EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

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

Research in Computing Science

Vol. 152 No. 10
October 2023

Research in Computing Science

Research in Computing Science, Año 22, Volumen 152, No. 10, octubre de 2023, es una publicación mensual, editada por el Instituto Politécnico Nacional, a través del Centro de Investigación en Computación. Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othon de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, Ciudad de México, Tel. 57 29 60 00, ext. 56571. https://www.rcs.cic.ipn.mx. Editor responsable: Dr. Grigori Sidorov. Reserva de Derechos al Uso Exclusivo del Título No. 04-2019-082310242100-203. ISSN: en trámite, ambos otorgados por el Instituto Politécnico Nacional de Derecho de Autor. Responsable de la última actualización de este número: el Centro de Investigación en Computación, Dr. Grigori Sidorov, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othon de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738. Fecha de última modificación 01 de octubre de 2023.

Research in Computing Science, year 22, Volume 152, No. 10, October 2023, is published monthly by the Center for Computing Research of IPN.

Volume 152(10)

Advances in Pattern Recognition

J. Arturo Olvera-López
J. Ariel Carrasco-Ochoa
J. Francisco Martínez-Trinidad
Ansel Y. Rodríguez-González
Humberto Pérez-Espinosa (eds.)

Instituto Politécnico Nacional, Centro de Investigación en Computación
México 2023

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

Shallow Convolutional Neural Network to Classify Microcalcifications Clusters in Digital Mammograms

Ricardo Salvador Luna-Lozoya[1], Humberto de Jesús Ochoa-Domínguez[1], Juan Humberto Sossa-Azuela[2], Vianey Guadalupe Cruz-Sánchez[1], Osslan Osiris Vergara-Villegas[1]

[1] Universidad Autónoma de Ciudad Juárez, Instituto de Ingeniería y Tecnología, Mexico

[2] Instituto Politécnico Nacional, Centro de Investigación en Computación, Laboratorio de Robótica y Mecatrónica, Mexico

al216618@alumnos.uacj.mx, hochoa@uacj.mx, hsossa@cic.ipn.mx
{vianey.cruz,overgara}@uacj.mx

Abstract. Convolutional Neural Networks (CNNs) have proven to be an efficient tool to classify medical image data. In this paper, we propose a new shallow CNN to classify into presence or absence of microcalcifications clusters in digital mammograms. The network consists of two convolutional layers of 6 and 16 filters of size 9×9, respectively without pooling layers. After, a GlobalPooling layer is used to reduce the dimensionality from 3D to 1D and to avoid the flattening operation and the dense layers. The output layer is a sigmoid function for binary classification purposes. The loss function used was the Binary Cross Entropy. The network was trained using the INbreast database. The overall accuracy of the network is 99.3% with 8,301 parameters with as compare to the MobileNetV2 network that achieves 99.8% with 67,797,505 parameters.

Keywords: Microcalcifications detection, shallow CNN, deep learning.

1 Introduction

Breast cancer is the most prevalent cancer among women being a significant problem of public health [10]. Microcalcifications (MCs) are the most significant indirect signs of early breast cancer and its detection represents a 99% survival at 5 years or more [3]. The MCs are small deposits of calcium typically in the range of 0.1 mm to 1 mm [4]. The microcalcifications clusters (MCCs) are found in up to 50% of mammograms with confirmed cancer and correspond to at least

three MCs per cm^2 [17, 19, 20]. Mammography is described as the most widely used technique for the detection of breast cancer in early stages [4, 7].

Currently, Deep Learning (DL) models [13], trained with large amounts of data, have achieved high degrees of accuracy. In this sense, the CNNs techniques are studied in the field of MCCs detection [4]. The problem is that the mammographs or X-ray scanners have specific configurations, making them different from each other. Consequently, each hospital or clinic would have to train their own model, which would be a complicated situation because of the CNN architectures have become more complex and deeper, requiring a lot of computational power, specialized equipment and time invested to train a model.

Automatic systems to discriminate MCCs from normal tissue are always demanded. Hence, we propose a shallow CNN to classify patches of digital mammograms into presence or absence of MCCs. The network consists of two convolutional layers of 6 and 16 filters of size 9×9, respectively without pooling layers. A GlobalPooling is used to reduce the dimensionality and to avoid the flattening operation and the dense layers. For binary classification, the output layer is a sigmoid function. The loss function used was the Binary Cross Entropy (BCE). This work is a continuation of the research presented in [12].

The main contribution of the paper is:

– A light and shallow CNN to classify MCCs in digital mammograms. The network is light because of its reduced number of parameters and shallow because it has two convolutional layers only.

The rest of the article is organized as follows: in Section 2, the state of the art. In Section 3, the material and methods are shown. In Section 4, the experiments and results are presented. In Section 5, the discussion of the results is presented. Finally, the conclusions are presented in Section 6.

2 State of the Art

In our perusal, around 90 journal papers were analyzed. This section describes the current evidence found related to the detection of MCCs using DL.

Hsieh et al. [9], implemented a VGG-16 network to find MCCs in mammograms. A Mask R-CNN was used to segment the MCCs of the previously found MCCs and remove background noise. Then, the InceptionV3 was used to classify MCCs into benign or malignant. Accuracy for classification and detection was 93%, for MCs labeling 95%, and classification 91%. Precision, specificity, and sensitivity of the entire method was 87%, 89%, and 90%, respectively.

Rehman et al. [15], proposed a Fully Connected Deep CNN (FC-DSCNN) diagnosis system to detect MCCs and classify them into the classes benign and malignant. The proposed system has four steps: 1) image processing and data augmentation, 2) RGB to a grayscale transformation, 3) suspicious regions segmentation and 4) MCCs classification. First, the mammogram is divided

into subregions and sent to the FC-DSCNN to classify them into malignant and benign class. A total of 6,453 mammograms, from the DDSM and PINUM databases, were used. The results showed a sensitivity, a specificity, a precision and a recall of 99%, 82%, 89% and 82%, respectively.

Valvano et al. [18], developed two CNNs, one to detect possible Regions of Interest (ROIS) containing MCs and the other to segment the ROIS. They used 283 mammograms with a resolution of 0.05 mm from a private database. Square patches of $n \times n$, overlapped by n/2 pixels were extracted from the mammogram. For each patch, a positive label was associated when the patch contains MCs and a negative label when it does not. Each patch is processed by a CNN that detects the presence or absence of MCs. Afterwards, the ROIS found are entered into a segmentation CNN, which returns a mask. The mask is analyzed by a labeling algorithm to locate the position of each MC within the mask. Both CNNs are built of six convolutional layers with kernel of 3×3 and stride of one. Patches of 29, 39 and 49 squared pixels were used. The last size yielded the best results with an accuracy of 98.22% for the detector and 97.47% for the segmenter.

Gómez et al. [8], proposed a methodology to preprocess digital mammograms from the mini-MIAS and the UTP databases to detect presence or absence of MCCs. First, they divided mammograms in 4,292 patches of size 101×101 pixels, 3,500 used for training and 792 for testing purposes. In total, 2,360 patches out of the 4,292 contained MCCs and 1,932 did not contained these lesions. The CNN proposed used seven hidden layers with 8, 16, 32, 64, 128, 256, and 512 filters, respectively and kernel size of 3×3. After each convolutional layer, a 2×2 MaxPooling layer and a layer of Rectified Linear Unit (ReLU) activation functions were added. A softmax layer was used for multiclass classification purposes. The model yielded an accuracy of 96.26% during training while during testing 95.83%.

Luna et al. [12], presented a comparison of the CNN architectures InceptionV3, DenseNet121, ResNet50, VGG-16, MobileNetV2, LeNet-5 and AlexNet for classifying MCCs. In the best testing accuracies, InceptionV3 achieved 99.71%, DenseNet121, ResNet50 and VGG-16 yielded 99.74%, MobileNetV2 obtained the best overall accuracy with 99.84%, LeNet-5 99.30% and AlexNet 99.40%.

3 Materials and Methods

The CNN model was created using the Google Colaboratory Integrated Development Environment (IDE) [6], Python 3.0 language, and TensorFlow framework 2.0 [1]. The IDE automatically allocates the necessary computational resources.

The INbreast public database [14] was used to train the model. It contains 410 digital mammograms of size $2,560 \times 3,328$ pixels and $3,328 \times 4,084$ pixels with 8-bit depth (grayscale). The mammograms are labeled with various types of lesions such as asymmetries, calcifications, microcalcification clusters, distortions, regions spiculate, masses or nodules, and pectoral muscle. These

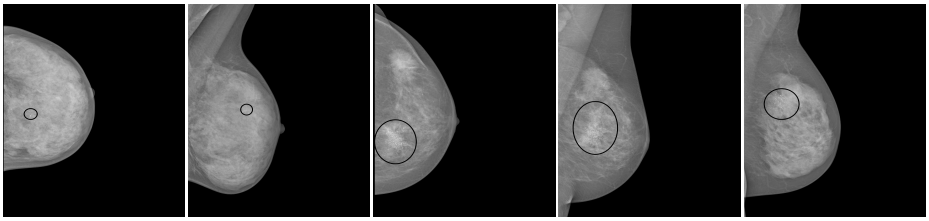
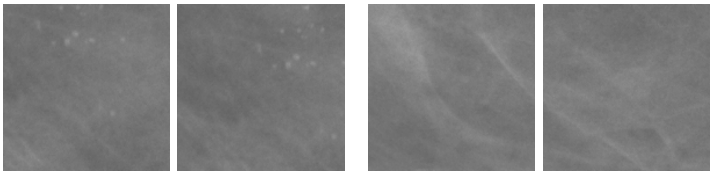


Fig. 1. Examples of mammograms showing MCCs highlighted within a circle.



(a) (b)

Fig. 2. Patches of digital mammograms (a) with MCCs and (b) normal tissue.

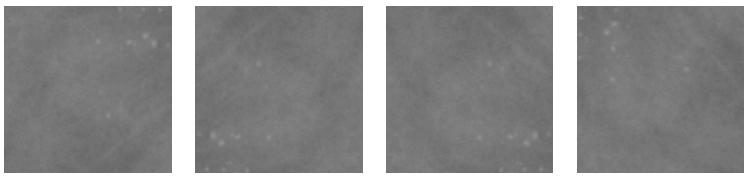
mammograms were acquired using the MammoNovation Siemens FFDM scanner. Each pixel in the image represents 70 microns. In this work, only cases labeled as MCCs were utilized. In Fig. 1, five example images labeled as MCCs from the database are shown.

3.1 Data Preparation

Digital mammograms in the INbreast database are stored in DICOM format. Therefore, we converted them into the PNG format. The labeling and coordinates of the breast lesions were searched in their corresponding XML file, independent of the images. To mark the lesions on the digital mammograms, it was necessary to develop a computer program to read the coordinates of the MCCs from the file.

Patch Extraction. The proposed CNN model processes the mammograms in patches of 1 cm^2. This corresponds to squares of area 144×144 pixels. We developed a computer program to extract the patches from the mammogram. Figs. 2(a) and 2(b) show two examples of patches with MCCs and normal tissue respectively. An expert radiologist doctor analyzed the patches manually discarding the wrong ones. She selected a total of 1,576 patches with MCCs and 1,692 without these injuries.

Data Augmentation. Due to the limited availability of mammograms labeled with MCCs in the INbreast database, there were not enough patches to adequately train, validate, and test the models. To address this issue, we augmented the database to enhance precision. In Fig. 3, we illustrate four



(a) (b) (c) (d)

Fig. 3. Examples of geometric transformations on one patch. (a) Reflection, (b) 180^o turn, (c) reflection and 180^o turn, and (d) 90^o turn.

geometric transformations applied to the patches. This augmentation process resulted in a total of 6,304 transformed patches with MCCs and 6,768 without MCCs. Only geometric transformations were employed to augment the data in order to ensure that the features of each patch remained unaltered. As a result, the augmented database comprises 7,880 patches with MCCs and 8,460 patches without MCCs, totaling 16,340 patches.

The Datasets. A total of 15,760 patches were collected, consisting of 7,880 patches containing MCCs and 7,880 patches representing normal tissue samples. It was determined by the Pareto's Principle [2] that 80% of the data would be used for both training and validation, while the remaining 20% would be reserved for testing purposes. To be specific, for training purposes, we utilized 64% (10,088 patches).

For validation, we allocated 16% (2,520 patches), and for testing, we reserved 20% (3,152 patches). This partitioning strategy proved effective in achieving optimal results. To ensure consistency, all patches were normalized by dividing their pixel values by 255, given that the pixel depth was eight bits.

The Proposed Architecture. In [12], we compared different CNN architectures and recommend the best performing layer types for classifying MCCs. From here, the CNN MobileNetV2 [16] yielded the highest overall accuracy of 99.8% with 67,797,505 parameters.

In the same work, the CNN LeNet-5 [11] yielded an accuracy of 99.3% with 2,233,365 parameters. The difference in accuracies is only 0.539%. However, the LeNet-5 is 30 times smaller. Therefore, to classify MCCs it is not necessary to implement Deep CNNs. Hence, after extensive testings and combinations of the layers suggested in [12], we obtained the shallow CNN model shown in Fig. 4, that comprises only two convolutional layers without any intervening pooling layers. To optimize the number of trainable parameters, the conventional combination of a flat layer with a fully connected layer was replaced with a GlobalPooling layer, resulting in a significant reduction of parameters.

The input layer extracts the characteristics of the patches x from mini batches of size 64. The layer comprises 6 filters W_0 of size 9×9 with biases B_0 and a layer of ReLU activation functions. The output can be defined as in Eq. (1):

$$F_0 = \max(0, W_0 x + B_0). \tag{1}$$

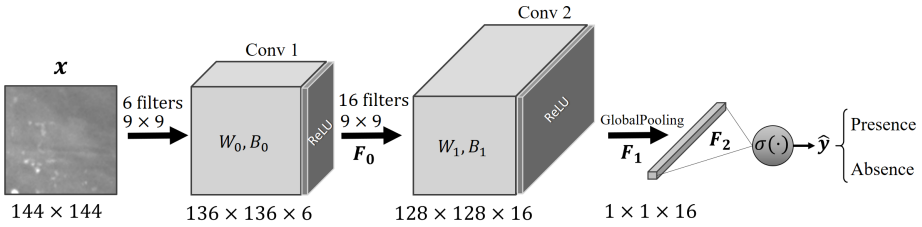


Fig. 4. Proposed CNN architecture to classify MCCs.

The second layer has 16 filters W_1 of 9×9 with another ReLU layer. Similarly, at the output of this layer we have Eq. (2):

$$F_1 = \max(0, W_1 F_0 + B_1). \tag{2}$$

The output volume F_1 represents the feature maps of the second layer which are further processed by a GlobalPooling layer, which extracts the highest value of each map as shown in Eq. (3):

$$F_2 = \max(F_1). \tag{3}$$

The features from the GlobalPooling are sent to the sigmoid to predict the probability of the binary variable as represented in Eq. (4):

$$\hat{y} = \sigma(F_2). \tag{4}$$

The BCE cost function is shown in Eq. (5):

$$\mathcal{L}(y, \hat{y}) = -\frac{1}{m} \sum_{i=1}^{m} \left[y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i) \right]. \tag{5}$$

Where m is the size of the training set used, y_i is the target value that can take two possible values, 0 or 1 (presence or absence of MCCs), and $\hat{y}_i$ is the predicted value. We used dropout to regularize the network.

Hyperparameter Tuning. The tuning of hyperparameters consists of choosing the values that achieve the maximum performance of the model. We used the random search method [5] for hyperparameter tuning, because the proposed network is very short and allows us to specify the number of models to train. Besides, we can base our search interactions on our computational resources (which is limited) or the time taken by iteration. The validation loss was monitored for up to 100 epochs. The resulting hyperparameters are shown in Table 1.

Table 1. Resulting hyperparameters for the proposed CNN architecture.

Hyperparameter	Value
First Convolutional Layer	
Number of features map	6
Kernel size	9 x 9
Activation funcion	ReLU
Second Convolutional Layer	
Number of features maps	16
Kernel size	9 x 9
Activation Function	ReLU
GlobalPooling Layer	
Number of features	16
Dense Layer	
Units	1
Activation Function	Sigmoid
Network Training	
Loss function	Binary Cross Entropy (BCE)
Optimization algorithm	Adaptive Moment Estimation (ADAM)
Learning rate	0.001
Batch size	64
Epochs	100
Dropout	Keep 80%

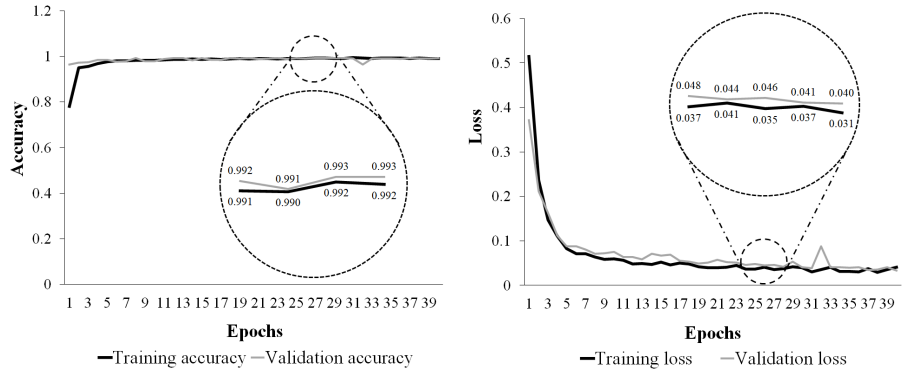
4 Experiments and Results

The output layer is a sigmoid with real output between 0 and 1. In our work, the range between 0 and 0.49 indicates '0' or absence and from 0.5 to 1 indicates '1' or presence of MCCs in a patch. All the models tested were trained with one hundred epochs. At the end of each epoch, the models were validated with the validation set to obtain the accuracy and the error. We selected the model with the highest accuracy. The dense layer is the output sigmod whose inputs are the 16 output features delivered by the GlobalPooling layer plus a bias term. Table 2 presents the number of parameters per layer, as well as the sum of these parameters after hyperparameter tuning.

The accuracy and loss plots during the training and validation are depicted in Fig. 5. Upon closer inspection, it becomes evident that both the training and validation results exhibit remarkable similarity. Table 3 shows the results of the proposed network compared with the state-of-the-art MobileNetV2 and LeNet-5.

Table 2. Parameters of the proposed CNN architecture.

Type of Layer	Parameters
Convolution 1	492
Convolution 2	7,792
Dense	17
Total	**8,301**



(a) (b)

Fig. 5. Proposed CNN performance during training and validation epochs. (a) shows the accuracy performance and (b) shows the loss performance.

5 Discussion

From Figure 5, it shows that the training and validation accuracies follow each other very closely. Also, the losses of training and validation are similar and the validation loss does not increase after a number of epochs, showing that no overfitting is present.

Table 3 shows that the accuracy of the MobileNetV2 is greater by only 0.5%. However, the number of parameters of the proposed network is much less even than the LeNet-5, which makes the proposed network light and shallow suitable for MCCs classification.

Clean data is important for optimal model performance. In this work, the data set was cleaned by an expert. She validated the model by observing the patch and checking the decision made by network (presence/absence).

The continuation of the present investigation is a faster residual network, with better performance, under the assumption that it is not necessary a deep neural network to classify MCCs. Also, the networks reported in [12] are being investigated to include other type of lesions.

Table 3. Performance comparison of the proposed CNN versus the MobileNetV2 and the LeNet-5.

Architecture	Accuracy	Parameters
MobileNetV2	**99.8**%	67,797,505
LeNet-5	99.3%	2,233,365
Proposed	99.3%	**8,301**

6 Conclusions

We presented a new shallow CNN architecture to classify MCCs using only two convolutional layers without pooling layers between the convolutions and a GlobalMaxPooling layer. The results of the proposed model yielded similar results to the deeper CNN MobileNetV2. This demostrate that for MCCs classification, shallow networks produce similar results to their deeper and more complex counterparts. The proposed model is already implemented in a web application that inspects digital mammograms. The application is been tested in collaboration with the Centro de Imagen e Investigación (Medimagen) of Chihuahua, México. Currently, we are compiling a database of Mexican mammograms to train shallow models that can work in hospitals and clinics of the country.

Acknowledgments. Ricardo Luna thanks the UACJ for the support provided and the CONACYT for the scholarship granted to pursue his doctoral studies. We would like to express our gratitude to the radiologist Dra. Karina Núñez, from the Salud Digna Clinical and Imaging Laboratory, for her support in carrying out this work.

References

1. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C. a.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015), https://www.tensorflow.org/
2. Ali, S., Mohammed, A., Hefny, H.: An enhanced deep learning approach for brain cancer MRI images classification using residual networks. Artificial Intelligence in Medicine, vol. 102, pp. 101779 (2020)
3. American Cancer Society: Tasas de supervivencia del cáncer de seno. cancer.org. https://www.cancer.org/es/cancer/cancer-de-seno/comprension-de-un-diagnostico-de-cancer-de-seno/tasas-de-supervivencia-del-cancer-de-seno.html, accessed May. 13, 2023
4. Basile, T., Fanizzi, A., Losurdo, L., Bellotti, R., Bottigli, U., Dentamaro, R.: Microcalcification detection in full-field digital mammograms: A fully automated computer-aided system. Physica Medica, vol. 64, pp. 1–9 (2019)
5. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization, vol. 13, pp. 281–305 (2012)

6. Bisong, E.: Building machine learning and deep learning models on Google Cloud Platform: A comprehensive guide for beginners. Apress Berkeley, CA, 1st edn. (2019)
7. Cronin, K., Lake, A., Scott, S., Firth, A., Sung, H., Henley, S.: Annual report to the nation on the status of cancer, part i: National cancer statistics: Annual report national cancer statistics. Cancer, vol. 124, pp. 2785–2800 (2018)
8. Gómez, A., Echeverry-Correa, D., Gutiérrez, A.: Automatic pectoral muscle removal and microcalcification localization in digital mammograms. hir, vol. 27, no. 3, pp. 222–230 (2021)
9. Hsieh, Y., Chin, C., Wei, C., Chen, I., Yeh, P., Tseng, R.: Combining VGG16, Mask R-CNN and Inception V3 to identify the benign and malignant of breast microcalcification clusters. In: 2020 IEEE International Conference on Fuzzy Theory and Its Applications (iFUZZY). pp. 1–4. IEEE, Hsinchu, Taiwan (2020)
10. International Agency for Research on Cancer: https://gco.iarc.fr/today/data/factsheets/cancers/20-Breast-fact-sheet.pdf, last accessed May. 08, 2023
11. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE, vol. 86, pp. 2278 – 2324 (1998)
12. Luna, R., Ochoa, H., Sossa, J., Cruz, V., Vergara, O.: Comparison of deep learning architectures in classification of microcalcifications clusters in digital mammograms. In: Pattern Recognition. pp. 231–241. Springer Nature Switzerland, Cham (2023)
13. Miotto, R., Wang, F., Wang, S., Jiang, X., Dudley, J.: Deep learning for healthcare: review, opportunities and challenges. Briefings in Bioinformatics, vol. 19, pp. 1236–1246 (2018)
14. Moreira, I., Amaral, I., Domingues, I., Cardoso, A., Cardoso, M., Cardoso, J.: INbreast: Toward a full-field digital mammographic database. Academic radiology, vol. 19, pp. 236–48 (2011)
15. Rehman, K., Li, J., Pei, Y., Yasin, A., Ali, S., Mahmood, T.: Computer vision-based microcalcification detection in digital mammograms using fully connected depthwise separable convolutional neural network. Sensors, vol. 21, pp. 4854 (2021)
16. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4510–4520. IEEE, Salt Lake City, UT, USA (2018)
17. Sickles, E., D'Orsi, C., Bassett, L.: ACR BI-RADS® mammography. In: ACR BI-RADS® atlas, breast imaging reporting and data system. American College of Radiology, 5th edn. (2013)
18. Valvano, G., Santini, G., Martini, N., Ripoli, A., C., I., Chiappino, D.: Convolutional neural networks for the segmentation of microcalcification in mammography imaging. Journal of Healthcare Engineering, vol. 2019, pp. 1–9 (2019)
19. Wang, J., Nishikawa, R., Yang, Y.: Global detection approach for clustered microcalcifications in mammograms using a deep learning network. Journal of Medical Imaging, vol. 4, pp. 024501 (2017)
20. Wang, J., Yang, Y.: A context-sensitive deep learning approach for microcalcification detection in mammograms. Pattern Recognition, vol. 78, pp. 12–22 (2018)

ISSN 1870-4069

Evaluation of View-wise ResNet on the Digital Database for Screening Mammography

Osmar Moreno-Rivas[1], Alfonso Rojas-Domínguez[1], Matías Alvarado[2], Manuel Ornelas-Rodríguez[1]

[1] Tecnológico Nacional de México,
campus León,
Mexico

[2] Instituto Politécnico Nacional,
Centro de Investigación y Estudios Avanzados,
Mexico

m22240203@leon.tecnm.mx, alfonso.rojas@gmail.com, matias@cs.cinvestav.mx, manuel.ornelas@leon.tecnm.mx

Abstract. Screening mammography aided by deep learning classifiers has demonstrated enhanced efficacy by reducing False Positives, consequently minimizing unnecessary recalls that cause anxiety among participants. However, the development of effective classifiers necessitates substantial computational resources and a vast amount of training data. Despite these requirements, it is generally assumed that these models possess a high level of generalization, enabling them to perform well on similar datasets to the ones they were trained on. In this study, we assess the performance of a ResNet-based model for screening mammography presented by Wu et al. (2019). This model was trained on an extensive dataset of over one million images and reported an Area Under the ROC curve (AUC) of 0.88. Previous studies have fine-tuned similar models using additional data, achieving AUC values around 0.9. However, these studies had limited sample sizes in their test sets, consisting of only a few hundred images, thereby restricting the applicability of their findings and conclusions. In contrast, our evaluation utilizes the DDSM, the largest publicly available dataset for screening mammography, containing over 10,000 images. The evaluated model achieved an AUC of approximately 0.50, significantly lower than the performance reported by other authors.

Keywords: Residual networks, deep learning, breast cancer screening.

1 Introduction

Breast cancer is the most commonly diagnosed cancer among women; the World Health Organization estimates that 7.8 million women were diagnosed with breast cancer in 2021. Moreover, it was estimated that in 2022 30% of diagnosed cancers in women would be breast cancer according to the National

Breast Cancer Foundation[3]. However, the lifetime breast-cancer survival rate increases with timely diagnosis; this is why new detection and diagnostic techniques for breast cancer are constantly being developed. The most common acquisition technique employed in screening tests for detection of breast cancer is mammography, because it is a non-invasive and low cost technique compared with other techniques like MRI (magnetic resonance imaging) or CT-scan (computerized tomography). Screening tests for early detection of breast cancer through mammography can be considered a first line of defense, from which a few cases that warrant further testing can be identified (see Fig. 1). Screening tests are indicated for women who have not presented any symptoms potentially indicative of breast cancer, but who belong to an age-range in which prevalence of the disease is the highest. These studies have the ability to detect abnormalities that cannot be felt through palpation or self-examination.

Computer-aided diagnosis (CAD) systems have been successfully used to support human decision-making in radiological image analysis and precision medicine in general [4]. Traditional approaches to breast cancer CAD involve extracting manually-designed features to detect breast masses and classify them as probably benign or malignant [5, 11]. However, the outputs from these CAD systems in conjunction with radiologists' reviews result in numerous false-positives, which can increase reading times [6]. Alternative approaches involve learning features directly from the full images through deep neural networks [8]; we can list a variety of such: Convolutional Neural Networks (CNNs), Residual Networks (ResNets), Dense Networks, among others [11].

In particular, ResNets have shown favorable results on detection and classification of breast cancer. Xiang Yu et al. 2020 [14] obtained an average accuracy of 95.74% correct classification on the MINI-MIAS and InBreast datasets. On the other hand, Y. Chen et al. [1] has reported an accuracy of 93% with a CCN-based model fine-tuned with a ResNet architecture on the CBIS-DDSM [9] database. It is worth mentioning that the databases that have been employed in those previous studies contain only a few hundred images, and thus the reported results are limited. Our contribution is the evaluation of a reportedly efficient model [13] on DDSM, a public dataset with over 10,300 mammographic images.

Honig et al. 2019 [7] conducted a study about impact factors of False Positives (FPs) in recall cases; their study found that 91.6% of 1,258 recalled cases were FPs. Thus, a vast majority of women who received a recall notification had not actually developed breast cancer, despite initial screening results suggesting otherwise, which can result in unnecessary procedures and psychological effects like elevated anxiety in women [2]. This justifies further research towards the improvement of breast cancer screening systems with tools like DL (Deep Learning) models, to ensure that women receive the most accurate and reliable information about their health. On the other hand, DL-CAD systems have proved the reduction of FPs per image to 69% in comparison with traditional CAD systems that often yield a higher number of FPs [10]. Moreover, DL-CAD

[3] https://www.nationalbreastcancer.org

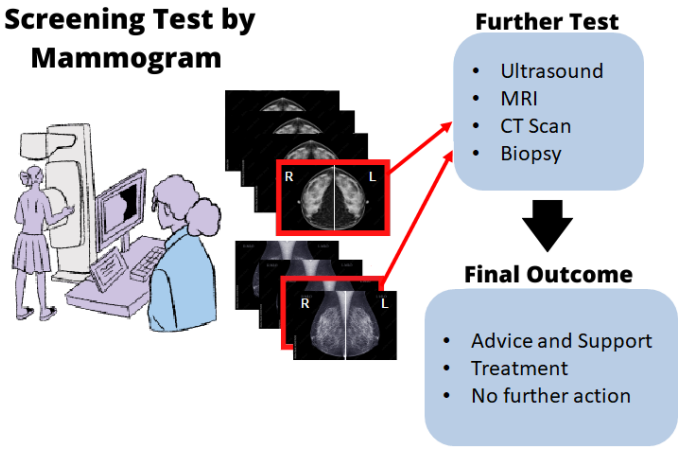


Fig. 1. Breast cancer mammography-based screening test. A very small portion of the cases (red outline) require further tests such as Ultrasound, MRI, or a biopsy. After further detection and diagnostic tests, a final outcome can be reached.

systems reduce 17% the reading time per case performed by radiologists in comparison when they used CAD systems [10], which in turn improves the benefit-cost ratio for massive studies of breast screening in women. Additionally, these systems provide more precise results and report them in less time.

2 Materials and Methods

2.1 DDSM Database and Inclusion Criteria

The Digital Database for Screening Mammography (DDSM) is a public resource that contains 2,588 exams for detection of breast cancer, including two standard anatomical views: Cranio-Caudal (CC) and Mediolateral Oblique (MLO). The images have an average size [height × width] in pixels, for CC: $[2,677 \times 1,942]$ and for MLO: $[2,974 \times 1,748]$. The optimal size stated by Wu et al. [13] to run their model is $[2,290 \times 1,890]$; thus the DDSM images mostly comply with these parameters. There are three possible outcomes for each study in the DDSM: *benign*, *malignant* and *no finding* (or *normal*).

An exam generally consists of four mammograms: L-CC (Left breast-CC view), R-CC (Right breast-CC view), L-MLO (Left breast-MLO view) and R-MLO (Right breast-MLO view). Nevertheless, five exams in the DDSM only include three images. We also found 209 studies that have more than one overlay in the same image (meaning that there is more than one abnormality present in one image). In 130 cases in which an abnormality could only be found in one of the two views (either CC or MLO, but not both). Finally, we found 5 studies with duplicated patient ID. After excluding the studies described above, we ended up with a total of 2,244 studies to be used in our evaluation.

2.2 Neural Model for Screening Mammography

In 2019 Wu et al. described a multi-view neural-network system for screening mammography based on ResNets [13] which consists of two core modules: (i) four view-specific columns that output a fixed-dimension hidden representation for each mammography view, and (ii) two fully-connected (FC) layers to map the hidden representations to the output predictions [13]. Depending on how the hidden representations are aggregated into a final prediction, four different models are produced: View-wise, Image-wise, Side-wise and Joint models.

According to Wu et al., the View-wise model obtained the best results among their models, with an Area under the ROC Curve (AUC) of approximately 88%. Consequently, this View-wise model is employed in our evaluation and is described below. A schematic representation of the model is shown in Fig. 2. In the view-wise model, the 256-dimensional hidden representations of the CC views (L-CC and R-CC) are concatenated together before going through the FC layers. Independently, the representations of the MLO views (L-MLO and R-MLO) are also concatenated together and pass through their own FC layers. This process produces independent predictions for CC and MLO views, which are averaged during inference to produce the breast-wise predictions [13]. At the top of Fig. 2 it can be seen that the model produces four numerical predictions (two for each breast) named Right-Benign: $\hat{y}_{R,b}$, Right-Malignant: $\hat{y}_{R,m}$, Left-Benign: $\hat{y}_{L,b}$ and Left-Malignant:$\hat{y}_{L,m}$. These predictions are to be compared against binary labels that correspond to the ground truth of the cases in the evaluation dataset.

2.3 Model Predictions

To evaluate the View-wise model its predictions are binarized and compared against binary labels that represent the pathology of each case in our evaluation dataset. The binary labels (two per breast, four per study) indicate the presence (1) or absence (0) of a finding, either Benign or Malignant, in the corresponding breast. Table 1 shows an example of the labels for one patient; in this example, a malignant finding is present in the left breast and a benign finding is present in the right breast. The view-wise model produces four numerical predictions between 0 and 1; binarization of the predictions is done by comparing these against a detection threshold, as illustrated in Fig. 3.

Whenever the value of a numerical prediction is equal to or above the detection threshold, it is assigned a value of 1, otherwise it is assigned a value of 0. If the values of both malignant and benign predictions of the same breast surpass the threshold, we assign 1 to the malignant prediction and 0 to the benign prediction; this is done to avoid contradictory predictions and to prioritize the detection of malignant findings over benign ones. Importantly, we test ten different threshold values, regularly spaced between 0 and 1, to generate ten different sets of results with which to evaluate the model and build a ROC curve.

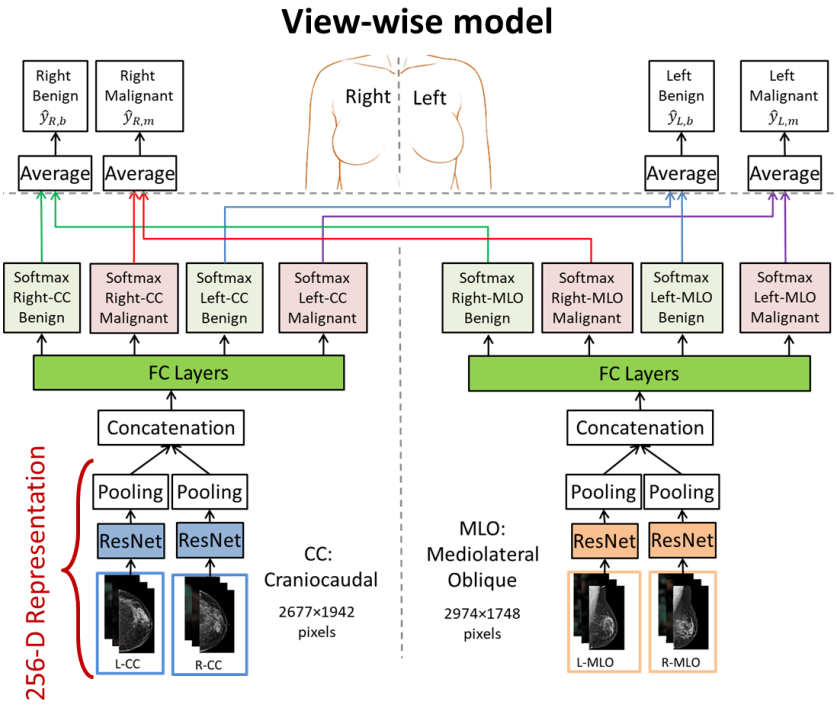


Fig. 2. Schematic representation of the View-wise model proposed by Wu et al. [13].

Table 1. Example of labels for one mammographic study.

Left Benign	Left Malignant	Right Benign	Right Malignant
0	1	1	0

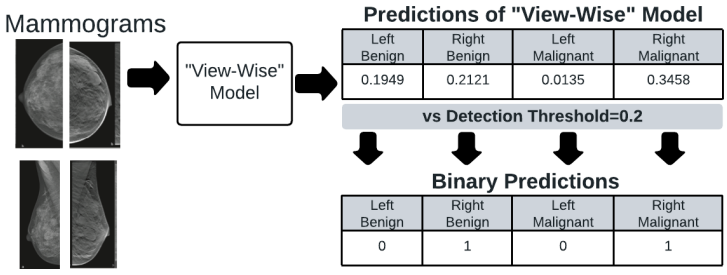


Fig. 3. Example of binary predictions generated by thresholding; a binary prediction is obtained for each side and type of finding. Notice that the threshold=0.2 is only used to illustrate the process; in the actual evaluation ten threshold values were employed.

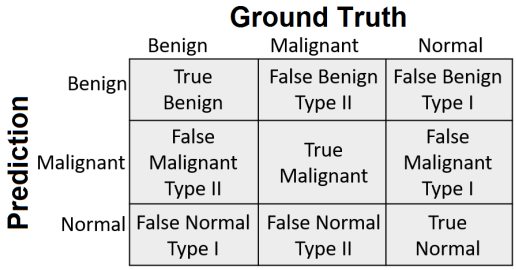


Fig. 4. Multi-class confusion matrix employed to evaluate the model; notice the different possible errors for each of the three classes: *Benign*, *Malignant* and *Normal*.

2.4 Model Evaluation

To evaluate the classification performance of the model, a multi-class confusion matrix is employed. There are three classes in our test dataset: *Benign*, *Malignant*, and *Normal* (see Fig. 4). For each class the model's prediction can be correct, or it can be one of two types of misclassification, depending on the Ground Truth (GT). For instance, if the model produces a *Benign* prediction, this may be a True Benign, a False Benign of Type I (GT indicates that the true class is *Normal*), or a False Benign of Type II (the actual class is *Malignant*).

To correctly compute each of the values in a confusion matrix, the binary predictions produced by the model need to be compared against the corresponding GT labels. Figure 5 shows a flowchart of the different comparisons that must be carried out to reach one (and only one) of the nine possible outcomes contained in a confusion matrix. Notice that the predictions and GT labels correspond to individual mammograms, while each confusion matrix corresponds to one of the patient's sides (Left or Right). In this work we tallied the classification results independently, for the Left side and for the Right side.

Complementarily to the multi-class evaluation, we also performed an evaluation of the model in which only the Malignant and Normal classes are considered (Benign cases were treated as Normal). Thus we can examine if there is a difference in the performance of the model when considering only two classes instead of three. In the work of Wu et al. [13] there is not sufficiently detailed information regarding how numerical predictions are binarized and on how the Benign predictions were treated to obtain their final results. Our best assumption is that Benign predictions were ignored in the computation of their ROC curves.

3 Results and Discussion

Table 2 shows a small portion of the results of the model obtained with different values of the detection threshold. As can be seen, the performance of the View-wise model is higher for the smaller values of the detection threshold and

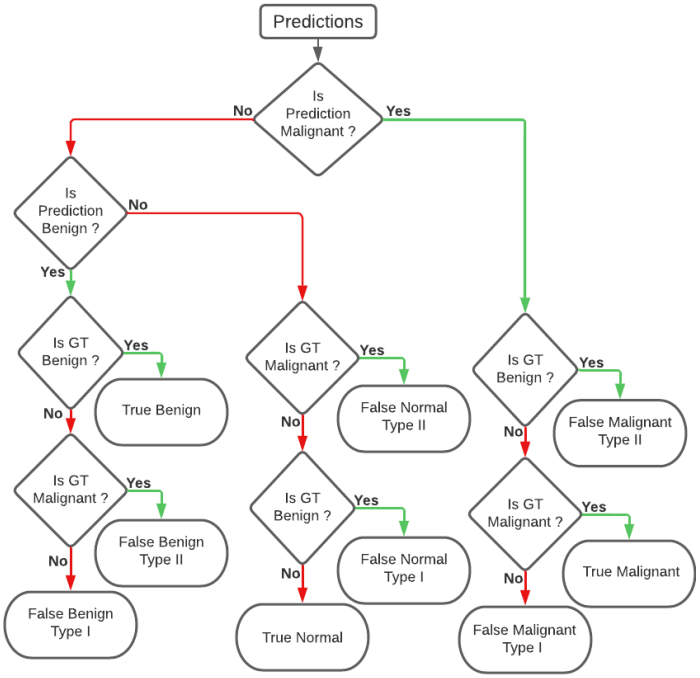


Fig. 5. Flowchart to generated the classifications of each type finding.

gradually decreases as the model is tested with larger threshold values. However, even for the smallest threshold value, the model shows a very low performance.

This low performance is visualized for both sides: in Table 2-a) and 2-d) we can see that class Benign has more instances correctly classified than classes Malignant and Normal. In contrast, Table 2-c) and 2-f) show that class Normal has more instances correctly classified than classes Malignant and Benign. This indicates that with larger threshold values the correct predictions fall much more into the Normal class, because the model does not detect as many abnormalities as with smaller threshold values. On the other hand, correct classifications for class Malignant are always very low, regardless of the threshold value applied. Specially it should be noticed that for larger threshold values there are no hits for instances on the Left side (Table 2-b) and 2-c)) and there is only four instances correctly classified among instances of the Right side (Table 2-e) and 2-f)).

Based on the confusion matrices obtained for the two-class evaluation (on classes Malignant and Normal) for different threshold values, which effectively represent a set of operating points, an ROC curve was obtained for the View-wise model on the DDSM dataset, with an AUC of about 0.5. Clearly this (around 50% correct classification) is not a desirable performance, as it is indicative that the model does not posses any predictive ability. Also, this was not the expected result, given that Wu et al. reported an ROC curve of this same model on their dataset with about 88% correct classification.

Table 2. Example confusion matrices; a) and d): threshold=0.1, b) and e): threshold=0.6, c) and f): threshold=0.9.

			Ground Truth				
	Left Side				**Right Side**		
Class	Benign	Malignant	Normal		Benign	Malignant	Normal
a)	405	338	128	d)	359	302	1232
Pred.	20	37	112		26	55	236
	33	37	1134		3	5	26
b)	0	5	10	e)	9	22	86
Pred.	0	0	1		0	3	2
	458	407	1363		379	337	1406
c)	0	0	0	f)	0	3	2
Pred.	0	0	0		0	1	0
	458	412	1374		388	358	1492

As can be observed, the evaluation on a different dataset (other than that with which it was trained) is not favorable to this View-wise model, as we obtained approximately 33% lower performance on the DDSM than what was previously reported on the NYU dataset [13]. Moreover, we can observe that in both views (CC and MLO) the DDSM images possess close to optimal sizes to be processed by the model, but we suspect that there are other properties that may affect the performance, such as poor contrast, different range of the intensity values, noise in the images etc. Although it is not very probable, image pre-processing could also affect the performance of the model.

Recent investigations have made similar observations. Frazer et al. [3] used models pretrained with the NYU dataset of Wu et al. [13] as the base for their whole-image classifier, observing poor performance (around 55% correct classification). Then the models were retrained with a small subset of DDSM and up to 87% correct classification was obtained. Similarly, Shen et al. [12] found that when applying transfer learning with around 239 images on a model pretrained with CBIS-DDSM, classification of the InBreast dataset improved. Because of this, we attempted to classify the DDSM database using the model proposed by Shen et al. [12]. This model is open source and has an architecture that enables it to classify small patches and extend the patch classifier to the entire image. We obtained similar results with the model proposed by Shen et al. as with the model of Wu et al., achieving an AUC of approximately 0.5. Subsequently, we applied transfer learning with a small subset of the DDSM database. Fig. 7 illustrates the training and validation curves. We observed that the model of Shen et al. exhibited good retraining; however, it did not improve the classification performance as expected. We suspect that this behavior is due to the heterogeneity of the images in the DDSM database, where some images have excessive contrast, while others have a noisy background, as illustrated in Fig. 6.

To address this issue, we clustered the mammograms to obtain image sets with reduced heterogeneity. We first performed segmentation to separate the breast from the background. This was done by applying a manually defined threshold of 128 (half of the grayscale range from 0 to 255). Alternatively we also utilized Otsu's method to determine the optimal threshold value for each

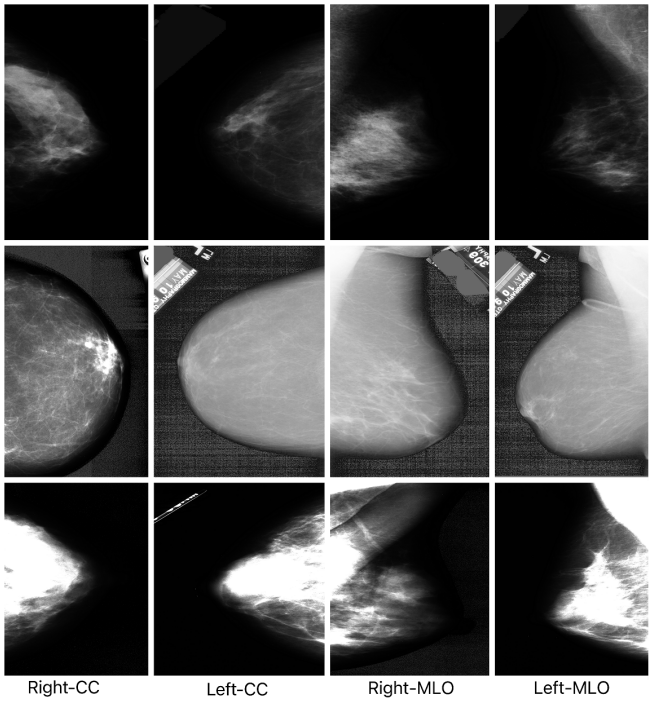


Fig. 6. A few images of DDSM to illustrate the heterogenity in the dataset.

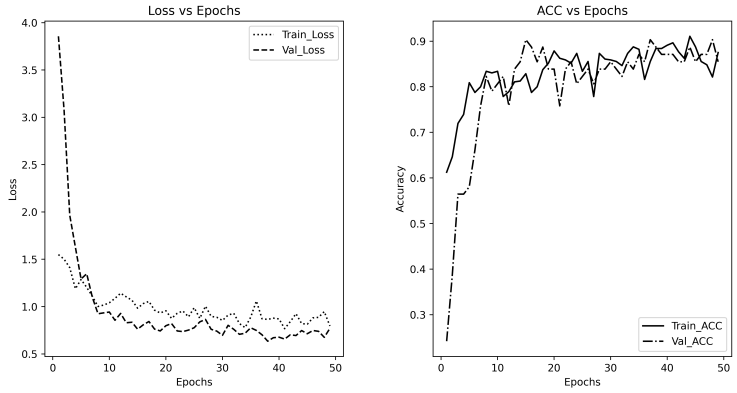


Fig. 7. Results of fine-tuning the model by Shen et al. with 600 images of DDSM.

image, resulting in more accurate segmentation (Fig. 8). Next, background and breast tissue features (mean and standard deviation of the pixel values) were obtained from the images and the K-means algorithm was employed to partition the DDSM images into subsets with similar properties.

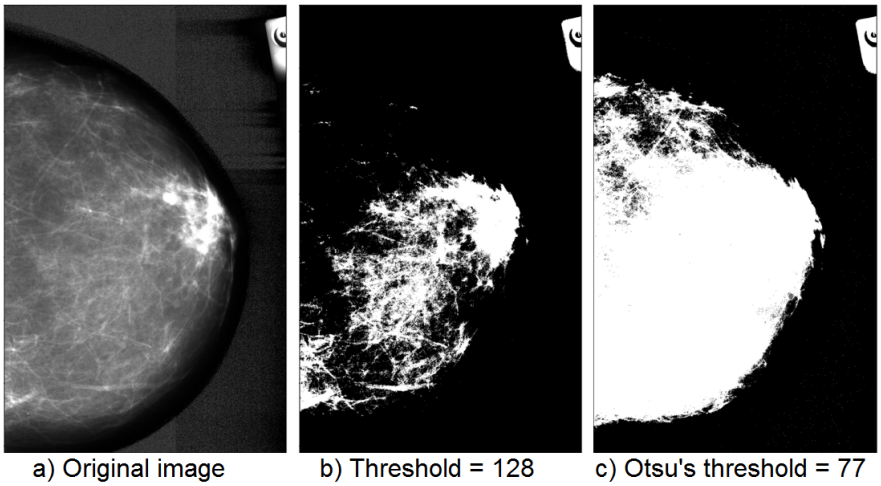


Fig. 8. Image segmentation with manual threshold and Otsu's threshold.

We applied different methods to determine the correct number of clusters: gap statistic, elbow method, and silhouette. All of these methods converge to the most suitable number of clusters, which is 4 (Fig. 9-a). The cluster points represented by the features extracted from the images using the Otsu's threshold are shown in Fig. 9-b.

Analysis of the data clusters in Figure 9-b reveals that cluster 1 comprises images with dark background (small background mean) and bright tissue (large tissue mean). Moreover, the grayscale values of the tissue exhibit significant variation (large standard deviation). Overall, these characteristics indicate well-equalized images.

Finally, the model was fine-tuned separately with 627 images per cluster, in proportions of 90% and 10% for training and validation, respectively, according to the methodology followed by Shen et al. [12], classification results reported in Table 3. The clusters generated by features extracted through Otsu's method display highly similar AUC scores, all above 0.60. Notably, cluster 3 exhibits the best performance in training, with an AUC=0.92. Conversely, the clusters generated via the single threshold method show greater discrepancies in AUC scores on the test set, ranging from 0.38 to 0.82. The lowest score is observed for cluster 2, while cluster 3 achieves the highest score among both methods, reaching an AUC=0.82. Additionally, cluster 3 when generated from a single threshold showcases the highest score in training, AUC=0.83, compared to the other clusters from the same method. The closest to this performance is cluster 1, with an AUC=0.82.

In conclusion, our analysis indicated that using four clusters yields the most appropriate split of the DDSM data. Features extracted through the Otsu's method demonstrate consistent performance across the clusters, while the single

Table 3. AUC results of fine-tuning the subsets on DDSM.

	with Otsu's threshold			**with threshold=128**		
Cluster	Images	Training	Test	Images	Training	Test
0	198	0.69	0.64	313	0.78	0.54
1	462	0.84	0.68	332	0.82	0.66
2	316	0.79	0.69	209	0.60	0.38
3	143	0.92	0.66	259	0.83	0.82

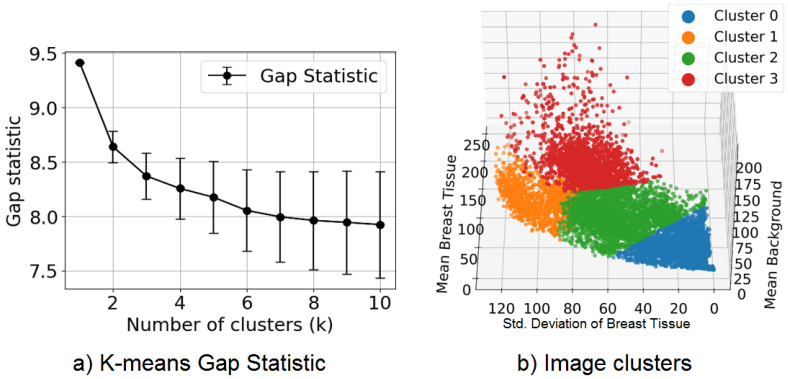


Fig. 9. Determinate the K-value and features representation for Otsu Threshold A) Gap statistic method applied to extracted features to determined best K-value B) extracted features from images, grouped into four clusters through K-means algorithm.

threshold method displays more variability. Specifically, cluster 3 consistently exhibits a good performance in both training and testing, regardless of the threshold method used. These results highlight the value of cluster analysis and careful feature selection for images to reduce heterogeneity in the DDSM database for classification tasks.

4 Conclusions

We conclude that evaluation of the model described by Wu et al. on the DDSM has not been favorable. As we have previously mentioned, we believe that there exist several properties of the images in the DDSM that may be negatively affecting the performance of the model. We also observed that evaluating only two types of classes (i.e. Malignant vs. Normal) does not produce any performance improvement when compared against the evaluation with three classes (Malignant, Benign and Normal). However, as future work we will test another public dataset (for instance, InBreast) to determine if the model's performance changes or not. We will also modify the preprocessing of the images to try to obtain better results. Our purpose is to evaluate the feasibility of employing a pretrained model directly on datasets of the same nature as that in which the model was trained originally (in this case, screening mammograms),

that is, to evaluate in practice the generality of such models. Moreover, it is necessary to experiment with more extensive feature selection for the DDSM database in order to improve the classification performance of the model of Shen et al., in this way we hope to obtain better results when applying fine-tuning and reduce the heterogeneity of the training images.

Acknowledgments. This work was partially supported by the National Council of Humanities, Science and Technology (CONAHCYT) of Mexico, via Postgraduate Scholarship 813768 (O. Moreno) and Research Grant CÁTEDRAS-2598 (A. Rojas).

References

1. Chen, Y., Zhang, Q., Wu, Y., Liu, B., Wang, M., Lin, Y.: Fine-tuning resnet for breast cancer classification from mammography. In: Proceedings of the 2nd International Conference on Healthcare Science and Engineering 2nd. pp. 83–96. Springer (2019)
2. El Hachem, Z., Zoghbi, M., Hallit, S.: Psychosocial consequences of false-positive results in screening mammography. Journal of Family Medicine and Primary Care, vol. 8, no. 2, pp. 419 (2019)
3. Frazer, H. M., Qin, A. K., Pan, H., Brotchie, P.: Evaluation of deep learning-based artificial intelligence techniques for breast cancer detection on mammograms: Results from a retrospective study using a breastscreen victoria dataset. Journal of medical imaging and radiation oncology, vol. 65, no. 5, pp. 529–537 (2021)
4. Giger, M. L., Chan, H.-P., Boone, J.: Anniversary paper: history and status of cad and quantitative image analysis: the role of medical physics and aapm. Medical physics, vol. 35, no. 12, pp. 5799–5820 (2008)
5. Giger, M. L., Karssemeijer, N., Schnabel, J. A.: Breast image analysis for risk assessment, detection, diagnosis, and treatment of cancer. Annual review of biomedical engineering, vol. 15, pp. 327–357 (2013)
6. Harvey, H., Karpati, E., Khara, G., Korkinof, D., Ng, A., Austin, C., Rijken, T., Kecskemethy, P.: The role of deep learning in breast screening. Current Breast Cancer Reports, vol. 11, pp. 17–22 (2019)
7. Honig, E. L., Mullen, L. A., Amir, T., Alvin, M. D., Jones, M. K., Ambinder, E. B., Falomo, E. T., Harvey, S. C.: Factors impacting false positive recall in screening mammography. Academic radiology, vol. 26, no. 11, pp. 1505–1512 (2019)
8. Huynh, B. Q., Li, H., Giger, M. L.: Digital mammographic tumor classification using transfer learning from deep convolutional neural networks. Journal of Medical Imaging, vol. 3, no. 3, pp. 034501–034501 (2016)
9. Lee, R. S., Gimenez, F., Hoogi, A., Miyake, K. K., Gorovoy, M., Rubin, D. L.: A curated mammography data set for use in computer-aided detection and diagnosis research. Scientific data, vol. 4, no. 1, pp. 1–9 (2017)
10. Mayo, R. C., Kent, D., Sen, L. C., Kapoor, M., Leung, J. W., Watanabe, A. T.: Reduction of false-positive markings on mammograms: a retrospective comparison study using an artificial intelligence-based cad. Journal of digital imaging, vol. 32, pp. 618–624 (2019)
11. Rojas-Domínguez, A., Puga, H., Rodríguez, M. O., Guerrero-Gasca, I.: Cad of breast cancer: A decade-long review of techniques for mammography analysis. Advances in Artificial Intelligence, vol. 115 (2020)

12. Shen, L., Margolies, L. R., Rothstein, J. H., Fluder, E., McBride, R., Sieh, W.: Deep learning to improve breast cancer detection on screening mammography. Scientific reports, vol. 9, no. 1, pp. 12495 (2019)
13. Wu, N., Phang, J., Park, J., Shen, Y., Huang, Z., Zorin, M., Jastrzebski, S., Févry, T., Katsnelson, J., Kim, E., et al.: Deep neural networks improve radiologists' performance in breast cancer screening. IEEE transactions on medical imaging, vol. 39, no. 4, pp. 1184–1194 (2019)
14. Yu, X., Kang, C., Guttery, D. S., Kadry, S., Chen, Y., Zhang, Y.-D.: Resnet-scda-50 for breast abnormality classification. IEEE/ACM transactions on computational biology and bioinformatics, vol. 18, no. 1, pp. 94–102 (2020)

ISSN 1870-4069

A General Overview of Language Pronunciation Analysis Based on Machine Learning

Eric Ramos-Aguilar[1], J. Arturo Olvera-López[1], Ivan Olmos-Pineda[1], Manuel Martín-Ortiz[2]

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

[2] Laboratorio Nacional de Supercómputo del Sureste de México, Mexico

eric.ramosag@alumno.buap.mx, {jose.olvera, ivan.olmos, manuel.martin}@correo.buap.mx

Abstract. Currently, pronunciation analysis is an area related to Natural Language Processing (NLP) which is based on machine learning methods in languages considered universal such as French, English, Mandarin, and Spanish; for low-resource languages such as indigenous languages, some machine learning techniques used to evaluate pronunciation are transfer learning, deep learning, or classic machine learning. Some methods have been applied for pronunciation evaluation, obtaining different levels of performance. This paper provides a review of different approaches, describing phases, languages, metrics, and other important features for the Indigenous Languages from Mexico.

Keywords: Low resource languages, audio analysis, machine learning.

1 Introduction

The goal of Natural Language Processing (NLP) is for computers to understand, interpret, and manipulate human language [26]; an important key is the analysis of pronunciation, which is the correct way in which a word or a language is spoken. This topic is of particular interest to researchers since pronunciation involves articulatory and auditory phonetics, which are sounds of a language that describe its physical aspects and the auditory part analyzes the qualities of sound and describes how it is perceived by the listener [30].

Pronunciation is linked to two skills: speaking and listening in a teaching-learning process; speaking considers the practical and phonological features of a target language (TL), on the other hand, listening refers to the interpretation of the TLs phonological features, which are segments (phonemes) and suprasegments (stress, rhythm and intonation) [30]. The latter are a great challenge for researchers when evaluating isolated words, creating a dilemma

between evaluating pronunciation or global intelligibility, where they diagnose pronunciation and provide feedback on the analyzed words [32].

Computer-Assisted Pronunciation Training (CAPT) applications use different evaluation metrics that help to analyze pronunciation with speaker level scoring methods calculated at the through expressions, words, or local phonemes. Due machine learning pronunciation analysis models and methods require different amounts of data, the analysis of different languages can be categorized into two language groups:

- Universal languages: These are languages that currently have a largest number of speakers worldwide, such as French, English, Mandarin, and Spanish, considering enough instances to be evaluated via machine learning [32].
- Low-resource languages: These are languages with less data, a unique writing system, limited web presence, little understanding of its linguistics, and minimal transcribed speech data and translation dictionaries [4].

In this work, a general overview of the methods used for the analyzing the pronunciation of different languages is presented, describing some corpus, features and classifiers.

2 Pronunciation Analysis

Data gathered on world languages is often unbalanced, considering its number of speakers the data collected on it, so the study and analysis of the NLP has been developed considering two language groups languages. The following section presents a review of the methods and elements used for universal and low-resource languages analysis.

2.1 Universal Languages

Currently, universal languages pronunciation analysis considers data with a high number of instances, in the literature, corpora of this kind contain more than 8,000 instances, with lexical and phonetic descriptors, audio recordings from people of different ages and genders, with acoustic and phonetic knowledge. These data are preprocessed and structured for use in automatic learning. Different corpora have been found in the literature for evaluation tasks as a reference for pronunciation, such as:

- TIMIT which is a corpus that brings together a series of broadband recordings of the English language of 630 speakers from the eight main dialects of American English and which is designed to be implemented in Atomatic Speech Recognition (ASR).
- VoxForge is a corpus composed of 6 languages (English, Spanish, French, German, Russian and Italian) with training, validation, and testing divisions and constant data updating.

Table 1. Some datasets used in related works for audio language analysis.

Paper	Language	Corpus name
Zou, et. al 2018	Mandarin	DidiCallcenter, DidiReading
Duan, et. al 2019	English	Wall Street Journal, LibriSpeech
Feng, et. al 2019	Indo-European	VoxForge, Lwazi
Wang, et. al 2019	Cantonese	CUChild127
Chakroun, et. al 2020	English	TIMIT
Arias-Vergara, et al 2021	German	Verbmobil
Mao, et al 2022	English	Speechocean762, LibriSpeech
Sancinetti, et. al 2022	English	EpaDB
Lin, et. al 2022	English	TIMIT, L2-ARCTIC

Other corpora used in pronunciation analysis are presented in Table 1. These datasets were created for promoting acoustic-phonetic knowledge and automatic speech recognition systems, using phonetic and lexical transcriptions of people of different ages and gender according to their language.

The aforementioned corpora are used to extract relevant features in NLP, in pronunciation, most of the reviewed analyzes use the Mel Frequency Cepstral Coefficients (MFCC), which are a scale that depends on the auditory scale and the coefficients depend on of perception, to be able to duplicate human ears [24] where spectrograms with speech characteristics are obtained, through the Fourier transform, energy power and filter bank that are processed by the discrete cosine transform obtaining cepstral coefficients.

The coefficients can consider different cepstral aspects, as in [7] where 13 MFCCs are extracted with first and second order derivatives representing speech velocity and acceleration, respectively, taking into account 39 cepstral coefficients and integrating Best Tree Encoding (BTE) which is a Wavelet Packet Decomposition (WPD) for ASR and a Image Normalize Encoder (INE).

On the other hand, 14 MFCC coefficients are used in [19] together with their first and second delta, energy (representation of amplitude variations), RMS (the square of the function that defines the continuous waveform), pitch (speed at which the vocal cords vibrate, when pressurized air from the lungs passes through the vocal cords), entropy (a measure of the signal's Fourier power spectrum concentration), spectral features (formants, the most widely used spectral feature, are commonly used to disambiguate vowels and consonants), zero crossing (time domain function which indicates how many times a signal has changed sign with respect to zero) and statistical features.

In [35], 40 MFCC filterbank features, 33-dimensional phonemic posterior features, and 71-dimensional composite posterior features are used for comparison with the deep learning system. The use of Log Phone Posterior (LPP) and Log Posterior Ratio (LPR) has been employed as a vector of phonetic features for the comparative evaluation of phonemes with a high rate of posterior probability per phoneme, proving it to be on par with other methods such as MFCC and with the Word Error Rate (WER) [21]. This feature

vector is used in [8] to train a transformer based on Goodness of Pronunciation features (GOPT) with multitask learning.

The methods used to classify the obtained pronunciation features were initially developed with CAPT based on Hidden Markov Models (HMM) which provide log-likelihood scores, log-posterior with high correlation of human scores and qualifying the pronunciation of a given phoneme and the segment duration score [14]. Some authors implement HMM to detect mispronunciation with the help of ASR based methods, as in [38] in which a model triphone (sequence of three phonemes) was ised to train a Gaussian Mixture Model (GMM) with HMM from left to right with three states.

Some methods for performing pronunciation analysis and feature classification are based on neural networks, one being Deep Neural Networks (DNN), a feedforward artificial neural network with more than one layer of hidden data units between its inputs and outputs. Each hidden unit generally uses the logistic function to map its total input from the lower layer to the scalar state, which it sends to the upper layer [10].

DNN is based on acoustic models, so the network is trained to identify acoustic senones linked to the triphone state. It uses a mixed selection method designed for acoustic modeling based on the GMM, selecting a subset of the senone in the DNN output layer to calculate the posterior probabilities. The senone selection strategy is obtained by grouping the acoustic inputs according to their linear outputs in the hidden upper layer [17].

In [5], DNNs with acoustic-phonetic models are used to detect non-native speech recognition and pronunciation errors and to diagnose articulation-level pronunciation errors of based on the GOP score by calculating the log-posterior ratio between the target canonical phoneme and its most competing phoneme, which has the highest posterior probability for Japanese English learners. Assessment metrics such as False Alarm Rate (FAR), Receiver Operating Characteristic (ROC curve), and Diagnostic Error Rate (DER) are used in this paper, whit 7.82% being the best line error reduction result.

A study carried out in [13] uses a DNN-HMM to improve ASR performance with the Punjabi language, obtaining a Word Error Rate (WER) of 5.32% as the best result. SoftMax uses a DNN to for automatic pronunciation error detection based on GOP; this network is used to carry out transfer learning in order to detect pronunciation errors in Mandarin language using acoustic models trained by different criteria, such as accuracy, F1-score, and Recall[12].

Another pronunciation evaluation method is Convolutional Neural Networks (CNN), which are deep learning architectures inspired by the natural visual perception mechanism of living creatures, where the convolutional layer aims to learn feature representations from inputs; for this type of analysis, this neural network considers digital audio data by feeding the convolution layers with time-frequency representations (spectrograms) of the signals that provide information about how the energy distributed in the frequency domain changes over time [9].

In [3], a CNN is trained to classify speech segments of people with cochlear implants (CI) and healthy control (HC) speakers in the German language, using a cross-validation of k=10 to train and evaluate the models; the performance is measured by means of Presicion (PR), Recovery and F1 score, with the best results obteined from spectrograms of three channels extracted from the compensated transitions, F1 = 0.84. Another study uses a CNN to detect the mispronunciation of phonemes, using forced aligners orthographic transcriptions aligned with the audio recordings automatically generating segmentation at the phoneme level; CNN uses an input channel, output channel, and kernel size of 64, 64, and 9; this analysis uses PR, Recall, and F1-score metrics for evaluating, with F-score showing the best performance with 63.04% [16].

The work proposed in [39] performs an alphabetical classification of the Arabic language, increasing the training data for a better performance of the neural network, adding 20 samples for each alphabet. CNN, Recurrent Neural Network (RNN), and Bidirectional Long Short-Term Memory (BLSTM) are also used; these learning models are accurate to within 91% to 98.5% using Support Vector Machine (SVM) classification.

RNN contains at least one feedback connection, so the activations can flow in a loop, this allows the networks to perform temporal processing and learn sequences [20]; the RNNs are included in CNN to perform a detection and diagnosis of a mispronunciation of the English language; two blocks of CNN and four of RNN with bidirectional LSTM are used, augmentintg the data with acoustic, phonetic, and linguistic embedding (APL) for increased performance; exceeding the baseline by 9.93%, 10.13%, and 6.17% in detection accuracy, DER, and F-measure, respectively [37].

2.2 Low-resource Languages

This section describes the methods used in low-resource languages pronunciation analysis. As mentioned above, this type of language considers a small amount of data, such as text and audio, do not contain enough computational pre-processing instances to be evaluated on their own in a machine learning environment, in some cases as many as a thousand per language.

The use of evaluated corpora in universal languages has been identified in the reviewed literature, with which the neural networks are trained and the subsequent evaluation is carried out with low-resource languages. For example, the LibriSpeech corpus is used in [27] to create ASR in the Tamil language from India and part of Siri Lanka. Another case is the analysis to identify African languages such as Afrikaans, isiNdebele, isiXhosa, isiZulu, Sepedi, Sesotho, Setswana, Siswati, Tshivenda, and Xitsonga, where the VoxForge corpus is used to train a neural network [6].

Another corpus used is PHOIBLE, which is a phoneme database for more than 2,000 languages and dialects, used to develop a tool called Allosaurus that recognizes the phonemes of some universal languages and has been put into practice for low-resource languages [15]. Another language with corpus

preprocessing is Uzbek which uses Common Voice Corpus 8.0 where recordings of sentences on Uzbek dialects are stored [22].

Due to the lack of data on low-resource languages, some authors (in addition to using corpora previously designed for computer environments) have opted to create databases in order to analyze these languages, as in [29], which reports a Tamil and Malay corpus containing 7,582 utterances. Another corpus built to implement a Tibetan speech recognizer contains 28,000 utterances by native speakers [36]. In [23] to recognize the Pashto dialect, a corpus of 900 audio expressions by 45 people was created.

Features similar to those extracted in universal languages can be obtained from audio recordings, such as MFCC [27, 22, 23, 15]. Other features are extracted to analyze low-resource languages with the aim of finding robust features such as the frame-level perceptual linear predictive (PLP) coefficients [6], which are representations conforming to a smoothed short-term spectrum that has been equalized and compressed in a manner similar to that of human hearing [11]. Other features are extracted by [36], where through acoustic features of the speech signal (frequency, amplitude and volume) are obtained through spectrograms.

The features are categorized in different ways when analyzing low-resource languages. One method is transfer learning, applied in [6], where a CNN is trained from a corpus containing 7 Indo-European languages; the weights of the last fully connected layers of the neural network architectures are adjusted using approximately 22 hours of the Lwazi corpus containing 11 African languages, yielding Equal Error Rate (EER) evaluations of 20%; this is a 10% improvement in the identification of African languages (comparison to other authors).

Another method, used for the Uzbek language and its dialects, is End-To-End (E2E) Deep Neural Network-Hidden Markov Model implemented in ASR; it computes the probability from the entire alphabet using the coefficients of MFCC, a deep voice method with CNN and RNN, removing pre-segmented data, and training E2E with Connectionist Temporal Classification (CTC) considering an input voice stream to an output token stream using a single network; this system is capable of training pronunciation, acoustics, and language simultaneously; this system is evaluated with WER, obtaining 16.4% and 17.6% on the Uzbek_Test and Hidden_Test sets, respectively [22].

In [27], a pronunciation analysis is carried out to recognize the Tamil language, where the dialects of the stop words are extracted through the Mel Scales, audio and voice signal features. The features are converted into vectors with LSTM/RNN model and clustered with CNN model to learn interestingness and detect outliers as noise from the original features. Patterns are stored to compare Character Error Rate (CER) and WER through the CNN model.

Another speech approach, applied to the Tibetan language, is that proposed in [36], where one encoder is based on deep CNN and another on a hybrid network of deep CNN and LSTM, it includes a 10-layer CNN architecture. This method uses acoustic features from spectrograms as inputs, computing a WER rating of 36.85%.

Typical classifiers for language recognition are HMM which model a sequence of events or hidden states, Support Vector Machines (SVM) that try to maximize the functional margin between the closest training data from a different class and build an optimal hyperplane, K-Nearest Neighbor (K-NN) which classifies new cases based on a measure of similarity. These classifiers were used for the Pashto dialect which for training and testing purposes are divided into 77% (35 speakers) and 23% (10 speakers), using these classifiers an accuracy of 88%, 84%, and 76% is obtained respectively [23].

Previously developed methods consider more than one language for evaluation and training, generating multilingual recognition methods based on phonetic annotation, such as Allosaurus (Allophone system of automatic recognition for universal speech). This method first calculates the distribution of phonemes using a standard ASR encoder; then, the allophone layer maps the phoneme distribution for each language. This model is trained from start to finish using standard phonemic transcriptions and a list of allophones created by phonetics. The allophone layer is first initialized with the allophone list and then further optimized during the training process.

Allosaurus selected 11 languages (English, Switchboard, Japanese, Mandarin, Tagalog, Turkish, Vietnamese, German, Spanish, Amharic, Italian, Russian) for training with more than 8,000 utterances in each corpus, with 5% of them for testing, the rest were used for validation and training. On the other hand, 2 African languages (Inukitut and Tusom) were used with one thousand randomly selected utterances each; a bidirectional LSTM encoder is used for these, and the phonemes for the training languages are assigned using the grapheme-phoneme tool, creating allophone assignments by specialists in phonetics.

When carrying out an evaluation with the Phoneme Error Rate (PER), accuracy is achieved for the Inuktitut language 84.1% and 77.3% for Tusom. While combined with other corpus (PHOIBLE), error rates are further improved to 73.1% and 64.2% respectively [15].

Indigenous Languages in Mexico. This subsection describes the methods used for the low-resource languages from Mexico, whose analysis has not used machine learning techniques as for African and Asian languages, so the process is still basic compared to the previous ones.

There are sixty-eight indigenous languages in Mexico, distributed throughout national territory, located mainly country's southern and central regions, with linguistic variants producing unique languages in every region. These languages are classified into the following eleven linguistic families: Algica, Yuto-nahua, Cochimi-yumana, Seri, Oto-mangue, Maya, Totonaco-tepehua, Tarasca, Mixe-zoque, Chontal, and Huave [1].

The analysis of indigenous languages from Mexico and other low-resource languages, with limited audio and text data requires different considerations, one of these being transfer learning, which can evaluate them with the support of other data. Another process is to use phonetic and acoustic embeddings from other languages or to perform neural network training using similar phonemes.

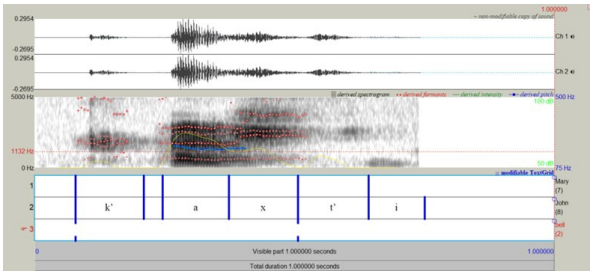


Fig. 1. Representation of analysis with the Praat software of the word yellow said in Otomi.

To carry out pronunciation evaluation processes on indigenous languages of Mexico, various researchers, mostly linguists, have made recordings of words or sentences of people from some communities. These recordings generate corpus of digital audio, with the number of people ranging from five to approximately thirty speakers, as in [25], which considers a corpus of six native Mixtec speakers; on the other hand, [34] uses a database of 8 people with audio recordings of interrogative and declarative sentences in the Otomi language from the Tultepec region in Queretaro; in [28] a corpus of 30 speakers of the Nahuatl language from the state of Puebla is used to carry out an analysis of the production of sonorous sounds; another study carried out by [33] takes into account audio recordings of words uttered by a single person.

Although different language corpora have been mentioned, the methods used for their analysis are similar. A common approach for processing provides as input the audio recordings (without pre-processing or feature extraction) to a software tool such as Praat which is opensource for recording and analyzing words or sentences, computing spectrogram, tone, intensity, volume, and cochleagram(Figure 1).

Another free analytical software is ELAN which represents digital audio in the time from audio or video recordings, it is used to segmentation and made textual annotations to identify phonemes or tones in a subjectively (figure 2); conclusions or evaluations are written qualitatively, commenting on references of tones or between word phoneme similarities of the same language or Spanish.

As it can be seen from figures 1 and 2, the annotations are made under the digital audio time representation and can represent phonemes, syllables, or phrases; unlike Praat, ELAN can process video to obtain audio and analyze complete input sentences; Praat, on the other hand, provides a spectrogram for representing formants as well as for inspecting tones, however, these annotations or analysis are carried out in a unitary and subjective manner by the analyst, how has their own analysis criteria.

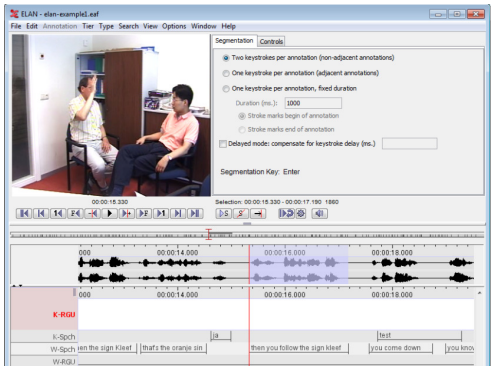


Fig. 2. Analysis representation with ELAN software (Image obtained from [31]).

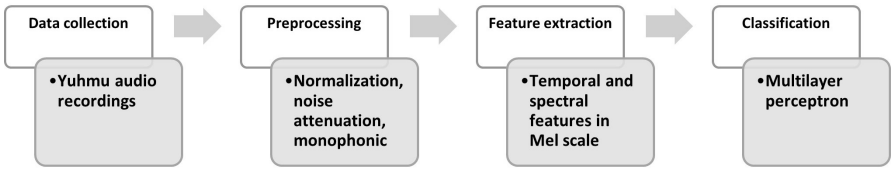


Fig. 3. Phases of the methodology.

3 Proposed Method

The following methodology is proposed for the evaluation of the pronunciation of an indigenous language of Mexico based on the analyzed literature, which considers four important phases (figure 3).

Digital audio recordings of native and non-native people of the Yuhmu indigenous language (a variant of the Otomi language of the State of Tlaxcala) are used in the data collection where Yumhu speakers of 330 words with 3 repetitions each of good pronunciation from [2] is considered, complementing with a corpus of the same words with poor pronunciation of creation own, carrying out a subsampling guided by clustering, removing words with relevant phonetic characteristics (number of phonemes and repetition of phonemes per word), obtaining a final sample of 622 words per category, which consider all the phonemes used in the Yuhmu language.

Within the preprocessing stage, digital audio enhancement is performed, with noise attenuation, amplifying the audio signal and using mono channel for analysis.

During the feature extraction, the use of algorithms based on spectrograms is proposed with experimentation of different parameters for the STFT (Short-time Fourier transform), from which the following are obtained: energy in bands and

Table 2. Audio classification results considering Time, Spectral and Time-Spectral features.

Features	Accuracy (%)
Time	90-91.8
Spectral	91-94.7
Time-Spectral	90-94% and 96-97.7%

shape features; time characteristics (RMS and ZCR), statistics (average and standard deviation) and Pitch.

Finally, a classification of good and bad pronunciation is carried out using a multilayer perceptron considering a "Grid Search" for the search for ideal hyperparameters (one hidden layer, 11 neurons, ReLU activation function, Momentum at 0.4, learning rate at 0.11, and a cross validation with k=5) that would yield the best classification results.

From the previous methodology, 21 characteristic sets are obtained for 4 types of window and overlap with 12 characteristics per set.

The windows used to carry out the experimentation were Hanning, Hamming, Gaussian, and Blackman-Harris, considered for the secondary lobes they have and that can help in the loss of information during the windowing. The window size range used is from 15 to 45 ms with steps of 5 ms, with an overlap of 25, 50, and 75%.

Three analyzes were carried out with different groups of characteristics, the first one considering temporal characteristics, the next one with spectral characteristics and finally using all the characteristics (table 2), to classify good and bad pronunciation.

The results shown in the table 2 present an evaluation of the accuracy of the classification of good and bad pronunciation where the temporal and spectral characteristics show a result of 90-94.7%, while all the characteristics consider two ranges, one similar to the previous ones and another of 96-97.7%, the first is the result of the 15 and 45 ms windows, while when performing a 20-40 ms window the best results are obtained, taking into account that stated in the literature [18]: the best window is within the latter mentioned.

4 Discussion

There are different machine learning processes for evaluating pronunciation due to each language having unique phonetic features, as well as variants in phoneme pronunciation (phono). Figure 4 depicts the methods explained in section 2, showing the researches and the methods used, considering the following:

- Method: Name of the pronunciation evaluation method used.
- Reference: Author(s) of the method.
- Assessment: Metrics used to evaluate the method (WER, PER, PR, EER, Acurracy, DER, CER, among other).

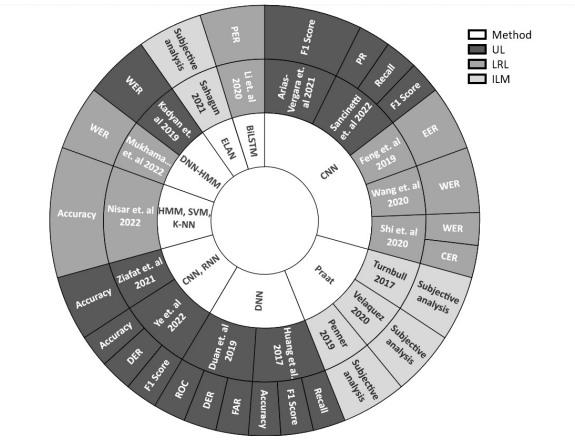


Fig. 4. Methods described in section 2.

– Group: The type of language being analyzed, Universal Languages (UL), Low-resource Languages (LRL), or Indigenous Languages in Mexico (ILM).

From the figure 4, it is evident that the use of neural networks to analyze the UL and LRL is common across all methods, using recurrent, convolutional, and deep neural networks, although in some cases two types of networks are used. The use of machine learning is common; however, as noted above, training is different for of these networks since low-resource languages do not have corpora with more than 8,000 instances as universal languages do. These are aided by a UL corpus for training and a model neural network proposal for classification.

The literature presents some works that apply machine learning for the analysis of indigenous languages for text-to-speech translation tasks, vowel/consonant recognition, however, there are no works that aim to evaluate the pronunciation of the different indigenous languages that exist in Mexico, which presents an area of opportunity in the development of approaches for low-resource languages of Mexico, specifically in pronunciation tasks, because this is still manually generated using software such as Praat or ELAN to perform audio segmentation or interpretation by an analyst, in this case, they are regularly linguists who try to study languages at a semantic level.

From the area referring to assessment, the level of error or accuracy that occurs when making different pronunciations is analyzed, in some cases as in [13, 22, 29, 36] are focused on carrying out an evaluation at the word level considering WER as its measurement, having a better result in the LRL evaluation of 16.4 %. Another metric considered to evaluate is the accuracy that becomes relevant within the methods analyzed, having results even of 98.5% when performing an analysis of universal languages [39].

It is notorious that when observing the type of measurement that is carried out in low-resource languages, a machine learning method to evaluate

pronunciation is not yet considered, this is because currently these processes for this type of language are still under development. Consequently, they are not yet processed like universal languages, due to the number of instances they have, which in some cases reaches a maximum of a thousand, for which reason they have relied on robust corpora for their analysis.

There is a difference between universal languages and low-resource languages, due to the fact that when considering a corpus of a greater number of instances, they are capable of providing a greater number of references for evaluation, which is why the difference in accuracy where the range for world languages is 80-99%, for a machine learning analysis; on the other hand, low-resource languages have results of 60-70% accuracy in their evaluations, having a difference of up to 40 percentage points with respect to universal languages; while the indigenous languages of Mexico have not currently carried out a precision analysis in their processes, so all the results have been concluded with qualitative descriptions.

5 Conclusions

Pronunciation analysis is still a problem of interest for researchers, due to the areas of opportunity that still exist to assess the intelligibility or pronunciation of words or sentences. This machine learning task currently considers two languages groups universal and of low resource, which have been described in this paper.

Different authors have proposed methods with CNN, DNN, RNN or transfer learning, using GOP, EER, PER or WER to define if a word or set of these are well pronounced as evaluation method. On the other hand, the feature extraction methods are similar, considering MFCC, spectrograms, frequencies, energy features, time, among others, of audio recordings. The analysis is carried out with segmentation or supra-segmentation of words or sentences that depend on the type of evaluation, considering phonemes, articulation zone or tone of the same.

It is worth noting that low-resource languages analysis does not consider a method of analysis through machine learning similar to that of universal languages where they can be evaluated autonomously, while the indigenous languages of Mexico are considered a challenge for researchers when evaluating pronunciation.

When carrying out the experimentation with the Yuhmu language, it is concluded that the characteristics used for other types of languages are considered useful to analyze this type of languages. Currently, the analysis of the pronunciation of the indigenous languages of Mexico would help the conservation, preservation and dignification of this kind of languages.

Acknowledgments. This work was supported by the National Council of Humanities Science and Technology (CONAHCYT) under the scholarship number 814401 and the 189 VIEP-BUAP project.

References

1. Catálogo de las lenguas indígenas nacionales (2022), https://www.inali.gob.mx/clin-inali, last accessed May 06, 2023
2. Alarcon Montero, R.: Manual para la escritura de los sonidos del yuhmu. INAH (2023)
3. Arias-Vergara, T., Klumpp, P., Vasquez-Correa, J. C., Nöth, E., Orozco-Arroyave, J. R., Schuster, M.: Multi-channel spectrograms for speech processing applications using deep learning methods. Pattern Analysis and Applications, vol. 24, pp. 423–431 (2021)
4. Besacier, L., Barnard, E., Karpov, A., Schultz, T.: Automatic speech recognition for under-resourced languages: A survey. Speech communication, vol. 56, pp. 85–100 (2014)
5. Duan, R., Kawahara, T., Dantsuji, M., Nanjo, H.: Cross-lingual transfer learning of non-native acoustic modeling for pronunciation error detection and diagnosis. IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 28, pp. 391–401 (2019)
6. Feng, K., Chaspari, T.: Low-resource language identification from speech using transfer learning. In: 2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP). pp. 1–6. IEEE (2019)
7. Gbaily, M. O.: Automatic database segmentation using hybrid spectrum-visual approach. The Egyptian Journal of Language Engineering, vol. 8, no. 2, pp. 28–43 (2021)
8. Gong, Y., Chen, Z., Chu, I.-H., Chang, P., Glass, J.: Transformer-based multi-aspect multi-granularity non-native english speaker pronunciation assessment. In: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 7262–7266. IEEE (2022)
9. Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J., et al.: Recent advances in convolutional neural networks. Pattern recognition, vol. 77, pp. 354–377 (2018)
10. Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., et al.: Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal processing magazine, vol. 29, no. 6, pp. 82–97 (2012)
11. Hönig, F., Stemmer, G., Hacker, C., Brugnara, F.: Revising perceptual linear prediction (plp). In: Ninth European Conference on Speech Communication and Technology (2005)
12. Huang, H., Xu, H., Hu, Y., Zhou, G.: A transfer learning approach to goodness of pronunciation based automatic mispronunciation detection. The Journal of the Acoustical Society of America, vol. 142, no. 5, pp. 3165–3177 (2017)
13. Kadyan, V., Mantri, A., Aggarwal, R., Singh, A.: A comparative study of deep neural network based punjabi-asr system. International Journal of Speech Technology, vol. 22, pp. 111–119 (2019)
14. Kim, Y., Franco, H., Neumeyer, L.: Automatic pronunciation scoring of specific phone segments for language instruction. In: Fifth European Conference on Speech Communication and Technology (1997)
15. Li, X., Dalmia, S., Li, J., Lee, M., Littell, P., Yao, J., Anastasopoulos, A., Mortensen, D. R., Neubig, G., Black, A. W., et al.: Universal phone recognition with a multilingual allophone system. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 8249–8253. IEEE (2020)

16. Lin, B., Wang, L.: Phoneme mispronunciation detection by jointly learning to align. In: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 6822–6826. IEEE (2022)
17. Liu, J.-H., Ling, Z.-H., Wei, S., Hu, G.-P., Dai, L.-R.: Cluster-based senone selection for the efficient calculation of deep neural network acoustic models. In: 2016 10th International Symposium on Chinese Spoken Language Processing (ISCSLP). pp. 1–5. IEEE (2016)
18. Liu, L., He, J., Palm, G.: Effects of phase on the perception of intervocalic stop consonants. speech communication, vol. 22, no. 4, pp. 403–417 (1997)
19. Maqsood, M., Habib, H. A., Nawaz, T.: An efficientmis pronunciation detection system using discriminative acoustic phonetic features for arabic consonants. Int. Arab J. Inf. Technol., vol. 16, no. 2, pp. 242–250 (2019)
20. Mikolov, T., Karafiát, M., Burget, L., Cernockỳ, J., Khudanpur, S.: Recurrent neural network based language model. In: Interspeech. vol. 2, pp. 1045–1048. Makuhari (2010)
21. Minh, N. Q., Hung, P. D.: The system for detecting vietnamese mispronunciation. In: Future Data and Security Engineering. Big Data, Security and Privacy, Smart City and Industry 4.0 Applications: 8th International Conference, FDSE 2021, Virtual Event, November 24–26, 2021, Proceedings 8. pp. 452–459. Springer (2021)
22. Mukhamadiyev, A., Khujayarov, I., Djuraev, O., Cho, J.: Automatic speech recognition method based on deep learning approaches for uzbek language. Sensors, vol. 22, no. 10, pp. 3683 (2022)
23. Nisar, S., Tariq, M.: Dialect recognition for low resource language using an adaptive filter bank. International Journal of Wavelets, Multiresolution and Information Processing, vol. 16, no. 04, pp. 1850031 (2018)
24. Pangaonkar, S., Panat, A.: A review of various techniques related to feature extraction and classification for speech signal analysis. In: ICDSMLA 2019: Proceedings of the 1st International Conference on Data Science, Machine Learning and Applications. pp. 534–549. Springer (2020)
25. Penner, K.: Prosodic structure in ixtayutla mixtec: Evidence for the foot, (2019)
26. Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N., Huang, X.: Pre-trained models for natural language processing: A survey. Science China Technological Sciences, vol. 63, no. 10, pp. 1872–1897 (2020)
27. Rajendran, S., Mathivanan, S. K., Jayagopal, P., Venkatasen, M., Pandi, T., Sorakaya Somanathan, M., Thangaval, M., Mani, P.: Language dialect based speech emotion recognition through deep learning techniques. International Journal of Speech Technology, vol. 24, pp. 625–635 (2021)
28. Sahagun, A. S.: Spanish VOT Production by L1 Nahuat Speakers. Ph.D. thesis, University of Saskatchewan (2021)
29. Shi, K., Tan, K. M., Duan, R., Salleh, S. U. M., Suhaimi, N. F. A., Vellu, R., Thai, N. T. H. H., Chen, N. F.: Computer-assisted language learning system: Automatic speech evaluation for children learning malay and tamil. In: INTERSPEECH. pp. 1019–1020 (2020)
30. Szyszka, M.: Pronunciation learning strategies and language anxiety. Switzerland: Springer, vol. 10, pp. 978–3 (2017)
31. Tacchetti, M.: User's guide for elan linguistic annotator. The Language Archive, MPI for Psycholinguistics, Nijmegen, The Netherlands.[Google Scholar], (2017)
32. Tejedor-García, C., Escudero-Mancebo, D., Cámara-Arenas, E., González-Ferreras, C., Cardeñoso-Payo, V.: Assessing pronunciation improvement in students of english using a controlled computer-assisted pronunciation tool. IEEE Transactions on Learning Technologies, vol. 13, no. 2, pp. 269–282 (2020)

33. Turnbull, R.: The phonetics and phonology of lexical prosody in san jerónimo acazulco otomi. Journal of the International Phonetic Association, vol. 47, no. 3, pp. 251–282 (2017)
34. Velásquez Upegui, E. P.: Entonación del español en contacto con el otomí de san ildefonso tultepec: enunciados declarativos e interrogativos absolutos. Anuario de letras. Lingüística y filología, vol. 8, no. 2, pp. 143–168 (2020)
35. Wang, J., Qin, Y., Peng, Z., Lee, T.: Child speech disorder detection with siamese recurrent network using speech attribute features. In: INTERSPEECH. vol. 2, pp. 3885–3889 (2019)
36. Wang, W., Yang, X., Yang, H.: End-to-end low-resource speech recognition with a deep cnn-lstm encoder. In: 2020 IEEE 3rd International Conference on Information Communication and Signal Processing (ICICSP). pp. 158–162. IEEE (2020)
37. Ye, W., Mao, S., Soong, F., Wu, W., Xia, Y., Tien, J., Wu, Z.: An approach to mispronunciation detection and diagnosis with acoustic, phonetic and linguistic (apl) embeddings. In: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 6827–6831. IEEE (2022)
38. Zhang, Z., Wang, Y., Yang, J.: Masked acoustic unit for mispronunciation detection and correction. In: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 6832–6836. IEEE (2022)
39. Ziafat, N., Ahmad, H. F., Fatima, I., Zia, M., Alhumam, A., Rajpoot, K.: Correct pronunciation detection of the arabic alphabet using deep learning. Applied Sciences, vol. 11, no. 6, pp. 2508 (2021)

ISSN 1870-4069

A Process for Topic Modeling via Word Embeddings

Diego Saldaña-Ulloa

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

diegos.ulloa13@gmail.com

Abstract. This work combines algorithms based on word embeddings, dimensionality reduction, and clustering. The objective is to obtain topics from a set of unclassified texts. The algorithm to obtain the word embeddings is the BERT model, a neural network architecture widely used in NLP tasks. Due to the high dimensionality, a dimensionality reduction technique called UMAP is used. This method manages to reduce the dimensions while preserving part of the local and global information of the original data. K-Means is used as the clustering algorithm to obtain the topics. Then, the topics are evaluated using the TF-IDF statistics, Topic Diversity, and Topic Coherence to get the meaning of the words on the clusters. The results of the process show good values, so the topic modeling of this process is a viable option for classifying or clustering texts without labels.

Keywords: Topic modeling, word embedding, dimensionality reduction, TF-IDF.

1 Introduction

Text processing in the digital age is widely linked to daily processes occurring in all ambits. These types of tools allow obtaining several features related to the semantics of a text that can later be used for a wide variety of purposes, mainly related to the correct categorization of a document. The processes involving textual categorization are one of the tools with the most applications in different areas [2], considering the growing increase in information generation. This previous statement drives the need to automatically categorize a text (or document) as a fundamental task.

The process of categorizing a document or extracting its related topics is called topic modeling. Topic modeling involves collecting a set of latent variables that help define concepts from documents [3]. This type of process can be helpful in sites with a large amount of information, for example, in databases of journals or articles, and even in content generated in social networks by users [4, 5].

The methods to extract topics from a document look for a way to access the semantic information of the text; in this way, different types of processes

that combine techniques and statistical algorithms can be designed. As a first step, using processes based on the count of words is common. However, these techniques could not consider the dependency between the document's different terms [6]. This is the reason for the necessity of techniques that consider both the words' context and the frequency of terms.

In this work, we propose a set of techniques and algorithms that consider the latent variables of a document, the context of the words (through word embeddings), and the frequency of the different terms present in a text to perform topic modeling. The organization that will be followed will be the one described below: Section 1 describes the background related to the topic modeling area as well as the related works, section 2 introduces the theory of the different methods and algorithms used, section 3 presents the description of the process for the extraction of topics from a document, in section 4 the experimental results are detailed, and finally, the conclusions are handled.

2 Related Work

Topic modeling is a set of techniques used to extract information from a document and define a set of ideas that characterize it. There are different methods to carry out this process, but many arise from the Vector Space Model (VSM) [7]. VSM is a mathematical model to represent texts that work considering the relevance (numerical weights) of different terms on a set of documents [8]. Relevance is generally assigned by a function related to the frequency of each term in the entire document. In this way, an n-dimensional vector type can be defined for each document, formed by the numerical weights of each word.

TF-IDF is one of the first methods used to consider the relevance of a term over the whole document. This method considers the frequencies of the terms (TF) over the total corpus size (IDF). The result of this procedure is precisely a matrix of weights for each of the terms in a document. The TF-IDF method is used by another technique called Latent Semantic Analysis (LSA) which is used in natural language processing tasks such as text classification. LSA factors the TF-IDF matrix using Singular Value Decomposition (SVD) and thus manages to reduce its dimensionality. In this process, part of the semantic information of the terms is captured, so LSA can be used to assign topics to a document based on the numerical weights of each term over the entire text [9].

Another method that is used in topic extraction is PLSA (Probabilistic Latent Semantic Analysis). It is based on LSA, and they differ in that PLSA considers that each term comes from a probability distribution given for each one of the topics. In this way, a set of probability distributions of a fixed set of topics characterizes each document. The model estimates the topic and word distributions that best explain a co-occurrence of terms in the corpus [3].

Latent Dirichlet Allocation (LDA) is another method used for topic modeling. It works similarly to PLSA, each document is considered a mixture of topics, and each topic comes from a probability distribution of the possible words in the

topic. The algorithm works by assigning words to topics and iteratively refining the assignment by maximizing the probability of the data. This is done through a Bayesian estimate of the probability distributions of words and topics given the observed data [9].

The methods described above are commonly used for topic modeling tasks. Several works use them as part of a text categorization process or through comparing methods [9–12]. However, LSA, PLSA, and LDA have several disadvantages, such as fixing the number of topics, lack of capturing non-linear relationships between words, and the assumption that a document can contain different topics, which may not be valid. For this reason, other proposals have been developed that consider the possible dependence on the context of words.

Deep learning is today's most widely used method for natural language processing tasks. Different models and architectures can capture the dependency between the words in a sentence. With this consideration, the task focuses on obtaining an n-dimensional representation of a word or sentence, i.e., an embedding. The use of word embeddings has become increasingly widespread nowadays. The advantage of an n-dimensional representation is that it can be used from different approaches [2].

Some works have focused on using word embeddings for topic modeling tasks [13–15]. They generally work by obtaining the embeddings for the texts of each document and then applying term frequency techniques, clustering, and combinations with LDA. In the present work, this approach is used, combining the extraction of word embeddings through a pre-trained BERT (Bidirectional Encoder Representations from Transformers) model, a UMAP dimensionality reduction technique [16], and the clustering of the elements as the final step to get a set of topics using K-means. Some methods and results of [13] inspired this work.

3 Word Embeddings, Dimensionality Reduction, and Topic Modeling

Word embedding represents a word or text as an n-dimensional vector considering the semantic and syntactic characteristics within the corpus [17]. This type of representation is commonly used in multiple NLP tasks, generally as features that feed some machine learning algorithm. The most usual way to obtain word embeddings is through different types of neural network architectures. It has been verified that better results are obtained using neural networks than techniques based on the frequency of terms [18].

3.1 Transformers and BERT Model for Word Embedding

One of the most used architectures currently in NLP tasks is the Transformers, particularly a derived model called BERT. Transformers are a type of deep neural network architecture that uses attention mechanisms (weights between input data elements) to select the most important parts of an input sequence,

catching the long-range dependencies of a sentence [19]. They work through an encoder-decoder structure that takes $X = \{x_1, x_2, ..., x_N\}$ sequences (tokenized values from the input text) and produces $X = \{z_1, z_2, ..., z_N\}$ representations. The encoder-decoder blocks use multi-head attention mechanisms to capture the dependencies between different tokens in the input sequence [20]. In general, this type of architecture uses an stack of encoder-decoder layer stacks; that is, the information is autoregressive and depends on the previous results of the computation.

BERT comprises a set of Encoder Transformer layers and an attention mechanism to capture relationships between all the words in a sentence. The difference from BERT is that this unit deals with the forwards and backward of a sentence [21]. This model is pre-trained on a large text corpus using two unsupervised learning tasks: masked language modeling and next-sentence prediction [19]. The first part consists of predicting the original masked tokens (tokens generated on the input data) based on the context of the surrounding words. This allows the model to understand relationships between words in a sentence. In the next sentence prediction, the model aims to predict if a pair of sentences follow each other. This helps BERT to understand the relationship between sentences in a document. Since pre-training aims to learn dependencies and relationships between words and sentences, the model can be fine-tuned to perform multiple tasks as word embedding [20].

3.2 UMAP for Dimensionality Reduction

The dimensionality reduction plays an essential role in data analysis and visualization tasks. Some of these techniques work through multiple transformations on the input data that are linear combinations of the initial information. Dimensionality reduction techniques generally do not consider the global characteristics of the input data, so the distance of a set of points in the original space will not necessarily be preserved in the reduced space [24].

Uniform Manifold Approximation and Projection (UMAP) is a dimension reduction technique that uses graph layout algorithms to arrange data in a low-dimensional space [16]. The main idea focuses on approximating the high-dimensional data manifold (curved surface embedded in a high-dimensional space). The embedding of the data is computed by searching for a low-dimensional projection of the data with the closest possible equivalent global shape and structure as the original dataset [24], i.e., UMAP preserves the local and global structure of the data. This method considers that there is a manifold on which the data would be uniformly distributed, and the main objective is to preserve the topological structure of this manifold [16].

3.3 K-means Clustering

The K-means algorithm is one of the best-known methods for solving clustering problems. K-means operates on a set of observations to try to group them into a certain number of sets, considering that the square of the distance between each

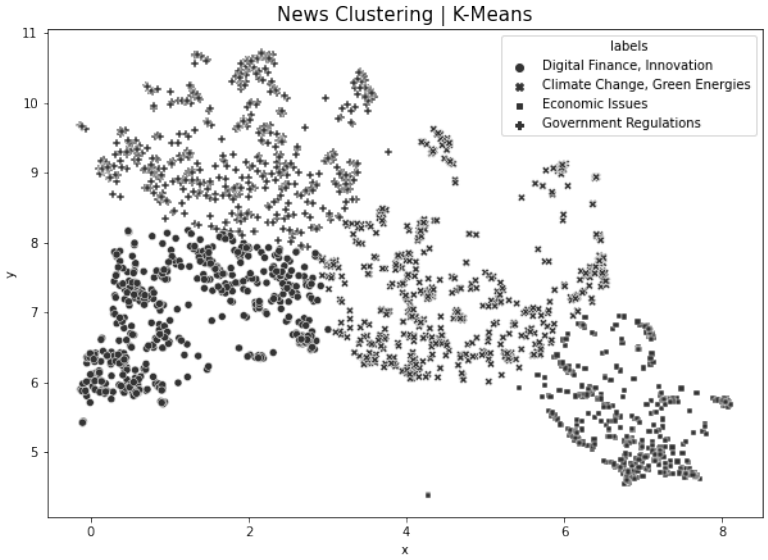


Fig. 1. Clustering of the reduced embeddings.

element and its cluster center is the minimum possible [25]. We can formally define the K-means problem as the next optimization problem.

K-means optimization problem: Given a set of elements $\mathbf{x}_n \in X$, $X \subset R^D$ and an integer $k \leq n$ such that there are $c_k \in C$ subsets. The objective of the K-means clustering is to partition n elements into k sets S to minimize the within-cluster sum of squares (WCSS), i.e.:

$$\min_{c \in C} \sum_{x \in X} \|x - c\|^2 . \tag{1}$$

During each iteration of the algorithm, the following steps are computed: first, the k cluster centroids are taken randomly from the elements of X. Next, each data element is assigned to its nearest cluster. New centroids are created by taking the average value of the elements assigned to the cluster. Finally, the difference between the new centroid and the previous one is calculated. The algorithm stops until there are no significant differences between these two values.

4 Topic Modeling with Word Embeddings

The process used for topic modeling consists of multiple of steps involving the algorithms described above. A pre-trained BERT model [26, 27] is used as a first

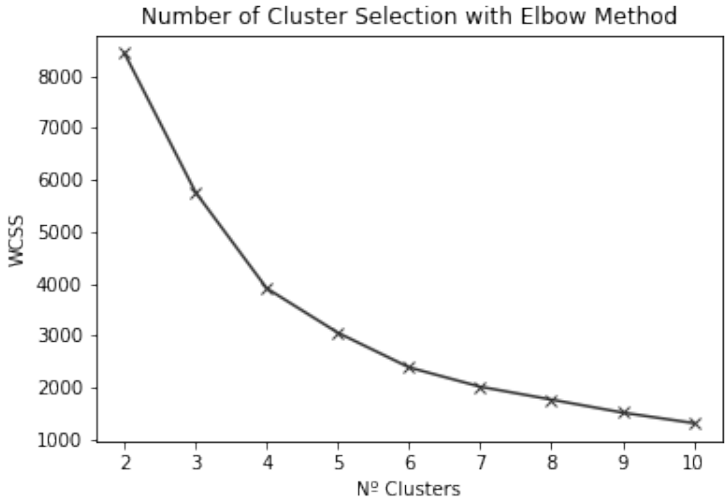


Fig. 2. Elbow method to select the number of clusters.

step to obtain the embeddings of a set of texts. Using a pre-trained model has an advantage, such as time and resource efficiency or a broad understanding of the language, because these models are trained on a large amount of text data. Since the pre-trained BERT models to generate embeddings result in high-dimensionality vectors [21], it is necessary to use dimensionality reduction techniques.

That is why as a second step, the UMAP algorithm is applied. The reduced dimensions preserve the original data's local and global structure through nonlinear transformations [16]. BERT generates embeddings considering the context of the words in the corpus; therefore, similar text strings must have similar embeddings in the final space. However, difficulties have been observed in vectors with high dimensionality when using distance metrics [21].

For this reason, using dimensionality reduction techniques such as the one described above is expected. By applying UMAP, this information manages to be preserved due to how the algorithm operates. To this point, generating embeddings with a reduced dimension for the text set to be described is possible. An additional step is needed to use this information to generate topics from a set of documents.

The third step of the process described in this work involves using clustering techniques to assign clusters to the embeddings obtained. K-means is used in this work due to its ease of use and interpretability. The clusters obtained by K-Means on each document's set of embeddings correspond to each text's topics.

5 Experimental Results

A dataset of 1212 news items in Spanish extracted from a website [28] was used for the experimental process. As part of the preprocessing, punctuation marks and special characters ($\$, \#, \$, \%, \&$, among others) were removed. Likewise, common words without significant relevance (stopwords) from Spanish were eliminated in concordance with [29]. For this process, a pre-trained BERT model was used in more than 50 languages, including Spanish. The BERT model, pre-trained in more than 50 languages, is designed to preserve the distance between words with the same meaning in different languages [26, 27]. This model is used because of the limited availability of BERT models trained exclusively in Spanish. Because of the characteristics of the pre-trained model (with English and Spanish languages included), a random sample composed of half of the news was taken, then its translation into English was obtained. This translated sample was incorporated into the original Spanish dataset to assess whether the resulting topics would contain similar words from both languages. The final size of the dataset was 2183 text news.

Once this process was completed, the embeddings were obtained (with the pre-trained BERT model in more than 50 languages) for each news text. The resulting embeddings had a dimensionality of 768, according to what was reported in the literature [21]. Subsequently, the UMAP dimensionality reduction algorithm was used, preserving two dimensions. The reasons for this choice focused on having a direct relationship between the two-dimensional visualization and the subsequent results, figure 1. The K-Means algorithm was used as a third step to obtain clusters on the embeddings of the texts. The heuristic called the elbow method was used to select the number of clusters [30]. The initial centroids required by the method were chosen according to the initialization of the K-Means++ algorithm [22]; this guaranteed that the distance between the initial centroids was distant, which ensured a higher probability of better results. In addition, the method was repeated 100 times, and the best result was selected according to [23]. Figure 2 shows the result of the elbow method applied; it is observed that the best number of clusters is four.

With the clusters obtained, the words of each text were grouped, and the TF-IDF statistic was applied. TF-IDF is a statistic used to evaluate the importance of a term in a document considering a corpus of words [8]. To calculate it, the term frequency TF is used, which is an estimate of the probability of occurrence of a term in the document [31], and the inverse document frequency (IDF), which is the change in the amount of information of a term above all the corpus [31] or the rarity of the term considering the corpus. TF-IDF is the product of TF and IDF components.

With this process, the most important words for each cluster can be obtained. This behavior is shown in 3. The clustering carried out by K-Means does an excellent job since the words seem related in their meaning. The results should also be attributed to the embeddings obtained by BERT and to the dimension reduction process with UMAP that manages to preserve the global and local information of the original BERT vector. Additionally, it is observed

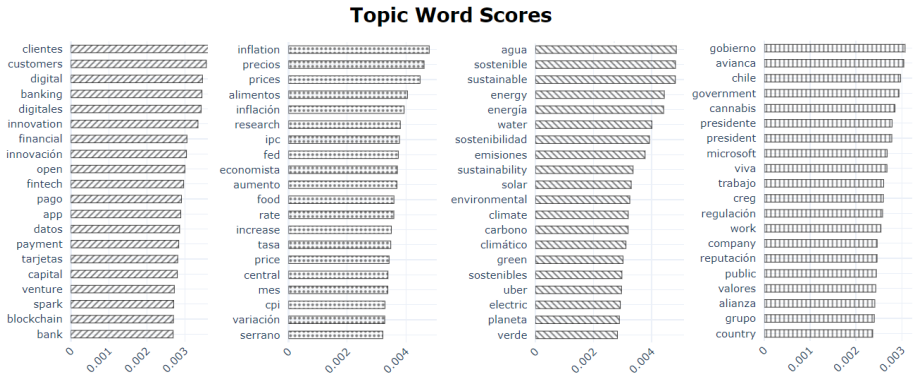


Fig. 3. Topics obtained with K-Means. The words and their TF-IDF score are shown.

that some topics contain the same words but in a different language due to the mentioned characteristics of the BERT, model pre-trained in more than 50 languages. Based on these results, topic 0 corresponds to subjects related to digital finance and innovation, topic 1 to economic issues, topic 2 to climate change and alternative energy, and topic 3 to types of government regulations or government agreements.

To statistically evaluate the results obtained by the combination of methods, two metrics extracted from [13] Topic Diversity and Topic Coherence were used. Topic Diversity is used to evaluate the redundancy of words in each document, that is, the number of unique terms with respect to a document or text. The range of values is between 0 and 1, where 1 indicates a great diversity of terms and 0 is more redundant terms or words [13]. Topic Coherence is a metric used to measure the association of terms in a corpus. It works by calculating the probabilities of occurrence of two terms in the same document [32]; this is done for all the terms of the same topic considering the corpus. The range of values goes from [-1,1], where 1 indicates a perfect association between the terms; that is, there is a co-occurrence (coherence) of the terms in the document.

Table 1 show the results of Topic Diversity and Topic Coherence for the process described in this work. A comparison with the method described in [13] is also shown. A significant difference with [13] is that the author of that work used a modified version of the DBSCAN clustering technique called HDBSCAN as the algorithm to get the clusters of the embeddings of the texts. Both methods show similar results regarding the Topic Diversity and Topic Coherence metrics. It is also observed that the general context of each topic is similar for both clustering techniques. The parameters selected for HDBSCAN were such that four topics were obtained (just like K-Means). In addition, figure 4 shows the topics created by the HDBSCAN algorithm.

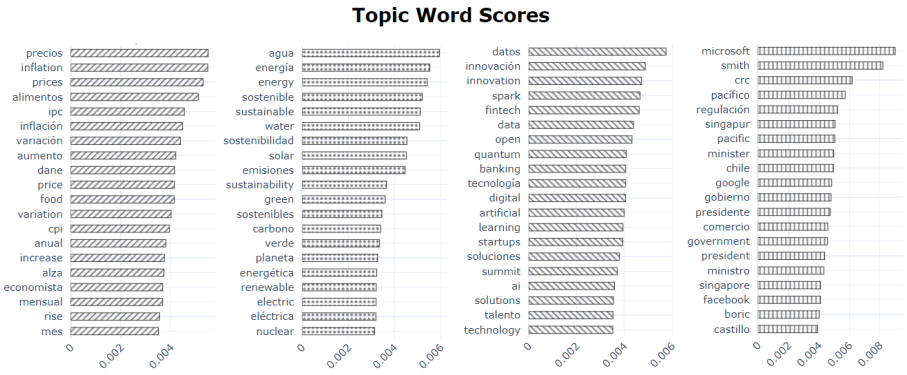


Fig. 4. Topics obtained with HDBSCAN. The words and their TF-IDF score are shown.

Table 1. Comparison between clustering methods.

Cluster method	Topics	Topic Diversity	Topic Coherence
K-Means	Digital finance, Innovation	0.25	0.53
	Economic issues	0.3	0.50
	Climate change, green energies	0.35	0.31
	Government regulation	0.425	0.11
HDBSCAN	Digital finance, technology	0.24	0.46
	Economic issues	0.32	0.4
	Climate change, green energies	0.38	0.31
	Government regulation	0.4	0.16

6 Conclusions

This work combined several algorithms to obtain the topics for a Spanish news dataset. The text was preprocessed, eliminating Spanish stopwords, punctuation marks, and special characters. Likewise, a random sample of half of the dataset was taken and translated into English, and this sample was incorporated into the original dataset. The later was done due to the type of model used to get the text embeddings. With the pre-processed text, a pre-trained BERT model trained on more than 50 languages was used to obtain each text's embeddings. Due to the high dimensionality of the resulting vectors, the UMAP dimensionality reduction technique was applied.

The K-Means algorithm was used to obtain a defined number of clusters. In this case, the number was selected by applying the cubit method. To visualize which sets of words were found within each cluster, the TF-IDF statistic was applied to obtain the most relevant words for each topic. Finally, two metrics related to the diversity of terms and coherence were applied to the topics found.

These results were compared with those obtained in a previous work [13]. A good relationship is observed between the topics and the coherence and diversity of the terms.

Through the process described in this work, good results are obtained in the topic modeling of a set of texts. The described technique could be used in another type of process in real applications where it is necessary to classify a set of unthematic texts.

References

1. Smith, T.F., Waterman, M.S.: Identification of Common Molecular Subsequences. J. Mol. Biol. **147**, 195-197 (1981)
2. Dieng, A., Ruiz, F., Ble, D.: Topic modeling in embedding spaces. Transactions of the Association for Computational Linguistics **8**, 439–453 (2020)
3. Vayansky, I., Kumar, S.: A review of topic modeling methods. Information Systems **94**, 1–15 (2020)
4. Blei, D.: Probabilistic topic models. Commun ACM **55**(4), 77–84 (2012)
5. Hong, L., Davison, B.: Empirical study of topic modeling in twitter. In: Proceedings of the First Workshop on Social Media Analytics -SOMA, pp. 80–88 (2010)
6. Ramage, D., Rosen, E., Chuang, J., Manning, C., McFarland, D.: Topic modeling for the social sciences. In: Workshop on Applications for Topic Models NIPS, Whistler, pp. 1–4 (2009)
7. Kherwa1, P., Bansal1, P.: Topic modeling: A comprehensive review. EAI Endorsed Transactions on Scalable Information Systems **7**(24) , 1–16 (2020)
8. Ferilli, S.: Automatic digital document processing and management: Problems, algorithms and techniques. Springer (2011)
9. Alghamdi, R., Alfalqi, K.: A survey of topic modeling in text mining. International Journal of Advanced Computer Science and Applications **6**, 147–153 (2015)
10. Kalepalli, Y., Tasneem, S., Teja, P., D, P., Manne, S.: Effective comparison of LDA with LSA for topic modelling. In: 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), pp. 1245–1250 (2020)
11. George, L., Birla, L.: A study of topic modeling methods, In: 2018 second international conference on intelligent computing and control systems (iciccs) IEEE, pp. 109-113 (2018)
12. Nallapati, R., Cohen, W.: Link-plsa-lda: A new unsupervised model for topics and influence of blogs. In: 2 (ed.) Proceedings of the International AAAI Conference on Web and Social Media, pp. 84–92 (2021)
13. Grootendorst, M.: Bertopic: Neural topic modeling with a class-based tf-idf procedure. Arxiv:2203.05794 (2022)
14. Suhyeon, K., Haecheong, P., L, J.: Word2vec-based latent semantic analysis (w2v-lsa) for topic modeling: A study on blockchain technology trend analysis. Expert Systems with Applications **152**, 1–12 (2020)
15. Moody, C.: Mixing dirichlet topic models and word embeddings to make lda2vec. ArXiv:1605.02019 (2016)
16. McInnes, L., Healy, J.: Umap: Uniform manifold approximation and projection for dimension reduction. ArXiv:1802.03426 (2018)
17. Yang, L., Zhiyuan, L., Tat-Seng, C., S, M.: Topical word embeddings. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, pp. 2418-2424 (2015)

18. Almeida, F., G, X.: Word embeddings: A survey. ArXiv:1901.09069 (2019)
19. Gillioz, A., Casas, J., Mugellini, E., Abou, O.: Overview of the transformer-based models for nlp tasks. In: Proceedings of the Federated Conference on Computer Science and Information Systems 21, pp. 179–183 (2020)
20. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, U., Polosukhin, I.: Attention is all you need. Advances in Neural Information Processing Systems **30**, 1–11 (2017)
21. Devlin, J., Chang, M., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACL-HLT, pp. 4171–4186 (2019)
22. Arthur, D., Vassilvitskii, S.: k-means++: the advantages of careful seeding. In: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1027–1035 (2007)
23. Pasi, F., Sami S.: How much can k-means be improved by using better initialization and repeats?. Pattern Recognition **93**, 95–112 (2019)
24. Vermeulen, M., Smith, K., Eremin, K., Rayner, G., Walton, M.: Application of uniform manifold approximation and projection (umap) in spectral imaging of artworks. Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy **252**, 1–15 (2021)
25. Arthur, D., Vassilvitskii, S.: k-means++: the advantages of careful seeding. In: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1027–1035 (2007)
26. Reimers, N., Gurevych, I.: Making monolingual sentence embeddings multilingual using knowledge distillation. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, pp. 4512–4525 (2020)
27. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. In: Conference on Empirical Methods in Natural Language Processing, pp. 3982-3992 (2019)
28. Kaggle: Spanish News Classification. https://www.kaggle.com/datasets/kevinmorgado/spanish-news-classification Accessed 15 May 2023
29. Steven, B., Loper, E., Klein, E.: Natural Language Processing with Python. O'Reilly Media Inc. (2009)
30. Thorndike, R.: Who belongs in the family?. Psychometrika **18**, 267–276 (1953)
31. Aizawa, A.: An information-theoretic perspective of tf–idf measures. Information Processing and Management **39**, 45–65 (2003)
32. Bouma, G.J.: Normalized (pointwise) mutual information in collocation extraction. In: Proceedings of the Biennial GSCL Conference, pp. 31-40 (2009)

ISSN 1870-4069

Pests Detection in Agricultural Crops using Computer Vision

Lauro Reyes-Cocoletzi[1], Miguel Angel Ortega-Palacios[2], Luis A. Cuecuecha-Sánchez[1]

[1] Universidad Autónoma de Tlaxcala, Facultad de Ciencias Básicas, Ingeniería y Tecnología, Mexico

[2] Benemérita Universidad Autónoma de Puebla, Complejo Regional Centro, San José Chiapa, Mexico

lauro.reyesco@uat.mx, miguel.ortega@correo.buap.mx, luisangel.cuecuecha.s@uatx.mx

Abstract. In the future, computer vision systems that connect machines with fungicides, insecticides and herbicides to be used on a regular basis will be needed. In the long term the systems will have autonomous monitoring of crop health and taking timely action against factors that damage crops. The problem to be solved consists of the recognition and classification of pest affectations in leaves of agricultural crops by means of machine learning algorithms ResNet18 based on the use of computer vision. Advances were made in a computer vision system for the detection of a specific pest that affects corn crops, Spodoptera frugiperda. In addition to the detection of the worm present in the captured images, plant damage is detected to infer the presence of the pest damaging the crop. This project also aims to make contributions in models of recognition systems and computer vision applicable in the prevention and reduction of the impact of pests present in agriculture of the countries.

Keywords: Computer vision, pest recognition, machine learning, plant damage.

1 Introduction

Agriculture plays an important role in food security, alleviating poverty and driving economic development. The world's population is expected to reach 9.7 billion by 2050 and 11.2 billion by the end of this century [9], so food production must increase despite various factors affecting crop yields, such as pests, weeds, pathogens, nutrients, water, sunlight, soil moisture, soil fertility nutrients, water, sunlight, soil degradation, environmental impact and scarcity of arable land.

Manual crop inspection is time consuming, prone to human error and some parts of the field may be difficult to access, reducing inspection efficiency.

Technology adaptation is crucial for food production, machine vision systems (MVS) can automate crop inspection with the help of on-site or off-site imaging techniques to improve overall crop yield [1]. Compared to human vision, MVS can predict crop problems more accurately by analyzing information acquired from images.

In recent years, the industrialized countries have had an accelerated growth in the use of applied agricultural technologies, which has led to the automation of plantations through the implementation of sensors for monitoring and determining the conditions in which crops are developing. The use of technology for automation, including novel precision agriculture techniques, has increased due to the need to determine and provide better conditions for good crop quality [1]. A pending problem to be solved is the increase of pest populations in agricultural crops (e.g., wheat, barley, corn and oats) because they can cause a significant reduction in grain yields due to their excessive spread.

A useful tool is the automation of industrial processes by means of computer vision, because it represents reliability, efficiency and speed of information processing of the environment, so the agricultural industry will use more this type of technology to monitor relevant aspects of crops.

Computer vision as mentioned is a powerful tool for crop health monitoring and with the right sensors it is possible to locate areas of affected crops in a given location in the field. With respect to specific tasks that machine vision can perform in agriculture, is the recognition and morphology of plants, so that new methods can be implemented for the detection of pests and diseases [11]. New methods for pest detection can be implemented, and advances in machine learning and high-performance computing can create efficient solutions for identifying crop diseases in order to create management, monitoring and control alternatives to reduce decision-making time once the pest or disease has been detected.

The proposal of this work focuses on computer vision techniques for the identification of pests in agricultural crops. Artificial neural networks (ANN) have a great potential in the identification of natural resources, precision agriculture, product quality assessment, sorting, grading, etc., ANN can recognize the color, shape, size and texture of an object and can find the point of interest from them (regions of interest).

The designed algorithm in this work of investigation is focused on the detection of the pest Spodoptera frugiperda (worm) and the detection of the damage caused by this animal in corn crops.

This paper is organized as follows, in section 2, relevant related works are mentioned, in section 3, details of the proposed methodology and relevant information are presented, in section 4, the experiments and the results obtained are described. Finally, Section 5 presents the conclusions and future work.

2 Related Work

The following are some related works to be taken into consideration with respect to the problem to be solved.

Yao et al. [14] propose a rice pest identification system using two 12MP digital cameras, the cameras are placed on a glass plate with 4 black light sources to attract the pests. The main objective was to detect four different rice pests of lepidopteran species. Their work achieves an accuracy of 90.5% without cross-validation and 97.5% by cross-validation. The main problem is the overlapping of insects, in these cases, manual separation is performed.

Vakilian et al. [3] developed a system to identify beet armyworm (Spodoptera Exigua), a pest of vegetable, field, and flower crops. Images were captured with a digital camera together with an illumination module, images utilized for training ANN classifier and remaining for evaluation. Convolutional neural network (CNN) classifier was able to classify armyworms with an accuracy of 90%.

Qing et al. [15] proposed a technique to measure the population density of white-backed grasshopper (WBPH) in rice paddies. A digital camera attached to an extendable pole was used to detect the pest on rice stalks. Detection was done in three-layer mechanism, the first layer is an AdaBoost classifier, the second is a support vector machine (SVM) classifier based on the histogram of oriented gradient (HOG), and the third layer used threshold based on one color and three shape features. They achieved a detection accuracy of 90.7% with 4.9% false detection rate.

Rajan et al. [16] proposed an automatic pest identification system to detect whiteflies, aphids, and cabbage moths. Digital camera was used to capture images of the crop which may have pests on their leaves. SVM classifier was used to train with threshold values and the slack variables of the images in the database collected. The threshold value was used to distinguish the object from the background and classification of the pests was done using slack variables. They achieved a detection accuracy of 95%.

The above-mentioned works are some of those already carried out; a summary of other proposals is shown in Table 1.

In this context, progress is presented in the research project to develop a system for the detection and subsequent monitoring of pests in agricultural fields by applying machine learning models through computer vision.

Specifically, the aim is to identify those pests that damage or alter the surface of the crop leaves, because the determination of the existence of a pest can be detected directly or indirectly by computer vision. Direct detection of the pest involves observing the insect or worm or feeding on crop leaves; on the other hand, indirect observation involves detecting damage or discoloration on crop leaves without the presence of the animal that causes it.

It is important remember that the artificial vision system designed in this work focuses on detecting a specific worm (Spodoptera frugiperda), in addition to detecting damage to the leaves of the crop (specifically corn) to infer the presence of pests damaging the plants.

Table 1. Computer vision based methods for pest detection.

Reference	Type of crop	Pest (name)	Method	Accuracy
[18]	Corn	Corn disease	ResNet	97.5%
[11]	Multiple crops	Beet armyworm	ANN	90.0%
[12]	Paddy	WBPH	AdaBoost & SVM Classifiers	85.2%
[13]	Paddy	Brown plant hopper (BPH)	One-way ANOVA	$< 70.5\%$
[15]	Multiple crops	Codling moth	ConvNets (CNN)	93.4%
[14]	Multiple crops	Whiteflies	SVM	95.0%
[17]	Strawberry	Thrips	SVM	$> 97.5\%$

3 Methodology

The methodological basis of this research project is the observation of the environment through the implementation of computer vision with previously captured images. The methodological basis of computational processing applies object detection and color segmentation algorithms for the analysis of the information of interest.

A Machine Learning approach is used for the analysis of the information of interest, which allows the classification of the observed in the crop. As mentioned the processed images include crop leaves in different health conditions (different colors), the number of images of the different classes will be balanced in an acceptable range of samples to validate the training of the ANN.

Ideally, pests should be detected as early as possible, but when their small size, e.g. at the egg stage, macro lenses have to be used to obtain images, this is not practical in field applications.

Preprocessing includes considerations of crop leaf damage distribution as they may occupy only a small portion of pixels in the captured images and may not be suitable enough for ANN model training. Regions of interest (ROI) are highlighted from the original images (data labeling).

Figure 1 shows in general the blocks corresponding to the algorithm to be developed, which is explained in more detail below.

3.1 Information Collection (dataset) and ROI

The system dataset consists of one general scenario, which consists of the identification of leaf, stem and fruit damage in the corn crop caused by the corn worm pest. IP102 dataset [5] has 737 images of interest, also we utilize the internet as the primary source to collect images, which is widely used to build datasets such as the ImageNet [12] and the Microsoft COCO [10]. The first collection step relies on common image search engines, including kaggle,the

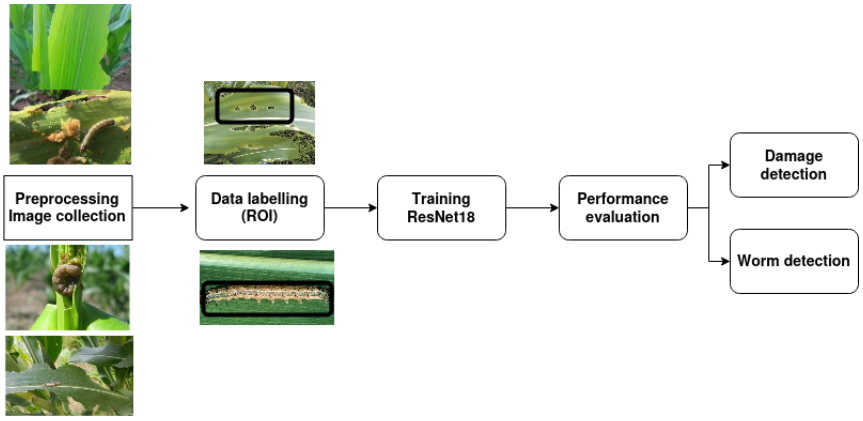


Fig. 1. Stages of the algorithm design.

(a) (b)

Fig. 2. Example of crop damage (a) leaf affected by worm, (b) worm in the stem.

total number of images collected that make up the data set is about 7,000. Figure 2 shows some of these samples.

Regions of interest are delimited in the images to train with samples of worm-damaged leaves as well as the presence of the worm in the stems or leaves, the identification (labeling) was done with labelImage [7]. The cropped images have sizes from 328 × 328 pixels to 600 × 600 pixels and the regions of interest can be clearly observed in the images.

According to the proposed methodology, the first step involves preprocessing the information to facilitate the characterization of the leaves of the crops, a filtering process is performed within the image obtained, with the purpose of eliminating as much noise as possible, present in the images after their acquisition. This stage focuses on suppressing excess lighting, unwanted shadows and elements that are not part of the leaf, highlighting in turn, the information necessary for further analysis.

3.2 Learning Stage

Characterization of the relevant aspects of the image, since this stage of the system is very important for an adequate final classification.

The basis for the detection of damage in agricultural crops is based on the implementation of a ANN. This network will focus on the detection of

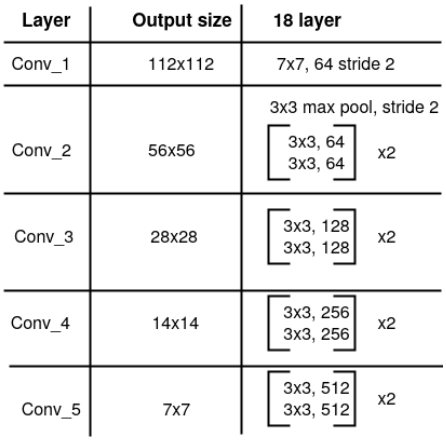


Fig. 3. Architecture ResNet18 [8].

affectations in crop leaves and presence of worm, as an initial development, the use of a Residual Network architecture based on ResNet18 [8] was proposed to deal with the vanishing gradient problem.

The core idea of ResNet18 is to introduce hop connections or residual connections, which allow network layers to learn differences rather than learning entire functions. These residual connections allow gradients to propagate more easily through the network, which helps to avoid the problem of gradient fading [4].

The ResNet architecture is based on residual blocks, each block contains a series of convolutional layers and can be stacked to form a deeper network [2]. The residual block has a shortcut path structure that adds the output of one layer to the output of a subsequent layer, known as the skip connection operation (Figure 3).

As usual after convolutional layers, convolutional filters are used to extract local features from the images, however, after passing through several convolutional and clustering layers, the resulting representation may still have local features and not be completely related to the final classification. Dense layers are added to perform a global classification and combine the extracted features into a more complete and global representation of the image.

The model assigns a probability to each class that represents its confidence that the example belongs to that class.

On the other hand, we have the actual labels that indicate the true class to which each example in the data set belongs; cross-entropy is used to quantify the difference between the prediction probabilities generated by the model and the actual probabilities or labels in the data set.

ResNet18 uses a training loop to adjust the network parameters using the training set, at each iteration, it performs the following steps:

– Passes a batch of images through the network to obtain the predictions.

- Calculates the loss using the predictions and the actual labels.
- Performs backpropagation of the error to compute the gradients.
- Updates the network parameters using the optimizer and the calculated gradients.

Cross entropy measures how different these two probability distributions are, the probabilities predicted by the model and the actual probabilities. In particular, it is a measure of the loss of information or uncertainty in the model prediction compared to the actual labels.

In the case of binary classification (worm, damaged leaf), the Equation 1 is applied:

$$H(p, q) = -[p \cdot log(q) + (1 - p) \cdot (log(1 - q))] \,, \tag{1}$$

where p is the actual probability of the class and q is the probability predicted by the model for the class.

The optimization during training used is the ADAM algorithm (Adaptive Moment Estimation) to adapt the size of the learning steps (learning rate) for each parameter as a function of the first and second moment estimates of the gradients [6]. The initialization of parameters and hyperparameters is of great importance to efficiently perform the iterative loop.

It is important to note that the choice of these hyperparameters is not trivial and requires adjustments by experimentation. In general, a hyperparameter search was performed using random search techniques to find combinations that work well for our specific problem resulting in $\beta_1 = 0.85$, $\beta_2 = 0.94$.

At each iteration, the gradient of the objective function with respect to the parameters is calculated, gradient indicates the direction in which the parameters should be adjusted to reduce the loss. Parameter updates are calculated using the corrected first-order and second-order moments and the adaptive learning rate, the moments are corrected to compensate for initial biases.

3.3 Performance Evaluation

Generally, the transfer learning method discards the last layer of a pre-trained model and adds a fully connected layer where the neurons correspond to the number of predicted classes. During the training stage the last layer is trained from scratch, while the others are initialized from the pre-trained model and updated.

To provide a direct observation of the classification results confusion matrices will be calculated in addition statistics will be used to evaluate the performance of the models, training accuracy as the percentage of correctly classified samples in the training data set and similarly validation on the data set given the epochs run when the model begins to converge.

Detect and classify the observed pest to assign a label to it compare with a number of relevant samples the system carried out.

Since the model returns probabilities for each class instead of direct labels, it is necessary to convert the probabilities to predicted labels. This is usually

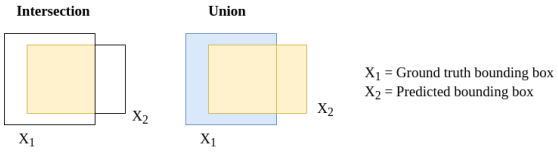


Fig. 4. Bounding box detection relation CM.

done by taking the index of the highest probability as the predicted label for each example.

The mean average precision (mAP) was used as the validation metric [18] for damage leaf and pest detection, mAP score was calculated as follows: average across the number of classes of the true positive divided by the true positives plus false positive as in the Equation 2:

$$mAP = \frac{1}{\#classes} \sum_{1}^{\#classes} \frac{\#TP}{\#TP + \#FP} \, . \tag{2}$$

In addition to mAP score, we also computed a confusion matrix (CM), for each detection, the algorithm mines all the ground-truth boxes and classes, along with the detected boxes, classes, and scores (probability of success in the bounding box). Only detections with a score ≥ 0.5 were considered and anything under this threshold were excluded. The list of matches was trimmed to remove duplicates (ground-truth boxes that match with more than one detection box or viceversa), if there are duplicates, the best match was continually selected.

The CM was updated to refect the resultant matches between ground-truth and detections, a detected box was refected as correct where the intersection over union (IoU) of that box and the corresponding ground-truth box was $\leq$ 0.5. Explanation for calculating IoU [17] is shown in Figure 4 and Equation 3, the CM was normalized:

$$IoU(X_1, X_2) = \frac{X_1 \cap X_2}{X_1 \cup X_2} \, . \tag{3}$$

4 Results

Implement a functional and reliable system for the detection and classification of pests affecting agricultural crop fields by means of novel machine learning techniques, which include the characterization of ResNet18 architecture. The convolutional neural network in addition to detecting the ROI (affected areas on the leaf) will also perform a membership prediction with respect to a number of possible classes to identify the type of affectation and/or pest.

This prediction is indicated through an enveloped frame of the object detected in the scene as well as the percentage of class membership using a

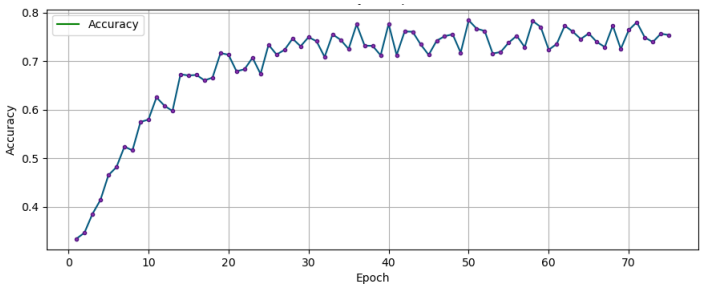


Fig. 5. Accuracy vs epochs.

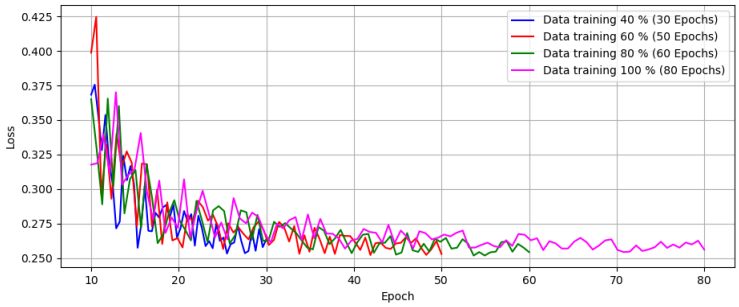


Fig. 6. Training loss curves of diferents epochs.

multi-label classification. During training, binary cross-entropy loss is used for the class predictions by classifying the class predictions by means of independent logistic classifiers.

After more than 60 epochs of training, Figure 5 and Figure 6 depict the accuracy and loss curves in training. With different amounts of training data-samples according to the number of epochs to be performed, in Figure 6 shows the loss value of ResNet18 reaching 0.26. After training, the maximum accuracy of our proposed method on the validation set can reach 0.75, and the minimum loss is 0.25.

Figure 7 shows examples of results obtained, showing the percentage and label of belonging to the two classes to be detected: worms and damaged leaves.

The performance shown by the proposed algorithm obtained for the detection of damaged leaves in the test data set a mAP rate of around 70%, for the case of worm detection in the case of detection of leaves damaged by pest the mAP rate obtained was 75%. Based on the results obtained on the test dataset, the confusion matrix associated with the results obtained is presented in Table 2.

Set the CM M_{ij}, in which each column (Table 2) of the matrix M_j (where i = 1, 2) represents the class prediction of the sample by the classifier, and each row of the matrix M_i (j= 1, 2) represents the ground truth to which the sample

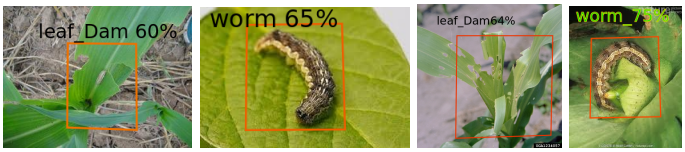


Fig. 7. Results obtained in pest detection.

Table 2. Results of confusion matrix.

	Prediction worn	Prediction damage leaf
Worn	656	86 Damage leaf
75	495	

belongs. Three general metrics for evaluating the performance of class models can be obtained from the CM.

The accuracy is percentage of correctly labeled samples in classified samples. It can reflect the classification performance of the model on data (Equation 4):

$$accuracy = \frac{\sum_{i=1}^{2} M_{ii}}{\sum_{i=1}^{2} \sum_{j=1}^{2} M_{ij}} \ . \tag{4}$$

Equation 5 defines precision, which measures the probability of correctly predicted samples in all predicted i-type samples. It denotes the classification effect of the algorithm:

$$precision = \frac{M_{ii}}{\sum_{j=1}^{2} M_{ij}} \ . \tag{5}$$

The recall is used to measure the probability that the prediction is correct in the instances labeled as i. It can express the effect of a certain type of recall, this calculation process is described in Equation 6:

$$recall = \frac{M_{ii}}{\sum_{i=1}^{2} M_{ij}} \ . \tag{6}$$

The f1-score (f1) is calculated by taking the weighted average of precision and recall (Equation 7). In other words, f1 conveys a balance between precision and recall. Although it is not as intuitive as accuracy, f1 is generally more valuable than accuracy, mainly when the class distribution is uneven:

$$f1 = 2 * \frac{precision * recall}{precision + recall} \ . \tag{7}$$

With the information from the CM, the results obtained (expressed in percentages) for the metrics of interest are listed below. For precision a value of 89% was obtained, for recall a value of 88% was obtained and finally for f1 a value of 88% was obtained.

It is worth mentioning that the confusion matrix presents the results where the mAP corresponding to the detection in the image of the detected class is greater than 65%, however there are images that were provided to the algorithm where it was not able to locate worm or damaged leaf when in fact there was.

According to the related work, an initial comparison can be made of the performance presented in this work vs. the works found in the state of the art. In general terms the performance of the proposed system has an accuracy of 75% and the work with the best performance has 97.5% [18], however the algorithm of our proposal in addition to detecting the animal (pest) additionally takes into consideration the damage that this pest causes in the plant.

These results are susceptible to improvement, the main problems to overcome being background noise in field environment, substantial overlapping of multiple leaves and scattered symptoms on different leaves. To handle these issues, we are currently collecting and labeling images of early stage damage leaf for improving the accuracy of the model, and the ability to generalize, because the dataset is not big enough.

5 Conclusions

This research work presents initial results with respect to crop pest detection, the evaluation metrics provide promising data feasible to improve by specific changes to the proposed algorithm.

Take the feature map (ResNet18) of the layers and increment by two, use a feature map from a previous network layer and merge with the particulars of the up-sampling using concatenation to predict a similar tensor, but now twice the size. This method will allow us to obtain more meaningful semantic information of the sampled distinguishing features and more detailed information of the previous feature map.

Increment more convolutional layers under the addition + concatenation model to predict boxes for the final scale. In this way, the predictions for the third scale benefit from all previous computation, as well as from the fine-grained characteristics of the first stages of the network.

With respect to the ANN, it is expected to predict boxes at 3 different scales in order to extract distinctive qualities of those scales using a concept similar to that of feature pyramid networks. From the base feature extraction, convolutional layers are added to extract the relevant qualities, in particular the last layer predicts a tridiagonal layer predicts a three-dimensional tensor that encodes the bounding box and class predictions.

ADAM optimization model used also has areas of opportunity such as momentum decay factors (β_1 and β_2) to aid model convergence without negatively affecting the rate (slower approximation to the learning rate).

The results obtained by ADAM are relevant, however, other optimizers such as SGD (stochastic gradient descent) could be used with momentum to verify the increase in algorithm performance.

Detection performance can be improved by using an architecture with more layers, in this case a network superior to ResNet18, such as ResNET34, ResNet50 or higher, ResNet18 was used because the capabilities of the computer equipment used are limited, with the right hardware a more robust architecture can be implemented.

At this point and with the obtained data, the results need to increase in terms of the performance obtained, the points mentioned with respect to the changes in the ANN will help improve the evaluation metrics of the algorithm.

As additional work, field tests or at least processing videos taken from real environments are also contemplated to verify and compare the results obtained with the images from the repositories.

The results obtained in this work for the moment are preliminary, they are advances corresponding to an initial stage of development of the proposed algorithm that can be improved.

References

1. Abbaspour-Gilandeh, Y., Aghabara, A., Davari, M., Maja, J. M.: Feasibility of using computer vision and artificial intelligence techniques in detection of some apple pests and diseases. Applied Sciences, vol. 12, no. 2, pp. 906 (2022)
2. Albanese, A., Nardello, M., Brunelli, D.: Automated pest detection with dnn on the edge for precision agriculture. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 11, no. 3, pp. 458–467 (2021)
3. Asefpour Vakilian, K., Massah, J.: Performance evaluation of a machine vision system for insect pests identification of field crops using artificial neural networks. Archives of phytopathology and plant protection, vol. 46, no. 11, pp. 1262–1269 (2013)
4. Barraza, J. A., Espinoza, E. J., Espinos, A. G., Serracin, J.: Precision agriculture with drones to control diseases in the rice plant, (2020)
5. Bojer, C. S., Meldgaard, J. P.: Kaggle forecasting competitions: An overlooked learning opportunity. International Journal of Forecasting, vol. 37, no. 2, pp. 587–603 (2021)
6. Ermoliev, Y. M., Wets, R.-B.: Numerical techniques for stochastic optimization. Springer-Verlag (1988)
7. Gao, B.-B., Zhou, H.-Y.: Learning to discover multi-class attentional regions for multi-label image recognition. IEEE Transactions on Image Processing, vol. 30, pp. 5920–5932 (2021)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
9. Júnior, T. D. C., Rieder, R., Di Domênico, J. R., Lau, D.: Insectcv: A system for insect detection in the lab from trap images. Ecological Informatics, vol. 67, pp. 101516 (2022)
10. Liu, H., Chahl, J. S.: A multispectral machine vision system for invertebrate detection on green leaves. Computers and Electronics in Agriculture, vol. 150, pp. 279–288 (2018)
11. Liu, H., Chahl, J. S.: Proximal detecting invertebrate pests on crops using a deep residual convolutional neural network trained by virtual images. Artificial Intelligence in Agriculture, vol. 5, pp. 13–23 (2021)

12. Muppala, C., Guruviah, V.: Machine vision detection of pests, diseases and weeds: A review. J. Phytol, vol. 12, pp. 9–19 (2020)
13. Partel, V., Nunes, L., Stansly, P., Ampatzidis, Y.: Automated vision-based system for monitoring asian citrus psyllid in orchards utilizing artificial intelligence. Computers and Electronics in Agriculture, vol. 162, pp. 328–336 (2019)
14. Qing, Y., Jun, L., Liu, Q.-j., Diao, G.-q., Yang, B.-j., Chen, H.-m., Jian, T.: An insect imaging system to automate rice light-trap pest identification. Journal of Integrative Agriculture, vol. 11, no. 6, pp. 978–985 (2012)
15. Qing, Y., Xian, D.-x., Liu, Q.-j., Yang, B.-j., Diao, G.-q., Jian, T.: Automated counting of rice planthoppers in paddy fields based on image processing. Journal of Integrative Agriculture, vol. 13, no. 8, pp. 1736–1745 (2014)
16. Rajan, P., Radhakrishnan, B., Suresh, L. P.: Detection and classification of pests from crop images using support vector machine. In: 2016 international conference on emerging technological trends (ICETT). pp. 1–6. IEEE (2016)
17. Selvaraj, M. G., Vergara, A., Ruiz, H., Safari, N., Elayabalan, S., Ocimati, W., Blomme, G.: Ai-powered banana diseases and pest detection. Plant methods, vol. 15, pp. 1–11 (2019)
18. Yu, H., Liu, J., Chen, C., Heidari, A. A., Zhang, Q., Chen, H., Mafarja, M., Turabieh, H.: Corn leaf diseases diagnosis based on k-means clustering and deep learning. IEEE Access, vol. 9, pp. 143824–143835 (2021)

ISSN 1870-4069

Steganography in Frequency Domain: Hiding Text through Audio Spectrogram

Luis Enrique Morales-Márquez

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

luise.morales@viep.com.mx

Abstract. Steganography aims to hide information in digital media so that it is imperceptible, among the most used methods is the use of LSB, and hiding bits of a message in spectral peaks at high frequencies can help improve imperceptibility. This study takes some test audios, obtains the spectrogram and hides bits of a message in the peaks at high frequencies, it is concluded that the performance in terms of MSE, SNR, PSNR and SSIM are generally good since they preserve the quality of the carrier audio of the information.

Keywords: Frequency domain, steganography, spectrograms, LSB.

1 Introduction

Steganography aims to hide secret messages or data via digital media such as images, audio clips, or video sequences. In this way, the information that needs to be discreetly transferred is protected from unauthorized individuals, while keeping the appearance of the multimedia file containing it unchanged. In all digital media, the most commonly used technique is embedding in the least significant bits (LSB) [1]. There are some key properties that must be addressed in these procedures:

- Embedding capacity: The amount of data that can be hidden in the cover media in relation to its size. More information cannot be hidden than the information that is contained in the cover itself.
- Undetectability: The data must be inserted in such a way that the secret message cannot be accidentally perceived when playing or observing the file containing it. If the message is detected at first glance, the steganography has failed.
- Robustness: Ability to withstand methods attempting to retrieve the secret message. The recovery of information should not be simple, but neither should it be complicated for the authorized recipient [2].

The success of audio steganography depends on the behavior of the Human Auditory System (HAS), as it is more sensitive to changes than the visual system [3]. Therefore, special care must be taken when using audio media for data hiding.

The hiding method proposed in this article involves obtaining the spectrogram of the audio file and selecting the elements with the highest energy at high frequencies to hide

bits of a text string in the 3 LSB of the high spectral peaks. The audio is then recovered from the modified spectrogram, obtaining the stego audio.

The structure of the article is as follows: Section 2 briefly presents related work on audio steganography; Section 3 details the proposed method and associated theoretical concepts, as well as metrics for evaluating the quality of steganography; Section 4 shows the results obtained, which are analyzed in Section 5; and finally, Section 6 presents the conclusions derived from this study.

2 Related Work

Information hiding in audio files has been widely explored, just as in image or video files. Below are some examples of audio steganography.

The technique of hiding information in the LSB is one of the most widely used in steganography for all types of media. Hussian, J., & Farhan, K. [4] designed a model in 2016 that generates a random sequence of bits, modifying only 2 bits in each audio window. They then generate a HASH of the key and apply an XOR operator with the bits of the data to be hidden. This result is embedded in the bits selected by a random generator.

On the other hand, Chua, T. et al. [1] in 2017 considered that, to maintain the imperceptibility of the embedded messages, it is appropriate to encrypt the secret message using the RC4 algorithm and a password defined by the sender. Subsequently, the bits of the encrypted message are inserted into the selected audio. The recipient must know the encryption password to obtain the secret message back.

In 2021, Zainab, N., & Ban, N. [5] proposed an indirect LSB insertion method, which consists of obtaining the lengths of the audio and the text to be attached. The message must not exceed an eighth part of the length of the audio. After encrypting the message, it is processed with the XOR operator with bits of several prime numbers. The encoded message is compared to the first bit of the position indicated in the audio of a sequence of numbers. If they are equal, the message bit is embedded in the least significant bit of the position indicated in the audio.

Using pre-trained neural networks, Galeta, M. et al. [6] in 2021 employed a residual network to encode an image selected as the secret message and add it to the audio spectrogram. The recovery of the attached image is also the responsibility of a residual neural network. This allows for a larger area for bit insertion than a one-dimensional signal, such as the traditional audio representation. The spectrogram is performed with the cosine transform.

Finally, Abood, E. et al. [7] in 2022 developed a hybrid model. The message is encrypted with a bit-swapping technique based on a key hidden in the audio in the time domain. The insertion of the key is done in LSB at random positions in this domain. Since the sampling rate in the time domain determines the number of points representing the signal, there are a large number of bins in which insertion can be performed.

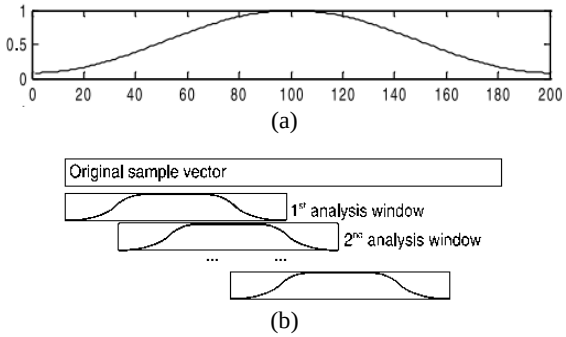


Fig 1. (a) Hamming window, the amplitude is represented on the *y* axis and the number of elements of the window on the *x* axis. (b) Overlapping of the blocks resulting from the windowing [11].

3 Proposed Method

LSB techniques are easy to understand and have been widely developed. The simplest way to apply this method is to sample the signal in the time domain and select the peaks with the highest amplitude. However, this idea may result in a relatively easy extraction of the hidden data. Therefore, in this work, we propose to insert the data inside the audio spectrogram by hiding them in the frequency domain, specifically at high frequencies. It is based on the fact that the human ear has a hearing range that goes from 15Hz to some value between 15kHz and 20kHz, depending on the individual.

In general, we should not be able to perceive audio with alterations at high frequencies, as even though an individual could exceed the 20kHz barrier, their hearing quality at such a high frequency is very poor [8]. This means that, in theory, an audio file with an embedded message should sound almost exactly the same as an unaltered audio and the changes should be detectable only through analysis, to achieve that objective, the following method is proposed.

3.1 Insertion Stage

First, the WAV audio file is selected and sampled at 44.1kHz, a standard sampling frequency that obeys the sampling theorem. This indicates that sampling must be performed at least twice the signal frequency to avoid aliasing phenomenon [9]. In this way, we slightly exceed twice the human audible limit of 20kHz. Then, the signal windowing process is made as follows: the audio is segmented into 256-unit windows using the Hamming-type function, which allows for greater frequency resolution and, therefore, better storage capacity at high frequencies [10]. We choose 256 bins in the window to maintain a balance between the number of blocks to work with and the block size. Additionally, we define a 50% block overlap, which ensures signal continuity during analysis and reconstruction [11] (see Figure 1).

We represent the energy value of a certain frequency at a specific time (see Figure 2), this is called a spectrogram and is obtained through the Short-Time Fourier Transform (STFT) applied to each signal block obtained in the previous step. The STFT

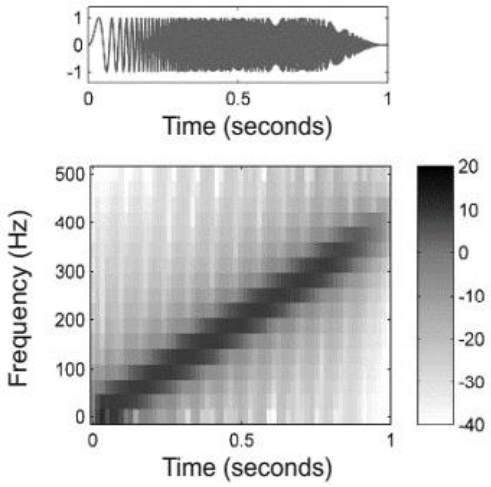


Fig 2. Spectrogram of an audio signal of 1 second duration using a window size of 32 bins and overlap of 16 bins [11].

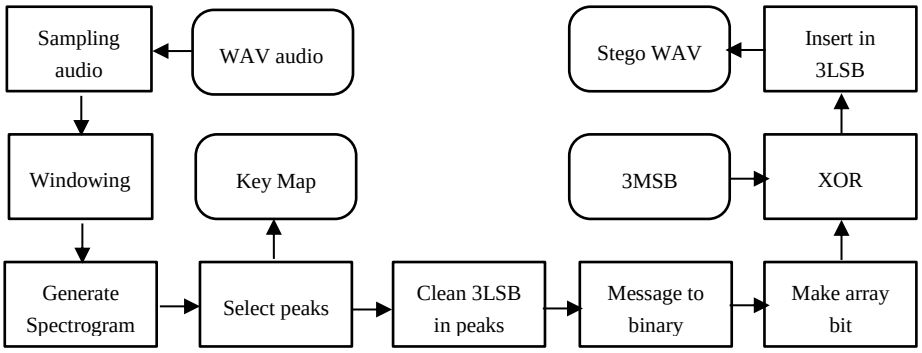


Fig. 3. Steps followed in the insertion stage.

of the nth block with length $K = i_e(n) - i_s(n) + 1$, where i_e and i_s are the final and initial times of the block, k is the frequency of interest and j is the imaginary component. The STFT is defined in Equation (1) [9]:

$$X(k,n) = \sum_{i=i_s(n)}^{i_e(n)} x(i) \exp\left(-jk\left(i - i_s(n)\right)\frac{2\pi}{K}\right). \quad (1)$$

The selection of frequency and initial energy level from which the points of the spectrogram will be chosen to hide the message bits. Starting from the selected frequency, a bit is written as 1 if the point is selected for insertion or a 0 if not, in a key file with which the positions with information are obtained during message recovery.

This is the time to clear the 3 LSBs of the value of selected spectrogram points, that is, the 3 LSB are set to '000'. After clearing, we count the number of bits available to insert information. Then a message that can be hidden in that number of bits is chosen

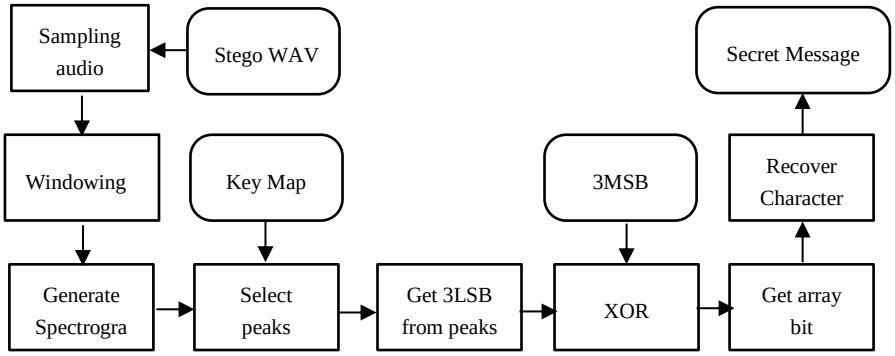


Fig. 4. Process to retrieve secret message from stego audio file.

considering a representation of 8 bits per character, subsequently, every character is converted into 8-bit string.

To reduce the risk of message extraction by unauthorized persons, the message bits are arranged in a vector constructed as follows: we place the first bit of each character, then the second bit of every character, and so on until the least significant bit is reached. For example, considering the string “EL” with ASCII values 69 and 76 for each character respectively, and binary values '01000101' and '01001100', the bit string to be inserted would be '0011000001110010'.

For insertion, the bit string is placed in batches of 3 into the cleaned LSBs, protecting them with an XOR masking with the 3 MSBs of the selected elements of the spectrogram. Finally, the recovery of the modified audio signal is done by applying the inverse STFT and writing it to a WAV file. This procedure can be schematically seen in Figure 3.

3.2 Recovery Stage

The process to retrieve the secret message is similar to the procedure described above: first we read the stego WAV file with a sampling frequency equal to that of the insertion stage, the signal windowing is made with the same parameters as in the insertion stage. Then the spectrogram is obtained using the STFT.

At this point, the key file containing the spectrogram points with information to be extracted is read. For each point, if was selected, we operate the 3 LSBs and the 3 MSBs of the real part using the XOR gate to generate a string. Finally, we extract each character from the string and write it to a text file, considering the order of the character bits from insertion stage. The previous process can be seen in Figure 4.

3.3 Evaluation of Hiding Quality

Traditional metrics are used to compare the original media and the stego media:

– Mean Square Error (MSE): Is the error between the original signal and the stego signal in the form of mean squared error, expressed as an average. Low values indicate insignificant changes in the audio and are given by Equation (2):

$$MSE = \frac{\sum_{i=1}^{M} ||x(i) - y(i)||^2}{M}, \tag{2}$$

where M is the number of points or moments considered in the signal, $x(i)$ is the value that the original signal takes at moment i, and $y(i)$ is the value that the stego signal takes at moment i, both $x(i)$ and $y(i)$ are evaluated in the time domain [7].

– Signal to Noise Ratio (SNR): The ratio between the signal power and the noise power, usually expressed in decibels (dB), is given by Equation (3):

$$SNR = 10 \log_{10} \left(\frac{\sum_{i=1}^{M} x(i)^2}{\sum_{i=1}^{M} [x(i) - y(i)]^2} \right), \tag{3}$$

where $M, x(i)$ and $y(i)$ behave in the same way as in Equation (2) [5].

– Peak Signal to Noise Ratio (PSNR): Used to calculate the quality of steganography, it is a metric that evaluates the distortion of the modified media, measured in dB. Generally, a value greater than 30 indicates that the hidden information will go unnoticed, and it is defined by Equation (4):

$$PSNR = 10 \log_{10} \left(\frac{\max\{x\}}{\sqrt{MSE}} \right), \tag{4}$$

where $max\{x\}$ is the maximum value of signal amplitude values in the time domain and MSE is the Mean Squared Error calculated with Equation (2) [7].

– Structural Similarity Index (SSIM): It is used to evaluate the similarity between the original media and the media with the inserted information. If the result is close to 1, then the altered media maintains good quality and is very similar to the original, and is given by the expression:

$$SSIM = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}, \tag{5}$$

where μ_x and μ_y are the mean values of the signal in the time domain of the original media x and the stego media$y$, σ_x and σ_y are the standard deviations of the signals and σ_{xy} is the covariance of the signals, in addition C_1 and C_2 adopt values of 0.01 and 0.03 respectively in order to avoid instability when the mean or standard deviation is close to zero [7].

4 Results

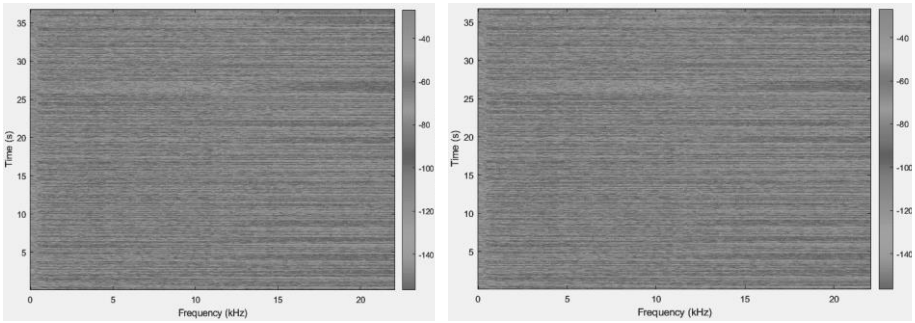
For the tests of the proposed method, 3 audios from the resource list of the Audio Content Analysis website [12] were used, those are "audio_MusicDelta_Britpop_Drum.wav" with a duration of 36 seconds, "audio_pop_excerpt.wav" with 14 seconds, and "audio_speech_excerpt.wav" also with 14 seconds duration. Chat GPT model [13] was asked to generate a random tale in spanish, from which the first necessary characters were extracted, the largest chunk of 141 letters obtained is *"Había una vez un pequeño pueblo ubicado en medio de un*

Table 1. Some parameters selected for insertion, payload capacity and duration.

Cover audio	Minimum Insertion Frequency (kHz)	Minimum Insertion Power (dB/Hz)	Number of Characters	Duration (sec)
audio_MusicDelta_Britpop_Drum	15	-70	33	36
audio_pop_excerpt	10	-70	141	14
audio_speech_excerpt	10	-80	121	14

Table 2. Results of the established metrics.

Cover audio	MSE	SNR (dB)	PSNR (dB)	SSIM
audio_MusicDelta_Britpop_Drum	0.0914	82.1428	94.9246	0.9999
audio_pop_excerpt	0.2824	84.7829	93.0542	0.9998
audio_speech_excerpt	0.1411	76.5768	89.9308	0.9996



(a) (b)

Fig. 5. (a) Spectrogram of the original audio. (b) Spectrogram of the stego audio.

bosque denso y frondoso. El pueblo estaba formado por pequeñas casas de madera, cada u", this text and some subchunks was hidden in audio files.

The audios were sampled at 44.1kHz with a Hamming window of 256 elements and 128 overlaps to ensure signal continuity, and a high frequency and energy level were chosen from which to select the points to hide information in the 3 LSBs, based on this, Table 1 is obtained.

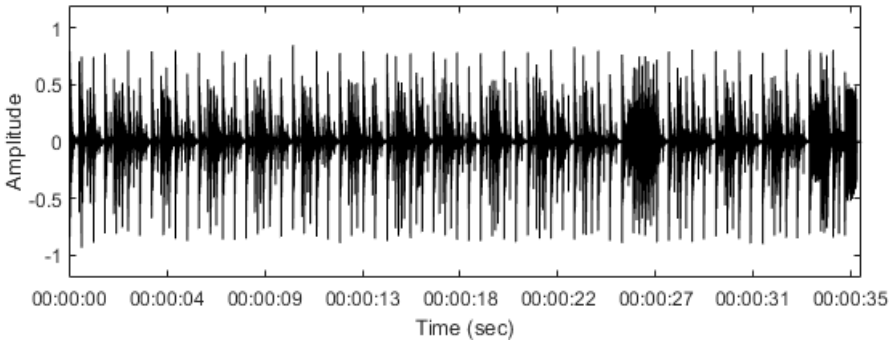
The maximum possible number of bits was hidden according to the method, and the metrics mentioned in section 3 were calculated; the results are reported in Table 2.

The original and stego spectrograms of the file "audio_MusicDelta_Britpop_Drum" are shown below in Figure 5a and 5b, respectively.

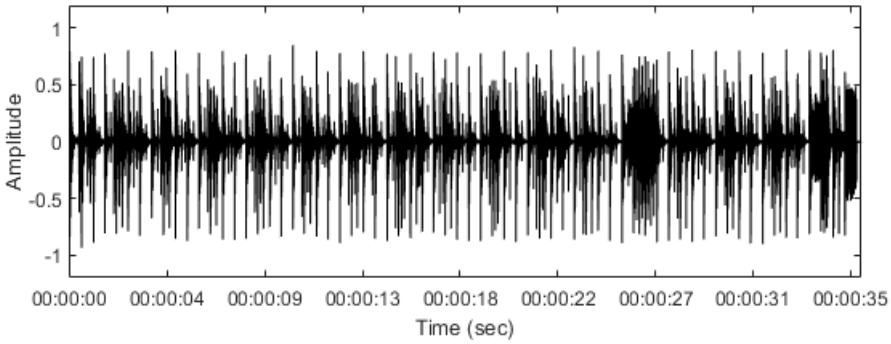
The spectrogram is not usually the common way to represent a signal, the most used way to show them graphically is as waves in the time domain, this representation of the same audio file in Figures 5a and 5b can be seen in Figure 6a and 6b.

Note that both audio files are practically the same when viewed in their full representation, which is to be expected considering the data shown in Table 2, so the difference between the original audio and the stego audio is shown in Figure 7.

A sample of the change can be seen in Figure 8, in (a) the change between the waves is observed in a high-frequency segment where text bits were inserted, while in (b) a low-frequency segment is shown where there is no change.



(a)



(b)

Fig. 6. (a) Original audio signal. (b) Stego signal. The *x* axis represents the index of the bins along the signal.

5 Analysis

The audio "audio_MusicDelta_Britpop_Drum" required an initial frequency of 15kHz, which is higher than the 10kHz of the other two audios. This is because there are higher energy levels at that high frequency, while the other audios had few bins with energy at that frequency, so the range was reduced to 10kHz. Similarly, it was sought to keep the minimum energy level in the region represented in green in the spectrogram. For the 3 audios, this value remains similar. The reason for choosing high energy levels is because these levels, which are the spectral peaks, are particularly useful, as they tend to be resistant to noise, which theoretically helps in preserving the hidden message. Evaluating the resistance to attack or compression of the stego file goes beyond the objective of this document, which is limited to proposing a method of hiding bits in the frequency domain.

Reviewing the data hiding evaluation metrics, it can be seen that the mean squared errors in the 3 files are considerably low, never exceeding 0.3 units with a minimum of 0.0914. The SNR remains of good quality, above 82 and close to each other for the files "audio_MusicDelta_Britpop_Drum" and "audio_pop_excerpt", while it drops a little to

just over 76 for "audio_speech_excerpt". However, it can be considered acceptable. The PSNR is, in general, high, above 89, and the structural similarity is very close to 1 in all cases. Therefore, it can be said that the steganography work has been carried out successfully and preserving the quality of the cover audio, in addition, it was possible to recover 100% of the hidden characters.

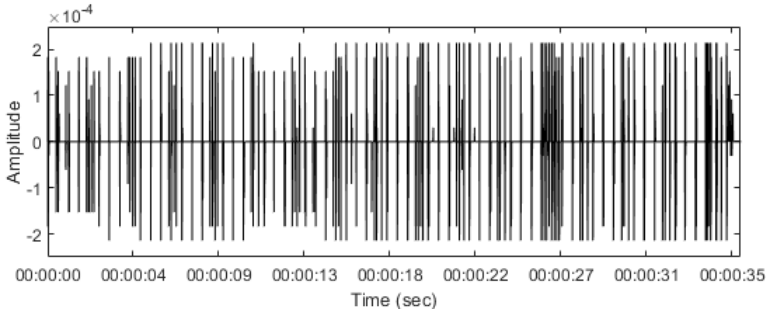


Fig. 7. Audio visible changes in time domain after message insertion.

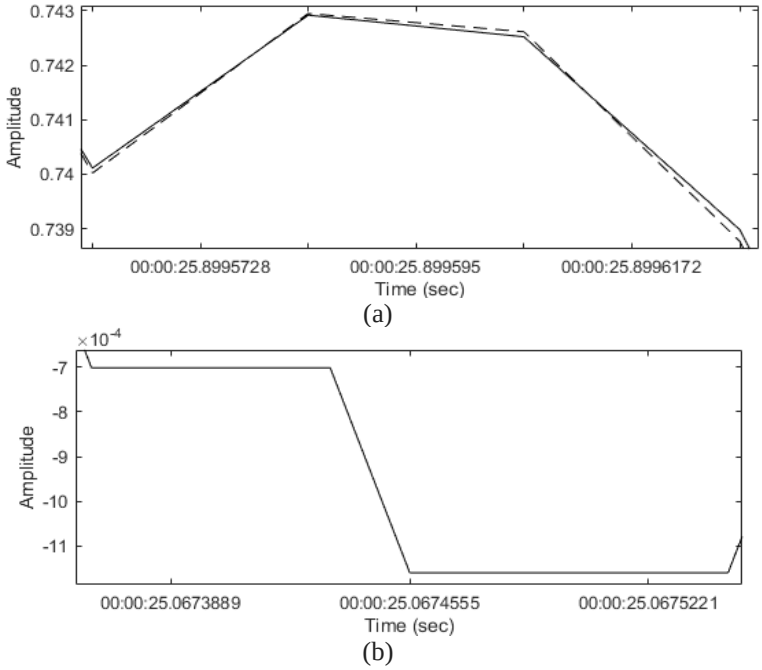


Fig. 8. (a) High frequency segment of the original (continuous line) and stego (dashed line) audios where there is a change. (b) Low frequency segment where there is no insertion.

It is worth noting that relatively small amounts of characters have been hidden. However, this is due to the use of audio files lasting only a few seconds. The use of steganography in longer audio files would allow the transport of much longer texts, supported by a different selection of initial frequencies and energy levels.

Regarding the robustness of the method, the proposed steganography procedure is not robust against lossy compression attacks such as MP3 compression, since a key file is used and due to the large number of parameters that can be used in the compression, it is very difficult to match the right points where information could be stored to recover, in addition to the nature of lossy compression that generates changes in the audio data, also, the alteration of the audio with white noise, due to the a wide range of frequencies that it covers, it alters the audio file making the hidden information impossible to recover, since it even affects high frequencies, which is where the secret message has been hidden.

6 Conclusions

The analysis of the results shows that the proposed method for hiding bits in the frequency domain is effective and manages to preserve the quality of the original audio. The choice of an appropriate initial frequencies and energy levels, as well as the use of resistant spectral peaks, contribute to the effectiveness and likely preservation of the hidden message.

The evaluation metrics, such as MSE, SNR, PSNR, and SSIM, indicate good performance in data hiding for the three audio files analyzed. Despite some variations in the metrics between the files, overall, the results are consistent and satisfactory.

It is worth noting that the main objective of this study was to propose and analyze an audio steganography method in the frequency domain by hiding information in the spectrogram, without addressing attack robustness or compression. Future research could focus on analyzing the robustness of the proposed method against different types of attacks or compressions.

In summary, this study demonstrates that audio steganography in the frequency domain is a viable and effective technique for hiding information in audio files without compromising their quality, opening up new possibilities for information security and communication in digital environments.

References

1. Jian, C., Wen, C., Rahman, N., Hamid, I.: Audio Steganography with Embedded Text. IOP Conference Series, 226, 012084 (2017) doi: 10.1088/1757-899x/226/1/012084.
2. Febryan, A., Purboyo, T. Saputra, R.: Steganography methods on text, audio, image and video: A survey. International Journal of Applied Engineering Research, 12, pp. 10485–10490 (2017)
3. Johri, P., Mishra, A., Das, S. Kumar, A.: Survey on steganography methods (text, image, audio, video, protocol and network steganography). In: 3rd International Conference on Computing for Sustainable Global Development (INDIACom), pp. 2906–2909 (2016)
4. Hussain, M., Rafat, K.: Enhanced Audio LSB Steganography for Secure Communication. International Journal of Advanced Computer Science and Applications, 7(1), pp. 340–347 (2016)
5. Zainab, N., Ban, N.: Image and audio steganography based on indirect LSB. Kuwait Journal of Science, 48(4), pp. 1–12 (2021)

6. Geleta, M., Punti, C., McGuinness, K., Pons, J., Canton, C., Giro-i-Nieto, X.: PixInWav: Residual Steganography for Hiding Pixels in Audio. In: ICASSP 2022 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 2485–2489 (2022)
7. Abood, E., Abdullah, A., Sibahee, M., Abduljabbar, Z., Nyangaresi, V., Kalafy, S. Ghrabta, M.: Audio steganography with enhanced LSB method for securing encrypted text with bit cycling. Bulletin of Electrical Engineering and Informatics, 11(1), pp. 185–194 (2022)
8. Pandey, M., Parmar, G., Patsariya, S.: An Effective Way to Hide the Secret Audio File Using High Frequency Manipulation. In: Communications in computer and information science. Springer Science+Business Media, pp. 125–130 (2011)
9. Lerch, A.: An Introduction to Audio Content Analysis. Wiley (2012)
10. National Instruments: Understanding FFTs and Windowing (2023) https://download.ni.com/evaluation/pxi/Understanding.
11. Müller, M.: Fundamentals of Music Processing. Springer (2015)
12. Lerch, A.: Audio Content Analysis. (2023) https://www.audiocontentanalysis.org/
13. OpenAI: ChatGPT (2023) https://chat.openai.com/.

ISSN 1870-4069

Integrating Radiograph Normalization Preprocessing and Discriminative Feature Selection for Efficient and Automated Pneumonia Detection

Salvador E. Ayala-Raggi, Angel Ernesto Picazo-Castillo, Aldrin Barreto-Flores, José Francisco Portillo-Robledo

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

{saraggi,a.picazo.2505}@gmail.com,{aldrin.barreto,
francisco.portillo}@correo.buap.mx

Abstract. This paper introduces a technique for the detection of Viral Pneumonia using automatic localization, followed by pose and scale normalization of the specific region of interest (lungs) in chest radiographs. This method employs PCA and weighted K-NN regression. Our proposed approach includes estimating corner positions within the region of interest through interpolation, then mapping the image within that identified region onto a standardized fixed-size template. The primary goal is to achieve uniformity among training images in terms of position, angular pose, scale, and contrast, effectively aligning them. Subsequently, the eigenfaces method is employed to extract a reduced set of principal features from the normalized images. Among these PCA-derived features, those exhibiting the highest between-class discrimination capability are chosen using the Fisher criterion. Our results highlight the effective synergy achieved by integrating our lung region alignment technique with the meticulous selection and weighting of the most discriminative PCA features. This synergy is sufficient to achieve peak accuracies of 95.6% and 97.3% in classifying Viral Pneumonia radiographs using conventional classifiers, specifically weighted K-NN and MLP, respectively. Notably, our findings demonstrate that, in contrast to convolutional neural networks, a simpler technique can yield comparable classification results.

Keywords: Image classification, fisher discriminant, viral pneumonia, K-Nearest neighbors, multilayer perceptron.

1 Introduction

Pneumonia is a lung disease caused by bacteria and viruses; a person can be infected through the air, saliva, or mucus. Furthermore, children and the

elderly are at a higher risk of contracting it, according to [6]. Currently, various methods exist for detecting this disease, such as tomography, chest X-rays, and ultrasounds. However, tomography is more expensive than an X-ray, and ultrasound is not always available or affordable. Hence, X-rays prove to be a more common detection method [1,2,23,38,27].

Presently, there exist datasets accessible containing labeled radiographs, which can be employed to train diverse machine learning algorithms [2]. The establishment of these repositories has been a cooperative endeavor involving establishments and domain-specialist medical professionals [33,30,26]. Nevertheless, the obstacle lies in the absence of consistency in the area of interest (pulmonary region) within these images. Some radiographs encompass redundant or unrelated data for categorization, such as supplementary bodily components or objects obscuring the thoracic area. This can negatively impact the precision metrics of categorization algorithms [5,10].

In this work, we aim to demonstrate the hypothesis that aligning the region of interest in both the training images and the test image, such that the anatomical structures within the lungs are positionally consistent across all images, can enable simple and conventional classification methods like K-NN or MLP to achieve better accuracy results, provided that a reliable feature reduction method like PCA is employed in conjunction with a feature selection process based on their discriminatory capability.

To this end, we propose applying two consecutive processes. The first process involves the detection and normalization of the lung region, ensuring that the images within the lung region exhibit the same alignment, location, scale, and improved contrast as much as possible. In the second process, the "Eigenfaces" method (PCA) will be applied to the aligned regions to obtain a reduced set of statistically independent features. Finally, based on the Fisher criterion [35], we propose performing a selection of the features that best discriminate between classes. Using this set of optimal features and a traditional classifier such as K-NN or MLP, the classification accuracy will be measured.

This work is divided into four parts. Part 1 discusses the related work and the utilized database. Part 2 describes and presents the algorithm called "Lung Finder Algorithm" (LFA) for the normalization procedure. Part 3 presents the theory of "Eigenfaces" and Fisher linear discriminant, as well as the feature weighting applied in our analysis for the normalized image features. Finally, in Part 4, the precision metrics are compared when utilizing our methodology with the weighted K-Nearest Neighbors (K-NN) classifier and the Multilayer Perceptron [9].

1.1 Related Work

Currently, various methodologies have been developed for classification of chest radiographs, as evidenced in previous studies [17,13,28,11,4,31,37]. These methodologies make use of deep learning algorithms or traditional machine learning classifiers [7,8], and have reported high levels of classification accuracy, greater than 96%. However, the architectures employed in these algorithms still

Table 1. Comparison of the different preprocessing methods from related works.

Authors	Image Normalization	Features selection	Classifier	Accuracy
Changawala et al., (2021)	Not Used	Not Used	MLP (Involution)	98.31%
Liu et al., (2023)	Not Used	Used	SVM	100%
Park et al.,(2015)	Not Used	Used	SVM	93.5%
Lv et al.,(2022)	Not Used	Used	KNN	96.14%
Gadermayr et al., (2017)	Used	Not Used	SVM	97%
Kociolek et al.,(2020)	Used	Used	SVM	96%

face challenges in achieving a reliable classification of COVID-19 [32], as their accuracy decreases when tested with other datasets different from those used for training. This raises the need of exploring new proposals for normalizing and aligning the lungs region before classifying, instead of just facing the problem by training classifiers like CNNs with a large number of of different datasets, to cope with the bias imposed by a particular one.

Efficient non CNN-based works have been proposed too, as in [3], where a Multilayer Perceptron (MLP) and an architecture based on image involution were used, which proposes kernels similar to CNNs but shares their weights dynamically in all dimensions, thus reducing the number of multiplications necessary for the calculations. The former obtained a maximum classification accuracy of 98.31%. Feature selection has proven to be effective in increasing classification accuracy in other works, as observed in a study on [20] which used support vector machines to recognize the orbit axis of the sensors, as in another study [29] where it was also possible to classify the frequencies of an encephalogram. Furthermore, in a work carried out by Chengzhe et al. [21], the K-NN algorithm was applied successfully.

Several studies have shown that image normalization improves classification results. In a study on kidney radiographs [10], the best results were obtained using CNN and image normalization techniques. Also, in another [19] work, different normalization techniques were used on different types of radiographs to improve image classification. It is important to highlight that the results of our work are not intended to devalue CNNs in image classification, but rather to present an alternative option, and to demonstrate that image alignment and a proper feature selection technique can produce results comparable to the most commonly used algorithms. in the state of the art. In the table 1 we show the comparison of the different pre-processing methods used in some published works [3,20,29,21,10,19].

1.2 Data Set of Radiographic Images

The database used for this work was "COVID-19 Radiography Database" [4,31] from kaggle. This data set was selected because it has been used in other similar works[25,15]. The content of this data set is 6012 images already labeled as pulmonary opacity (other lung diseases), 1345 as viral pneumonia, 10192 as normal, and finally 3616 as COVID-19.

2 Overview of the Lung Finder Algorithm (LFA)

The goal of this algorithm is to locate the lungs in the radiographs, and it consists of a training and testing stage, as shown in figure 1. During the training stage, 400 images from the Pneumonia, COVID-19, and Normal classes were randomly selected from the data set. Histogram equalization (HE) [12,24] was applied to all images and regions of interest were manually labeled by placing 4 provisional landmarks easily located by a human user. It was agreed that two of them would be located, one in the middle of the cervical vertebrae just at the upper limit of the lungs, and the other also on the spine but below where the lung region ends. The other two provisional landmarks are forced to the user to place them on a imaginary straight line perpendicular to the spine that intersects it just in the middle of the two previous landmarks.These last two landmarks are located in the left and right sides of lung region. Finally, and by using these 4 provisional positions, we compute 4 final and permanent landmarks at the corners of the rectangular lung region. On the other hand, ten new images randomly rotated and displaced were then generated for each labeled image to increase the data set and have an *augmented dataset*. Next, a dimensionality reduction to this set of 4400 images was applied using the "Eigenfaces" method based on Principal Component Analysis (PCA) [39,18].

During the test stage, and after a contrast improvement (H.E.), a new image is projected to the "Eigenfaces" linear subspace in order to convert it to a compact few dimensions vector which is compared via euclidean distance with each of the 4000 examples contained within the augmented dataset to find k nearest neighbors $k - NN$. The landmarks associated with these k most similar images from the augmented dataset are used to estimate the 4 landmarks of the test image by interpolation. These predicted landmarks are the coordinates of the corners of the lung ROI which can be used to warp the inside region to a standard template of fixed size.

2.1 Coordinates Labeling for the LFA Training Stage

Each of the images selected for this stage requires a manual labeling where the region of interest of the lungs is delimited by a set of coordinates. These points or landmarks become the labels used by a regression weighted K-NN to predict the corner coordinates of the novel image. The coordinates the lung region are shown in figure 2, and consist of four points: Q1(x1,y1), Q2(x2,y2), Q3(x3,y3)

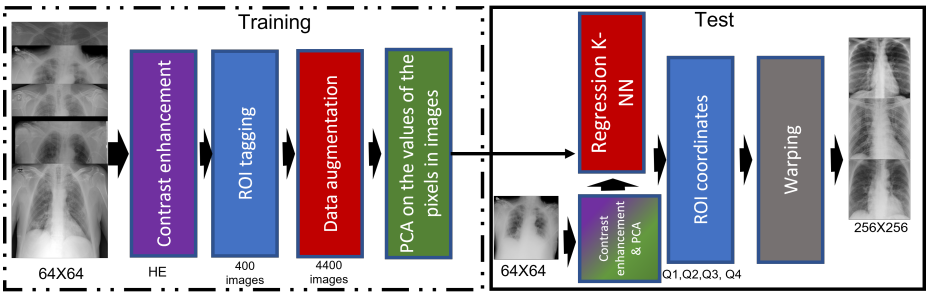


Fig. 1. Lung Finder Algorithm description. During the training phase, 400 images were tagged with their coordinates. PCA was applied to reduce the dimensionality of the images. In the testing phase, an example radiograph is provided as input, and the algorithm extracts the region of interest as the output. During the test phase, the test image is compared with its nearest neighbors to interpolate its coordinates. Finally, the algorithm outputs the extracted region of interest in a new image.

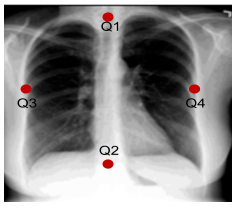


Fig. 2. Example of an array of coordinates Q1, Q2, Q3, and Q4 on a radiograph.

and Q4(x4,y4). Q1 and Q2 represent the length of the lungs, while Q3 and Q4 represent their width. In total, 400 images were labeled manually.

The labeling process is shown in figure 3. First, the Q1 point at the top of the lungs is manually located, using the spine as reference. The Q2 point is then placed at the bottom of the lungs. When the points Q1 and Q2 are placed, a straight line connecting them automatically appears, and at the midpoint of this line a perpendicular line is drawn containing the points Q3 and Q4. These last two points are constrained to be placed by the user only along the perpendicular line, and may have a different distance from the midpoint of the Q1Q2 line, due to the fact that the lungs are not symmetrical to each other.

2.2 Data Augmentation

Data augmentation is used in various machine learning tasks, such as image classification, to expand a limited database and avoid overfitting [22,34,19]. In the case of our algorithm, we have used a large dataset [4,31]. However, in order to have a set with ROI coordinates sufficiently varied we decided to generate artificial examples based on a randomly selected set, 400 images extracted from original set. The additional artificial images were generated by producing

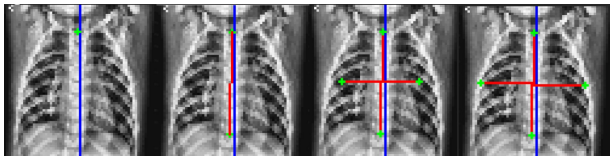


Fig. 3. Sequential placement of the points Q. First Q1 is placed, then Q2 so that Q3 and Q4 appear on the perpendicular line that crosses the midpoint of the line Q1Q2. Finally, Q3 and Q4 are adjusted.

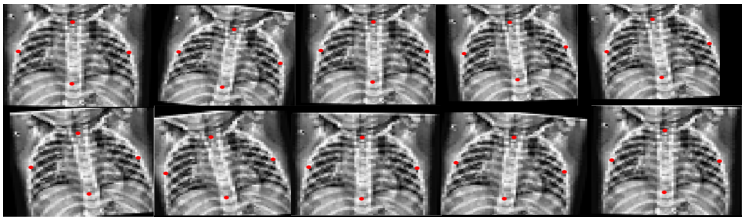


Fig. 4. Example of artificial images during data augmentation, applying translation and rotation operations.

random translations and rotations of the original images. Ten additional artificial images were created from each of the original 400, resulting in a total of 4400 images. First, it was necessary to define the range of operations on the images. For rotation, we set a range of -10 to 10 degrees, suggested by [31], and for translation a range from -5 to 5 pixels. These values were calculated by analyzing the coordinates of the 400 manually labeled images. In summary, the LFA training set contains 4400 images where the coordinates of the landmarks are normal distributed. Figure 4 shows an example of artificial images with their corresponding landmarks.

2.3 Estimating the Corner Coordinates of the Lung Region by Regression

As shown in figure 1, in the test stage a new image is introduced from which it is desired to obtain its region of interest. Contrast enhancement and feature reduction are automatically applied to the test image by projecting it onto the "Eigenfaces". The weights obtained in this projection are used in the "weighted regression K-NN" algorithm to find the most similar neighbors in the "Eigenfaces" space, using the Euclidean distance. In order to reduce the computational cost, the calculations are performed in a 64x64 resolution.

Once the nearest neighbors have been identified, a regression is performed using the coordinates of the ROIs of these neighbors with the aim of predicting the coordinates of the lungs in the test image. For this, the regression equations (1 and 2) are used, which are applied to each coordinate, either x or y, of each

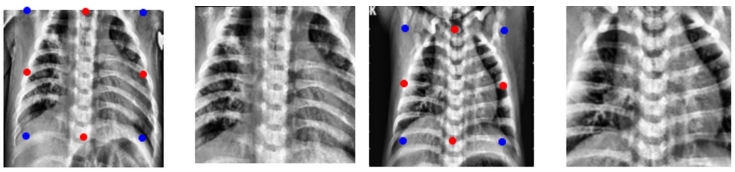


Fig. 5. Two examples of new images with their estimated ROI coordinates used to warp the inside region towards a fixed and normalized template.

Q landmark, until completing the entire set of landmarks (Q1, Q2, Q3 and Q4). The regression equations are detailed below:

$$x_i = \frac{1}{k} \sum_{i=1}^{k} x_{ni}, \tag{1}$$

$$y_i = \frac{1}{k} \sum_{i=1}^{k} y_{ni}. \tag{2}$$

2.4 Image Warping

Once the coordinates are obtained through regression, a Warping operation [36] is used to extract the region of interest. In figure 5, examples of different test radiographs from the data set are presented along with their automatically estimated ROI coordinates of provisional landmarks (red dots). The calculated coordinates are geometrically transformed to obtain the corners of the ROI (final landmarks depicted as blue dots) that are used in the Warping operation towards a standard fixed size template. On the right side of each image the normalized image resulting from the LFA is shown.

3 Feature Reduction and Selection

After using the LFA on all radiographs in the data set to extract all regions of interest, these new images undergo additional preprocessing before being processed by a classifying algorithm. For our work, we propose the use of [39,18] Eigenfaces as a feature reduction method. In addition, we incorporated a statistical analysis of these features using Fisher's linear discriminant in order to preserve only the most discriminating features and weighing each of them according to their power of discrimination between classes. Together these two methods ensure obtaining a reduced number of discriminant features suitable for efficient classification using traditional classifiers.

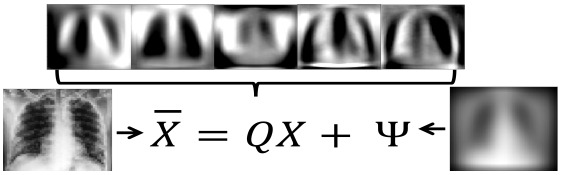


Fig. 6. Reconstructed image (left) is computed as a linear combination of the columns of matrix Q (in the middle) plus the mean image (right).

3.1 Eigenfaces for Dimensionality Reduction

Eigenfaces [39,18] is based on principal component analysis (PCA) and its objective is to reduce the dimensionality of the images in the [16] dataset. Because each pixel becomes a dimension or feature to be analyzed, processing 256x256 images can be time consuming. On the other hand, a large number of features, in comparison to a smaller number of training examples, could produce missclassification when euclidean distance based approaches as k-NN are used.

The resulting eigenfaces are sorted according to the greater variances of the training set, and can be used to reconstruct every image in the training set as a linear combination of them. Because the greatest amount of variance is concentrated in the first eigenfaces, we can use only a few number of them to efficiently represent all the training images and even novel ones. Thus, every normalized image from the training set can be represented with this compact set of features. Figure 6 shows the Eigenfaces equation, and the matrix Q which columns are the Eigenfaces. The $eigenfaces$ method works better and is capable of concentrating more variance in a less number of eigenfaces when training images are more similar. In our case, the normalized images are more similar to each other than the original images from the dataset. For this reason, the number of useful PCA features is necessarily reduced when using the proposed LFA.

3.2 Using the Fisher Discriminant to Reduce the Number of Useful Features

Fisher discriminant criterion also known as Fisher ratio FR has been used in Linear Discriminant Analysis for finding a linear projection of features that maximizes the separation between classes. Typically, only one important feature survives this process in two classes problems. However, since the PCA features are to some degree independent, we can use, in a naive fashion, the fisher ratio as a measure of separation between classes for a each feature.

This process is done by evaluating each feature individually, and making sure that the means of the observations in each class are as far apart as possible, while the variances within each class are as small as possible. Using this analysis, it is possible to select a number greater than 2 of those features obtained by the Eigenfaces method that best discriminate the classes in the data set [35].

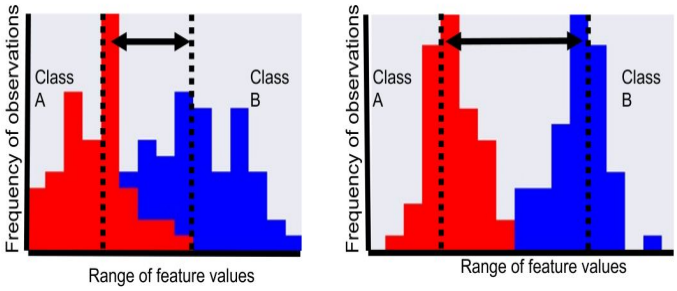


Fig. 7. Example of frequency distributions for each class. The discriminative capability of a feature can be visually assessed by the separation between the means of the histograms. The pair of histograms on the right shows a greater separation, indicating higher discrimination between classes. Conversely, the pair of histograms on the left exhibits lower discrimination.

The FR has been used in works such as the one mentioned in [14], and we denoted it as J. The FR formula is found in equation 3:

$$J_i = \frac{(\mu_{ic_0} - \mu_{ic_1})^2}{\sigma^2_{ic_0} + \sigma^2_{ic_1}}. \tag{3}$$

3.3 The Fisher Ratio as a Weight for each Feature

We propose to use the FR value as a weigh for each feature, in such a way that those features that possess a greater capacity for discrimination are amplified.

As a first step we standardize all selected features in order to give them a uniform relevance. Then, we calculate $\rho_K = \sqrt{J_K}$ for each feature k. Next, we normalize ρ_K as shown in equation 4:

$$\varrho_k = \frac{\rho_k}{\sum_{i=1}^{k} \rho_i}. \tag{4}$$

Finally, each ϱ_k is used to weigh all the standardized observations for the feature k.

4 Experiments Setup

In this work, the weighted K-NN and MLP algorithms were used for classification. Several experiments were conducted to compare the impact of different image preprocessing and feature enhancement algorithms on classification accuracy. The algorithms used in the training and testing stages included LFA for image normalization and preprocessing, Eigenfaces for dimensionality reduction, FR for selection of the best features, and W for

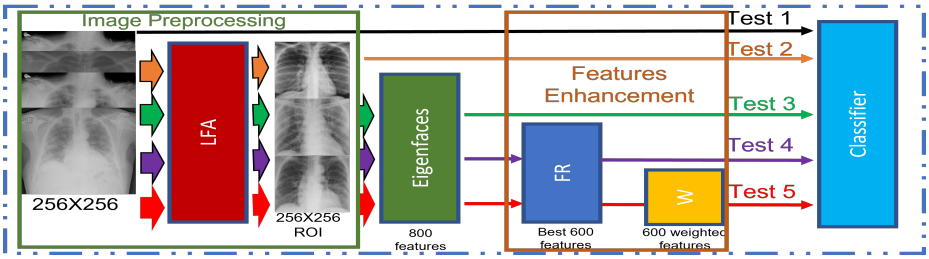


Fig. 8. Graphical representation of the different experiments conducted in image preprocessing. Each arrow represents a sequence of algorithms that may include image preprocessing or feature enhancement. A classification accuracy value is calculated for each arrow.

Table 2. Results of the Weighted K-NN and the MLP for the experiments using different preprocessing methods.

Classifier	E1	E2	E3	E4	E5
Weighted K-NN	82%	88%	88.3%	92.3%	95.6%
MLP	86%	90.8%	91%	93%	97.3%

weighting the features based on their discriminative capacity between classes. These two algorithms together aim to improve the discriminative ability of the features across classes. A total of five experiments were conducted for each classifier, which are described in Figure 8.

A total of 1300 COVID-19 images and 1300 normal images, all of size 256x256 pixels, were used. The region of interest was extracted from these images using the LFA algorithm, forming a bank of normalized images. The images were divided into 2000 training images, with 1000 from each class. For the testing phase, 600 images were selected, with 300 from each class. In experiments 1 and 2, 65,536 pixels, which constitute all the pixels of the images, were used. For experiments 3, 4, and 5, 600 features were employed.

For the MLP topology, 4 hidden layers with 120 neurons each and a single neuron in the output layer were utilized. The training was conducted for 100 epochs.

5 Experimental Results

Various values were tested for the parameter K in the weighted K-NN, and it was determined that the optimal value is 11. Conversely, experiments were conducted with various topologies and number of epochs in the MLP, yet no notable enhancements in classification precision were detected. The classification accuracy results for all experiments of each classifier are displayed in Table 2.

Additional tests were conducted in Experiment 5, varying the number of features for both classifiers. However, it was found that 600 is the optimal

Table 3. Results of Weighted K-NN and MLP for cross-validation.

Classifier	Test 1	Test 2	Test 3	Test 4	Test 5	Mean	Std
Weighted K-NN	95.6%	94.8%	95%	94%	95.2%	94.92%	0.593
MLP	97%	97.3%	96.4%	97%	96.8%	96.9%	0.331

number of features for both classifiers. Furthermore, Experiment 5 underwent cross-validation to demonstrate the consistency of the proposed set of algorithms in this work. Table 3 displays the results of the 5 tests, along with the average and standard deviation for each classifier.

6 Discussion of Results

For both classifiers, the following statements can be made regarding the experiments conducted in image preprocessing:

-Experiment one, where images undergo no preprocessing, generally displays the worst results.

-Experiment two illustrates that image normalization improves results compared to experiment one.

-In experiment three, where an image representation is projected onto the Eigenfaces space, no noteworthy enhancement is discernible.

-Experiment four highlights the importance of feature selection that effectively separates classes using FR, resulting in improved accuracy.

-Experiment five showcases the effectiveness of our algorithm sequence, which includes image normalization, feature selection, and weighting, yielding the best results.

Furthermore, the results exhibit robust consistency with minimal variability during cross-validation. Finally, the MLP achieved accuracy results that can compete with other state-of-the-art algorithms for classifying chest X-ray images.

7 Conclusions

In this paper, we have introduced a technique for the automatic detection and normalization of the Region of Interest (ROI) in chest radiographs. This approach is complemented by a feature selection method grounded in Fisher's criterion (FR) and utilizes PCA for automated COVID-19 detection. Through this approach, a reduced set of highly discriminative features is extracted. The outcomes underscore that the combination of both ROI alignment and feature selection processes leads to a significant improvement in classification accuracy. Notably, this improvement is evident when utilizing conventional classifiers such as weighted K-NN and MLP. These enhanced features demonstrate a notably superior classification capacity in comparison to the original pixel values. The reliability of the reported results is further solidified by the incorporation of cross-validation techniques in their acquisition.

The contributions of this study encompass a method for normalizing the ROI in lung images and a technique for selecting highly discriminative features using FR. Our approach achieves accuracy values that compete with other state-of-the-art works employing CNN-based techniques.

For future work, the ROI normalization technique can be applied to other databases and for the detection of other lung diseases. Additionally, the feature selection and weighting approaches can be tested to enhance the accuracy of other classification algorithms. Finally, we provide the link to download and use the LFA code (https://github.com/picazo07/LFA.git).

References

1. Alzahrani, S. A., Al-Salamah, M. A., Al-Madani, W. H., Elbarbary, M. A.: Systematic review and meta-analysis for the use of ultrasound versus radiology in diagnosing of pneumonia. Crit Ultrasound Journal, vol. 9, no. 1, pp. 1–11 (2017)
2. Amatya, Y., Rupp, J., Russell, F. M., Saunders, J., Bales, B., House, D. R.: Diagnostic use of lung ultrasound compared to chest radiograph for suspected pneumonia in a resource-limited setting. Int J Emerg Med, vol. 11, no. 8, pp. 1–5 (2018)
3. Changawala, V., Sharma, K., Paunwala, M.: Averting from convolutional neural networks for chest x-ray image classification. In: 2021 IEEE International Conference on Signal Processing, Information, Communication and Systems (SPICSCON). pp. 14–17 (2021)
4. Chowdhury, M. E. H., Rahman, T., Khandakar, A., Mazhar, R., Kadir, M. A., Mahbub, Z. B., Islam, K. R., Khan, M. S., Iqbal, A., Emadi, N. A., Reaz, M. B. I., Islam, M. T.: Can AI help in screening viral and COVID-19 pneumonia? IEEE Access, vol. 8, pp. 132665–132676 (2020)
5. Cleophas, T., Zwinderman, A.: Machine Learning in Medicine: Part Two. Machine Learning in Medicine, Springer Netherlands (2013)
6. in Data, O. W.: Covid-19 data explorer. https://ourworldindata.org/explorers/coronavirus-data-explorer (2022), accessed: February 2022
7. Do, T.-N., Le, V.-T., Doan, T.-H.: Svm on top of deep networks for covid-19 detection from chest x-ray images. Journal of information and communication convergence engineering, vol. 20, pp. 219–225 (2022)
8. El-kenawy, E.-S., Mirjalili, S., Ibrahim, A., Alrahmawy, M., Elsaid, M., Mounir, R., Eid, M.: Advanced meta-heuristics, convolutional neural networks, and feature selectors for efficient covid-19 x-ray chest image classification. IEEE Access, vol. 9, pp. 36019 – 36037 (2021)
9. Ertel, W., Black, N.: Introduction to Artificial Intelligence. Undergraduate Topics in Computer Science, Springer International Publishing (2018)
10. Gadermayr, M., Cooper, S. S., Klinkhammer, B., Boor, P., Merhof, D.: A quantitative assessment of image normalization for classifying histopathological tissue of the kidney. In: Pattern Recognition: 39th German Conference, GCPR 2017. pp. 3–13. Springer (2017)
11. Gazda, M., Plavka, J., Gazda, J., Drotar, P.: Self-supervised deep convolutional neural network for chest x-ray classification. IEEE Access, vol. 9, pp. 151972–151982 (2021)
12. González, R., Woods, R.: Digital Image Processing, Global Edition. Pearson Education (2018)

13. Hamza, A., Attique Khan, M., Wang, S.-H., Alhaisoni, M., Alharbi, M., Hussein, H. S., Alshazly, H., Kim, Y. J., Cha, J.: Covid-19 classification using chest x-ray images based on fusion-assisted deep bayesian optimization and grad-cam visualization. Frontiers in Public Health, vol. 10, pp. 1–17 (2022)
14. Ibis, E.: Sistema de aprendizaje automático para la detección de neumonía. Master's thesis, Benemérita Universidad Autónoma de Puebla, Puebla, México (2022)
15. Islam, N., Ebrahimzadeh, S., Salameh, J.-P., Kazi, S., Fabiano, N., Treanor, L., Absi, M., Hallgrimson, Z., Leeflang, M. M., Hooft, L., van der Pol, C. B., Prager, R., Hare, S. S., Dennie, C., Spijker, R., Deeks, J. J., Dinnes, J., Jenniskens, K., Korevaar, D. A., Cohen, J. F., Van den Bruel, A., Takwoingi, Y., van de Wijgert, J., Damen, J. A., Wang, J., McInnes, M. D.: Thoracic imaging tests for the diagnosis of covid-19. Cochrane Database Syst Rev, vol. 3, no. 3, pp. 1–145 (2021)
16. Jolliffe, I.: Principal Component Analysis. Springer Series in Statistics, Springer (2002)
17. Khan, A., Khan, S., Saif, M., Batool, A., Sohail, A., Khan, M.: A survey of deep learning techniques for the analysis of covid-19 and their usability for detecting omicron. Journal of Experimental and Theoretical Artificial Intelligence, vol. 2023, pp. 1–43 (2023)
18. Kirby, M., Sirovich, L.: Application of the karhunen-loeve procedure for the characterization of human faces. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, no. 1, pp. 103–108 (1990)
19. Kociolek, M., Strzelecki, M., Obuchowicz, R.: Does image normalization and intensity resolution impact texture classification?. Computerized Medical Imaging and Graphics, vol. 81, pp. 1–17 (2020)
20. Liu, W., Zheng, Y., Zhou, X., Chen, Q.: Axis orbit recognition of the hydropower unit based on feature combination and feature selection. Sensors, vol. 23, no. 6, pp. 1–18 (2023)
21. Lv, C., Lu, Y., Lu, M., Feng, X., Fan, H., Xu, C., Xu, L.: A classification feature optimization method for remote sensing imagery based on fisher score and mrmr. Applied Sciences, vol. 12, pp. 1–19 (2022)
22. Mikołajczyk-Bareła, A., Grochowski, M.: Data augmentation for improving deep learning in image classification problem. In: 2018 international interdisciplinary PhD workshop (IIPhDW). pp. 117–122. IEEE (2018)
23. Moberg, A., Taléus, U., Garvin, P., Fransson, S.-G., Falk, M.: Community-acquired pneumonia in primary care: Clinical assessment and the usability of chest radiography. Scandinavian journal of primary health care, vol. 34, pp. 1–7 (2016)
24. Moeslund, T. B.: Introduction to Video and Image Processing: Building Real Systems and Applications. Undergraduate Topics in Computer Science, Springer London (2012)
25. Muljo, H. H., Pardamean, B., Purwandari, K., Cenggoro, T. W.: Improving lung disease detection by joint learning with covid-19 radiography database. Communications in Mathematical Biology and Neuroscience, vol. 2022, no. 1, pp. 1–24 (2022)
26. Mustafa Ghaderzadeh, M. A., Asadi, F.: X-ray equipped with artificial intelligence: Changing the COVID-19 diagnostic paradigm during the pandemic. BioMed research international, vol. 2021, pp. 1–16 (2021)
27. Niederman, M. S.: Community-acquired pneumonia. Annals of Internal Medicine, vol. 163, no. 7, pp. 1–16 (2015)

28. Nillmani, Sharma, N., Saba, L., Khanna, N. N., Kalra, M. K., Fouda, M. M., Suri, J. S.: Segmentation-based classification deep learning model embedded with explainable ai for covid-19 detection in chest x-ray scans. Diagnostics, vol. 12, no. 9, pp. 1–32 (2022)
29. Park, S.-H., Lee, S.-G.: A method of feature extraction on motor imagery eeg using fld and pca based on sub-band csp. Journal of KIISE, vol. 42, pp. 1535–1543 (2015)
30. Qin, C., Yao, D., Shi, Y., Song, Z.: Computer-aided detection in chest radiography based on artificial intelligence: a survey. BioMedical Engineering OnLine, vol. 17, no. 1, pp. 1–23 (2018)
31. Rahman, T., Khandakar, A., Qiblawey, Y., Tahir, A., Kiranyaz, S., Abul Kashem, S. B., Islam, M. T., Al Maadeed, S., Zughaier, S. M., Khan, M. S., Chowdhury, M. E.: Exploring the effect of image enhancement techniques on covid-19 detection using chest x-ray images. Computers in Biology and Medicine, vol. 132, pp. 1–16 (2021)
32. Ridzuan, M., Bawazir, A. A., Navarette, I. G., Almakky, I., Yaqub, M.: Self-supervision and multi-task learning: Challenges in fine-grained covid-19 multi-class classification from chest x-rays. In: Annual Conference on Medical Image Understanding and Analysis. pp. 234–250. Springer (2022)
33. Salvatore, C., Interlenghi, M., Monti, C. B., Ippolito, D., Capra, D., Cozzi, A., Schiaffino, S., Polidori, A., Gandola, D., Alì, M., Castiglioni, I., Messa, C., Sardanelli, F.: Artificial intelligence applied to chest x-ray for differential diagnosis of covid-19 pneumonia. Diagnostics, vol. 11, no. 3, pp. 1–12 (2021)
34. Shorten, C., Khoshgoftaar, T. M.: A survey on image data augmentation for deep learning. Journal of Big Data, vol. 6, no. 1, pp. 1–48 (2019)
35. Silva, T. S.: An illustrative introduction to fisher's linear discriminant. https://sthalles.github.io/fisher-linear-discriminant/ (2019)
36. Szeliski, R.: Computer Vision: Algorithms and Applications. Springer-Verlag (2010)
37. Talaat, A., Yousri, D., Ewees, A., Al-qaness, M. A. A., Damaševičius, R., Elsayed Abd Elaziz, M.: Covid-19 image classification using deep features and fractional-order marine predators algorithm. Scientific reports, vol. 10, pp. 15364 (2020)
38. Ticinesi, A., Lauretani, F., Nouvenne, A., Mori, G., Chiussi, G., Maggio, M., Meschi, T.: Lung ultrasound and chest x-ray for detecting pneumonia in an acute geriatric ward. Medicine, vol. 95, no. 27, pp. 1–7 (2016)
39. Turk, M., Pentland, A.: Eigenfaces for Recognition. Journal of Cognitive Neuroscience, vol. 3, no. 1, pp. 71–86 (1991)

ISSN 1870-4069

Feature Analysis for Stress Detection on Text Posts

Erick Barrios-González

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

erick.barrios@alumno.buap.mx

Abstract. This paper examines stress detection in social networks, specifically focusing on the Dreaddit corpus. The study utilizes a Naive Bayes classifier with tagging from the Spacy tool. Two grid searches are performed to identify optimal features for the classifier. The evaluation of results using the F_1 metric shows superior performance compared to other Naive Bayes models. Features based on n-grams, POS tagging, lemma, and stem were analyzed. Useful features were found from an approach where their frequency of occurrence in the corpus was evaluated, and, likewise, other features were discarded.

Keywords: Stress detection, naive bayes, N-grams.

1 Introduction

Stress is defined as the reaction to pressures, existing demands, and future demands [1]. It is the natural response of the human being to situations of fear, tension, or danger [8]. Excessive stress can be harmful to the mind and body. Stress is a normal part of our lives, and in small amounts, it can have positive effects. However, excessive stress can cause negative alterations in our organism and mind [9], making the individual prone to physical and psychological illnesses.

Every day, social media networks are becoming more common in our daily lives, and it is increasingly normal for people to continuously turn to social media platforms like Twitter and Reddit to share their feelings and express their stress. This interest in sharing feelings on social media is the main reason why analyzing texts posted on social media is useful for stress detection.

The main objective of stress detection on social media is to determine which users may be suffering from stress, to have more information about people with this condition, or to implement solutions that can help individuals with their stress levels.

Reddit is a social media platform where users post in specific topic communities (subreddits), and other users comment and vote on these posts. The extensive nature of these posts makes Reddit an ideal source of information for studying the nuances of phenomena like stress [2].

The forthcoming sections are arranged as follows: Related work, Speech emotion recognition algorithm, experiments, and conclusions.

2 Related Work

Stress detection in social media, especially on Reddit, is not as well-explored as depression detection. Therefore, there are few corpora available for this task. The most widely used corpus for stress detection is Dreaddit [2]. However, some works create their corpora for this task, such as [10, 11].

Dreaddit is a corpus collected from Reddit with the purpose of facilitating the development of models for stress detection. Dreaddit consists of a set of posts annotated by humans as either stress or non-stress. This corpus collects posts from various subreddits where stress-related topics could be discussed [2]. Additionally, this corpus provides an extensive set of features, primarily based on lexical diversity using the categories of Linguistic Inquiry and Word Count (LIWC). The posts in this corpus range from 3 to 300 words, with the majority of posts being above 26 words.

Several works address stress detection using the Dreaddit corpus. The creators of Dreaddit evaluated various baseline models and obtained the best result (BERT-base) with an F_1 score of 0.8065. On the other hand, [12] evaluated multiple models and achieved an F_1 score of 0.84 with the MentalRoBERTa^{FT} model (which is the model from [3] with features from [12]), while [3] achieved an F_1 score of 0.819 with MentalRoBERTa.

BERT-based models have achieved the best results, with an F_1 score above 0.8. However, several approaches have achieved values between 0.75 and 0.80 in F_1 score. For example, [12] implements two Bayesian models (Bernoulli NB and Multinomial NB) with F_1 scores of 0.75 and 0.76, respectively. Additionally, [6] and [13] apply logistic regression algorithms, obtaining F_1 scores between 0.77 and 0.7980. Furthermore, [13] implements a Random Forest classifier, resulting in an F_1 score of 0.78.

In the literature on stress detection in social media, Bayesian approaches have not been fully explored. For instance, [12] implements two Naive Bayes models with features based on their proposed Monte Carlo Tree Search, which allows for targeted keyword searching. Another study [12] implements two Naive Bayes models with TFIDF and BERT-based features, with F_1 scores below 0.69.

2.1 Contribution

In this work, we propose to explore features for a Naive Bayes classifier, such as publication time, subreddit in which the post was made, n-grams for tokenized text, n-grams for POS tagging, n-grams for lemmatization, and n-grams to stem words. Furthermore, with this exploration, we aim to identify features that can be used for training in other models.

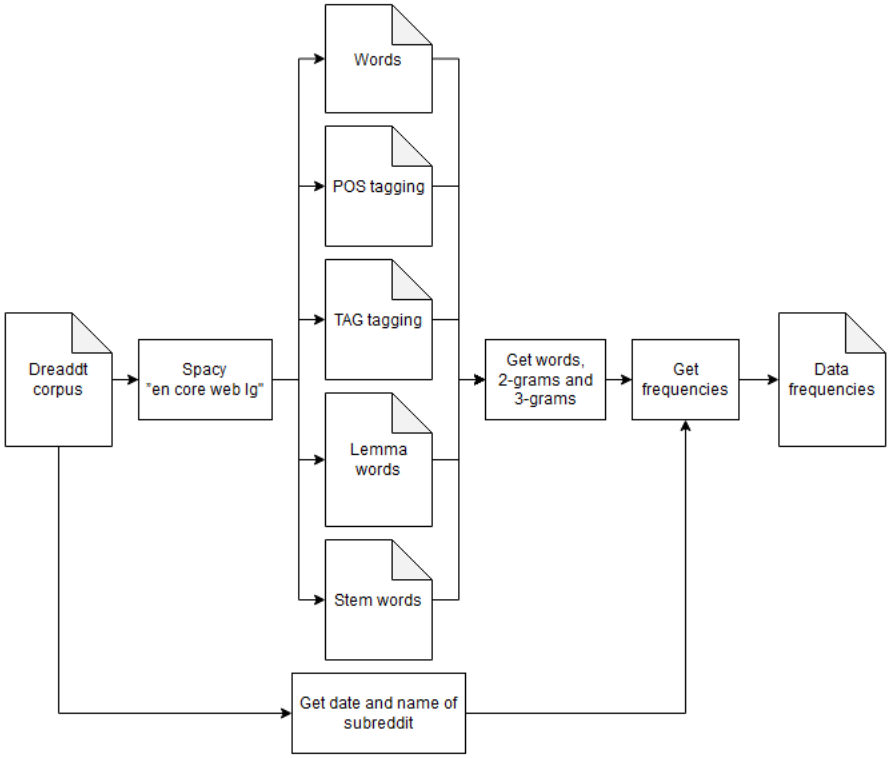


Fig. 1. Pre-processing used to obtain the features and data frequencies.

3 Stress Detection Algorithm

In this section, the process that was carried out to perform stress detection will be described, the following points will be described: pre-processing, feature selection, and classification with Naive Bayes algorithm.

3.1 Pre-processing

The pre-processing involved tokenizing the words in each post, POS tagging, lemmatizing, and stem words. Additionally, only the hour was extracted from the publication dates, and the subreddit name in which the post was made was extracted. Later, 2-grams and 3-grams were created from the tokenized words, POS tags, lemmatized words, and stem words, subsequently, the frequency of appearance of each n-gram and each word was obtained, the frequency in which each word or n-gram appears will be used to build the Naive Bayes model (calculating the probabilities). In Fig. 1 the previously mentioned process is observed.

The corpus pre-processing was programmed with Python 3.9, and the Spacy tool (using the "en_core_web_lg" pipeline for english, this "lg" version is the most complete that spacy can offer us for labeling tasks).

3.2 Feature Selection

To find the best features, a grid search has been implemented considering different combinations between different types of features. The combinations are made considering combinations between 7 types of data (Timestamp, subreddit, words, POS labels, TAG labels, lemmas and stems), in total there are 4096 combinations, that are the product of the lists of characteristics shown below:

- Timestamp (2 features): "social timestamp" (consider the time and date of publication), "without timestamp" (no date and time).
- Subreddit (2 features): "Name subreddit" (probability per name), "without subreddit" (name is not considered).
- Words (4 features): "One word" (probability per word), "2-grams" (probability per 2-gram), "3-grams" (probability per 3-gram), "without words" (words are not considered).
- POS labels (Simple part-of-speech tag, 4 features): "One word" (probability per label), "2-grams" (probability per 2-gram), "3-grams"(probability per 3-gram), "without POS" (POS labels are not considered).
- TAG labels (Detailed part-of-speech tag, 4 features): "One word" (probability per label), "2-grams" (probability per 2-gram), "3-grams" (probability per 3-gram), "without TAG" (TAG labels are not considered).
- Lemma (4 features): "One word" (probability per word), "2-grams" (probability per 2-gram), "3-grams" (probability per 3-gram), "without lemma" (lemma words are not considered).
- Stem (4 features): "One word" (probability per word), "2-grams" (probability per 2-gram), "3-grams" (probability per 3-gram), "without stem" (stem words are not considered).

For the grid search, combinations of 7 elements each were made (4096 combinations in total). In the Naive Bayes algorithm's main processing load lies in calculating the frequencies of occurrence during the pre-processing stage, for that reason the grid search only performed sum of probabilities to calculate the results for each combination. The approximate execution time was 2 seconds per combination, that is, about 8,192 seconds (136.53 minutes), running on a single core with a Ryzen 7 5800X processor.

Subsequently, to improve the results, other grid search was conducted to select the best TAG labels, filtering out the most relevant labels for stress detection.

In the labeling of the posts, 17 different labels were found (for TAG labels). To create combinations of these labels, combination sizes from 2 (136 combinations), 3 (680 combinations), 4 (2,380 combinations), 5 (6,188 combinations), 7 (12,376 combinations), 8 (24,310 combinations), 9 (24,310 combinations) and 10 (19,448

combinations) were considered, with the purpose of discarding tags that do not provide any information for classification. It is important to mention that combinations with more than 10 elements were not performed because the evaluation of results started to decrease.

3.3 Classification

As mentioned previously, a Naive Bayes classifier will be used for classification. To perform this classification, the probabilities were calculated in the following ways:

In equation 1, the calculation of the probability of a specific hour appearing in a class x ("Stress" and "Not stress") is shown. Here, FA represents the frequency of that hour appearing in class x, IC is the number of instances in class x, and H is the total number of hours in a day (i.e. 24 hrs.):

$$P = (FA + 1)/(IC + H). \tag{1}$$

In equation 2, the calculation of the probability of a specific subreddit appearing in a class x ("Stress" and "Not stress") is shown. FA represents the frequency of that subreddit appearing in class x, IC is the number of instances in class x, and N is the total number of subreddits considered in the corpus:

$$P = (FA + 1)/(IC + N). \tag{2}$$

In equation 3, the calculation of the probability of a specific word, tag, lemma, stem, or n-gram appearing in a class x ("Stress" and "Not stress") is shown. FA represents the frequency of that word appearing in class x, VC is the vocabulary size in class x, and VL is the vocabulary size of the specific feature being calculated (word vocabulary, POS tag vocabulary, n-gram vocabulary, etc.):

$$P = (FA + 1)/(VC + VL). \tag{3}$$

4 Experiments and Evaluation

In this section will be shown, the datasets used, the metrics used for evaluation, and the cross-validation process are described.

4.1 Dataset

The corpus used for evaluation is Dreaddit, which has a binary labeling with the tags "Stress" and "Not stress". It consists of 3,553 instances, out of which 1,696 are labeled as "Not stress" and 1,857 as "Stress". Table 1 shows the distribution of instances per subreddit.

As observed in the Table 1, instances have been counted for each subreddit thread. The threads with fewer instances are Food pantry, Stress, and Almost homeless. To ensure that each fold contains at least 8 instances of "Not stress" from the "Food pantry" subreddit, two folds were created for the cross-validation experiment.

Table 1. Number of instances in Dreaddit corpus by subreddit.

Subreddit	Label	Instances	Total
Relationships	Not stress	387	694
	Stress	307	
Anxiety	Not stress	234	650
	Stress	416	
PTSD	Not stress	297	711
	Stress	414	
Assistance	Not stress	229	355
	Stress	126	
Homeless	Not stress	139	220
	Stress	81	
Almost homeless	Not stress	40	99
	Stress	59	
Domestic violence	Not stress	139	388
	Stress	249	
Survivors of abuse	Not stress	172	315
	Stress	143	
Stress	Not stress	33	78
	Stress	45	
Food pantry	Not stress	26	43
	Stress	17	

4.2 Evaluation

For the evaluation of the system, the main metric used was F_1. This metric allows for a proper comparison with related works, as studies using the Dreaddit corpus also present their results using these metrics. The following are the cases used to calculate the metrics of recall, precision, and F_1:

- True positives (TP): Correct detection of the "Stress" label.
- True negatives (TN): Correct detection of the "Not stress" label.
- False positives (FP): Incorrect detection of the "Stress" label.
- False negatives (FN): Incorrect detection of the "Not stress" label.

In equation 4, the formula for calculating recall is shown, while in equation 5, the formula for calculating precision is shown. These two metrics are necessary to calculate the F_1 score:

$$Recall = (TP)/(TP + FN), \tag{4}$$

$$Precision = (TP)/(TP + FP). \tag{5}$$

In equation 6, the formula for calculating the F_1 score is presented, which combines precision and recall measurements into a single value:

Table 2. Number of created instances per set.

Set Set	"Stress" instances	"Not stress" instances	Total instances
Test	367	344	711
Fold 1	734	687	1,421
Fold 2	734	687	1,421

Table 3. Best five results F_1, mean and standard deviation, for the different combinations of characteristics.

Fold 1 F_1	Fold 2 F_1	Mean	Test F_1	Features
0.7518	0.7582	0.7550	**0.7605**	words, TAG, lemma
0.7524	**0.7584**	**0.7554**	0.7581	words, TAG, lemma, stem, subreddit
0.7517	**0.7584**	0.7550	0.7581	words, TAG, lemma, stem, hour, subreddit
0.7524	0.7572	0.7548	0.7581	words, TAG, lemma, stem
0.7512	0.7576	0.7544	0.7581	words, TAG, lemma, stem, hour

$$F_1 = 2((PrecisionRecall)/(Precision + Recall)). \quad (6)$$

For the evaluation of the experiments, the corpus was divided into 80% for creating two folds for cross-validation, and the remaining 20% was used as the final test set. Table 2 shows the number of instances for the test set and each created fold.

During the second implementation of grid search for TAG label filtering, folds 1 and 2 were used together and split into 80% for training and 20% for testing, with the aim of finding an improvement in the results. However, in the final evaluations, the previously described cross-validation approach was continued to be used.

5 Results

In this section, the results obtained with the folds and the test set will be shown.

In Table 3, the results with the F_1 metric can be observed. The results are ordered based on the combinations of features that have the best F_1 results. As seen in the table, the model with the best performance in terms of the mean is the second result, with an F_1 score of 0.7584. On the other hand, the model that achieved the best result on the test set is the first result, with an F_1 score of 0.7605, which also has the fewest implemented features. Another detail to note is that the features "words," "TAG," and "lemma" are constant in all the best models.

Additionally, an experiment was conducted to improve the results obtained in Table 3. This experiment involved applying a grid search to filter out less relevant

Table 4. Results F_1, mean and standard deviation in folds, for the best combinations of characteristics, filtering tags.

Fold 1 F_1	Fold 2 F_1	Mean	Features
0.7545	**0.7565**	**0.7554**	words, TAG (filtered), lemma, stem, subreddit
0.7534	**0.7560**	0.7550	words, TAG (filtered), lemma, stem, hour, subreddit
0.7520	0.7569	0.7544	words, TAG (filtered), lemma, stem, hour
0.7513	0.7573	0.7543	words, TAG (filtered), lemma, stem
0.7523	0.7539	0.7531	words, TAG (filtered), lemma

Table 5. Results for the test set, in each model filtering tags.

Precision	Recall	F_1	Features
0.7259	**0.8333**	**0.7759**	words, TAG (filtered), lemma
0.7149	0.8225	0.7650	words, TAG (filtered), lemma, stem, hour, subreddit
0.7159	0.8198	0.7644	words, TAG (filtered), lemma, stem, subreddit
0.7102	0.8172	0.7600	words, TAG (filtered), lemma, stem
0.7102	0.8172	0.7600	words, TAG (filtered), lemma, stem, hour

TAG labels. The grid search used different combination sizes, the combination that yielded the best results was of size 4, and the labels it contained were as follows: 'NNPS' (noun, proper plural), 'UH' (interjection), 'MD' (verb, modal auxiliary), and 'NFP' (superfluous punctuation).

Table 4 presents the results of this experiment on the folds. The results were sorted from highest to lowest using the average, and it can be observed that the lowest result corresponds to the model with fewer features, while the second position in Table 3 now takes the first place.

When we consider the results with tag filtering on the test set in Table 5, we can observe that the best model is still the one with fewer features. Another notable detail is the improvement in the test results.

Finally, in Table 6, a comparison of the proposed model with most of the models seen in the literature can be observed. The proposed model is better than other Naive Bayes-based approaches. It is worth noting that the proposed model achieved higher precision than other Naive Bayes algorithm-based models. However, the proposed model does not manage to position itself among the top-performing models.

6 Conclusions

In this paper, the stress detection task in social networks was reviewed specifically for the Dreaddit corpus. The task was addressed using a Naive Bayes classifier, and the tagging provided by the Spacy tool for Python was utilized. Additionally, two grid searches were conducted to find the best features for this type of classifier.

Table 6. F_1 scores of the most relevant models reviewed in the literature.

Model	Paper	Precision	Recall	F_1
MentalRoBERTaFT	[12]	0.780	0.900	0.840
KC-Net	[1]	0.841	0.833	0.835
MentalRoBERTa	[3]	0.821	0.818	0.819
RoBERTa	[4]	0.812	0.813	0.813
EMO_INF	[5]	0.817	0.817	0.817
Random Forest (BERT)	[13]	0.720	0.850	0.780
Naive Bayes	**Proposed**	**0.725**	**0.833**	**0.775**
LR+Features	[6]	0.735	0.810	0.770
Logistic Reg.	[13]	0.750	0.790	0.770
n-grams + features*	[2]	0.747	0.794	0.770
Multinomial NB	[12]	0.680	0.870	0.760
Bernoulli NB	[12]	0.690	0.840	0.750
BiLSTM_Att	[7]	0.727	0.720	0.720
Naive Bayes (TFIDF)	[13]	0.650	0.740	0.690

The results were evaluated using the F_1 metric, which allowed for a comparison with the works found in the literature. Moreover, the obtained results surpassed those achieved by other models that employed the Naive Bayes algorithm.

Useful features were found from an approach where their frequency of occurrence in the corpus was evaluated, and, likewise, other features were discarded. The use of n-grams to identify potential stress patterns in detection was discarded. Similarly, the use of simple POS tagging from Spacy was also discarded. Instead, the use of detailed tagging (TAG) is suggested, particularly with the identified tags ('NNPS', 'UH', 'MD', and 'NFP'), as they have proven to be useful for stress detection.

Furthermore, as future work, exploring dependency parsing in Spacy is recommended to identify common dependency pairs in texts expressing stress. Additionally, using the discovered features to experiment and investigate if models in the literature can further improve their results.

References

1. Yang, K., Zhang, T., Ananiadou, S.: A mental state Knowledge–aware and Contrastive Network for early stress and depression detection on social media. Information Processing & Management **59**(4), 1–16 (2022)
2. Turcan, E., McKeown, K.: Dreaddit: A Reddit Dataset for Stress Analysis in Social Media. In: Proceedings of the Tenth International Workshop on Health Text Mining and Information Analysis (LOUHI 2019), pp. 97–107 (2019)
3. Ji, S., Zhang, T., Ansari, L., Fu, J., Tiwari, P., Cambria, E.: MentalBERT: Publicly Available Pretrained Language Models for Mental Healthcare. In: Proceedings of the Thirteenth Language Resources and Evaluation Conference, pp. 7184–7190 (2022)

4. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, Y.: RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692. (2019)
5. Turcan, E., Muresan, S., McKeown, K.: Emotion-Infused Models for Explainable Psychological Stress Detection. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 2895–2909 (2021)
6. Tadesse, M., Lin, H., Xu B., Yang, L.: Detection of Depression-Related Posts in Reddit Social Media Forum. IEEE Access **7**, 44883–44893 (2019)
7. Ren, L., Lin, H., Xu, B., Zhang, S., Yang, L., Sun, S.: Depression Detection on Reddit With an Emotion-Based Attention Network: Algorithm Development and Validation. JMIR Med Inform **9**(7), 1–13 (2021)
8. Capdevila, N., Segundo, M.: Estrés. Offarm: farmacia y sociedad **24**(8), 96–104 (2005)
9. Calcia, M.A., Bonsall, D.R., Bloomfield, P.S.: Stress and neuroinflammation: a systematic review of the effects of stress on microglia and the implications for mental illness. Psychopharmacology 233, 1637–1650 (2016)
10. Gong, C., Saha, K., Chancellor, S.: The Smartest Decision for My Future: Social Media Reveals Challenges and Stress During Post-College Life Transition. In: Proceedings of the ACM on Human-Computer Interaction, pp. 1–29 (2021)
11. Rastogi, A., Liu, Q., Cambria, E.: Stress Detection from Social Media Articles: New Dataset Benchmark and Analytical Study. In: 2022 IEEE International Joint Conference on Neural Networks (IJCNN), pp. 1–8 (2022)
12. Swanson, K., Hsu, J., Suzgun, M.: Monte Carlo Tree Search for Interpreting Stress in Natural Language. In: LTEDI 2nd Workshop on Language Technology for Equality, Diversity and Inclusion, pp. 107–119 (2022)
13. Selvadass, S., Malin Bruntha, P., Priyadharsini, K.: Stress Analysis in Social Media using ML Algorithms. In: 4th IEEE International Conference on Smart Systems and Inventive Technology (ICSSIT), pp. 1502–1506 (2022)

ISSN 1870-4069

Automatic Image-based Galaxy Classification: An Approach Using Fractal Dimension Analysis

Jorge de la Calleja[1], Elsa de la Calleja[2], Hugo Jair Escalante[3], Eduardo López-Domínguez[4], María Auxilio Medina-Nieto[1], Marco Aurelio Nuño-Maganda[5]

[1] Universidad Politécnica de Puebla, Mexico

[2] Universidad Nacional Autónoma de México, Instituto de Investigaciones en Materiales, Mexico

[3] Instituto Nacional de Astrofísica, Óptica y Electrónica, Computer Science Department, Puebla, Mexico

[4] National Polytechnic Institute, Department of Computer Science, Center for Research and Advanced Studies, Mexico

[5] Universidad Politécnica de Victoria, Tamaulipas, Mexico

jorge.delacalleja@uppuebla.edu.mx, elsama79@gmail.com, hugojair@inaoep.mx, eduardo.lopez.dom@cinvestav.mx, mnunom@upv.edu.mx

Abstract. The fractal dimension is a measure of complexity which provides structural and spatial information of an object. This physical measurement has been used in the evaluation of several structural properties of the objects and phenomena of the Universe. The Universe contains billions of galaxies, which are large systems of stars and cloud of gases; galaxy classification permits to understand the origin and evolution of the Universe. In this work we present an experimental study for image-based galaxy classification using features extracted with principal component analysis, and combining them with the measure of *Haussdorf-Besicovich* fractal dimension. The classification stage was performed using well-known machine learning algorithms: C4.5, k-nearest neighbors, random forest and support vector machines; considering the three main types of galaxies: elliptical, spiral and irregular. Experimental results using 10-fold cross-validation show that the fractal dimension value allows to improve the galaxy classification yielding an accuracy of 86.71% using the random forest classifier.

Keywords: Fractal dimension, machine learning, galaxy classification.

1 Introduction

Astronomy has a long history of acquiring and analyzing enormous quantities of data. As many other fields, this science has become very data-rich due to advances in telescope, detector, and computer technology. Recently, numerous digital sky surveys across a wide range of wavelengths are producing very large image databases of astronomical objects. For example, the Large Synoptic Sky Survey will produce billions of galaxy images.

Therefore, there is a need to build robust and automated tools for processing astronomical data, particularly for the analysis of the morphology of celestial objects such as galaxies. Galaxy classification is the first step towards a greater understanding of the origin and evolution process of the Universe, and to discover physical properties related to dark matter [33]. Edwin Hubble in 1926 devised a formal galaxy classification scheme, known as the Hubble tuning-fork [1]. This scheme grouped the galaxies based on their shape into three main types: elliptical, spiral and irregular. Elliptical galaxies have the shape of an ellipsoid. Spiral galaxies are divided into ordinary and barred: ordinary spirals have an approximately spherical nucleus, while barred spirals have an elongated nucleus that looks like a bar. Finally, irregular galaxies do not have an elliptical or spiral shape [1].

On one hand, visual inspection for classifying galaxies has been done traditionally by experts, but this time-consuming process requires several skills and high experience. On the other hand, automatic classification methods allow to analyze thousands of images in seconds, also these approaches are more objective and without of prejudices that probably are present in human methodology when looking at galaxy images [2].

Several approaches have been proposed for automatic image analysis and galaxy classification using machine learning and computer vision techniques. Many of this research work has been focused on artificial neural networks [2, 7, 17, 31], decision trees [24, 28], instance-based methods [30], kernel methods [16], among others. Recently, some interesting works have been introduced using new approaches. For example, the sparse representation technique and dictionary learning [11], rotation invariant descriptors [9], quaternion polar complex exponential transform moments [21], and deep neural networks [12, 25, 8, 22].

In this work, we hypothesized that *fractal dimension* quantification can be used in order to improve accuracy for classification of some types of galaxies and then justify their study in depth. Thus, we use the following methodology, composed by three stages, to perform galaxy classification: image processing; feature extraction using fractal dimension analysis and principal component; and classification using machine learning algorithms.

The paper is organized as follows. The next section provides a theoretical background on the fractal dimension analysis. Section 3 introduces the methodology for image-based galaxy classification. Section 4 describes experimental results and Section 5 presents a discussion. Finally, Section 6 outlines conclusions and directions for future work.

2 Fractal Dimension

Fractal dimension provides structural and spatial information of an object which could be a result of reaction or aggregation processes [23, 13, 3]. The *Haussdorf-Besicovich fractal dimension* (HB-fd) by box counting method [13, 19, 18, 20, 27] is a technique to provide information of the complexity of universe objects. It is necessary to use a spectrum of dimensional measures to characterize the total geometry for huge clusters of bright objects.

The fractal dimension formalism is based on the definition of the so-called multifractal spectra, this describes the evolution of the probability distribution of fractal structures. The analysis is performed on an image which is divided into small boxes until ε_0, then the probability of decomposition of the each box (i,Q) is calculated by:

$$P_{i,Q}(\varepsilon) = \frac{x_{i,Q}}{\sum x_{i,Q}} \propto \varepsilon^{\alpha}, \tag{1}$$

where $x_{i,Q}$ is the average height of shapes deposition inside the box of size ε, and α is the singularity of the subset of probabilities. It is suggested that the number of times that α in $P_{i,Q}$ takes a value between α' and $d\alpha'$, defined as $d\alpha'\rho(\alpha')\varepsilon^{-f(\alpha')}$ where $f(\alpha')$ is a continuous function. Then, the number of boxes of ε with the same probability $P_{i,Q}(\varepsilon)$ is given by:

$$N_{\alpha}(\varepsilon) \propto \varepsilon^{f(\alpha)}, \tag{2}$$

where $f(\alpha)$ is the fractal dimension of the subset α [13, 19, 20, 27, 6]. After that, the probability $P_{i,Q}(\varepsilon)$ gives the rise of the partition function:

$$I(Q,\varepsilon) = \sum_{i=1}^{N(\varepsilon)} [P_{i,Q}(\varepsilon)]^{Q} = \varepsilon^{\tau(Q)}, \tag{3}$$

where Q is the moment order. We used the scaling exponent defined by Halsey et al. [19, 18] where $\tau(Q)$ can take a width range of values measuring different regions of the set. The standard procedure [6] takes into account the generalized box-counting dimension defined as:

$$D_Q = \frac{1}{1-Q} \lim_{\varepsilon \to 0} \frac{lnI(Q,\varepsilon)}{ln(\varepsilon_0/\varepsilon)} = \frac{\tau(Q)}{Q-1}. \tag{4}$$

This spectrum generated by an infinite set of dimensions, measures the scaling structure as a function of the local pattern density. If *Q=0* the generalized fractal dimension represent the classic fractal dimension, i.e. $D_f = D_{Q=0}$. The exponent $\tau(Q)$ can be obtained from the slope of $lnI(Q,\varepsilon)$- $ln\varepsilon$ curve. Details of multi-fractal spectrum measures are described in [6, 5].

We select the case of $D_f = D_{Q=0}$ as the parameter of order in the images, where ϵ is the size of the box which acquire successively smaller values of length until the minimum value of ϵ_0. Then, the probability to find is given by:

$$I(Q,\epsilon) = \sum_{i}^{N(\epsilon)} [P_{i,Q}]^{Q}, \tag{5}$$

where Q is a parameter which gives the width of the spectrum and when $Q = 0$ the generalized fractal dimension represents the classical fractal dimension. In this work, the method was performed on gray scale images and using default sampling sizes. The distribution of particles at mesoscopic scales [35, 14, 10] or on macroscopic scales such as the famous fractality of the Britain island [23] were also taken into account for the parametrization.

3 Methodology for Galaxy Classification

The process to perform galaxy classification is divided into three main stages: 1) image processing, 2) feature extraction and 3) classification. In order to standardize the image data set, the images were rotated, centered and cropped, as we have already introduced in [7]. After that, features were extracted by calculating the fractal dimension, and principal component analysis. Finally, the numerical vectors were used as input parameters for the machine learning algorithms to classify the galaxies according to the main three types. Next subsections describe each stage in detail.

3.1 Image Processing

An image processing stage was performed to create a standardized image data set, which permits to extract some useful information from it. This process has been introduced in early work [7], therefore we only give a brief description.

The first step is to distinguish the galaxy contained in the image, then, a threshold is applied to obtain the pixels that form the galaxy: values greater than the threshold. Later, the images are rotated considering their main axis, which is given by the largest eigenvalue of the covariance matrix of the points in the galaxy image. Finally, the images are resized to 128x128 pixels. Figure 1 shows examples of original and standardized images for each type of galaxy.

3.2 Feature Extraction

Before performing classification, galaxy imagery must be represented as numerical vectors (features), which contain meaningful information. However, one of the main challenges when performing this task is to find the best method for characterizing the structural or geometrical properties of galaxies. In this study we have calculated the fractal dimension for each galaxy, which is considered as one of the attributes. Also, we used principal component analysis (PCA) to reduce the dimensionality of the images and to find a set of significant features.

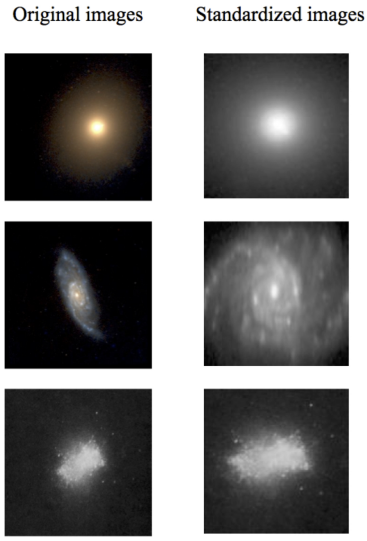


Fig. 1. Galaxy images used for experiments. Left: Original images, Right: Standardized images.

Fractal Characterization. The generalized fractal dimension was measured on different samples of the three main types of galaxies, following the procedure described in section 2.

Figure 2(a) presents the fractal dimension of 17 elliptical galaxies obtaining a fractal dimension value between $1.80 - 1.83$. The behavior of 104 spiral galaxies is presented in Figure 2(b), with values between $1.77 - 1.78$. Finally, Figure 2(c) shows the behavior of 10 irregular galaxies with values between $1.774 - 1.776$.

The ranks of the fractal dimension of these three types of galaxies are evidently different. The most of *HB-fd* of spiral galaxies are in a very similar range; irregular galaxies exhibit one dimension which correspond to structural properties of homogeneity[23]; while *HB-fd* on elliptical galaxies indicates structural wealth. The fractal dimension value for each galaxy was used as a parameter in the classification stage.

Principal Component Analysis. Principal component analysis (PCA) is a mathematical method that converts a (large) data set into a smaller number of variables called principal components (patterns). PCA, in machine learning, is an unsupervised method that reduces data while retaining meaningful patterns. These patterns are considered as a set of attributes that permit to differentiate the objects [32]. The first principal component accounts the largest variability in the data, and each subsequent component accounts the remaining

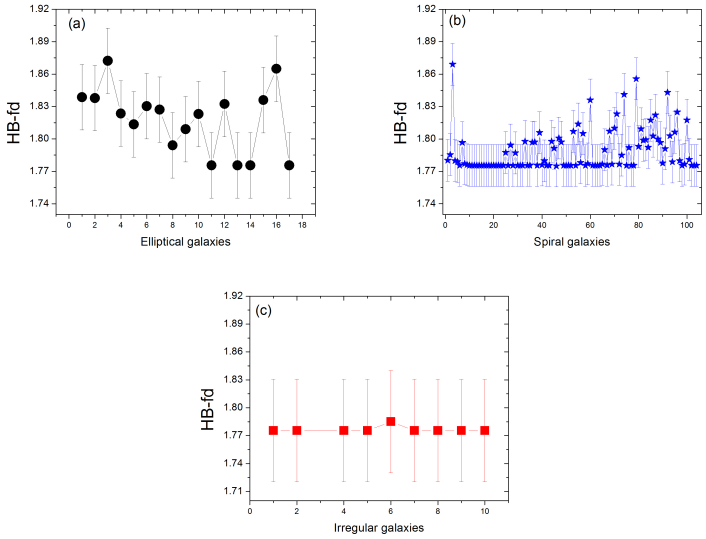


Fig. 2. In figure (a) black dots show the behavior of fractal dimension of elliptical galaxies. In figure (b) blue stars show the behavior of spiral galaxies. In figure (c) red squares presents the corresponding behavior related to irregular galaxies.

ones. This variability permits to rank the principal components according to their usefulness.

In our study, we have used 8 and 12 principal components, which represent about 80% of the information in original and standardized images, respectively; and 21 and 29 principal components, which represent about 90% of the information in the same way (see Figure 1 3). The projection of these principal components onto the original galaxy images were used as parameters for the classification stage.

3.3 Classification

For the classification stage we considered four representative algorithms of the supervised machine learning literature: a decision tree classifier (C4.5), an instance-based method (k-nearest neighbors), an ensemble classifier based on bootstrapping (random forest) and a linear discriminant (support vector machines).

Starting from a sample of labeled images, classification methods learn a function that aims to map unseen images to labels. In this study the labels are associated to the three main galaxy types of the Hubble sequence (i.e., elliptical, spiral or irregular). Before feeding images into classifiers, they must be represented as numerical vectors, thus, two representations were evaluated in this study: (i) The projection of images onto the first principal components, and (ii)

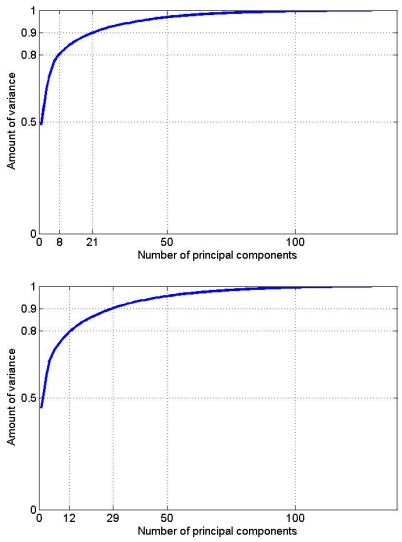


Fig. 3. Amount of cumulative variance of the principal components. Left: Original images. Right: Standardized images.

The same projection plus the *Haussdorf-Besicovich* fractal dimension (HB-fd) value, as an additional feature. Since the HB-fd is a physical measurement employed to describe structural properties of complex systems, our hypothesis is that including it in the representation of images will lead to better classification performance [23, 3]. The remainder of this section briefly describes the considered classification algorithms.

C4.5. It is an algorithm for learning decision tree classifiers [29]. Nodes of a decision tree are associated to thresholds on the attribute values in such a way that the tree induces a partition of the input space. Leaf nodes correspond to samples of the same class. When an unseen image has to be classified its associated vector representation is passed throughout the tree, the label corresponding to the leaf node reached by the feature vector is assigned to the image. The reader is referred to [29] for a more detailed explanation of this classifier.

k-Nearest Neighbors. The k-nearest neighbors (k-nn) classifier is an intuitive method that relies on similarities among instances to determine their class. k-nn memorizes a training set of labeled instances by storing them. When an unseen image needs to be classified, its feature vector is compared to the stored ones, then the k-most similar instances (the k-nn) are identified and used to determine the class of the image. Commonly the memory-based class of the nearest neighbors is assigned to a new instance. The similarity among instances

can be measured in many different ways, in this work we used the inverse of the Euclidean distance as similarity measure. An extensive treatment of k-nn is available in [26].

Random Forest. Random Forest (RF) is a committee or ensemble classifier formed by decision trees. Ensemble classifiers combine the outputs of many individual models trained for the same task, but that focus on different aspects of the problem. In the case of RF, decision trees are considered, each trained on a different subset of samples and dimensions of the feature vectors. When classifying a new instance, all of the individual models provide a prediction and RF returns the average output of the individual learners. RF, as other ensemble models, have theoretical bounds that guarantee the committee of learners outperforms individual models. Further information about RF can be found in [4].

Support Vector Machines. Support Vector Machines (SVMs) is a type of linear discriminant that guarantees obtaining the optimal hyperplane in the input-vector space that separates instances form two classes [34]. Linear discriminant aim at learning a linear function in the input space that separates examples of two classes. SVM finds such a function by maximizing the margin that separates instances from different classes. It provides a sparse solution as the decision function depends only in a subset of instances (the support vectors), which are the closest instances to the decision margin. SVM guarantees obtaining the optimal separating hyperplane in training data when the problem is linear separable. When linear separability does not hold, the *kernel trick* is used to map the original input space into another high-dimensional one where a linear function can separate the classes.

4 Experimental Results

The data set consisted of 131 images of galaxies: 17 elliptical, 104 spiral and 10 irregular; which were taken from different data bases of the web.

The experiments were carried out using Weka, a software package that implements machine learning algorithms for performing classification tasks [15]. We tested the following algorithms: decision trees, k-nearest neighbors, random forest and SVM. For the case of decision trees, and random forest we used default parameters. For the case of k-nn we used three neighbors with weighted distance, and we used a two-degree polynomial kernel for SVM.

In order to measure the overall accuracy of the machine learning algorithms, we used 10-fold cross-validation for all the experiments; that is, the original data set is randomly divided into ten equally sized subsets and performed 10 experiments, using in each experiment one of the subsets for testing and the other nine for training.

Tables 1 and 2 show the accuracy for each learning algorithm using the original images and the standardized ones, respectively. These results were

Table 1. Accuracy for original images using different number of principal components (PCs) and using the PCs plus the fractal dimension value (FDV). The best results are in bold.

	PCs		PCs + FDV	
Algorithm	8	21	9	22
C4.5	71.29	70.83	71.29	70.52
3-nn	77.55	**81.82**	79.22	**81.67**
RF	78.31	80.91	80.30	80.60
SVM	79.38	79.53	79.38	79.99
mean	76.63	78.27	77.54	78.20

Table 2. Accuracy for standardized images using different number of principal components (PCs) and using the PCs plus the fractal dimension value (FDV). The best results are in bold.

	PCs		PCs + FDV	
Algorithm	12	29	13	30
C4.5	77.55	74.34	78.61	76.33
3-nn	81.06	72.81	78.92	75.87
RF	**85.95**	85.33	85.94	**86.71**
SVM	79.84	73.27	85.49	83.20
mean	81.10	76.44	82.24	80.52

obtained by averaging the results of five runs of 10-fold cross-validation for each algorithm. On one hand, as we can observe from table 1, the best results were obtained by 3-nearest neighbors, with 81.82% accuracy using only PCs, and 81.67% accuracy using PCs plus the fractal dimension value. On the other hand, we can see from Table 2 that random forest obtained the best results with 85.95% and 86.71% accuracy, using PCs and PCs plus the fractal dimension value, respectively.

Tables 3 and 4 present the accuracy of the algorithms using only one feature, that is, 1 principal component (1-PC) and the fractal dimension value. Also, we show the results using 1-PC plus the fractal dimension value. From these tables, we can observe that random forest obtained five of the best results, while C4.5 obtained the other one. In addition, we can see that, on average, classification using the fractal dimension value is better than using 1-PC, considering standardized images.

5 Discussion

Figure 4 presents a summarizing of the *HB-fd* values for the three types of galaxies. As we can observe from this Figure, the characterization by fractal dimension helps to distinguish the type of galaxy.

Results presented in Tables 1 and 2 show that the best results are obtained when standardized images and fractal dimension are used, particularly using random forest with 29 PCs plus the fractal dimension value. In addition, it

Table 3. Accuracy for original images using 1 principal component (1 PC) and the fractal dimension value (FDV). The best results are in bold.

Algorithm	1 PC	FDV	1 PC + FDV
C4.5	**79.68**	77.09	**79.38**
3-nn	76.02	70.22	74.04
RF	74.34	65.79	74.34
SVM	79.38	**79.38**	**79.38**
mean	77.35	73.12	76.78

Table 4. Accuracy for standardized images using 1 principal component (1 PC) and the fractal dimension value (FDV). The best results are in bold.

Algorithm	1 PC	FDV	1 PC + FDV
C4.5	78.92	78.46	75.56
3-nn	74.49	78.16	77.24
RF	68.39	75.11	76.17
SVM	**79.38**	**79.22**	**78.62**
mean	75.29	77.74	76.90

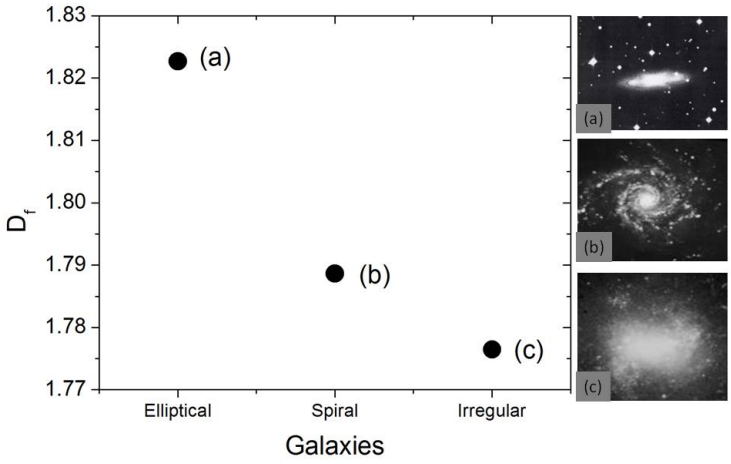


Fig. 4. Fractal dimension for three galaxies: (a) shows an example of an elliptical galaxy with fractal dimension value of $HB - fd = 1.8226$; (b) shows an image of an spiral galaxy, its fractal dimension value is $HB - fd = 1.7886$; finally, (c) presents an example of an irregular galaxy with a value of $HB - fd = 1.7764$.

can be observed that SVM was the algorithm with the largest improvement of accuracy when using PCs plus the fractal dimension value. Specifically, a better evaluation (from 73.27% to 85.29%) is obtained by using 29 and 30 features.

Table 5. Confusion matrix for the best algorithm to classify elliptical galaxies: 3-nearest neighbors.

Galaxy type	Elliptical	Spiral	Irregular	Accuracy per type
Elliptical	15	2	0	88.9 %
Spiral	12	92	0	88.4 %
Irregular	1	9	0	0 %

Table 6. Confusion matrix for the best algorithm to classify spiral galaxies: Random forest.

Galaxy type	Elliptical	Spiral	Irregular	Accuracy per type
Elliptical	11	6	0	64.7 %
Spiral	0	104	0	100.0 %
Irregular	0	10	0	0 %

Table 7. Confusion matrix for the best algorithm to classify irregular galaxies: C4.5.

Galaxy type	Elliptical	Spiral	Irregular	Accuracy per type
Elliptical	9	8	0	52.9 %
Spiral	8	91	5	87.5 %
Irregular	2	4	4	40.0 %

In fact, in average among the different classifiers, the performance classification improved by more than 4% when including the fractal dimension value as a feature.

In Tables 5, 6 and 7 we present the confusion matrix for the best result obtained to classify elliptical, spiral and irregular galaxies, respectively. From these results we can see that 3-nearest neighbors was the best algorithm to classify elliptical galaxies with 88.9% accuracy; random forest was able to classify with 100% accuracy of the spiral galaxies; while C4.5 was the best algorithm to classify irregular galaxies with 40% accuracy. We can also observe that none of the best results for elliptical and spiral galaxies have classified irregular galaxies correctly. On the other hand, when irregular galaxies are classified correctly, the accuracy of elliptical decreases significantly; while the accuracy for spiral galaxies remains about 87% accuracy.

6 Conclusions

In this paper we have introduced the fractal dimension analysis to perform image-based galaxy classification. The fractal dimension value contributes to improve the classification accuracy for the three main types of galaxies, despite using a small data set of images for training the classifiers. The best results were obtained by 3-nearest neighbors and random forest using standardized images with PCs and the fractal dimension value. Directions for future work includes to identify more types of galaxies and testing some approaches of deep learning with fractal dimension analysis.

References

1. Ball, N.: Morphological Classification of Galaxies Using Artificial Neural Networks. Master's thesis, University of Sussex (2002)
2. Ball, N., Loveday, J., Fukugita, M., Nakamura, O., Okamura, S., Brinkmann, J., Brunner, R. J.: Galaxy types in the sloan digital sky survey using supervised artificial neural networks. Monthly Notices of the Royal Astronomical Society, vol. 348, pp. 1038–1046 (2004)
3. Barnsley, M.: Fractals everywhere. Dover publications (2012)
4. Breiman, L.: Random forests. Machine Learning, vol. 45, pp. 5–32 (2001)
5. De la Calleja, E., Bazoni, R., Rocha, M., Barbosa, M.: Topology of dna: a honeycomb stable structure under salt effect. arxiv.org/abs/1706.02685, (2018)
6. De la Calleja, E., Cervantes, F., De la Calleja, J.: Order-fractal transitions in abstract paintings. Annals of Physics, vol. 371, pp. 313–322 (2016)
7. De la Calleja, J., Fuentes, O.: Machine learning and image analysis for morphological galaxy classification. Monthly Notices of the Royal Astronomical Society, vol. 349, pp. 87–93 (2004)
8. Cavanagh, M. K., Bekki, K.: Bars formed in galaxy merging and their classification with deep learning. Astronomy and Astrophysics, vol. 641 (2020)
9. Cecotti, H.: Rotation invariant descriptors for galaxy morphological classification. International Journal of Machine Learning and Cybernetics, vol. 11, pp. 1839–1853 (2020)
10. Chhabra, A., Meneveau, C., Jensen, R., Sreenivasan, K.: Direct determination of the f(α) singularity spectrum and its application to fully developed turbulence. Phys. Rev. A, vol. 40, pp. 5284 (1989)
11. Diaz-Hernandez, R., Ortiz-Esquivel, A., Peregrina-Barreto, H., Altamirano-Robles, L., Gonzalez-Bernal, J.: Automatic approach to solve the morphological galaxy classification problem using the sparse representation technique and dictionary learning. Experimental Astronomy, vol. 41, pp. 409–426 (2016)
12. Dieleman, S., Willett, K., Dambre, J.: Rotation-invariant convolutional neural networks for galaxy morphology prediction. Monthly Notices of the Royal Astronomical Society, vol. 450, pp. 1441–1459 (2015)
13. Feigenbaum, M., Jensen, M., Procaccia, I.: Time ordering and the thermodynamics of strange sets: Theory and experimental tests. Phys. Rev. Lett., vol. 57, pp. 1503 (1986)
14. Ferreira, T., Rasband, W.: Imagej user guide. http::/rsb.info.nih.gov/ij/docs/guide/user-guide.pdf (2013)
15. Frank, E., Hall, M., Witten, I.: The WEKA Workbench. Online Appendix for Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann (2016)
16. Freed, M., Lee, J.: Krylov iterative methods for support vector machines to classify galaxy morphologies. Journal of Data Analysis and Information Processing, vol. 03, pp. 72–86 (2015)
17. Goderya, S., Lolling, S.: Morphological classification of galaxies using computer vision and artificial neural networks: A computational scheme. Astrophysics and Space Science, vol. 279, pp. 377–387 (2002)
18. Halsey, T., Jensen, M., Kadanoff, L., Procaccia, I., Shraiman, B.: Erratum: Fractal measures and their singularities: The characterization of strange sets. Physical Review A, vol. 34, pp. 1601 (1986)

19. Halsey, T., Jensen, M., Kadanoff, L., Procaccia, I., Shraiman, N.: Fractal measures and their singularities: The characterization of strange sets. Phys. Rev. A, vol. 33, pp. 1141 (1986)
20. Hentschel, G., Procaccia, I.: The infinite number of generalized dimensions of fractals and strange attractors. Physica D, vol. 8, pp. 435–444 (1983)
21. Hosny, K., Elaziz, M., Selim, I., Darwish, M.: Classification of galaxy color images using quaternion polar complex exponential transform and binary stochastic fractal search. Astronomy and Computing, vol. 31, pp. 1–13 (2020)
22. Lalit, M. G., Maanak, A., Tushar, P., Mamta, M.: Morphological classification of galaxies using conv-nets. Earth Science Informatics, vol. 13, pp. 1427–1436 (2020)
23. Mandelbrot, B.: How long is the coast of britain? statistical self-similarity and fractional dimension. Science, vol. 156, pp. 636–638 (1967)
24. Marin, M., Sucar, L., Gonzalez, J., Diaz, R.: A hierarchical model for morphological galaxy classification. FLAIRS Conference, (2013)
25. Misra, D., Mohanty, S. N., Agarwal, M., Gupta, S. K.: Convoluted cosmos: Classifying galaxy images using deep learning. Data Management, Analytics and Innovation, vol. 1, pp. 569–579 (2020)
26. Mitchell, T.: Machine learning. McGrawHill (1997)
27. Ott, E.: Chaos in Dynamical Systems. Cambridge University Press, United States of America (1993)
28. Owens, E., Griffiths, R., Ratnatunga, K.: Using oblique decision trees for the morphological classification of galaxies. Monthly Notices of the Royal Astronomical Society, vol. 281, pp. 153–157 (1996)
29. Quinlan, J.: Induction of decision trees. Machine Learning, vol. 1, pp. 81–106 (1986)
30. Shamir, L.: Automatic morphological classification of galaxy images. Monthly Notices of the Royal Astronomical Society, vol. 399, pp. 1367–1372 (2009)
31. Sodré, L., Storrie-Lombardi, M., Lahav, O., Storrie-Lombardi, L.: Morphological classification of galaxies by artificial neural networks. Monthly Notices of the Royal Astronomical Society, vol. 259, pp. 8–12 (1992)
32. Turk, M., Pentland, A.: Face recognition using eigenfaces. In: IEEE Conference on Computer Vision and Pattern Recognition., pp. 586–591 (1991)
33. Van Dokkum, P., Danieli, S., Cohen, Y., Merritt, A., Romanowsky, A., Abraham, R., Brodie, J., Conroy, C., Lokhorst, D., Mowla, L., E., O., Zhang, J.: A galaxy lacking dark matter. Nature, vol. 555, pp. 629–632 (2018)
34. Vapnik, V.: The nature of statistical learning theory. Springer (1995)
35. Weitz, D., Oliveria, M.: Fractal structures formed by kinetic aggregation of aqueous gold colloids. Phys. Rev. Lett, vol. 52, pp. 313–322 (1984)

ISSN 1870-4069

Analysis of Walking Paths from Pedestrian Tracking in Real-Time Using Deep Learning

Ana L. Ballinas-Hernández[1], Carlos E. Hernández-Inzunza[2], M. Claudia Denicia-Carral[1], Maricruz Rangel-Galván[3]

[1] Benemérita Universidad Autónoma de Puebla,
Complejo Regional Centro,
Mexico

[2] Universidad Autónoma de Sinaloa,
Facultad de Informatica,
Mexico

[3] Benemérita Universidad Autónoma de Puebla,
Facultad de Ciencias Químicas,
Mexico

analuisa.ballinas@correo.buap.mx, carloshinzunza2@gmail.com,
claudia.denicia@correo.buap.mx, maricruz.rangelgalvan@viep.com.mx

Abstract. The study of pedestrian walking has a crucial role in design of safe and comfortable public spaces in urban areas and in smart cities. Analysis of walking paths is a task with great challenges in various fields to understand and characterize pedestrian behavior. In this work, real-time detection and tracking algorithms of pedestrians are applied to recover walking paths in urban environments. The main contribution of this work is the recovery of microscopic parameters and frequency distribution analysis of average speed to evaluate the counterflow pedestrian walking dynamics, as well as the description of paths generated from real experiments.

Keywords: Pedestrian detection, pedestrian tracking, deep learning, walking paths.

1 Introduction

Pedestrian walking dynamics has been a subject of wide interest in recent years and is currently an open field of study. The importance of analyzing this system is mainly oriented towards safe and comfortable design of urban spaces, as well as the construction of pedestrian zones with adequate infrastructure in smart cities.

Pedestrian detection and tracking has various applications in the field of computer vision, such as autonomous driving, driver assistance, video surveillance systems, robotics, among others [8]. Traditional object tracking techniques can be inefficient and unreliable, especially in challenging scenarios where there are changes in lighting in the environment, partial obstructions,

variation of postures, areas with high density of crowds, etc. Several algorithms have been developed to solve these situations, however, many problems remain to be solved.

Walking styles vary in each geographic region because pedestrians take into account the cultural aspects of their environment, their physical characteristics, and their walking preferences. It is important to consider these aspects to build reliable pedestrian detection and tracking systems in different scenarios.

In the current work, the detection and monitoring of pedestrians is done in real time, in regular walking conditions, considering different environments, people with different physical characteristics, different walking preferences and at low, medium and high densities. Walking paths and the microscopic parameters of each pedestrian with respect to their position and speed are recovered. In addition, walking patterns and the average speed obtained by following pedestrians are analyzed to understand the group behavior.

2 Related Work

In the work of Camara et al. a review of the current state of the art in the field of detection, recognition, monitoring and prediction of pedestrian paths has been done [3]. Review is organized into five levels, the lower levels being pedestrian detection methods that rely on machine vision and robotics models to detect pedestrians, tracking their positions and speeds over time. At the highest levels, other factors intervene, among them psychological and pedestrian personality, through which it is possible to predict their movements and actions. At these levels, psychological information is inferred from body language, gestures, and demographic information. Models like YOLO (You Only Look Once) have been widely used in object detection techniques.

In the work of Sundararaman et al. a head tracker in high densities comparable to traditional pedestrian trackers is introduced. Results show their method is comparable with traditional algorithms [14]. Also they present a metric known as EDEucl, two methods for head detection, and a useful model for crowd counting and movement analysis. A work has been done for the tracking of multiple objects by means of the detection and tracking of objects in a scene through a SORT algorithm (simple online real-time tracking) that includes an identification module [1]. Experiments were done with the MOT17 and MOT20 sets, relevant data sets in the field of study. Results obtained show that SORT-based algorithms for pedestrian tracking are a good option and can be easily integrated into other tracking trackers for detection.

Video surveillance systems have been developed to detect unusual events through pedestrian monitoring in order to maintain crowd safety [6]. A comparison is made between different computational vision techniques for the detection and tracking of pedestrians and some pedestrian video databases obtained from different repositories are described. In addition, the main problems that occur in human detection and tracking are identified, such as occlusion, variation in postures and areas of high crowd density that generate errors.

An occlusion management strategy has been developed through modeling the relationships between occlusions and occluded tracks, unlike traditional feature-based approaches [13]. This strategy is used for multiple pedestrian detection, focuses on lane management and is capable of working for bidirectional tracking with results higher than those reported for the MOT17, MOT20 and MOT16 sets.

Several experimental works have been done to understand real pedestrian walking dynamics through analysis of their paths and quantification of the system under controlled environmental conditions in different walking situations [11, 17]. In the work of Zanlungo et al. the effect of real pedestrian walking in groups with bidirectional flow was analyzed where a line grouping algorithm was built for its characterization [16]. The experimental results of the physical system were compared with the simulation model where the speed of pedestrians, the number of collisions, the number of sidewalks and the ratio of pedestrians are quantified.

In the work of Feliciani et al. controlled experiments of pedestrian walking in chaotic scenarios were done [5]. Authors analyzed paths, collision avoidance mechanisms and the fundamental diagram for different experimental data. From description of experiments, a simulation model based on particle gases was built where the interactions between pedestrians were modeled as physical forces. Authors concluded that when people walk in chaotic conditions with minimal influence from the environment, a simple model is sufficient to describe the system general behavior.

In present work, pedestrian walking paths recovered from videos of real experiments are analyzed. For pedestrian detection, a YOLOv3 algorithm based on convolutional neural networks and SORT algorithm for pedestrian tracking are applied [18]. The contribution of this work, with respect to the related work, is the recovery of microscopic parameters of pedestrians that characterize their walking and analysis of generated paths and average speed of the group.

3 Proposed Methodology

To generate walking paths that pedestrians follow, the methodology shown in Figure 1 is proposed. The detection and monitoring of pedestrian walking in corridors under regular situations in corridors is proposed. Some microscopic parameters of pedestrians are identified and the group behavior patterns obtained are analyzed.

3.1 Video Collection

In this phase, a set of videos of urban streets where pedestrians walk at different densities is built. Data is obtained from the public repository Caltech Pedestrian Dataset of people who walk under regular conditions with a total of 10 hours of video with a resolution of 640 x 480 px [4]. In addition, videos of pedestrians walking against the flow were captured with a total of 5,100 seconds of video in

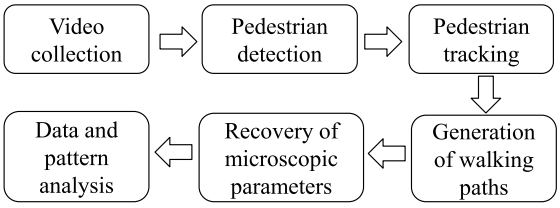


Fig. 1. Proposed methodology for trajectory analysis of pedestrian walking.

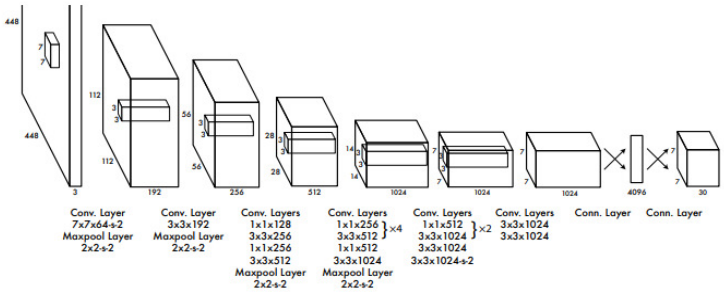


Fig. 2. YOLO architecture for object detection based on convolutional neural networks, Obtained from [10].

25 different corridors with a resolution of 1,280 x 720 px. Videos were captured during the day at different times with low, medium and high pedestrian densities.

3.2 Pedestrian Detection

Videos are divided into frames for the recognition of pedestrians in each image. The pretrained model YOLO is used for the fast detection of multiple objects that uses a CNN (convolutional neural network) following the architecture of Figure 2 [12]. The CNN has 24 convolutional layers and 2 fully connected layers from a set of images rescaled to a size of 448 × 448. For automatic detection and classification of pedestrians, the pretrained model YOLOv3 is used, which generates feature maps at three different scales. The algorithm predicts bounding boxes for each image cell to recognize objects and calculates the loss function that is compared with the ground truth where the highest IoU (Intersection of Union) is chosen to recognize pedestrians [18].

3.3 Pedestrian Tracking

For tracking by pedestrian detection in consecutive frames from videos, SORT algorithm is applied [2]. The algorithm combines object detection with Hungarian algorithms and Kalman filters for tracking. YOLOv3 and SORT algorithms were tested overall with 4,327 frames with 6 different classes and

obtained an average accuracy of 0.724, an average precision of 0.8068, and a recall of 0.909. These algorithms allow pedestrian dynamics to be tracked in real time by assigning an ID to each pedestrian.

3.4 Generation of Walking Paths

Based on pedestrian tracking between input video frames, walking paths of each pedestrian are automatically retrieved in a new image. Each path is painted in a color for each pedestrian for easy identification. These paths allow to see the evolution of pedestrian walking from an initial position to a final position where pedestrians walk against the flow in corridors reaching the exit.

3.5 Recovery of Microscopic Parameters

Some parameters are recovered at microscopic level where pedestrians are treated as individuals. These parameters are positions (x, y) of each pedestrian, taking reference system the size of videos. In addition, average speed of each pedestrian v from the first to the last frame is retrieved. Speed relates distance travelled by pedestrians to the time elapsed as a scalar quantity. These data are stored in .CSV files. Speed is a relevant parameter to understand the way pedestrian crowds walk at different densities and under different situations. The parameters obtained are:

$$P = \{x, y, v\}. \tag{1}$$

3.6 Data and Pattern Analysis

Final phase of the methodology consists in analyzing the microscopic parameters recovered by means of a frequency distribution histogram of walking speed to quantitatively analyze the pedestrian speed behavior at different densities. In addition, obtained paths are analyzed qualitatively to understand the evolution of the group behavior on a macroscopic level. The objective is to recognize patterns of self-organization that appear in walking dynamics of pedestrian crowd.

4 Results

Results obtained in this work are: construction of a set of videos of corridors, where pedestrian crowds walk against flow under regular flow conditions; path generation obtained by detection and monitoring of pedestrians in real time; qualitative description of walking paths as a manifestation of the group self-organization on a macroscopic level; data retrieval from microscopic parameters of crowd dynamics. Three different crowd density areas are tested: low, medium, and high.

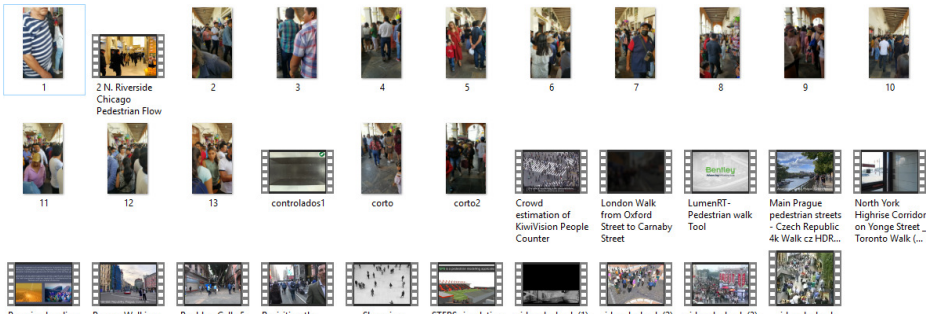


Fig. 3. Sample of the dataset that contains public repositories and own videos of pedestrians walking.

Table 1. Evaluation of detection based pedestrian tracking model.

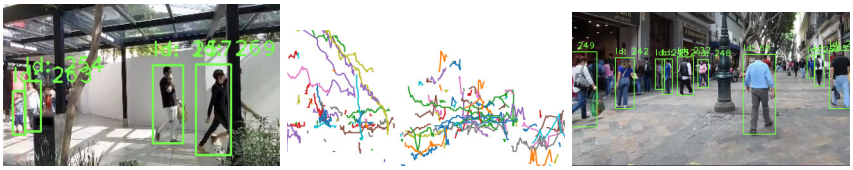
Video	Total frames	Pre-processing time	Inference time	Average ground truth
Low density	197	0.7ms	42ms	0.7433
Medium density	310	1.1ms	42.5ms	0.7633
High density	156	1.2ms	53.8ms	0.5959

Total of 41,100 seconds of videos of pedestrians walking against the flow in corridors under regular conditions were collected (see a sample in Figure 3). Videos are divided into frames to process the image sequences to be processed.

Tests of tracking algorithms for pedestrian detection are done using a computer with an AMD Core i5 processor at 2.2GHz with 12 GB of RAM and Windows operating system. Python programming language and machine learning libraries are used to test the predictive models. For the tests, a sample of three videos each with 350 frames is considered. Left section of Figure 4 shows results of the YOLO model application for the pedestrian detection where a bounding box is drawn with the ID of each pedestrian at three densities: low, medium, and high. Right section of Figure 4 shows results of the SORT algorithm application for pedestrian tracking, where walking paths are generated from the initial video frame to the final frame. As can be seen in the figure, it was possible to recover and draw the pedestrian walking paths in videos of real experiments based on tracking by pedestrian detection at low, medium and high densities.

Table 1 shows average pre-processing times required for video frames at low, medium and high densities. In addition, the average ground truth of each video is calculated to evaluate results of predictive model with respect to real data.

Results of the pedestrian detection tracking demonstrate an acceptable performance of algorithms with an average ground truth of 0.7433 at low concentration densities and 0.7633 at medium densities. However, at high densities, the algorithms present a ground truth of 0.5959, which is unreliable due to the frequent occlusions of large crowds and because the separation between pedestrians is limited, causing algorithms have high detection errors.



(a) Pedestrian detection at low density (b) Generation of walking paths at low density (c) Pedestrian detection at medium density

(d) Generation of walking paths at medium density (e) Pedestrian detection at high density (f) Generation of walking paths at high density

Fig. 4. Generation of walking paths obtained by tracking pedestrians in real time.

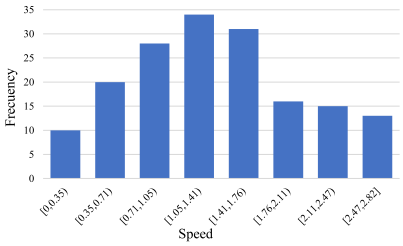
X	204	202	201	201	200	200	199	197	194	192	191	190	189	187	185	183	181	179	178	176	176	175	175	176	167	166	165	164	157	157	156	156	156	156	72
Y	200	199	199	199	199	200	199	199	200	200	199	199	199	199	198	199	199	198	198	198	198	200	200	200	199	200	199	199	198	199	199	199	199	200	207
V	1.46	1.94	0.47	1.41	1.31	1.12	1.47	1.2	1.29	1.24	1.95	2.36	1.2	1.45	2.51	0.75	2.16	0.96	1.51	1.2	2.03	2.63	0.67	1.03	2.59	1.41	2.24	0.63	1.61	0.71	1	1.52	0.41	1.01	2.24

Fig. 5. Position (x, y) and speed v obtained from pedestrian tracking.

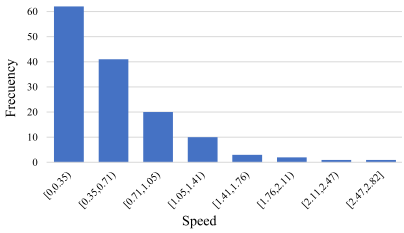
Figure 4.b shows walking paths obtained from a video at low densities. These paths present irregular patterns because, as there are few encounters, pedestrians apply turns for convenience and their walking is free. Free walking is a very commonly reported rule in the literature at low densities [9]. Figure 4.d shows paths of bidirectional pedestrian walking where line formation occurs, which is a very frequent phenomenon in the simulation of pedestrian walking as a manifestation of self-organization [7]. Figure 4.f shows the pedestrian walking paths at high densities. Despite the errors in crowd detection, it can be seen that walking lines are regular because pedestrians are not free to turn for convenience and follow a line until they reach their goal. From this description of paths, it is possible to qualitatively understand the pedestrian crowd dynamics at three different densities under regular walking conditions.

Figure 5 presents a sample of position (x, y) and speed v evolution of a pedestrian recovered from his walking paths. These are the microscopic parameters of each individual pedestrian that were managed to be stored in .CSV files to characterize the group walking. These data are stored for each pedestrian detected along frames with respect to coordinate system of videos scaled to a size of 640 x 480 px, where lower left corner being coordinate $(0, 0)$ and frames are equivalent to the first quadrant of Cartesian coordinate system.

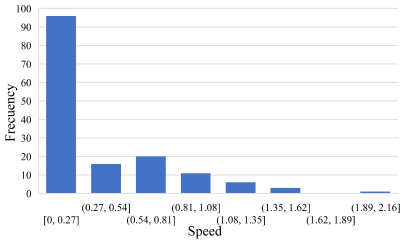
From positions and speed of pedestrians retrieved from real videos, histogram of speed distribution function shown in Figure 6 is obtained. As can be seen, crowd behavior at low densities in Figure 6.a shows a behavior similar



(a) Video at low density with $\mu = 1.34m/s$ and $\sigma = 0.67m/s$.



(b) Video at medium density with $\mu = 0.85m/s$ and $\sigma = 0.82m/s$.



(c) Video at high density with $\mu = 0.38m/s$ and $\sigma = 0.5m/s$.

Fig. 6. Frequency distribution of crowd average speed at low, medium, and high densities.

to the Gaussian function with mean $\mu = 1.34m/s$ and standard deviation $\sigma = 0.67m/s$ because pedestrians have few encounter conflicts with others. At medium densities, the behavior resembles a decreasing exponential function because present various encounters between pedestrians with $\mu = 0.85m/s$ and $\sigma = 0.82m/s$ (see Figure 6.b). At high densities, their behavior does not have a uniform behavior and most pedestrians have a concentrated speed at the lowest value between $0m/s$ and $0.27m/s$ with $\mu = 0.38m/s$ and $\sigma = 0.5m/s$ because crowd gatherings occur and pedestrians do not have enough space to maintain their social distance (see Figure 6.c).

5 Conclusions

This paper proposes a methodology for real-time video pedestrian tracking by applying an object detection algorithm based on convolutional neural networks. Walking paths between frames and some microscopic parameters of pedestrians are recovered to understand their evolution.

The main contribution of this work is description of walking paths and frequency distribution analysis of average speed to understand pedestrian crowd dynamics at different densities under regular walking conditions.

As future work, application of tracking algorithms capable of recognizing pedestrians occluded by obstacles and detecting multiple pedestrians in

high densities is proposed to achieve more reliable results. In addition, the methodology can be extended to include other scenarios such as emergency exits or panic situations.

References

1. Aharon, N., Orfaig, R., Bobrovsky, B.-Z.: Bot-sort: Robust associations multi-pedestrian tracking. arXiv:2206.14651, (2022)
2. Bathija, A., Sharma, G.: Visual object detection and tracking using yolo and sort. International Journal of Engineering Research Technology, vol. 8, no. 11, pp. 705–708 (2019)
3. Camara, F., Bellotto, N., Cosar, S., Nathanael, D., Althoff, M., Wu, J., Ruenz, J., Dietrich, A., Fox, C. W.: Pedestrian models for autonomous driving part i: low-level models, from sensing to tracking. IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 10, pp. 6131–6151 (2020)
4. Dollár, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: A benchmark. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 304–311. IEEE (2009)
5. Feliciani, C., Murakami, H., Nishinari, K.: A universal function for capacity of bidirectional pedestrian streams: Filling the gaps in the literature. PloS one, vol. 13, no. 12, pp. 1–31 (2018)
6. Gawande, U., Hajari, K., Golhar, Y.: Pedestrian detection and tracking in video surveillance system: issues, comprehensive review, and challenges. Recent Trends in Computational Intelligence, pp. 1–24 (2020)
7. Murakami, H., Feliciani, C., Nishinari, K.: Lévy walk process in self-organization of pedestrian crowds. Journal of The Royal Society Interface, vol. 16, no. 153, pp. 1–10 (2019)
8. Pal, S. K., Pramanik, A., Maiti, J., Mitra, P.: Deep learning in multi-object detection and tracking: state of the art. Applied Intelligence, vol. 51, pp. 6400–6429 (2021)
9. Rangel-Huerta, A., Ballinas-Hernández, A., Muñoz-Meléndez, A.: An entropy model to measure heterogeneity of pedestrian crowds using self-propelled agents. Physica A: Statistical Mechanics and its Applications, vol. 473, pp. 213–224 (2017)
10. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 779–788 (2016)
11. Seer, S., Brändle, N., Ratti, C.: Kinects and human kinetics: A new approach for studying pedestrian behavior. Transportation research part C: emerging technologies, vol. 48, pp. 212–228 (2014)
12. Srazhdinova, A., Anvarov, S., et al.: Detection and tracking people in real-time with yolo object detector. Challenges of Science, vol. 2020, no. 3, pp. 69–75 (2020)
13. Stadler, D., Beyerer, J.: Improving multiple pedestrian tracking by track management and occlusion handling. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10958–10967 (2021)
14. Sundararaman, R., De Almeida Braga, C., Marchand, E., Pettre, J.: Tracking pedestrian heads in dense crowd. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3865–3875 (2021)
15. Xiao, Y., Zhou, K., Cui, G., Jia, L., Fang, Z., Yang, X., Xia, Q.: Deep learning for occluded and multi-scale pedestrian detection: A review. IET Image Processing, vol. 15, no. 2, pp. 286–301 (2021)

16. Zanlungo, F., Crociani, L., Yücel, Z., Kanda, T.: The effect of social groups on the dynamics of bi-directional pedestrian flow: A numerical study. In: Traffic and Granular Flow 2019. pp. 307–313. Springer (2020)
17. Zanlungo, F., Feliciani, C., Yücel, Z., Nishinari, K., Kanda, T.: Macroscopic and microscopic dynamics of a pedestrian cross-flow: Part II, modelling. Safety science, vol. 158, pp. 4–24 (2023)
18. Zuo, X., Li, J., Huang, J., Yang, F., Qiu, T., Jiang, Y.: Pedestrian detection based on one-stage yolo algorithm. Journal of Physics: Conference Series, vol. 1871, no. 1, pp. 1–7 (2021)

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

http://rcs.cic.ipn.mx

Centro de Investigación
en Computación