Scientific Paper / Artículo Científico

UNIVERSIDAD POLITÉCNICA SALESIANA ECUADOR
INGENIUS
Revista de Ciencia y Tecnología
https://doi.org/10.17163/ings.n32.2024.06
pISSN: 1390-650X / eISSN: 1390-860X

Diagnosis of oral cancer using deep learning algorithms

Diagnóstico de cáncer oral mediante algoritmos de aprendizaje profundo

Mayra Alejandra Dávila Olivos[1,*], Henry Miguel Herrera Del Águila[1], Félix Melchor Santos López[1]

Received: 15-05-2023, Received after review: 20-05-2024, Accepted: 28-05-2024, Published: 01-07-2024

Abstract

Objective. The aim of this study was to use deep learning for the automatic diagnosis of oral cancer, employing images of the lips, mucosa, and oral cavity. A deep convolutional neural network (CNN) model, augmented with data, was proposed to enhance oral cancer diagnosis. Materials and methods. We developed a Mobile Net deep CNN designed to detect and classify oral cancer in the lip, mucosa, and oral cavity areas. The dataset comprised 131 images, including 87 positive and 44 negative cases. Additionally, we expanded the dataset by varying cropping, focus, rotation, brightness, and flipping. The diagnostic performance of the proposed CNN was evaluated by calculating accuracy, precision, recall, F1 score, and area under the curve (AUC) for oral cancer. Results. The CNN achieved an overall diagnostic accuracy of 90.9% and an AUC of 0.91 with the dataset for oral cancer. Conclusion. Despite the limited number of images of lips, mucosa, and oral cavity, the CNN method developed for the automatic diagnosis of oral cancer demonstrated high accuracy, precision, recall, F1 score, and AUC when augmented with data.

Keywords: Automatic diagnosis, convolutional neural network, data augmentation, dental health, oral cancer, oral disease

Resumen

Objetivo. El propósito de este estudio fue diagnosticar automáticamente el cáncer oral en imágenes de labios, mucosa y cavidad oral utilizando aprendizaje profundo. Se propuso un modelo de red neuronal convolucional (CNN) profunda con aumento de datos para el diagnóstico de enfermedades bucodentales. Materiales y métodos. Se desarrolló una CNN profunda de MobileNet para detectar y clasificar la enfermedad de cáncer oral en la zona de los labios, mucosa y cavidad oral. El conjunto de datos de 131 imágenes de labios, mucosa y cavidad oral estaba compuesto por 87 casos positivos y 44 casos negativos. Además, el número de imágenes se multiplicó mediante cambios de corte, enfoque, rotación, brillo y volteo. Se evaluó el rendimiento de diagnóstico de la CNN propuesta a través del cálculo de la exactitud, la precisión, la recuperación, la puntuación F1 y el AUC (Área bajo la curva) para la enfermedad de cáncer oral. Resultados. El rendimiento general del diagnóstico de la enfermedad de cáncer oral alcanzó el 90.9% de exactitud y 0.91 AUC usando la CNN con el conjunto de datos. Conclusión. El método CNN desarrollado para diagnosticar automáticamente el cáncer oral en imágenes de labios, mucosa y cavidad oral usando aumento de datos mostró una alta exactitud, precisión, recuperación, puntaje F1 y AUC a pesar del número limitado de imágenes de labios, mucosa y cavidad oral utilizadas.

Palabras clave: Diagnóstico automático, aumento de datos, cáncer oral, enfermedad bucodental, red neuronal convolucional, salud dental

[1,*]Faculty of Systems Engineering and Informatics, National University of San Marcos, Perú.
Corresponding author ✉: 12200133@unmsm.edu.pe.

Suggested citation: Dávila Olivos, M. A.; Herrera Del Águila, H.M. ; Santos López, F.M. "Diagnosis of oral cancer using deep learning algorithms," *Ingenius, Revista de Ciencia y Tecnología*, N.° 32, pp. 58-68, 2024, DOI: https://doi.org/10.17163/ings.n32.2024.06.

1. Introduction

Oral diseases pose a significant global public health challenge, particularly affecting less privileged populations due to their widespread occurrence. Treatment costs are often prohibitively high and remain unattainable in many low- and middle-income countries. According to the World Health Organization, managing oral diseases ranks as the fourth most expensive health condition in heavily urbanized nations. Given its profound impact on overall health, oral health is an essential determinant of human well-being and a critical component of healthcare. Furthermore, the presence of oral diseases increases the risk of chronic conditions such as diabetes, respiratory issues, and cardiovascular and cerebrovascular diseases [1].

Machine learning (ML), a subset of artificial intelligence (AI), employs statistical, probabilistic, and optimization techniques that enable machines to learn from historical data, acquire information, and make predictions regarding new data based on the learned information [2,3]. Within dental clinical decision-making, methods grounded in deep learning (DL), another subset of AI, streamline processes and address intricate challenges. Among these methods, a deep convolutional neural network (CNN), a well-defined algorithm in DL, has proven highly effective for organ segmentation and the classification and detection of organs and diseases within medical images [4–6].

ML has demonstrated remarkable accuracy and precision, surpassing human judgment in predicting medical outcomes [2]. DL techniques offer advantages over feature-based methods in medical image analysis, consistently outperforming healthcare professionals in disease identification [7].

In the field of oral cancer diagnosis, DL has yielded promising results in automated pathology analysis, oral cavity imaging, confocal laser endomicroscopy imaging, and fluorescence imaging. These advancements facilitate the prediction of cancer risk and patient diagnostic outcomes, enabling the identification of subtle patterns within large, noisy datasets. The ultimate goal is to develop tools to improve public dental health [2], [7,8].

This article aims to implement a model for diagnosing oral cancer using high-performance DL algorithms. The proposed model can potentially be an asset in the decision-making process for diagnosing this disease.

1.1. Literature review

The systematic review of the literature was conducted using the PRISMA methodology. This approach facilitated a comprehensive understanding of the research background, supported this work, and demonstrated proficiency in machine learning (ML) and deep learning (DL) approaches, thereby ensuring the study's relevance [9].

The study presented in [2] develops and validates four ML models to predict the occurrence of lymph node metastases in early-stage Oral Tongue Squamous Cell Cancer (OTSCC), both before and after surgery. The random forest and support vector machine models exhibit superior predictive performance compared to traditional methods based on the depth of invasion, neutrophil-to-lymphocyte ratio, or tumor budding.

Another study [7] explores an efficient imaging method utilizing smartphones and a DL algorithm for the automatic detection of oral diseases. The researchers introduce the HRNet DL network and assess its efficacy in detecting oral cancer. The proposed HRNet model demonstrates marginally improved performance over the VGG16, ResNet50, and DenseNet169 models across 455 test images based on sensitivity, specificity, precision, and F1-score metrics.

In a separate research endeavor [4], scientists automate the diagnosis of odontogenic cysts and tumors in both jaws using panoramic X-rays and. They enhance a modified deep CNN derived from YOLOv3 to detect and classify these conditions. The overall disease classification performance improves when using CNN with an augmented dataset compared to a non-augmented dataset.

1.2. MobileNet

MobileNet uses depth separable convolutions, a technique that significantly reduces the number of parameters compared to networks that use regular convolutions with the same depth. This reduction in parameters allows for the creation of lightweight deep neural networks. Developed by Google as an open-source CNN class, MobileNet is an excellent base for training classifiers. The classifiers generated using MobileNet are not only compact and fast, but also help minimize model size and computational requirements. This is accomplished by replacing standard convolution filters with deep and point convolutions [10].

1.3. Performance metrics

1. **Receiver operating characteristic (ROC) curves:** ROC curves are graphical representations widely employed to assess and compare the performance of classifiers. These two-dimensional graphs illustrate the trade-off between sensitivity and specificity in a classifier's predictions. They visually demonstrate the classifier's performance across various discrimination thresholds, facilitating the classification and selection of classifiers according to specific user requirements. These requirements frequently encompass considerations of differential error costs and precision demands [11].

2. **Area under the curve (AUC):** AUC is a single scalar value that provides a comprehensive measure of the overall performance of a binary classifier. The AUC value ranges from 0.5 to 1.0, where the minimum value indicates the performance of a random classifier, and the maximum value corresponds to that of a perfect classifier. In Figure 1, the ROC curves for two scoring classifiers, A and B, are presented. In this example, classifier A has a higher AUC value than classifier B [12].

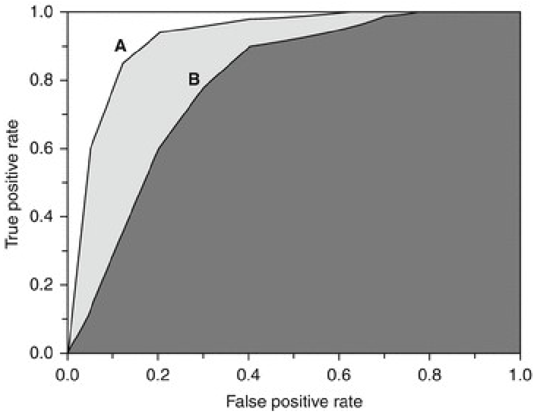
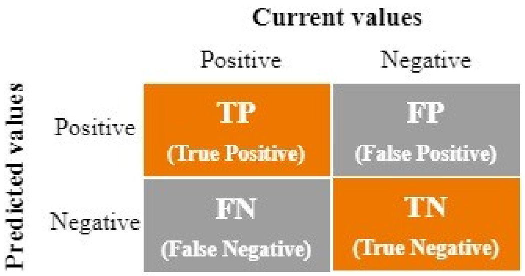


Figure 1. Area under the curve [11]

3. **Confusion matrix:** A confusion matrix is a table-like representation that shows the true and predicted class of each case in the test set. This matrix is essential for evaluating the performance of a model in a classification problem. By presenting a clear breakdown of true positives, true negatives, false positives, and false negatives, it enables a detailed understanding of the model's performance across different classes. Figure 2 provides a visual representation of the confusion matrix, specifically designed for a binary class classification problem [13].



		Current values	
		Positive	Negative
Predicted values	Positive	**TP** **(True Positive)**	**FP** **(False Positive)**
	Negative	**FN** **(False Negative)**	**TN** **(True Negative)**

Figure 2. Binary confusion matrix [12]

4. **Accuracy:** A test method is considered accurate when it precisely measures what it is intended to measure. In other words, it can effectively determine a substance's accurate amount or concentration within a sample [14].

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \cdots \quad (1)$$

5. **Precision:** Precision in a test method is achieved when repeated determinations or analyses on the same sample yield consistent results. In the context of accuracy, a precise test method exhibits minimal random variation, enhancing trust in its reliability. The ability of the test method to consistently reproduce results over time underscores its dependability [14].

$$Precision = \frac{TP}{TP + FP} \cdots \quad (2)$$

6. **Recall:** Recall, also referred to as sensitivity, denotes the capability of a diagnostic test to correctly detect individuals afflicted with a particular disease or disorder. A test with high sensitivity minimizes instances of 'false negatives,' wherein the test fails to identify the presence of a disease despite its actual existence [14].

$$Recall = \frac{TN}{TN + FP} \cdots \quad (3)$$

7. **F1 score:** The F1 score integrates accuracy and recall measurements into a unified metric, facilitating a comparative assessment of overall performance across diverse solutions. The F1 score operates under the assumption that both accuracy and recall hold equal significance [15].

$$F1Score = 2 * \frac{Precision * Recall}{Precision + Recall} \cdots \quad (4)$$

1.4. Oral cancer

Oral cancer includes malignant tumors that affect the lip, different areas of the mouth, and the oropharynx, as shown in Figure 3. This form of cancer is more common in men and older individuals, with significant differences associated with socioeconomic status. Interestingly, in some countries in Asia and the Pacific, oral cancer is among the top three cancers with the highest occurrence [16].

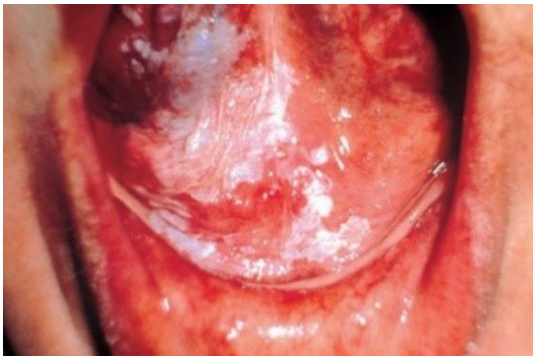


Figure 3. Oral cancer sore [16]

2. Materials and methods

2.1. Data acquisition

The dataset used in this study was acquired from the Kaggle web platform [17], which offers open access to downloadable data. The original data format comprised jpg images. The dataset included 131 cases, with 87 featuring images of lips, mucous membranes, and the oral cavity indicative of oral cancer and the remaining 44 showcasing images without oral cancer. Figure 4 visually summarizes the research methodology, clearly outlining the phases. The implementation of this stage is detailed in Section 2.6.1.

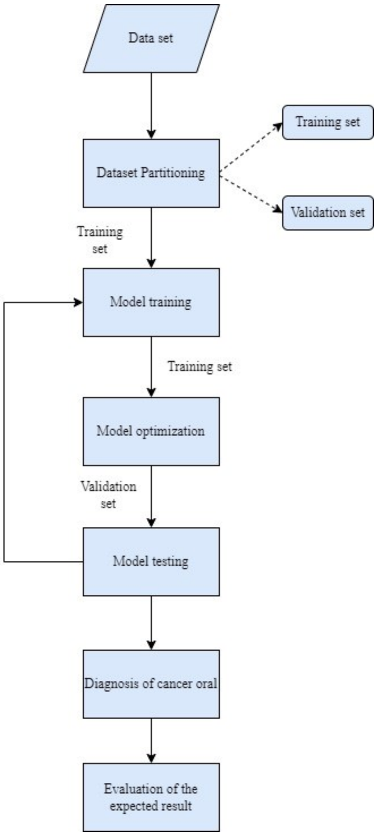


Figure 4. Model development methodology

2.2. Dataset partitioning

The partitioning of the dataset entails a non-overlapping division of the available data into two distinct subsets: the training dataset and the validation dataset. This separation provides one subset for analytical purposes and another for model verification.

1. **Training set:** The training dataset constitutes 90% of the total dataset, encompassing 118 images of lips, mucosa, and the oral cavity. This includes 78 images indicative of oral cancer and 40 images without signs of oral cancer.

2. **Validation set:** The validation dataset comprises 10% of the total dataset, featuring 13 images of lips, mucosa, and the oral cavity. This includes 9 images indicative of oral cancer and 4 images without signs of oral cancer.

The implementation of this stage is detailed in Section 2.6.2.

2.3. Model training, optimization, and tests

The CNN model proposed in this study was implemented and trained on the Kaggle platform, utilizing Python as the programming language due to its extensive capabilities in machine learning (ML) and deep learning (DL). These features make Python particularly well-suited for managing the complexities of the task. During this phase, the training dataset served not only for the initial training but also for the pre-validation of the model, thereby laying the groundwork for further optimization and testing.

When validating the developed model, which exhibits performance metrics ranging from 0.5 to 1.0, areas requiring improvement are identified to enhance its performance. Adjustments are made using the training dataset. Upon achieving satisfactory evaluation metrics, the model is subjected to testing to verify its effectiveness and reliability.

To evaluate the proposed model, the validation dataset is employed to confirm its high performance. If the outcomes differ from expectations, additional iterations of training, optimization, and testing are conducted until the desired results are achieved. This iterative process is documented in Sections 2.6.3 and 2.6.4.

2.4. Diagnosis of oral cancer disease

Based on the performance results obtained from the model tests, the capability for automatic diagnosis of oral cancer is determined. This diagnosis pertains to the images of the lips, mucosa, and oral cavity utilized in the model. The implementation of this stage is detailed in Section 2.6.5.

2.5. Evaluation of the expected result

Following the diagnosis of oral cancer, the results obtained are evaluated by comparing their accuracy, precision, recall, and F1 score. Through this comparison, it is determined that the diagnosis provided by the proposed model yields satisfactory results. The implementation of this stage is detailed in Section 2.6.6.

2.6. Implementation

2.6.1. Loading libraries and read in data

The development of the solution commences with the loading of essential libraries, such as matplotlib, NumPy, and pandas. Global parameters are defined, and images of the lips, mucosa, and oral cavity, whether indicative of the disease or not, are obtained. Key parameters include:

1. **Size:** Input size [18].
2. **Epochs:** The number of iterations over the entire dataset [19].
3. **Batch size:** Division of the dataset into multiple smaller batches [19].
4. **Fold:** The number of folds the dataset will be divided into [20].

2.6.2. Generating dataset

A dataset comprising images of lips, mucous membranes, and the oral cavity, with or without the disease, is created. The images are resized, and their quantity is augmented through various alterations, including cropping, focus adjustment, rotation, brightness modification, and flipping.

The dataset, initially unbalanced with 44 cases not presenting oral cancer and 87 cases that do, undergoes class balancing. Classes are determined to identify the presence or absence of disease, and the images are segmented based on whether they depict oral cancer.

2.6.3. Creating a model

The proposed CNN model is established using "MobileNet" as the chosen architecture. Additionally, the following attributes are employed:

1. **Early Stopping:** Configured with a patience of “10” training cycles , this attribute monitors a specified metric for any signs of improvement before concluding [21].
2. **Adam:** This optimizer implements Adam’s algorithm, a stochastic gradient descent method based on the adaptive estimation of first- and second-order moments [22].
3. **Sequential:** This attribute provides training and inference functions for the model [23].
4. **Conv2D:** A 2D convolution layer that generates a convolution kernel rolled over input layers, producing an output tensor [24].
5. **Relu:** Applied to activate the rectified linear unit activation function [25].
6. **MaxPooling2D:** This attribute performs a maximum pooling operation for 2D spatial data [26].
7. **Flatten:** Used to flatten the input without affecting the batch size [27].
8. **Dense:** This layer applies weights to all nodes of the preceding layer [28].
9. **Dropout:** During training, this attribute randomly sets input units to 0 with a specified frequency at each step, helping to prevent overfitting [29].
10. **SoftMax:** It converts a vector of values into a probability distribution [25].
11. **Compile:** A method that accepts a metric argument and a list of metrics [30].
12. **Categorical cross entropy:** This attribute calculates the cross-entropy loss between labels and predictions [31].
13. **Accuracy:** It calculates the frequency with which predictions match [32] the labels.

2.6.4. Training with K-fold

Model training is executed utilizing the following attributes:

1. **K-fold:** The dataset is divided into K folds, with each fold serving as the test set while the remaining dataset is used as the training set [33, 34].
2. **Stratified Fold:** This attribute ensures further cross-validation by preserving the distribution of classes in the dataset across both training and validation splits [35].
3. **To categorical:** It converts a class vector (integers) into a binary class array [36].
4. **Image Data Generator:** This attribute facilitates the generation of training blocks and performs data augmentation by increasing the number of images through modifications such as zooming, scaling, horizontal flipping, etc [37].
5. **Fit:** This function is used to train the model for a fixed number of epochs (iterations on a dataset) [38].

6. **Model Checkpoint:** It serves as a callback to save the Kera's model or model weights at specified intervals [39].

7. **Create model:** This function is responsible for creating and training a new model instance [40].

Similarly, another CNN is proposed to create the model, utilizing "ResNet152V2," "DenseNet121," and "EfficientNetB6" as the chosen architectures. Table 1 illustrates a comparison of the model's performance using these deep learning architectures:

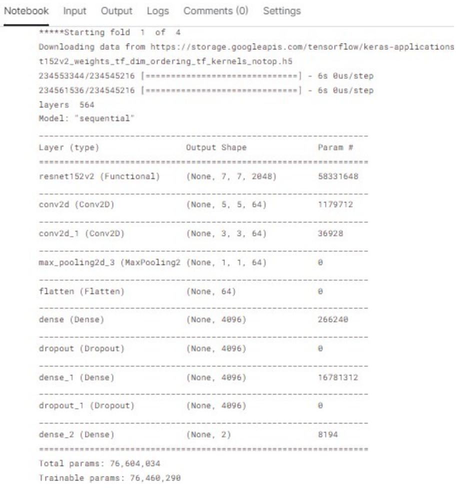
Table 1. Comparison of deep learning architectures

ResNet152V2	Precision (%)	Recall (%)	F1 Score (%)
With oral cancer	90	93	92
No oral cancer	85	80	82
Accuracy		0.8855	
AUC		0.863244514106583	
DenseNet121	Precision (%)	Recall (%)	F1 Score (%)
With oral cancer	85	94	90
No oral cancer	86	68	76
Accuracy		0.8550	
AUC		0. 812173458725183	

Regarding the use of "Efficient Net B6", a memory problem arises at the beginning of fold 3 during model training, preventing the completion of the process.

Similarly, although the results of the model using the "ResNet152V2," "DenseNet121," and "MobileNet" architectures on the same dataset are comparable, the number of hyperparameters used in MobileNet is smaller. Figures 5, 6, and 7 illustrate the number of hyperparameters obtained for each architecture:

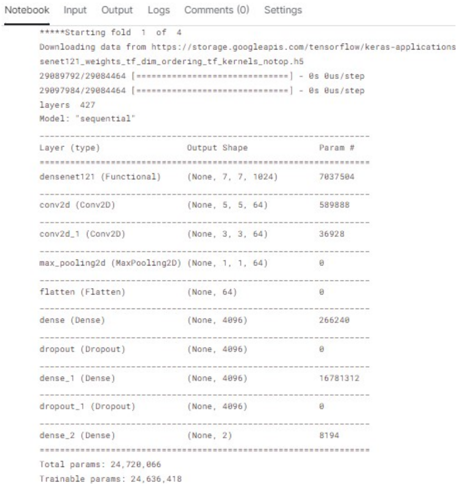
"ResNet152V2" used 76MM, as depicted in Figure 5.



```
*****Starting fold  1  of  4
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications
t152v2_weights_tf_dim_ordering_tf_kernels_notop.h5
234553344/234545216 [==============================] - 6s 0us/step
234561536/234545216 [==============================] - 6s 0us/step
layers  564
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
resnet152v2 (Functional)     (None, 7, 7, 2048)        58331648
_________________________________________________________________
conv2d (Conv2D)              (None, 5, 5, 64)          1179712
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 3, 3, 64)          36928
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 1, 1, 64)          0
_________________________________________________________________
flatten (Flatten)            (None, 64)                0
_________________________________________________________________
dense (Dense)                (None, 4096)              266240
_________________________________________________________________
dropout (Dropout)            (None, 4096)              0
_________________________________________________________________
dense_1 (Dense)              (None, 4096)              16781312
_________________________________________________________________
dropout_1 (Dropout)          (None, 4096)              0
_________________________________________________________________
dense_2 (Dense)              (None, 2)                 8194
=================================================================
Total params: 76,604,034
Trainable params: 76,460,290
```

Figure 5. ResNet152V2 hyperparameter number

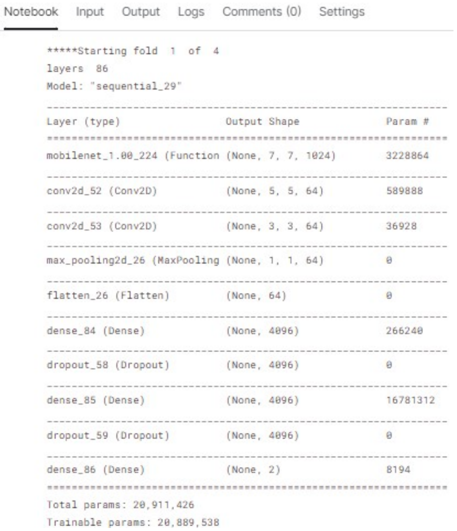
"DenseNet121" used 24MM, as depicted in Figure 6.



```
*****Starting fold  1  of  4
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications
senet121_weights_tf_dim_ordering_tf_kernels_notop.h5
29089792/29084464 [==============================] - 0s 0us/step
29097984/29084464 [==============================] - 0s 0us/step
layers  427
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
densenet121 (Functional)     (None, 7, 7, 1024)        7037504
_________________________________________________________________
conv2d (Conv2D)              (None, 5, 5, 64)          589888
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 3, 3, 64)          36928
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 1, 1, 64)          0
_________________________________________________________________
flatten (Flatten)            (None, 64)                0
_________________________________________________________________
dense (Dense)                (None, 4096)              266240
_________________________________________________________________
dropout (Dropout)            (None, 4096)              0
_________________________________________________________________
dense_1 (Dense)              (None, 4096)              16781312
_________________________________________________________________
dropout_1 (Dropout)          (None, 4096)              0
_________________________________________________________________
dense_2 (Dense)              (None, 2)                 8194
=================================================================
Total params: 24,720,066
Trainable params: 24,636,418
```

Figure 6. DenseNet121 hyperparameter number

"MobileNet" used 24MM, as depicted in Figure 7.



```
*****Starting fold  1  of  4
layers  86
Model: "sequential_29"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
mobilenet_1.00_224 (Function (None, 7, 7, 1024)        3228864
_________________________________________________________________
conv2d_52 (Conv2D)           (None, 5, 5, 64)          589888
_________________________________________________________________
conv2d_53 (Conv2D)           (None, 3, 3, 64)          36928
_________________________________________________________________
max_pooling2d_26 (MaxPooling (None, 1, 1, 64)          0
_________________________________________________________________
flatten_26 (Flatten)         (None, 64)                0
_________________________________________________________________
dense_84 (Dense)             (None, 4096)              266240
_________________________________________________________________
dropout_58 (Dropout)         (None, 4096)              0
_________________________________________________________________
dense_85 (Dense)             (None, 4096)              16781312
_________________________________________________________________
dropout_59 (Dropout)         (None, 4096)              0
_________________________________________________________________
dense_86 (Dense)             (None, 2)                 8194
=================================================================
Total params: 20,911,426
Trainable params: 20,889,538
```

Figure 7. MobileNet hyperparameter number

2.6.5. Ground truth

The model verification process is conducted using the entire validation dataset, employing the following attributes:

1. **Evaluate:** This function returns the loss value and metric values of the model in test mode [38].

2. **Predict:** It generates output predictions for the input samples [38].

3. **Confusion matrix:** Calculating the confusion matrix is used to evaluate the accuracy of a classification [41].

4. **Subplot:** This attribute obtains the index position in a grid with "n" rows and "n" columns [42].

5. **Heatmap:** It is utilized to obtain a class activation heatmap for an image classification model [43].

6. **Set-ticklabels:** This function sets the target names for the confusion matrix.

7. **Roc curve:** This attribute calculates the ROC [44].

8. **Roc-auc-core:** It calculates the ROC AUC from the prediction scores [45].

2.6.6. Plotting AUC curves

Figure 8 depicts a plot of the true positive rate against the false positive rate, illustrating the AUC through lines. This visualization allows for the observation of the relationship between these two variables.

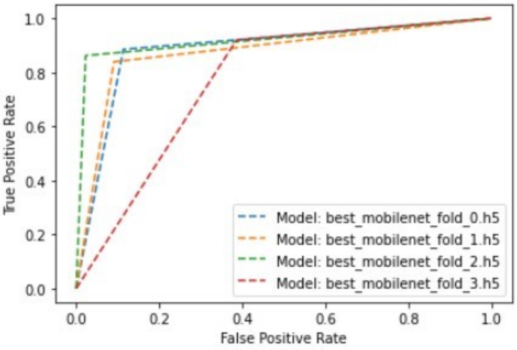


Figure 8. Plotting AUC curves

It is confirmed that the proposed CNN model demonstrates high performance in classifying the presence and absence of oral cancer.

3. Results and discussion

Figure 9 illustrates the plot of accuracy values against the number of epochs, using lines to visualize the relationship between these two variables.

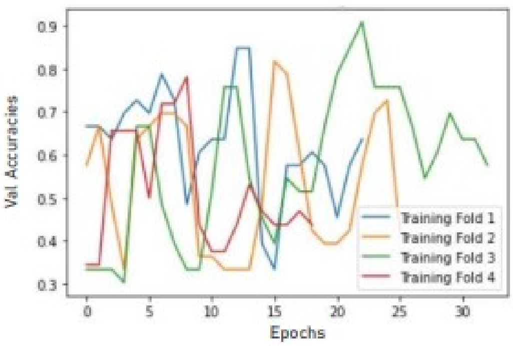


Figure 9. Plotting model accuracy per fold

When training the model, accuracy is obtained for each fold:

In the first training, with fold 1, a “value accuracy” of 0.84848 is achieved. In the second training, with fold 3, a "value accuracy " of 0.81818 is achieved. In the third training, with fold 3, a "value accuracy " of 0.90909 is achieved. In the fourth training, with fold 4, a "value accuracy " of 0.78125 is achieved.

It is observed that fold 1 attains a good "value accuracy", fold 2 decreases the "value accuracy", fold 3 achieves the highest "value accuracy" with a value of 0.90909, and fold 4 decreases the " value accuracy".

During the model validation, the following metrics are obtained for the model saved as “best mobilenet fold 0.h5,” as illustrated in Table 2.

Table 2. Simple model performance metrics with fold 1

	Precision (%)	Recall (%)	F1 Score (%)
With oral cancer	80	89	84
No oral cancer	94	89	91
AUC	0.8857105538140021		

The “best mobilenet fold 1.h5” achieves the following rankings, as depicted in Table 3.

Table 3. Simple model performance metrics with fold 2

	Precision (%)	Recall (%)	F1 Score (%)
With oral cancer	74	91	82
No oral cancer	95	84	89
AUC	0.8740856844305119		

The “best mobilenet fold 2.h5” achieves the following rankings, as depicted in Table 4.

Table 4. Simple model performance metrics with fold 3

	Precision (%)	Recall (%)	F1 Score (%)
With oral cancer	78	98	87
No oral cancer	99	86	92
AUC	0.9196708463949843		

The “best mobilenet fold 3.h5” achieves the following rankings, as depicted in Table 5.

Table 5. Simple model performance metrics with fold 4

	Precision (%)	Recall (%)	F1 Score (%)
With oral cancer	79	61	69
No oral cancer	82	92	87
AUC	0.7665882967607105		

It is concluded that the “best mobilenet fold 2.h5” model stands out as the optimal choice, exhibiting the highest precision (78% and 99%), recall (98% and 86%), F1 score (87% and 92%), and AUC (0.9196708463949843) for both non-cancerous and cancerous cases, surpassing the performance of other models.

Comparatively, among the architectures "ResNet152V2," "DenseNet121," and "MobileNet," it is demonstrated that the "MobileNet" architecture is optimal in terms of resource optimization, utilizing 20 million hyperparameters, whereas the "ResNet152V2" and "DenseNet121" architectures use a significantly greater number of hyperparameters.

In addition to the performance metrics of the “ResNet152V2”, “DenseNet121” and “EfficientNetB6” architectures evaluated, the model featured in [7] demonstrated a precision of 84.3%, recall of 83.0%, F1 score of 83.6%, and AUC of 0.8974. In contrast, the model proposed in this study exhibited improvements, achieving a precision of 88.5%, recall of 92.0%, F1 score of 89.5%, and AUC of(0.9196708463949843). Thus, a noticeable enhancement in overall performance ranging between 2% and 9% is observed.

4. Conclusions

This study highlights the potential of AI in addressing oral health issues, particularly oral cancer, which affects a significant portion of the population. The research emphasizes the effectiveness of DL and concludes that CNNs are a suitable DL algorithm for processing images of the mucosa and the oral cavity. CNNs take these images as input and assign weights to specific elements to distinguish between them. The choice of the MobileNet CNN is justified due to its ability to reduce model size and computation by replacing standard convolution filters with deep, pointwise convolutions. The study utilizes the open-source Kaggle platform and implements the model using the Python programming language. The evaluation of various performance metrics yields an accuracy of 0.90909, confirming that the proposed CNN model demonstrates high diagnostic performance for oral cancer. Regarding image quantity, the study specifies that using more images enhances the proposed DL model’s results. Additionally, evaluating different CNN architectures helps to understand their performance, facilitating the determination of the most optimal model. Ultimately, this research asserts that the developed model is ready for practical application, offering valuable support for dental decision-making in real-time diagnostic scenarios.

4.1. Future work

Continued efforts in collecting more images of lips, mucosa, and the oral cavity depicting various oral conditions, including oral cancer, will be a focal point for future studies. Augmenting the image dataset and collaborating with public or private clinical institutions for evaluation are believed to significantly improve the results and facilitate the practical application of the model. Recognizing the pivotal role of large datasets in optimizing DL algorithms, the current results are promising and serve as a first step to advance this line of research. Furthermore, upcoming research endeavors will aim to evaluate the performance of the proposed CNN method in diagnosing a broader spectrum of oral diseases.

References

[1] L. A. Zanella-Calzada, C. E. Galván-Tejada, N. M. Chávez-Lamas, J. Rivas-Gutiérrez, R. Magallanes-Quintanar, J. M. Celaya-Padilla, J. I. Galván-Tejada, and H. Gamboa-Rosales, “Deep artificial neural networks for the diagnostic of caries using socioeconomic and nutritional features as determinants: Data from NHANES 2013–2014,” *Bioengineering*, vol. 5, no. 2, 2018. [Online]. Available: https://doi.org/10.3390/bioengineering5020047

[2] J. Shan, R. Jiang, X. Chen, Y. Zhong, W. Zhang, L. Xie, J. Cheng, and H. Jiang, “Machine learning predicts lymph node metastasis in early-stage oral tongue squamous cell carcinoma,” *Journal of Oral and Maxillofacial Surgery*, vol. 78, no. 12, pp. 2208–2218, 2020. [Online]. Available: https://doi.org/10.1016/j.joms.2020.06.015

[3] A. M. Bur, A. Holcomb, S. Goodwin, J. Woodroof, O. Karadaghy, Y. Shnayder, K. Kakarala, J. Brant, and M. Shew, “Machine learning to predict occult nodal metastasis in early oral squamous cell carcinoma,” *Oral Oncology*, vol. 92, pp. 20–25, 2019. [Online]. Available: https://doi.org/10.1016/j.oraloncology.2019.03.011

[4] O. Kwon, T.-H. Yong, S.-R. Kang, J.-E. Kim, K.-H. Huh, M.-S. Heo, S.-S. Lee, S.-C. Choi, and W.-J. Yi, “Automatic diagnosis for cysts and tumors of both jaws on panoramic radiographs using a deep convolution neural network,” *Dentomaxillofacial Radiology*, vol. 49, no. 8, p. 20200185, Dec 2020. [Online]. Available: https://doi.org/10.1259/dmfr.20200185

[5] X. Zhang, Y. Liang, W. Li, C. Liu, D. Gu, W. Sun, and L. Miao, “Development and evaluation of deep learning for screening dental caries from oral photographs,” *Oral Diseases*, vol. 28, no. 1, pp. 173–181, 2022. [Online]. Available: https://doi.org/10.1111/odi.13735

[6] H.-J. Chang, S.-J. Lee, T.-H. Yong, N.-Y. Shin, B.-G. Jang, J.-E. Kim, K.-H. Huh, S.-S. Lee, M.-S. Heo, S.-C. Choi, T.-I. Kim, and W.-J. Yi, “Deep learning hybrid method to automatically diagnose periodontal bone loss and stage periodontitis,” *Scientific Reports*, vol. 10, no. 1, p. 7531, May 2020. [Online]. Available: https://doi.org/10.1038/s41598-020-64509-z

[7] H. Lin, H. Chen, L. Weng, J. Shao, and J. Lin, “Automatic detection of oral cancer in

smartphone-based images using deep learning for early diagnosis," *Journal of Biomedical Optics*, vol. 26, no. 8, p. 086007, 2021. [Online]. Available: https://doi.org/10.1117/1.JBO.26.8.086007

[8] W. Li, Y. Liang, X. Zhang, C. Liu, L. He, L. Miao, and W. Sun, "A deep learning approach to automatic gingivitis screening based on classification and localization in RGB photos," *Scientific Reports*, vol. 11, no. 1, p. 16831, Aug 2021. [Online]. Available: https://doi.org/10.1038/s41598-021-96091-3

[9] M. A. Dávila Olivos and F. M. Santos López, "Prediction models of oral diseases: A systematic review of the literature," in *Emerging Research in Intelligent Systems*, G. F. Olmedo Cifuentes, D. G. Arcos Avilés, and H. V. Lara Padilla, Eds. Cham: Springer Nature Switzerland, 2024, pp. 309–322. [Online]. Available: https://doi.org/10.1007/978-3-031-52255-0_22

[10] A. Pujara, "Image classification with mobilenet," *Analytics Vidhya*, 2020. [Online]. Available: https://n9.cl/coutpg

[11] F. Melo, *Receiver Operating Characteristic (ROC) Curve.* New York, NY: Springer New York, 2013, pp. 1818–1823. [Online]. Available: https://doi.org/10.1007/978-1-4419-9863-7_242

[12] ——, *Area under the ROC Curve.* New York, NY: Springer New York, 2013, pp. 38–39. [Online]. Available: https://doi.org/10.1007/978-1-4419-9863-7_209

[13] H. Rhys, *Machine Learning with R, the tidyverse, and mlr.* Manning Publications, 2020. [Online]. Available: https://n9.cl/q3hijw

[14] LabTests Online UK. (2018) Accuracy, precision, specificity & sensitivity. Association for Laboratory Medicine. [Online]. Available: https://n9.cl/8cvyg

[15] J. Martínez Heras. (2018) Machine learning lectures esa. GitHub, Inc. [Online]. Available: https://n9.cl/7yim6

[16] OMS. (2021) Salud bucodental. Organización Mundial de la Salud. [Online]. Available: https://n9.cl/zpz0f

[17] Kaggle. (2020) Oral cancer (lips and tongue) images. Kaggle. [Online]. Available: https://n9.cl/7ftbq

[18] Keras. (2022) Mobilenet, mobilenetv2, and mobilenetv3. Keras. [Online]. Available: https://n9.cl/dcvs2

[19] S. Sharma. (2022) Epoch vs batch size vs iterations. Medium. [Online]. Available: https://n9.cl/wlxncj

[20] S. Manna. (2022) K-fold cross validation for deep learning models using keras. Medium. [Online]. Available: https://n9.cl/hmyvr

[21] Keras. (2022) Earlystopping. Keras. [Online]. Available: https://n9.cl/undx7

[22] ——. (2022) Adam. Keras. [Online]. Available: https://n9.cl/x9m53

[23] ——. (2022) The sequential class. Keras. [Online]. Available: https://n9.cl/yi56j

[24] GeeksforGeeks. (2022) Keras.conv2d class. Geeks for Geeks. [Online]. Available: https://n9.cl/6bemi

[25] Keras. (2022) Layer activation functions. Keras. [Online]. Available: https://n9.cl/d9yeb

[26] ——. (2022) Maxpooling2d layer. Keras. [Online]. Available: https://n9.cl/51sbk

[27] ——. (2022) Flatten layer. Keras. [Online]. Available: https://n9.cl/cufk4

[28] I. Hull, *Dense layers – Introduction to tensorflow in Python.* DataCamp. [Online]. Available: https://n9.cl/hny28

[29] Keras. (2022) Dropout layer. Keras. [Online]. Available: https://n9.cl/02hdv

[30] ——. (2022) Metrics. Keras. [Online]. Available: https://n9.cl/jmihj

[31] ——. (2022) Probabilistic metrics. Keras. [Online]. Available: https://n9.cl/b3w2a

[32] ——. (2022) Accuracy metrics. Keras. [Online]. Available: https://n9.cl/7l3dt

[33] Data Science Team. (2020) Validación cruzada k-fold. Data Science Team. [Online]. Available: https://n9.cl/c2i0bp

[34] R. Delgado. (2018) Introducción a la validación cruzada (k-fold cross validation) en R. Amazonaws. [Online]. Available: https://n9.cl/ijyq

[35] S. Yildirim. (2020) How to train test split : Kfold vs stratifiedkfold. Medium. [Online]. Available: https://n9.cl/ymp9q

[36] Keras. (2022) Python & numpy utilities. Keras. [Online]. Available: https://n9.cl/zrhgh

[37] J. Utrera Burgal. (2019) Tratamiento de imágenes usando imagedatagenerator en keras. Knowmad mood. [Online]. Available: https://n9.cl/5gobr

[38] Keras. (2022) Model training apis. Keras. [Online]. Available: https://n9.cl/4gjr6

[39] ——. (2022) Modelcheckpoint. Keras. [Online]. Available: https://n9.cl/wvut7

[40] Tensorflow. (2022) Guardar y cargar modelos. TensorFlow. [Online]. Available: https://n9.cl/cjflnu

[41] Scikit Learn. (2022) sklearn.metrics.confusion matrix. Scikit-learn developers. [Online]. Available: https://n9.cl/ya6b1h

[42] Matplotlib. (2022) matplotlib.pyplot.subplot. Matplotlib. [Online]. Available: https://n9.cl/vlv1fe

[43] Keras. (2022) Grad-cam class activation visualization. Keras. [Online]. Available: https://n9.cl/r5l7k

[44] Scikitlearn. (2022) sklearn.metrics.roc curve. Scikitlearn. [Online]. Available: https://n9.cl/qles5

[45] ——. (2022) sklearn.metrics.roc auc score. Scikitlearn. [Online]. Available: https://n9.cl/1zf6r