, precisión, sensibilidad y F1-score de los modelos entrenados con mamografías de bases de datos públicas, como el CBIS-DDSM, y de una base de datos local del Hospital General de Ensenada. Este análisis busca identificar diferencias en la detección de anomalías en diferentes entornos clínicos y tipos de población.
- **Detectar y analizar sesgos y limitaciones en los modelos**: Examinar cómo los sesgos de los datos, como el desbalance de clases, la variabilidad en la calidad de las imágenes o las características demográficas, pueden afectar el rendimiento de los modelos. Esto incluirá la identificación de patrones que limiten la generalización de los modelos en la detección de cáncer de mama en diversas poblaciones.
- **Proponer mejoras para aumentar la generalización y aplicabilidad de los modelos**: Con base en los resultados de la evaluación y análisis de los modelos, se desarrollarán recomendaciones específicas para mejorar la robustez y adaptabilidad de los modelos de RNCP. Estas sugerencias estarán enfocadas en mejorar la precisión de los modelos en la detección de cáncer de mama en diferentes poblaciones y condiciones clínicas, asegurando su eficacia en la práctica médica real.

1.2. Hipótesis

La hipótesis principal de este estudio es que los modelos de RNCP entrenados en bases de datos públicas pueden no ser igualmente precisos cuando se aplican a mastografías de una población específica como la del Hospital General de Ensenada.

2. Trabajo relacionado

2.1. Modelos de RNCP en el diagnóstico mamario

Las redes neuronales convolucionales se utilizan principalmente para la clasificación de imágenes, y su eficiencia mostrada en sus resultados, es una de las razones principales por las que el aprendizaje profundo y el aprendizaje automático han tenido un nuevo auge en la investigación. Las RNCP aprenden características discriminatorias automáticamente y su arquitectura está particularmente adaptada para aprovechar la estructura 2D de la imagen de entrada, pero lo que es más importante, una de sus características más impresionantes es que generalizan sorprendentemente bien otras tareas de reconocimiento de patrones. En los últimos años, se ha logrado un progreso significativo del reconocimiento de patrones en imágenes, en diferentes dominios [6, 7], a través de RNCP. Se ha demostrado que el desempeño de los métodos de aprendizaje profundo, en términos de precisión, para el problema de reconocimiento en imágenes puede sobrepasar el desempeño del humano [8, 9]. Sin embargo, todavía hay una serie de problemas por resolver, como la alta complejidad computacional de los algoritmos de aprendizaje (puede durar hasta semanas), el requerimiento de un conjunto grande de muestras para entrenar, el deterioro del desempeño del reconocimiento en función de los cambios en la base de datos de entrenamiento, etc. A pesar de estos retos, en 2020 se reportó un sistema basado en aprendizaje profundo, que resultó ser tan bueno como los médicos especialistas en la predicción del cáncer de mama al analizar las mastografías, en donde se explica que gracias a estas técnicas hubo una reducción en los falsos positivos y los falsos negativos [10]. Los modelos de RNCP han revolucionado el campo de la imagen médica, ofreciendo herramientas avanzadas para la detección automática de enfermedades. Diversos estudios han demostrado la capacidad de estas redes para identificar anomalías en imágenes de mamografías con una precisión comparable a la de los radiólogos expertos [6,11]. Por ejemplo, el uso de técnicas de Transferencia de Aprendizaje y Pseudocolor en un Sistema CADx para la clasificación de cáncer de mama ha mostrado resultados favorables en comparación con métodos del estado del arte, utilizando métricas de calidad como Exactitud, Especificidad, Sensibilidad y Medida-F [12].

2.2. Bases de datos de mastografías

Las bases de datos públicas, como el Digital Database for Screening Mammography (DDSM), proporcionan un recurso valioso para entrenar modelos de RNCP. Sin embargo, la variabilidad en la calidad de imagen y las características demográficas de las pacientes puede afectar la representatividad de los modelos. El Hospital General de Ensenada, por otro lado, tiene bajo su resguardo una base de datos específica de su población, lo que nos permitirá evaluar la generalización de los modelos en un entorno clínico real.

2.3. Comparación de resultados en diferentes entornos

Estudios previos han destacado las discrepancias en la eficacia de los modelos de redes neuronales convolucionales profundas (RNCP) cuando se aplican a diferentes bases de

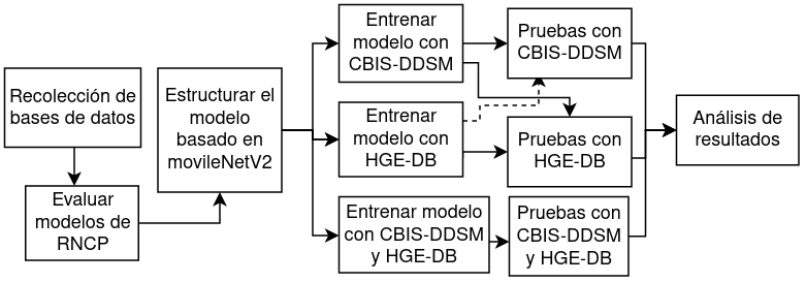

Fig. 1. Flujo del proceso metodológico utilizado en este estudio.

datos. Estas variaciones pueden deberse a diferencias en la calidad de imagen, el equipo utilizado y las características demográficas de los pacientes.

Por ejemplo, la precisión de los modelos en la detección de cáncer de mama puede variar significativamente según la calidad y diversidad demográfica de las bases de datos, con precisiones reportadas entre 74 % y 98 % [13].

En la detección de glaucoma, un modelo entrenado en un conjunto de datos específico mostró alta precisión dentro de ese conjunto pero una notable disminución cuando se aplicó a datos externos, subrayando la importancia del ajuste específico de los datos [14].

Además, en la detección y cuantificación de enfermedades pulmonares intersticiales, con enfoques de aprendizaje profundo se destaca la influencia de la preparación de datos y la integración de imágenes de múltiples canales, como la tomografía computarizada (CT), en el rendimiento de los modelos predictivos [15].

En conclusión, aunque los modelos de RNCP han demostrado ser herramientas poderosas para la detección de enfermedades a partir de imágenes médicas, su eficacia puede variar significativamente dependiendo de la base de datos utilizada y las características específicas de la población. Es crucial continuar evaluando y ajustando estos modelos para garantizar su aplicabilidad y precisión en diversos contextos clínicos.

3. Metodología

Este estudio comparativo de enfoque cuantitativo tiene como objetivo evaluar la precisión de los modelos de redes neuronales convolucionales profundas (RNCP) en distintas bases de datos de mamografías. La Fig. 1 muestra el flujo metodológico seguido para la evaluación de estos modelos. En el proceso, se entrenaron y probaron los modelos utilizando imágenes de mamografías procedentes del conjunto de datos CBIS-DDSM y del Hospital General de Ensenada. Las métricas empleadas para la evaluación incluyen exactitud, precisión, sensibilidad y el F1-score, con el fin de analizar la representatividad y eficacia de los modelos en diferentes contextos y poblaciones.

3.1. Selección de datos

En este paso se recopilan las imágenes de mastografías de 2 fuentes distintas: la base de datos pública CBIS-DDSM y la base de datos clínica del Hospital General de Ensenada (HGE-DB):

- **Base de datos CBIS-DDSM** La primera base de datos utilizada fue la CBIS-DDSM (Curated Breast Imaging Subset of DDSM), este conjunto de datos está compuesto por 6,773 imágenes de mamografías, que incluyen tanto casos benignos como malignos, con una clasificación basada en el sistema BI-RADS, es una base de datos pública que se ha utilizado extensamente en investigaciones sobre detección de cáncer de mama. En esta base de datos, las mastografías están clasificadas en 2 categorías: benignas y malignas, lo que permite entrenar y evaluar los modelos de RNCP en la detección de tumores mamarios.
- **Base de datos del hospital general de ensenada (HGE-DB)** La segunda fuente de datos es del Hospital General de Ensenada, en Baja California, México, este conjunto de datos contiene 69,950 imágenes mamográficas. Las mastografías en esta base de datos están clasificadas según la escala BI-RADS (Breast Imaging-Reporting and Data System), que evalúa el riesgo de cáncer de mama en una escala del 1 al 5. Esta clasificación proporciona una evaluación más detallada, desde normal correspondiente al BI-RADS 1 hasta altamente sospechoso de malignidad correspondiente al BI-RADS 5, ofreciendo una categorización más precisa de las imágenes mamarias.

Al combinar datos públicos y clínicos, el estudio busca comparar y evaluar la representatividad y eficacia de los modelos de RNCP en diferentes contextos y poblaciones. La selección de estas bases de datos permite analizar cómo las características de los datos afectan el rendimiento y la capacidad de generalización de los modelos.

3.2. Aprobación ética

El proyecto fue sometido y aprobado por el Comité de Ética del Hospital General de Ensenada, el cual autorizó el uso de las imágenes mamográficas del conjunto de datos HGE-DB. Garantizando de esta forma la confidencialidad de las pacientes y cumpliendo con las regulaciones éticas y de privacidad correspondientes conforme a los principios del Tratado de Helsinki [16].

3.3. Divisiones de entrenamiento y prueba

Para entrenar y evaluar los modelos, los datos fueron divididos en conjuntos de entrenamiento y prueba:

- **CBIS-DDSM**: 2,446 imágenes se utilizaron para el entrenamiento y 642 imágenes se utilizaron para la fase de prueba.

Tabla 1. Arquitectura utilizada.

Etapa	Metrica	CNN	DenseNet	Inception	MobileNet	ResNet	VGG
Entrenamiento	Exactitud	.62	.69	.66	.69	.54	.72
	Precisión	.58	.61	.62	.67	.49	.68
	Sensibilidad	.57	.82	.59	.61	.70	.72
	F1-score	.57	.70	.61	.64	.58	.70
Prueba	Exactitud	.60	.60	.63	.65	.48	.61
	Precisión	.50	.50	.55	.57	.41	.52
	Sensibilidad	.51	.74	.57	.56	.66	.55
	F1-score	.51	.60	.56	.56	.51	.53

– **HGE-DB**: Debido al desbalance de clases (con menos imágenes de las categorías de BI-RADS más altas), se seleccionaron 1,077 imágenes para el entrenamiento y 678 para la prueba.

3.4. Modelos evaluados

Durante el experimento, se probaron varios modelos de RNCP (ver Tabla 1). Entre los modelos evaluados, MobileNetV2 mostró los mejores resultados tanto en exactitud como en precisión. Otros modelos evaluados, como ResNet y VGG16, no lograron superar a MobileNetV2 en términos de rendimiento.

En el contexto médico de la detección del cáncer de mama, es fundamental que los modelos presenten altos niveles de precisión y exactitud:

– **Precisión**: Es esencial para evitar falsos positivos, es decir, la predicción de cáncer en pacientes que no lo tienen. Minimizar los falsos positivos reduce el número de pruebas adicionales invasivas y la ansiedad en las pacientes, lo que hace que la precisión sea una métrica crítica para el uso clínico.

– **Exactitud**: Aunque abarca tanto falsos positivos como negativos, una alta exactitud asegura que el modelo clasifica correctamente la mayoría de las imágenes, lo cual es crucial para su confiabilidad general en la práctica clínica. Esto asegura que tanto los casos positivos como los negativos se detecten con una alta tasa de éxito.

3.5. Descripción de los modelos de RNCP utilizados

El modelos de RNCP empleados fueron entrenados utilizando una arquitectura popular llamada mobileNet, con ajustes específicos para el aprendizaje de la misma.

MobileNetV2 es una arquitectura de red neuronal convolucional optimizada para dispositivos móviles y otros entornos con recursos limitados.

Introducida por Google en 2018 [17], mejora la eficiencia y el rendimiento de su predecesora MobileNetV1 mediante el uso de:

Tabla 2. Arquitectura utilizada.

Capa	Tipo	Salida
mobilenetv2_1.00_224	Funcional	(None, 16, 16, 1280)
Flatten	Aplanar	(None, 327680)
Fully_Connected	Secuencial	(None, 5) HGE-DB (None, 2) CBIS-DDSM

- **Bloques residuales invertidos:** Estas estructuras permiten que la información fluya más fácilmente a través de la red, mejorando la capacidad de aprendizaje y la precisión del modelo.
- **Conexiones lineales de cuello de botella:** Se utilizan para mantener la eficiencia computacional y reducir la cantidad de operaciones necesarias.

En la Tabla 2 se muestra la arquitectura de red neuronal convolucional profunda utilizada. Comienza con MobileNetV2, una red pre entrenada eficiente que extrae características de las imágenes de entrada con dimensiones de 512 × 512, produciendo una salida con dimensiones de 16 × 16 y 1280 canales de características. Luego, esta salida se aplana en un vector unidimensional de 327680 elementos mediante una capa de aplanado. Finalmente, este vector se pasa a través de una capa completamente conectada que reduce la dimensionalidad a 5 o 2 unidades, adecuada para clasificar las imágenes en una de cinco categorías posibles para el caso del clasificador BI-RADS o en 2 categorías posibles para la clasificación de las mastografías etiquetadas como benigno o maligno.

3.6. Entrenamiento del modelo

Se utilizó un modelo basado en MobileNetV2 ya pre-entrenado con ImageNet. Este modelo se entrena con la base de datos CBIS-DDSM y posteriormente se prueba utilizando un conjunto de imágenes de la misma base de datos. Esto permite evaluar la precisión y eficacia del modelo en un entorno controlado. El modelo se afina utilizando las imágenes de la base de datos HGE-DB. Este entrenamiento adicional permite al modelo adaptarse a las características específicas de la población atendida en el Hospital General de Ensenada. Se realizan pruebas del modelo entrenado con HGE-DB utilizando un conjunto de imágenes de la misma base de datos. Este paso es crucial para evaluar la representatividad del modelo en un entorno clínico real. Finalmente, se analizan los resultados obtenidos de las pruebas realizadas con ambas bases de datos.

3.7. Procedimiento de evaluación

Los modelos fueron evaluados mediante métricas estándar como exactitud, precisión, sensibilidad y F1-score. Estas métricas proporcionan una evaluación integral del rendimiento de los modelos en términos de su capacidad para identificar correctamente las anomalías en las mastografías y minimizar los errores de clasificación. La descripción de estas métricas es la siguiente:

Tabla 3. Matriz de confusión de modelo entrenado y probado con CBIS-DDSM, 2 clases.

	Clase verdadera	
Benigno	279 (73%)	102 (27%)
Malingo	115 (44%)	145 (56%)
Predición	Benigno	Maligno

Tabla 4. Métricas del modelo entrenado y probado con CBIS-DDSM, 2 clases.

Clase	**Exactitud**	**Precisión**	**Sensibilidad**	**F1-score**
Benigno	.66	.71	.73	.72
Maligno	.66	.59	.56	.57

- **Exactitud.** Mide el porcentaje de predicciones correctas realizadas por el modelo sobre el total de casos evaluados.
- **Precisión.** Proporción de verdaderos positivos entre todos los casos clasificados.
- **Sensibilidad.** Proporción de verdaderos positivos entre todos los casos positivos.
- **F1-Score.** Media armónica de precisión y sensibilidad.

Este análisis de evaluación permite determinar la eficacia y capacidad de generalización del modelo en diferentes contextos y poblaciones.

4. Resultados

Para evaluar la hipótesis de que los modelos de RNCP entrenados en bases de datos públicas pueden no ser igualmente precisos cuando se aplican a mamografías de una población específica como la del Hospital General de Ensenada, se utilizaron matrices de confusión y se realizó un análisis detallado de las métricas de rendimiento.

4.1. Desempeño de los modelos en mastografías públicas

En la Tabla 3 se presenta la matriz de confusión del modelo entrenado y probado con la base de datos CBIS-DDSM. Esta matriz muestra el número de casos correctamente e incorrectamente clasificados en las categorías benigno y maligno.

En la Tabla 4 se resumen las métricas de exactitud, precisión, sensibilidad y F1-score para cada clase, calculadas a partir de la matriz de confusión. Los resultados muestran que el modelo tiene una mayor precisión y sensibilidad en la clasificación de mastografías benignas en comparación con las malignas. La precisión y el F1-score para la clase benigna son 0.71 y 0.72, respectivamente, mientras que para la clase maligna son 0.59 y 0.57, indicando una menor capacidad del modelo para identificar correctamente las mastografías malignas. Estos hallazgos sugieren que, aunque el modelo es razonablemente efectivo en la detección de mastografías benignas, existe una necesidad de mejorar su capacidad para clasificar correctamente las mastografías malignas, posiblemente ajustando los parámetros del modelo o incorporando más datos

Tabla 5. Matriz de confusión de modelo entrenado y probado con CBIS-DDSM, 5 clases.

		Clase verdadera			
BI-RADS 1	0	0	0	0	2 (100%)
BI-RADS 2	0	63 (64%)	6 (6%)	14 (14%)	16 (16%)
BI-RADS 3	0	13 (14%)	26 (28%)	38 (41%)	16 (17%)
BI-RADS 4	0	21 (6%)	27 (8%)	216 (66%)	62 (19%)
BI-RADS 5	0	15 (12%)	14 (12%)	50 (41%)	42 (35%)
Predición	BI-RADS 1	BI-RADS 2	BI-RADS 3	BI-RADS 4	BI-RADS 5

Tabla 6. Métricas del modelo entrenado y probado con CBIS-DDSM, 5 clases.

Clase	Exactitud	Precisión	Sensibilidad	F1-score
BI-RADS 1	1	0	0	0
BI-RADS 2	.87	.56	.64	.60
BI-RADS 3	.82	.36	.28	.31
BI-RADS 4	.67	.68	.66	.67
BI-RADS 5	.73	.30	.35	.32

de entrenamiento específicos para esta clase. También se podría entrenar el modelo con las regiones de interés (ROI) contenidas en la base de datos pública, sin embargo, la base de datos proporcionada por el hospital general de ensenada no cuenta con esta misma información, por lo que, para desarrollar este análisis comparativo, ambas bases de datos deben contar con las misma información.

En la Tabla 5 se presenta la matriz de confusión del modelo entrenado y probado con la base de datos CBIS-DDSM. Esta matriz muestra el número de casos correctamente e incorrectamente clasificados con la escala BI-RADS. Se puede observar rápidamente que el modelo no aprendió a clasificar correctamente el BI-RADS 1, esto debido al desbalance drástico de esta clase.

En la Tabla 6 se resumen las métricas de exactitud, precisión, sensibilidad y F1-score para cada clase, calculadas a partir de la matriz de confusión. Los resultados muestran que el modelo tiene una mayor exactitud en la clasificación del BI-RADS 2 y 3.

Sin embargo la precisión en su clasificación es muy baja. Esto debido a que la distribución de esta base de datos fue creada para clasificar calcificaciones y masas como benignas y malignas, por lo que al tratar de redistribuir los datos ahora con la clasificación BI-RADS, esta nueva distribución de clases está desbalanceada.

4.2. Desempeño de los modelos en mastografías del Hospital General de Ensenada

En la Tabla 7 y 8 se presenta la matriz de confusión y las métricas del modelo entrenado y probado con la base de datos del Hospital General de Ensenada (HGE-DB), con las clases agrupadas según la escala BI-RADS.

Tabla 7. Matriz de confusión de modelo entrenado y probado con HGE-DB, 5 clases.

			Clase verdadera		
BI-RADS 1	70 (36%)	120 (61%)	2 (1%)	3 (2%)	1 (1%)
BI-RADS 2	55 (27%)	139 (69%)	4 (2%)	3 (1%)	0 (0%)
BI-RADS 3	44 (24%)	121 (67%)	4 (2%)	10 (6%)	2 (1%)
BI-RADS 4	8 (11%)	13 (18%)	10 (14%)	41 (56%)	1 (1%)
BI-RADS 5	0 (0%)	7 (32%)	3 (14%)	4 (18%)	8 (36%)
Predición	BI-RADS 1	BI-RADS 2	BI-RADS 3	BI-RADS 4	BI-RADS 5

Tabla 8. Métricas del modelo entrenado y probado con HGE-DB, 5 clases.

Clase	**Exactitud**	**Precisión**	**Sensibilidad**	**F1-score**
BI-RADS 1	.65	.40	.36	.38
BI-RADS 2	.52	.35	.69	.46
BI-RADS 3	.71	.17	.02	.04
BI-RADS 4	.92	.67	.56	.61
BI-RADS 5	.97	.67	.36	.47

Tabla 9. Matriz de confusión de modelo entrenado y probado con HGE-DB en grupos, 2 clases.

	Clase verdadera	
BI-RADS 1,2,3	572 (99%)	6 (1%)
BI-RADS 4,5	81 (85%)	14 (15%)
Predición	BI-RADS 1,2,3	BI-RADS 4,5

Dado el bajo rendimiento inicial, se decidió reagrupar las clases en 2 categorías: BI-RADS 1, 2, 3 (benigno) y BI-RADS 4, 5 (maligno). Esta reagrupación también se decidió realizar para permitir una comparación directa con los resultados obtenidos del modelo creado a partir de la base de datos CBIS-DDSM entrenado con 2 clases.

La Tabla 9 muestra la matriz de confusión resultante de este nuevo enfoque, observando un sobre ajuste en la primera clase, esto debido a que estas clases cuentan con un mayor número de imágenes. En la Tabla 10 se presentan las métricas de exactitud, precisión, sensibilidad y F1-score para esta nueva agrupación de clases. Los resultados muestran una mejora significativa en la precisión y sensibilidad para la categoría BI-RADS 1, 2, 3 (benigno), pero el rendimiento sigue siendo limitado para la categoría BI-RADS 4, 5 (maligno).

Los resultados indican que, aunque el modelo es altamente eficaz para clasificar las mastografías benignas (BI-RADS 1, 2, 3), tiene dificultades para identificar correctamente las mastografías malignas (BI-RADS 4, 5).

Esto subraya la importancia de considerar las características específicas de los datos de entrenamiento y su representatividad para mejorar la precisión en diferentes contextos clínicos.

Tabla 10. Métricas del modelo entrenado y probado con HGE-DB en grupos, 2 clases.

Clase	Exactitud	Precisión	Sensibilidad	F1-score
BI-RADS 1,2,3	.87	.88	.99	.93
BI-RADS 4,5	.87	.70	.15	.24

Tabla 11. Matriz de confusión de modelo entrenado con CBIS-DDSM y probado con HGE-DB, 2 clases.

	Clase verdadera	
BI-RADS 1,2,3 (Benigno)	570 (99%)	8 (1%)
BI-RADS 4,5 (Maligno)	93 (98%)	2 (2%)
Predición	Benigno	Maligno

Tabla 12. Métricas del modelo entrenado con CBIS-DDSM y probado con HGE-DB, 2 clases.

Clase	Exactitud	Precisión	Sensibilidad	F1-score
BI-RADS 1,2,3 (Benigno)	.85	.86	.99	.92
BI-RADS 4,5 (Maligno)	.85	.20	.02	.04

4.3. Comparación y discusión

Para evaluar la representatividad de los modelos de RNCP entrenados con datos públicos, se entrenó el modelo con la base de datos CBIS-DDSM y se probó con la base de datos del Hospital General de Ensenada (HGE-DB). Los resultados se presentan en las Tablas 11 y 12; En la Tabla 11 se muestra la matriz de confusión del modelo entrenado con CBIS-DDSM con 2 clases y probado con HGE-DB con 2 grupos de clases; La Tabla 12 resume las métricas de exactitud, precisión, sensibilidad y F1-score derivadas de la matriz de confusión, proporcionando una evaluación cuantitativa del rendimiento del modelo en la base de datos HGE-DB.

Los resultados indican una alta precisión y sensibilidad del modelo para las categorías benignas (BI-RADS 1, 2, 3), con un F1-score de 0.91. Sin embargo, el desempeño en la detección de categorías malignas (BI-RADS 4, 5) es notablemente bajo, con una precisión de 0.20, una sensibilidad muy baja de 0.02 y un F1-score de 0.04. Estos hallazgos sugieren que los modelos entrenados con datos de CBIS-DDSM no son suficientemente representativos cuando se aplican a datos del Hospital General de Ensenada, destacando una falta de generalización en diferentes contextos clínicos. La discrepancia en el rendimiento del modelo subraya la importancia de utilizar datos de entrenamiento que reflejen las características demográficas y clínicas específicas de la población objetivo para mejorar la precisión y eficacia de los modelos de RNCP.

En la Tabla 13 se muestra la matriz de confusión del modelo entrenado con CBIS-DDSM con 5 clases y probado con HGE-DB con 5 clases, en donde puede se puede observar de igual manera la falta de la representatividad de la clase etiquetada como BI-RADS 1. Así también, en la Tabla 14 se puede observar la falta de representatividad de las demás clases observando la baja precisión y sensibilidad en los resultados.

Tabla 13. Matriz de confusión de modelo entrenado con CBIS-DDSM y probado con HGE-DB, 5 clases.

			Clase verdadera		
BI-RADS 1	0 (0%)	126 (64%)	2 (1%)	67 (34%)	1 (1%)
BI-RADS 2	0 (0%)	133 (66%)	1 (0%)	67 (33%)	0 (0%)
BI-RADS 3	0 (0%)	103 (57%)	2 (1%)	74 (41%)	2 (1%)
BI-RADS 4	0 (0%)	38 (52%)	4 (5%)	31 (42%)	0 (0%)
BI-RADS 5	0 (0%)	8 (36%)	3 (14%)	10 (45%)	1 (5%)
Predición	BI-RADS 1	BI-RADS 2	BI-RADS 3	BI-RADS 4	BI-RADS 5

Tabla 14. Métricas del modelo entrenado con CBIS-DDSM y probado con HGE-DB, 5 clases.

Clase	**Exactitud**	**Precisión**	**Sensibilidad**	**F1-score**
BI-RADS 1	.71	0	0	0
BI-RADS 2	.49	.33	.66	.44
BI-RADS 3	.72	.17	.01	.02
BI-RADS 4	.61	.12	.42	.19
BI-RADS 5	.96	.25	.05	.08

Tabla 15. Matriz de confusión de modelo entrenado con HGE-DB y probado con CBIS-DDSM, 5 clases.

			Clase verdadera		
BI-RADS 1	0 (0%)	0 (0%)	0 (0%)	2 (100%)	0 (0%)
BI-RADS 2	0 (0%)	9 (9%)	4 (4%)	81 (82%)	5 (5%)
BI-RADS 3	1 (1%)	8 (9%)	2 (2%)	79 (85%)	3 (3%)
BI-RADS 4	1 (0%)	12 (4%)	17 (5%)	292 (90%)	4 (1%)
BI-RADS 5	1 (1%)	6 (5%)	4 (3%)	110 (91%)	0 (0%)
Predición	BI-RADS 1	BI-RADS 2	BI-RADS 3	BI-RADS 4	BI-RADS 5

Tabla 16. Métricas del modelo entrenado con HGE-DB y probado con CBIS-DDSM, 5 clases.

Clase	**Exactitud**	**Precisión**	**Sensibilidad**	**F1-score**
BI-RADS 1	.99	0	0	0
BI-RADS 2	.82	.26	.09	.13
BI-RADS 3	.82	.07	.02	.03
BI-RADS 4	.52	.52	.90	.66
BI-RADS 5	.79	0	0	0

Tabla 17. Matriz de confusión de modelo entrenado y probado con CBIS-DDSM y HGE-DB, 5 clases.

			Clase verdadera		
BI-RADS 1	82 (42%)	48 (24%)	41 (21%)	25 (13%)	0 (0%)
BI-RADS 2	67 (33%)	63 (31%)	36 (18%)	35 (17%)	0 (0%)
BI-RADS 3	59 (33%)	58 (32%)	29 (16%)	35 (19%)	0 (0%)
BI-RADS 4	6 (8%)	2 (3%)	16 (22%)	49 (67%)	0 (0%)
BI-RADS 5	2 (9%)	0 (0%)	5 (23%)	9 (41%)	6 (27%)
Predición	BI-RADS 1	BI-RADS 2	BI-RADS 3	BI-RADS 4	BI-RADS 5

En la Tabla 15 se muestra la matriz de confusión del modelo entrenado con HGE-DB con 5 clases y probado con CBIS-DDSM con 5 clases, en donde se puede observar como un posible sobre ajuste de la clase con la etiqueta BI-RADS 4, sin embargo esta clase es de las que tiene un número menor de imágenes de entrenamiento del modelo generado con HGE-DB. Así como también la representatividad de las clases debería de ser aproximadamente compatible con los resultados obtenidos anteriormente con este mismo modelo, ya que la escala BI-RADS es una escala estándar en el mundo de la radiología. Y como podemos observar tanto en la Tabla 16 como en la Tabla 8, la falta de representatividad de los datos obtenidos de distintas fuentes de datos

Para finalizar con esta sección se muestra en la Tabla 17 la matriz de confusión de los resultados obtenidos al combinar estas dos bases de datos, con la finalidad de observar el comportamiento de un modelo que es entrenado con distintas fuentes de datos, y si la representatividad del conocimiento de la misma mejora o no.

En la Tabla 18 en comparación con la Tabla 8, se puede observar una disminución en las métricas de exactitud, precisión, sensibilidad y F1-score al entrenar el modelo combinando la base de datos HGE-DB y la CBIS-DDSM, en comparación a los obtenidos al entrenar el modelo solo con HGE-DB. Esta disminución podría deberse a diversos factores:

- **Falta de representatividad de los datos**: Es posible que al combinar dos bases de datos con características poblacionales y demográficas diferentes, el modelo no sea capaz de generalizar correctamente, ya que los datos pueden estar representando diferentes distribuciones de las características clave (tipo de cáncer, calidad de imagen, etc.). Esto genera que el modelo tenga dificultades para aprender patrones comunes entre las bases de datos, afectando su rendimiento.
- **Diferencias en la calidad de las imágenes**: Las imágenes de ambas bases de datos pueden tener diferentes resoluciones, configuraciones de adquisición o protocolos médicos. Estas discrepancias en la calidad y estandarización de las imágenes pueden influir en el desempeño del modelo, que no puede adaptarse correctamente a las distintas fuentes de datos.
- **Dificultades de integración**: La combinación de bases de datos heterogéneas a veces genera "ruido" en los datos que hace más difícil que el modelo pueda identificar correctamente las anomalías. Este ruido puede surgir de diferencias en los etiquetados,

Tabla 18. Métricas del modelo entrenado y probado con CBIS-DDSM y HGE-DB, 5 clases.

Clase	Exactitud	Precisión	Sensibilidad	F1-score
BI-RADS 1	.63	.38	.42	.40
BI-RADS 2	.63	.37	.31	.34
BI-RADS 3	.63	.23	.16	.19
BI-RADS 4	.81	.32	.67	.43
BI-RADS 5	.98	1	.27	.43

los estándares de clasificación o incluso en los métodos de diagnóstico utilizados en los diferentes centros.

En conclusión, una falta de representatividad en los datos combinados es una explicación probable, aunque también puede deberse a problemas con la calidad de los datos o desbalances en las clases que no se están abordando adecuadamente.

5. Discusión y conclusiones

Los resultados de este estudio indican que los modelos de RNCP entrenados en bases de datos públicas, como CBIS-DDS, pueden no generalizar bien a poblaciones específicas, como las del Hospital General de Ensenada (HGE-DB). La comparación entre estas 2 bases de datos reveló una disminución significativa en la exactitud, precisión y sensibilidad del modelo cuando se aplicó a la base de datos HGE-DB, lo cual sugiere que las diferencias en la calidad de imagen y las características demográficas pueden contribuir a esta variabilidad en el rendimiento.

5.1. Limitaciones del estudio

Entre las limitaciones del estudio se encuentran:

- **Tamaño de la muestra:** El tamaño de la muestra de ambas bases de datos puede no ser representativo de todas las posibles variaciones en las imágenes de mastografías.
- **Diferencias en los equipos de adquisición de imagen:** Las diferencias en los mastógrafos utilizados en la adquisición de las mamografías pueden afectar la consistencia de los datos.
- **Posibles sesgos en la selección de datos:** La selección de datos para entrenamiento y prueba puede introducir sesgos que afectan la representatividad y generalización del modelo.

5.2. Implicaciones y recomendaciones

Los hallazgos resaltan la necesidad de desarrollar modelos de RNCP que consideren la diversidad poblacional y las variaciones en la calidad de las imágenes. Es crucial para la implementación efectiva de estas tecnologías en entornos clínicos diversos que

consideren estas variaciones. Para mejorar la generalización y precisión de los modelos de RNCP, se recomienda:

- **Incluir diversidad en los datos de entrenamiento:** Ampliar la base de datos utilizada para el entrenamiento del modelo para incluir una mayor diversidad de fuentes, asegurando que las características demográficas y de calidad de imagen sean representativas de las poblaciones objetivo.
- **Mitigar sesgos en el entrenamiento:** Desarrollar enfoques y técnicas para identificar y mitigar posibles sesgos en los datos de entrenamiento.
- **Estudios adicionales:** Realizar estudios adicionales que incluyan una mayor variedad de bases de datos para evaluar y mejorar la representatividad y eficacia de los modelos de RNCP en diferentes contextos clínicos.

Este estudio contribuye al entendimiento de la representatividad de los modelos de RNCP y subraya la importancia de incluir datos diversos en el entrenamiento de estos modelos. Los resultados evidencian que los modelos entrenados en bases de datos públicas mostraron una disminución significativa en precisión y sensibilidad cuando se aplicaron a mastografías del Hospital General de Ensenada, lo que reafirma la necesidad de enfoques más inclusivos en el desarrollo de estas tecnologías.

Como conclusión final, la representatividad de los modelos de RNCP es crucial para su efectividad en diferentes poblaciones. Es necesario un enfoque más diverso en el desarrollo y entrenamiento de modelos de RNCP para asegurar su generalización y precisión en entornos clínicos reales.

Referencias

1. World Health Organization.: Cáncer. (2020) https://www.who.int/news-room/factsheets/detail/cancer.
2. Instituto Nacional de Estadística y Geografía.: Estadísticas a propósito del día mundial de la lucha contra el cáncer de mama (2021) https://inegi.org.mx/app/salaDeprensa/noticia.html?id=6844.
3. Ferlay, J., Ervik, M., Lam, F.: Global Cancer Observatory: Cancer Today. International Agency for Research on Cancer (2024) https://gco.iarc.who.int/today.
4. Rodríguez-Ruiz, A., Krupinski, E., Mordang, J.J.: Detection of Breast Cancer with Mammography: Effect of an Artificial Intelligence Support System. Radiology, 290(2), 305–314 (2018) doi: 10.1148/radiol.2018181371.
5. Sawyer-Lee, R., Gimenez, F., Hoogi, A.: Curated Breast Imaging Subset of Digital Database for Screening Mammography (CBIS-DDSM). The Cancer Imaging Archive (2016) doi: 10.7937/K9/TCIA.2016.7O02S9CY.
6. Benítez-Mata, B., Castro, C., Castañeda, R.: Prediction of Breast Cancer Diagnosis by Blood Biomarkers Using Artificial Neural Networks. CLAIB. IFMBE, 75 (2020) doi: 10.1007/978-3-030-30648-9-7.
7. González-Lozoya, S.M., de la Calleja, J., Pellegrin, L.: Recognition of Facial Expressions based on CNN Features. Multimed Tools Applications, pp. 1–21 (2020) doi: 10.1007/s11042-020-08681-4.
8. Rajpurkar, P., Irvin, J., Ball, R.L.: Deep Learning for Chest Radiograph Diagnosis: A Retrospective Comparison of the CheXNeXt Algorithm to Practicing Radiologists. PLoS Medicine, 15(11), e1002686 (2018) doi: 10.1371/journal.pmed.1002686.

9. Hannun, A.Y., Rajpurkar, P., Haghpanahi, M.: Cardiologist-level Arrhythmia Detection and Classification in Ambulatory Electrocardiograms Using a Deep Neural Network. Nature Medicine, 25(1), pp. 65–69 (2019) doi: 10.1038/s41591-018-0268-3.
10. McKinney, S.M., Sieniek, M., Godbole, V.: International Evaluation of an AI System for Breast Cancer Screening. Nature, 577(7788), pp. 89–94 (2020) doi: 10.1038/s41586-019-1799-6.
11. Rodríguez-Ruiz, A.: One View or Two Views: Comparison between DBT and Mammography Using an AI-Based Breast Cancer Detection System. Radiology, 290(2), pp. 493–500 (2019)
12. García-Ávila, O., Almaraz-Damián, J.A., Ponomaryov, V.: Sistema CADx para la clasificación de cáncer de mama basado en técnicas de Transfer Learning y Pseudocolor. Research in Computing Science, 150(5), pp. 65–76 (2021)
13. Thakur, A., Gupta, M., Sinha, D.K.: Transformative Breast Cancer Diagnosis Using CNNs with Optimized ReduceLROnPlateau and Early Stopping Enhancements. International Journal of Computational Intelligence Systems, 17(1), pp. 14 (2024) doi: 10.1007/s44196-023-00397-1.
14. Ko, Y.C., Chen, W.S., Chen, H.H.: Widen the Applicability of a Convolutional Neural-Network-Assisted Glaucoma Detection Algorithm of Limited Training Images Across Different Datasets. Biomedicines, 10(6), pp. 1314 (2022) doi: 10.3390/ biomedicines10061314.
15. Baba, T., Ogura, T.: Effects of Automatic Deep-Learning-Based Lung Analysis on Quantification of Interstitial Lung Disease: Correlation with Pulmonary Function Test Results and Prognosis. Diagnostics, 12(12), pp. 3038 doi: 10.3390/diagnostics12123038.
16. World Medical Association: WMA Declaration of Helsinki – Ethical principles for medical research involving human subjects. (2022) https://www.wma.net/policies-post/wma-declaration-of-helsinki-ethical-principles-for-medical-research-involving-human-subjects/
17. Sandler, M., Howard, A., Zhu, M.: Mobilenetv2: Inverted Residuals and Linear Bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4510–4520 (2018)

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

http://rcs.cic.ipn.mx

Centro de Investigación
en Computación