Online-Journals.org

iJOE International Journal of Online and Biomedical Engineering

iJOE | eISSN: 2626-8493 | Vol. 20 No. 8 (2024) | OPEN ACCESS

https://doi.org/10.3991/ijoe.v20i08.48421

PAPER

Mobile Application Based on Convolutional Neural Networks for Pterygium Detection in Anterior Segment Eye Images at Ophthalmological Medical Centers

Edward Jordy Ticlavilca-Inche[1], Maria Isabel Moreno-Lozano[1], Pedro Castañeda[1](✉), Sandra Wong-Durand[1], Alejandra Oñate-Andino[2]

[1]Universidad Peruana de Ciencias Aplicadas (UPC), Lima, Perú

[2]Escuela Superior Politécnica de Chimborazo (ESPOCH), Riobamba, Ecuador

pcsipcas@upc.edu.pe

ABSTRACT

This article introduces an innovative mobile solution for Pterygium detection, an eye disease, using a classification model based on the convolutional neural network (CNN) architecture ResNext50 in images of the anterior segment of the eye. Four models (ResNext50, ResNet50, MobileNet v2, and DenseNet201) were used for the analysis, with ResNext50 standing out for its high accuracy and diagnostic efficiency. The research, focused on applications for ophthalmological medical centers in Lima, Peru, explains the process of development and integration of the ResNext50 model into a mobile application. The results indicate the high effectiveness of the system, highlighting its high precision, recall, and specificity, which exceed 85%, thus showing its potential as an advanced diagnostic tool in ophthalmology. This system represents a significant tool in ophthalmology, especially for areas with limited access to specialists, offering a rapid and reliable diagnosis of Pterygium. The study also addresses the technical challenges and clinical implications of implementing this technology in a real-world context.

KEYWORDS

automatic pterygium classification, deep learning system, photograph of anterior segment of the eye, pterygium detection

1 INTRODUCTION

Pterygium is an ocular disease that involves the abnormal growth of tissues covering the corneal regions, preventing light from passing through the pupil partially or completely, which can lead to discomfort, blurred vision, and even a predisposition to other chronic diseases such as blindness [1]. Currently, there are highly effective methods for pterygium treatment; however, these methods are inefficient due to numerous limitations. One major limitation is the need to have a specialized ophthalmologist with extensive experience in the field, who must manually classify

Ticlavilca-Inche, E.J., Moreno-Lozano, M.I., Castañeda, P., Wong-Durand, S., Oñate-Andino, A. (2024). Mobile Application Based on Convolutional Neural Networks for Pterygium Detection in Anterior Segment Eye Images at Ophthalmological Medical Centers. *International Journal of Online and Biomedical Engineering (iJOE)*, 20(8), pp. 115–138. https://doi.org/10.3991/ijoe.v20i08.48421

Article submitted 2024-02-07. Revision uploaded 2024-03-18. Final acceptance 2024-03-18.

the areas of the eye image, resulting in a tedious and time-consuming process [2]. Regardless of the ophthalmologist's experience, both classification and interpretation of results are subject to a certain degree of subjectivity, potentially affecting the diagnosis [3]. In terms of cost, the ophthalmological examination process and diagnostic tools are expensive and time-consuming, limiting accessibility to these resources, especially in rural areas [3], [4].

In this context, Artificial Intelligence (AI), specifically deep learning, has gained prominence in recent years, as it is characterized as a unique category of machine learning approach that employs multiple layers to gradually extract more sophisticated features from the original input. Which, in [5], show that it is advantageous, since when working with data in matrix format, such as pixels in an image or frames in a video, it allows a more complex representation of the information. This technology, in the field of ophthalmology, has given rise to numerous research efforts focused on the detection of ocular diseases using Convolutional Neural Networks (CNN). In [6], the detection of pterygium was proposed through an RFRC model (Faster RCNN based on ResNet101) and the SRU-Net model (U-Net based on SE-ResNeXt50) for its segmentation. Additionally, in [7], the use of deep learning algorithms for automatic pterygium detection was proposed, employing the VGG16 model. However, there are few studies employing ResNext50 model, which has shown successful results in detecting other eye diseases, such as the categorization of patients with diabetic retinopathy [8]. Moreover, none of the mentioned studies demonstrate the implementation of the detection model in mobile applications, which is crucial to provide solutions to medical centers that need them.

In this article, we present the development of a system aimed at detecting Pterygium disease, which consists of a mobile application, using a detection model in the anterior segment of the eye images based on the ResNext50 Convolutional Neural Network (CNN) architecture trained with an external dataset provided by the Peruvian Pterygium Center [9] and the authors of the articles [4], [10], [11], [12]. Through this application, users can take a photograph using their phone's camera, allowing the detection of the presence and severity of pterygium in the patient (mild or advanced). The development also involves the selection and training of the CNN model among 4 architectures (ResNext50, ResNet50, MobileNet v2, and DenseNet201) following the KDD methodology. The results indicate that the ResNeXt-50 architecture achieved a maximum accuracy of 86%, a maximum F1 score of 88%, a maximum recall of 90%, a maximum specificity of 95%, and an accuracy of 89% for our dataset.

The rest of the article is structured as follows: In the second section, a literature review will be presented. The design of the final solution is described in section 3. Section 4 details the model implementation. This includes description and architecture, dataset creation, experimental environments definition and results, confusion matrix and ROC curve. The integration into the mobile application will be detailed in Section 5. Finally, the discussion and conclusion will be presented in sections 6 and 7, respectively.

2 RELATED WORK

Currently, there is a wide variety of research on the performance of Artificial Intelligence (AI) and Deep Learning (DL) in different areas, such as education [13] and work [14]. The health field is particular due to the different specialties that exist and the rigor that must be applied in research aimed at DL solutions so that doctors trust

their results. Thus, in the field of ophthalmology, a wide range of research has been carried out on ocular diseases, including pterygium (see Table 1), oriented towards different approaches. Among the found models, they can be categorized into two groups: segmentation and classification. In segmentation, the group of pixels in the anterior segment images that exclusively correspond to the pterygium-infected area is separated or highlighted from the rest. Notable architectures in this group include classic segmentation models already applied to pterygium, such as U-Net, DeepLab, Fully Convolution Network (FCN), SegNet, PSPNet, ENet, DenseNet [1], [15], obtaining efficient results. In [1], an accuracy of 93.33% is achieved, while in [15], a maximum of 91.67% is reported. Additionally, models based on architectures commonly used for detection, as in [16] comparing AlexNet, ResNet, and VGG16, achieve an accuracy of 99% for the latter. Another segmentation application for pterygium is presented in [2], where an accuracy of 91.27% is achieved using SVM.

On the other hand, classification involves determining whether the input anterior segment photographed image (ASPI) belongs to a class, usually corresponding to the severity of the disease. In this group, numerous applications focus on detecting various eye diseases such as diabetic retinopathy [17], glaucoma [18], macular degeneration [19], among others. Most of these investigations use models like ResNet50, AlexNet, LSTM, and Inception, which have also been applied to pterygium. Common models include VGG16 [4], [7], [20], [21], ResNet, in some of its variants such as ResNet18, ResNet50, or ResNet101, [4], [20], [21], AlexNet and GoogleNet [4], [20]. Other architectures not as common but also relevant include DenseNet201 [4], the EfficientNet family [22] and the MobileNet family. Among all these models, the ones that stood out the most in their respective studies were VGG16 with a maximum specificity of 100% for a dataset of 772 ASPIs, EfficientNet-B6 with a specificity of 99.64% for a dataset of 1220 ASPIs, and MobileNet2 with a specificity of 98.43% for a dataset of 436 ASPIs.

Additionally, a relatively new architecture called ResNext50, a variant of ResNet50, was found with interesting results. This architecture has been applied outside the ocular domain, as seen in [23], [24], [25], where it achieved better results than other cited models. In the study [23], the model achieved a 99% accuracy and a maximum of 100% precision and recall in detecting harmful algae, surpassing MobileNet-V2, VGG16, and AlexNet. In [25], ResNext101 was combined with DenseNet169 and XceptionNet41, resulting in an F1 score of 87.74% and an accuracy of 95.75% in detecting alopecia. In [24], on the other hand, it achieved an accuracy of 99.32%, specificity of 99.01%, and recall of 100% in detecting COVID-19, surpassing AlexNet, GoogleNet, ResNet-50, Se-ResNet-50, DenseNet121, Inception V4, and Inception ResNet V2. In the field of ophthalmology, it has also been applied to diabetic retinopathy in [8], outperforming AlexNet, GoogleNet, Inception V4, and Inception ResNet V2 with an accuracy of 97.53%, specificity of 97.92%, and recall of 95.35% for a dataset of 5333 fundus images. Although this model has been applied to pterygium before, a modification of it was used for disease segmentation in [6], where SRU-Net (U-Net based on SE-ResNeXt50) is proposed and achieved an F1 score of 89.81%, sensitivity of 87.09%, specificity of 96.68%, and AUC of 92.95% in pterygium segmentation with 20,987 images taken with slit-lamp and 1,094 images taken with smartphones.

According to the research carried out, ResNext50 has demonstrated significant potential both within and beyond the ophthalmological field, as evidenced by results that even surpass several well-known models. Therefore, this research proposes the development of a model based on ResNext50 for the detection and classification of pterygium. The following Table 1 provides a summary of the research conducted to date.

Table 1. List of studies reviewed during the state-of-the-art phase

Item	Illness	Approach	Architectures	Dataset	Best Result	Resource
1	Pterygium	Segmentation and Classification	Artificial Neural Network (ANN) and Support Vector Machine (SVM)	3017 ASPIs	**Segmentation** Accuracy: 0.9127 **Classification** Sensitivity: 88.7% Specificity: 88.3% AUC: 95.6%	[2]
2	Pterygium	Detection	VGG 16	1366 with pterygium and 1566 without pterygium	Sensitivity: 87.2% Specificity: 99.4% AUC: 99.7%	[7]
3	Pterygium	Segmentation	Group-PPM-Net (based on FC-DenseNet) DeepLab V3 Stacked U-Net PSP-Net FCN FC-DenseNet U-Net DeepLab V2 SegNet	328 ASPIs	**Group-PPM-Net:** Accuracy: 0.9330 Intersection over union: 0.8640 Hausdorff distance: 11.5474 Jaccard index: 0.7966	[1]
4	Pterygium	Classification and Segmentation	**CLASSIFICATION** RFRC (based on ResNet101) **SEGMENTATION** SRU-Net (based on SE-ResNeXt50)	20987 from slit lamp and 1094 from smartphone	**CLASSIFICATION** Accuracy: 95.24% Average: 0.9563 Intersection over union: 0.9100 **SEGMENTACION** F1 score: 0.8981 Sensitivity: 0.8709 Specificity: 0.9668 AUC: 0.9295	[6]
5	Pterygium	Detection	VggNet16-wbn (based on VggNet16) AlexNet VggNet16 VggNet19 GoogleNet ResNet101 DenseNet201	772 ASPIs	**VggNet16-wbn:** Accuracy: 99.22% Sensitivity: 98.45% Specificity: 100.0% AUC: 100%	[4]
6	Pterygium	Classification	Based on EfficientNet-B6	1220 ASPIs	Accuracy: 94.68% Kappa: >85% Sensitivity: 100% Specificity: 99.64% F1-scores: 99.74% AUC: 100%	[22]
7	Pterygium	Classification	MobileNet2 MobileNet1 AlexNet VGG16 ResNet18	436 ASPIs	**MobileNet2:** Sensitivity: 96.72% Specificity: 98.43% F1-scores: 96.72% AUC: 0.976 Accuracy: 85.11% Kappa: 77.64%	[21]

(Continued)

Table 1. List of studies reviewed during the state-of-the-art phase *(Continued)*

Item	Illness	Approach	Architectures	Dataset	Best Result	Resource
8	Pterygium	Segmentation	AlexNet VGG16 ResNet18 ResNet50	367 with Pterygium and 367 without Pterygium	**VGG16:** Accuracy: 99% Kappa: 98% Sensitivity: 98.67% Specificity: 99.33% F1-score: 99%	[16]
9	Pterygium	Detection and classification	Ensemble DL (Resnet18, Alexnet, Googlenet y Vgg11)	172 ASPIs	**Ensemble DL:** Accuracy: 94,2% AUC: 0,978%	[20]
10	Pterygium	Segmentation	Unet	176 with Pterygium and 61 without Pterygium	Sensitivity: 80%–91,67% Specificity: 91,67%–100% F1score: 81,82%–94,34% Accuracy: 86,67%–91,67%	[15]
11	Macular Degeneration	Classification	HDF-Net ResNet50 AlexNet	No info	**HDF-Net:** AUC: 98.9% Accuracy: 93.6% Sensitivity: 93.3% Specificity: 93.8%	[19]
12	Diabetic retinopathy	Classification	Xception IncResNet ResNet Inception	35126 fundus images	Accuracy: 89%–95% AUC: 95%–98% Sensitivity: 74%–86% Specificity: 93%–97%	[17]
13	Glaucoma	Classification	LSTM	11242 fundus images	Accuracy: 91%–93% AUC: 0.89%–0.93%	[18]
14	Harmful bloom-forming algae	Classification	MobileNet V-2 VGG-16 AlexNet ResNeXt-50	450 algal images	**ResNeXt-50:** Accuracy: 99% F1 score: 100% Precision: 100% Recall: 100%	[23]
15	Alopecia	Classification	ResNet ResNeXt DenseNet XceptionNet	18249 alopecia images	**ResNext101 + DenseNet169 + XceptionNet41:** F1 score: 87.74% Accuracy: 95.75%	[25]
16	Covid-19	Classification	AlexNet GoogleNet ResNet-50 Se-ResNet-50 DenseNet121 Inception V4 Inception ResNet V2 ResNeXt-50 Se-ResNeXt-50	728 confirmed Covid-19 images	**Se-ResNeXt-50:** Accuracy: 99.32% Recall: 100% Specificity: 99.01% Precision: 97.83% F1: 98.9%	[24]
17	Diabetic retinopathy	Classification	AlexNet GoogleNet Inception V4 Inception ResNet V2 ResNeXt-50	5333 fundus images	**ResNeXt-50:** Accuracy: 97.53% Precision: 89.13% Specificity: 97.92% Recall: 95.35% F1 score: 86.31%	[8]

3 OVERVIEW OF THE PROPOSED SYSTEM

The system allows ophthalmological specialists to capture images of the anterior segment of patients' eyes through a mobile application developed in Flutter, compatible with both Android and iOS devices. Captured images are sent over the Internet and processed in the AWS (Amazon Web Services) cloud, where the following components are located:

1. **Backend:** Responsible for managing communication with the database and the detection model. This module transmits the captured images to the detection model and receives the results for further processing.
2. **Dockerized Container:** Flask web service that uses the trained classification model with the ResNext50 architecture. This module receives the ASPIs for processing and returns the results.
3. **Database:** Responsible for storing information about specialists and patients, including captured images and detection results.

The flow depicted in Figure 1 provides a visual representation of the mentioned components and how they are interconnected.

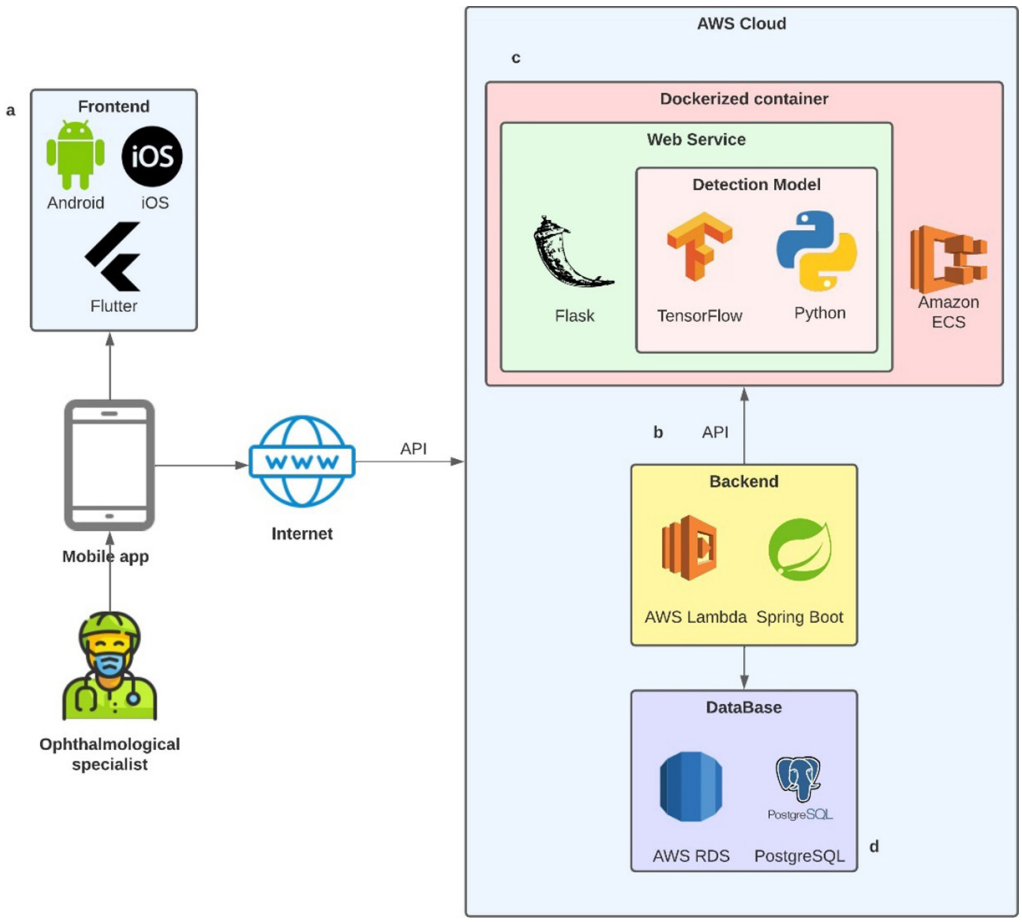

Fig. 1. Integrated architecture of the proposed system. (a) Mobile application used by a specialist. (b) Backend. (c) Dockerized container with the detection model. (d) Database

4 MODEL

4.1 Methodology

The methodology employed, Knowledge Discovery in Databases (KDD), is illustrated in Figure 2, comprising 5 stages. In the Selection stage (1), the data (ASPI) is identified, and then, in the Preprocessing stage (2), data cleaning, handling of missing values, data division by class, etc., are performed. Following this, in the Transformation stage (3), data balancing is carried out using libraries that generate images based on variations of the existing image data. In the fourth stage, Data training is conducted by assigning different weights to classes inversely proportional to the amount of data per class. For this phase, pre-trained prediction models such as ResNext50, ResNet50, MobileNet v2, and DenseNet201 were utilized and compared. Finally, in the last stage, the results obtained by the models are evaluated and interpreted.

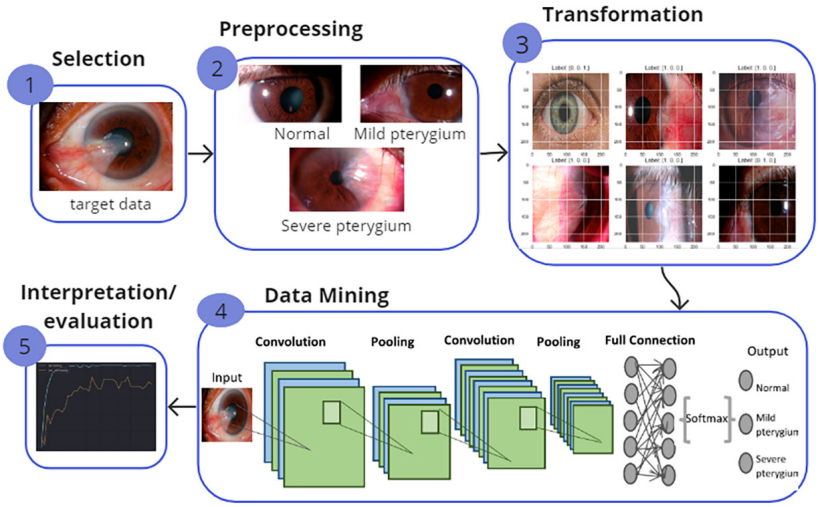

Fig. 2. Implementation flow of the model using KDD methodology

4.2 Data description

The set of ASPI images was provided by the Peruvian Pterygium Center [9] and the authors of the articles [4], [10], [11], [12]. There are 534 ASPIs, of which 474 were captured using a slit lamp and 60 were captured by smartphones. These were classified by an ophthalmic surgeon into 3 classes as shown in Figure 3: Normal (101), Mild Pterygium (167), and Severe Pterygium (266), where the latter suggests undergoing the surgery process.

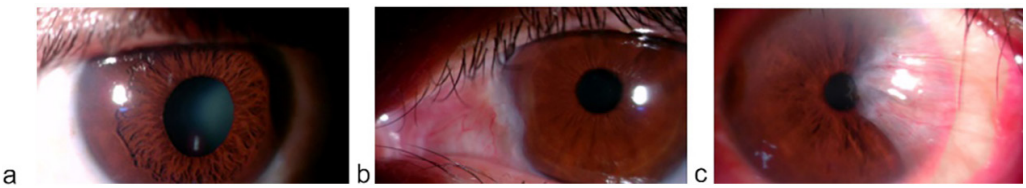

Fig. 3. Classification of ASPIs: (a) Normal. (b) Mild Pterygium. (c) Severe Pterygium

4.3 Convolutional neural network architecture

Four pre-trained CNN architectures (ResNext50, ResNet50, MobileNet v2, and DenseNet201) were selected with ImageNet weights. The results obtained were evaluated to identify the architecture with the best performance, which will be implemented in the proposed mobile solution. The fine-tuning technique will be used to take advantage of the prior knowledge of the pre-trained models during the classification of ASPIs.

ResNext50. This architecture was proposed in 2016 as an improvement to the ResNet50 architecture [26]. It is characterized by a simple and highly modular structure. Additionally, it introduces the concept of "cardinality" associated with the dimensions of depth and width of the transformation step, enhancing its capability in specific tasks such as image classification. The structure comprises five distinct sections, including convolutional blocks and identity blocks. Each convolutional block consists of three convolutional layers, while each identity block incorporates three transformation stages. The SE block is employed as a calculation unit that converts inputs into feature maps and is compatible with various convolutional neural network architectures and residual networks. This SE block is placed before performing the sum, increasing the computational load but, in return, enhancing the accuracy of the ResNext-50 model compared to the ResNet-50 model. The complete architecture of ResNext50 is illustrated in Figure 4.

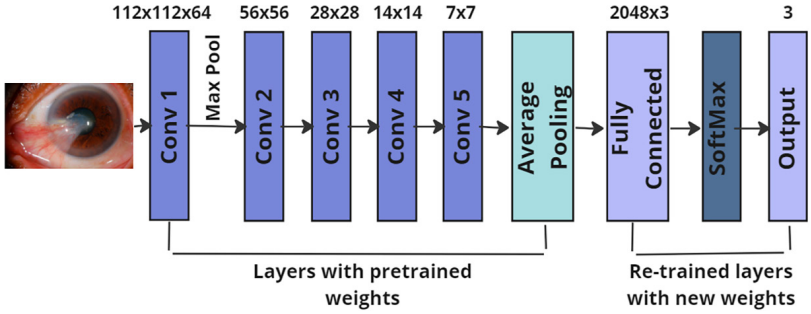

Fig. 4. Pre-trained architecture of ResNeXt-50

ResNet50. In this architecture, the layers of the network are redesigned through learning, employing residual functions related to the inputs of each layer. ResNet, also known as a residual network, introduces the concept of skip connections to address the issue of vanishing gradients [24]. This technique prevents distortions that may arise when increasing the depth and complexity of the network, facilitating the creation of deeper Convolutional Neural Networks (CNNs) without impacting accuracy. ResNet was one of the first CNNs to implement batch normalization. ResNet-50 consists of 50 layers and includes a convolutional layer, four convolutional blocks, a max pooling layer, and an average pooling layer, designed to mitigate the decrease in accuracy. The ResNet-50 architecture is illustrated in Figure 5.

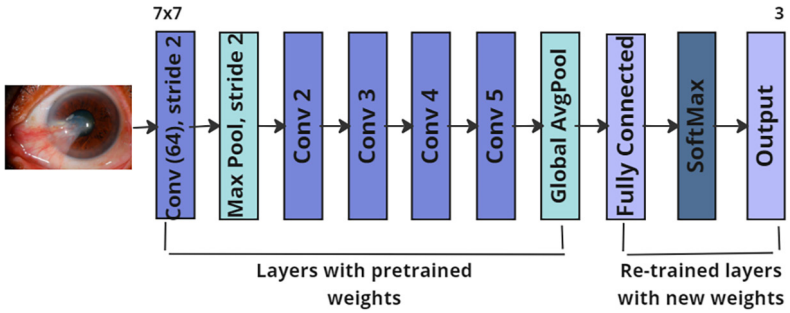

Fig. 5. Pre-trained architecture of ResNet-50

MobileNet v2. MobileNet is a simplified architecture that employs depth-wise separable convolutions to make deep convolutional neural networks lightweight, providing an efficient model for flexible and embedded vision applications. MobileNet relies on depth-wise separable convolutions, consisting of two inner core layers: depth-wise convolutions and point-wise convolutions [23]. Moreover, it is specifically designed for mobile and embedded devices. This study focuses on transfer learning using the MobileNet model with an inverted residual structure. Its basic network structure mainly includes convolutional layers, bottleneck layers, and an average pooling layer. The MobileNet v2 architecture is illustrated in Figure 6.

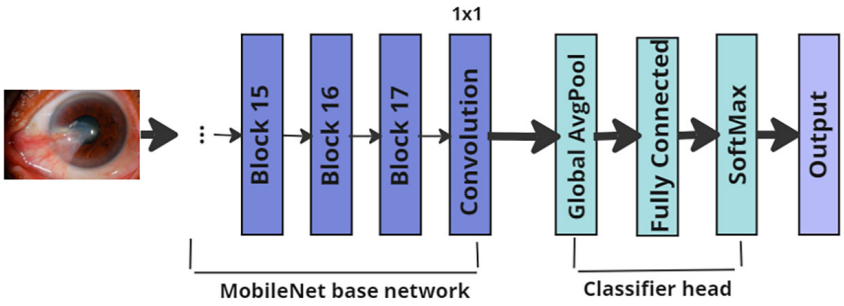

Fig. 6. Pre-trained architecture of MobileNet v2

DenseNet201. DenseNet is often compared to ResNet, as this architecture is the logical extension of ResNet. There are advantages in terms of significantly reducing the number of parameters, improving feature propagation, and avoiding gradient vanishing [25]. DenseNet is composed of convolutional layers, max pooling layers, global average pooling layers, dense blocks, and transition layers. Dense blocks employ 1×1 and 3×3 convolutions to decrease the number of parameters required for model optimization. Transition layers combine a 1×1 convolutional layer with a 2×2 average pooling layer. DenseNet addresses challenges related to training time by using the output of each layer as the input for the next one. Additionally, it leverages gradient values from the loss function to facilitate a reduction in time and computational cost. The architecture of DenseNet201 is illustrated in Figure 7.

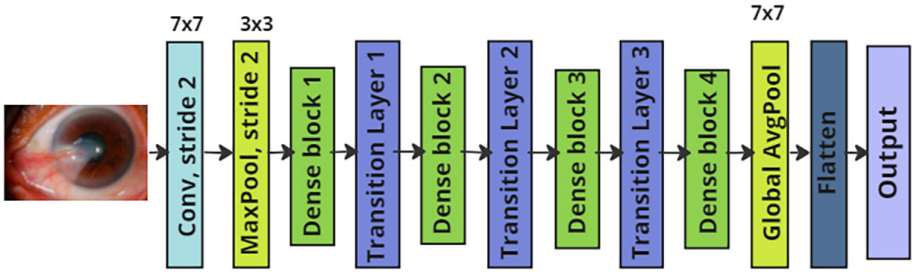

Fig. 7. Pre-trained architecture of DenseNet201

4.4 Valuation metrics

In [27], a series of indices and formulas commonly used for the evaluation of ophthalmic diagnostic models with artificial intelligence are established. In this list, there are Sensitivity, Specificity, Accuracy, Precision, and F1 score formulas, which will be used to evaluate the results of the different architectures mentioned:

The **F1 score** is employed to validate the outcome of preprocessing techniques, as it is a weighted harmonic mean of precision (*Pr*) and recall (*Rc*) [28]. Additionally, this metric considers true positives and false positives (*TP* and *FP*), as well as true negatives and false negatives (TN and FN). Therefore, a higher *F*1 score indicates greater predictability of the system. The following equation represents mathematically what has been described.

$$F1 = (2 \times Pr \times Rc) \div (Pr + Rc) \tag{1}$$

Precision focuses on the quality of positive predictions, as it measures the proportion of successfully predicted positive cases relative to the total predicted positive cases. A high value of this metric indicates that the model has a low false positive rate, that is, when it predicts a case as positive, it is likely to be correct. The following equation represents mathematically the definition of **precision** as the ratio of true positives (*TP*) over the sum of true positives and false positives (*FP*).

$$Pr = TP \div (TP + FP) \tag{2}$$

The recall or sensitivity evaluates the model's ability to successfully identify the positive cases existing in the dataset. A high recall value indicates that the model is good at detecting most positive cases, minimizing false negatives. It is mathematically represented by the following equation:

$$Rc = TP \div (TP + FN) \tag{3}$$

Accuracy is another metric used to measure the models' precision in classification tasks. It is calculated as the ratio between the number of correct predictions and the total number of observations or examples. The following equation shows the formula for calculating this metric.

$$Accuracy = (TP + TN) \div (TP + TN + FP + FN) \tag{4}$$

Specificity is employed to assess a model's ability to correctly identify true negatives. It is useful in contexts where minimizing false positives is crucial. The following equation mathematically represents the definition.

$$Especificidad = (TN) \div (TN + FP) \tag{5}$$

Finally, the Area Under the ROC Curve (Receiver Operating Characteristic), a curve that plots the true positive rate (sensitivity) against the false positive rate (1 – specificity) for different cutoff points. AUC is the area enclosed by the curve and the X-axis. It can be used to measure the performance of classification models, with values typically ranging between 0.5 and 1. A higher AUC value suggests better classification performance.

4.5 Creating the data set

A custom dataset composed of images of the anterior segment of the eye is created, intended for classification of the severity level of pterygium. The images, previously labeled by a specialist ophthalmologist from the Peruvian Pterygium Center, are distributed for training, testing and validation in proportions of 70%, 15%, and 15%, respectively, as detailed in Figure 8.

To improve the generalization capacity of the model, a data augmentation approach has been implemented that allows including strategic variations such as brightness adjustments, contrast, introduction of Gaussian noise, sharpening, gamma variation and manipulation of RGB channels. The choice of this technique is based on replicating varied capture conditions, simulating the diversity in lighting and other factors present in the images captured using a smartphone camera, which will serve as the main interface for the mobile application.

Introducing variability into the training images makes it easier for the model to learn more robust patterns and thus generalize more effectively to new images. Other generic methods such as horizontal or vertical flip may not capture the complexity of variability in ophthalmic image capture, which could limit the model's ability to make accurate predictions in the real-world mobile application environment.

Additionally, a class weighting approach was adopted to mitigate the imbalance in class distribution. This strategy was applied during model training to assign specific weights to each class, ensuring that the model was not biased by the prevalence of each pterygium severity category relative to the others. Finally, the images were normalized to a standard resolution of 224 pixels wide by 224 pixels high.

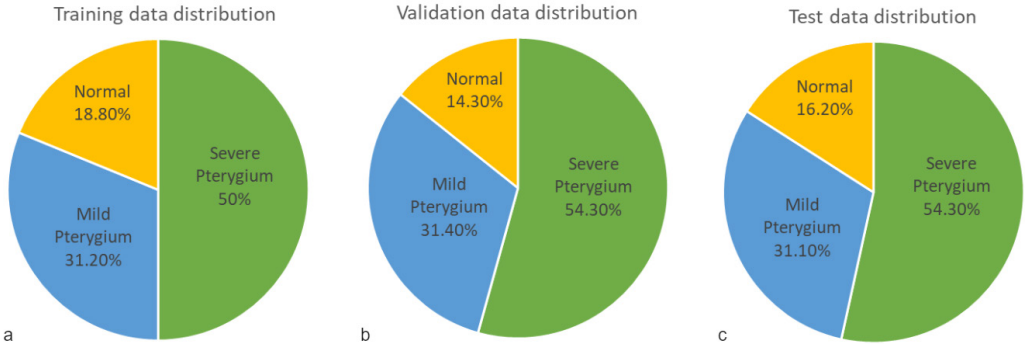

Fig. 8. Distribution of the data set for each class

4.6 Experimental environments

The experimental process was based on 4 pre-trained architectures with ImageNet weights. All experiments were conducted using the Stochastic Gradient Descent (SGD) optimizer and the categorical cross-entropy loss function, with a batch size of 32, 35 epochs, and a softmax output layer.

4.7 Experimental result

The methodology proposed in this work was based on evaluating the models using F1, precision, recall, and accuracy metrics. The F1 score varies across different classes (Normal, Mild Pterygium, and Severe Pterygium). For the Normal class, MobileNet v2 has the highest F1 score (86%); for the Mild Pterygium class, ResNext50 has the highest score (85%), and for the Severe Pterygium class, ResNext50 and DenseNet201 generated the highest value (93%). Table 2 displays the F1 scores with data augmentation for each model, where ResNext50 has the highest average

F1 score (88%) compared to other models. Data augmentation was applied to both training and validation data.

Table 2. F1 scores for each model with data augmentation

Architecture	Class			Total Score
	Normal	Mild Pterygium	Severe Pterygium	
ResNext50	0.85	0.85	0.93	0.88
ResNet50	0.75	0.72	0.87	0.78
MobileNet v2	0.86	0.74	0.86	0.82
DenseNet201	0.83	0.82	0.93	0.86

For precision, in the Normal class, ResNext50 has the highest value (79%); for the Mild Pterygium class, DenseNet201 has the highest score (86%), and for the Severe Pterygium class, ResNet50 generated the highest value (100%). Table 3 shows the precisions with data augmentation for each model, where ResNext50 has the highest average value (86%) compared to the other models.

Table 3. Precision of each model with data augmentation

Architecture	Class			Total Score
	Normal	Mild Pterygium	Severe Pterygium	
ResNext50	0.79	0.83	0.97	0.86
ResNet50	0.6	0.71	1	0.77
MobileNet v2	0.75	0.80	0.87	0.81
DenseNet201	0.71	0.86	0.97	0.85

For the recall metric, in the Normal class, there was a tie between the ResNet50, DenseNet201, and MobileNet v2 models with the maximum score (100%); for the Mild Pterygium class, ResNext50 has the highest score (87%), and for the Severe Pterygium class, both ResNext50 and DenseNet201 generated the highest value (90%). Table 4 shows the recall with data augmentation for each model, in which ResNext50 has the highest average value (90%) compared to the other models.

Table 4. Recall of each model with data augmentation

Architecture	Class			Total Score
	Normal	Mild Pterygium	Severe Pterygium	
ResNext50	0.92	0.87	0.90	0.90
ResNet50	1	0.74	0.77	0.84
MobileNet v2	1	0.70	0.85	0.85
DenseNet201	1	0.78	0.9	0.89

For the specificity metric, in the Normal class, ResNet50 achieved the maximum score (95%); for the Mild Pterygium class, DenseNet201 has the highest value (94%), and for the Severe Pterygium class, ResNet50 generated the highest value (100%). Table 5 shows the specificity with data augmentation for each model, where ResNext50 has the highest average value (95%) compared to the other models.

Table 5. Specificity of each model with data augmentation

Architecture	Class			Total Score
	Normal	Mild Pterygium	Severe Pterygium	
ResNext50	0.95	0.92	0.97	0.95
ResNet50	0.87	0.86	1	0.91
MobileNet v2	0.94	0.92	0.86	0.91
DenseNet201	0.92	0.94	0.97	0.94

Finally, an accuracy of 89% was achieved for ResNext50, 80% for ResNet50, 82% for MobileNet v2, and 88% for DenseNet201. The performance of the architectures during the training phase is shown in Figure 9. This figure represents the accuracy and loss value of the studied models based on the number of epochs. The loss graph is calculated using the categorical cross-entropy function and indicates the discrepancy between the model's prediction and the actual labels in the dataset. The lower the prediction's perfection, the further the loss value will be from zero, and the model will try to adjust its weights to reduce this value. In (a), (c), (e), and (g) of Figure 9, the losses of the 4 models are shown, where it can be observed that all of them obtained good results with the training data. However, in the validation data, ResNext50 achieved better results, followed by DenseNet201, ResNet50, and MobileNet V2, respectively. The accuracy, on the other hand, indicates the proportion of correct predictions relative to the total predictions. This should increase steadily and should not have sudden drops, as these could indicate issues such as overfitting or underfitting during training. In (b), (d), (f), and (h) of Figure 9, as in the loss graphs, all models perform well on the training data. In the validation data, however, differences are shown, where ResNext50 and DenseNet201 stand out, followed by ResNet50 and MobileNet V2.

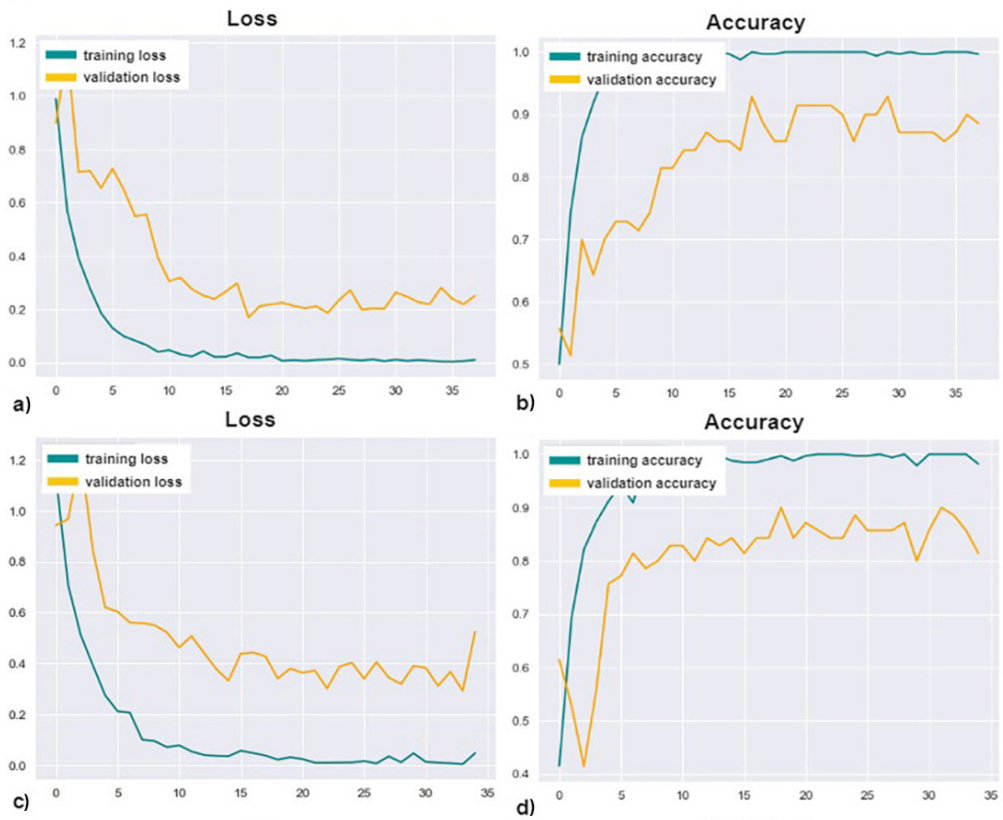

Fig. 9. *(Continued)*

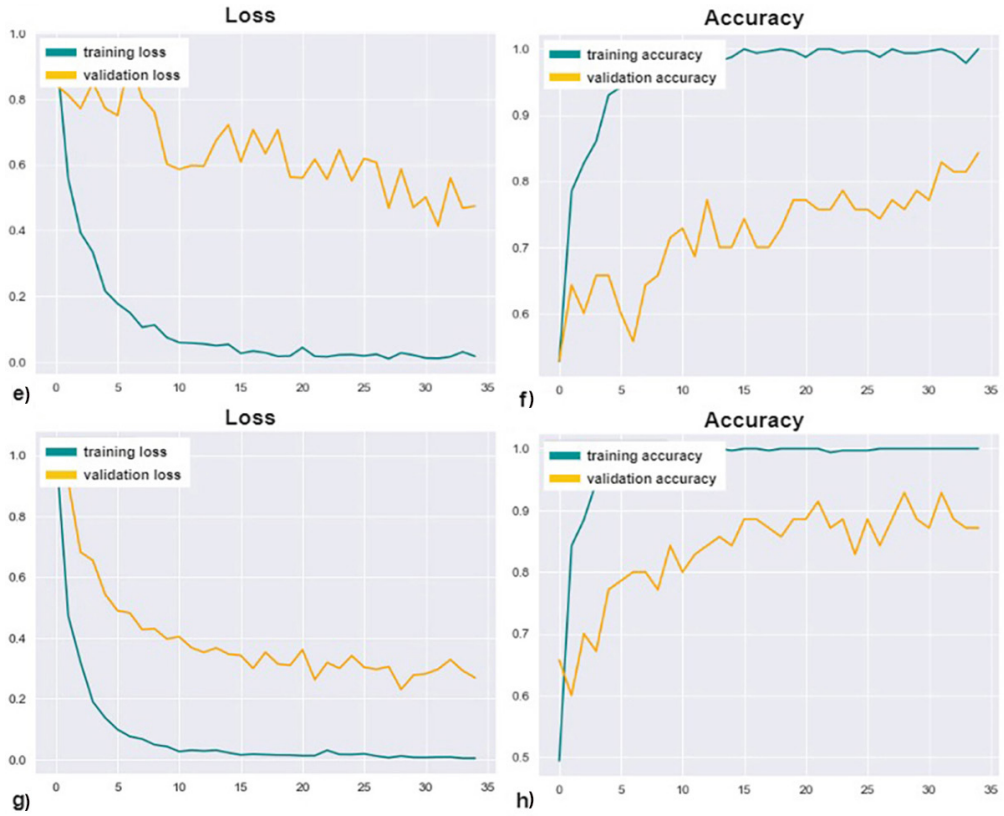

Fig. 9. Loss (a, c, e, g) and Accuracy (b, d, f, h) Graphs for ResNext50, ResNet50, MobileNet V2, and DenseNet201 Architectures, respectively

4.8 Confusion matrix

The confusion matrix has a tabular structure with columns and rows representing the actual classes and the classes predicted by the model, respectively. In Figure 10, the confusion matrix is observed, where the elements on the main diagonal represent the number of instances that the model correctly classified for each class, while the off-diagonal elements represent classification errors.

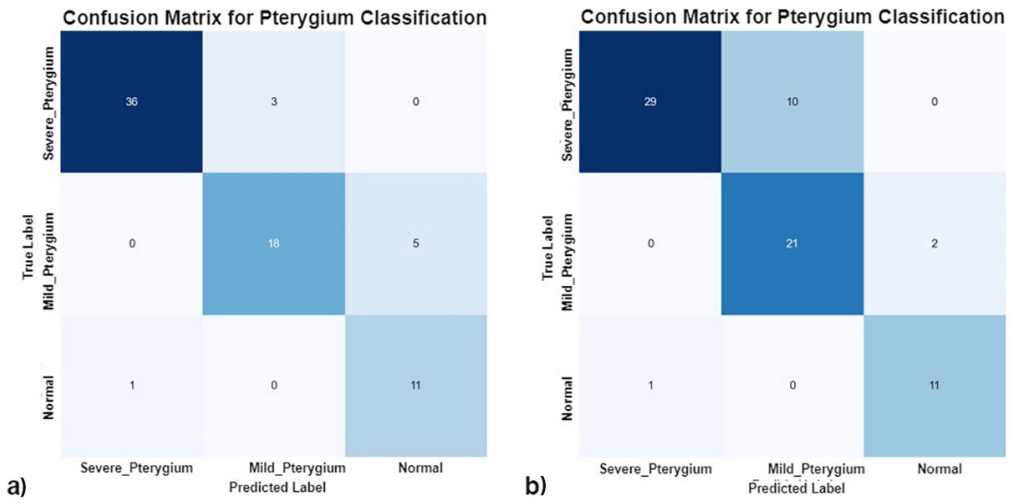

Fig. 10. *(Continued)*

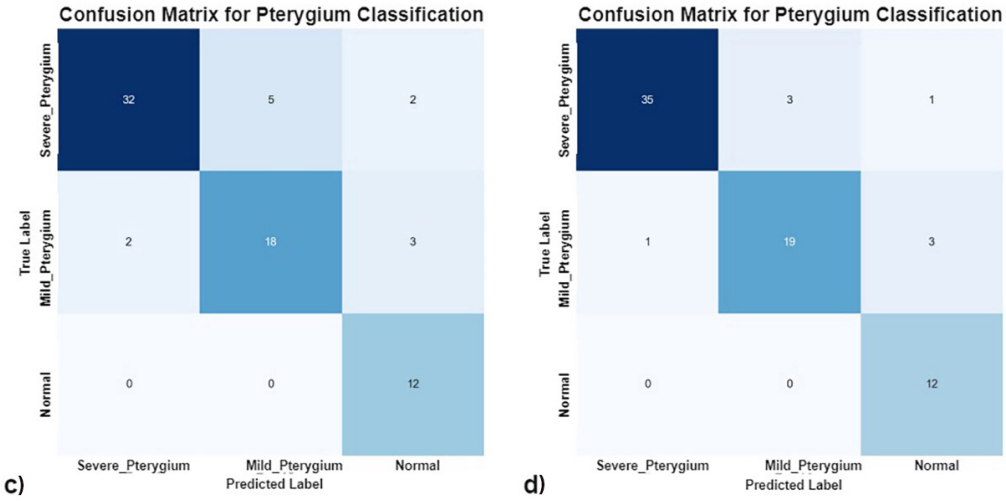

Fig. 10. Confusion matrix for the architectures (a) ResNext50, (b) ResNet50, (c) MobileNet V2, and (d) DenseNet201

From the confusion matrices, it can be concluded that the ResNext50 model identified 3 ASPIs belonging to the Mild Pterygium class as Severe Pterygium, 5 ASPIs belonging to the Normal class as Mild Pterygium, and 1 ASPI belonging to the Severe Pterygium class as Normal. Similarly, ResNet50 identified 10 ASPIs belonging to the Mild Pterygium class as Severe Pterygium, 2 ASPIs belonging to the Normal class as Mild Pterygium, and 1 ASPI belonging to the Severe Pterygium class as Normal. On the other hand, the MobileNet V2 and DenseNet201 models had more inaccuracies. The first one identified 5 ASPIs belonging to the Mild Pterygium class as Severe Pterygium, 2 ASPIs belonging to the Normal class as Severe Pterygium, 2 ASPIs belonging to the Severe Pterygium class as Mild Pterygium, and 3 ASPIs belonging to the Normal class as Mild Pterygium. The second one identified 3 ASPIs belonging to the Mild Pterygium class as Severe Pterygium, 1 ASPI belonging to the Normal class as Severe Pterygium, 1 ASPI belonging to the Severe Pterygium class as Mild Pterygium, and 3 ASPIs belonging to the Normal class as Mild Pterygium.

4.9 Curve Receiver Operating Characteristic (ROC)

The ROC curve is a graphical tool that represents the relationship between the true positive rate (sensitivity) and the false positive rate (1 – specificity) across a continuous range of decision thresholds. This graph visually provides an assessment of the models' performance. In Figure 11, the ROC curves for the proposed architectures ResNext50, ResNet50, MobileNet v2, and DenseNet201 are detailed. These curves resulted in AUC values of 0.98, 0.96, 0.93, and 0.98, respectively. This implies that ResNext50 and DenseNet201 have a greater ability to distinguish among the three classes rather than providing a random result. Although ResNet50 and MobileNet v2 exhibit less optimal results, they still possess a high capacity for discriminating between classes.

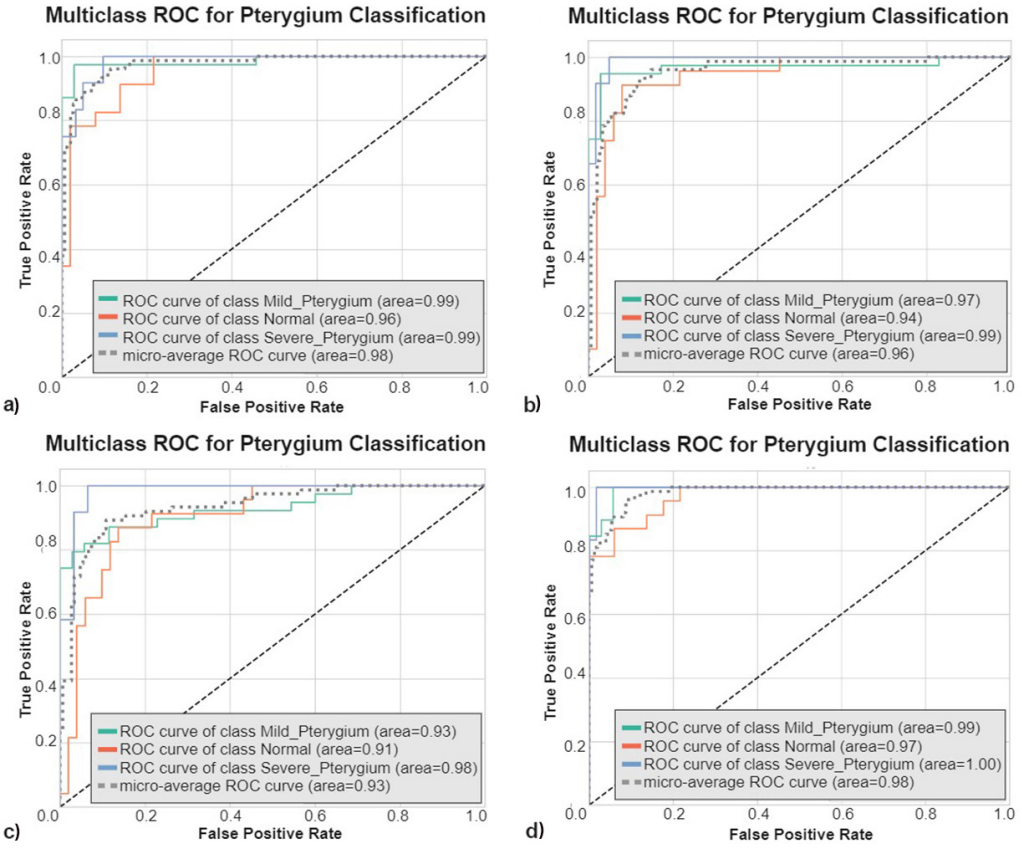

Fig. 11. ROC curves for the architectures (a) ResNext50, (b) ResNet50, (c) MobileNet V2, and (d) DenseNet201

5 INTEGRATION INTO THE MOBILE APPLICATION

In the previous section, it was determined that the optimal model for pterygium detection is ResNext50 due to its outstanding performance in the described metrics. Now, we will focus on the integration of this model with a mobile application developed in Flutter. The choice to use mobile technology for this research is relevant due to its ability to provide quick and efficient access to medical diagnostic tools, especially in resource-limited clinical and community settings such as rural areas. In this section, the details of the integration of the model with the other components detailed in point 3 and the flow of the application for the classification of pterygium using an ASPI will be described.

5.1 Model deployment and integration with the other components

Next, it will be described how the model was deployed and integrated with the other components as described in Figure 1:

Detection model. The model trained in point 4 was exported to two files to facilitate its use: a file with the extension '.hdf5' to store the weights and another with the extension '.json' to store the structure of the model, which will be loaded by a simple web service made in Flask that will be responsible for receiving the ASPIs from the backend and returning the result after having processed it with

the model. To facilitate portability and dependency management, the web service was uploaded to a container using docker, which would be uploaded to the AWS ECS service, where it would expose the web service endpoint. Table 6 shows the details of this.

Table 6. List of endpoints exposed by the model web service

Endpoint	Method Type	Input Data	Output Data
/predict	POST	Text string representing the base64 encoded ASPI	Int number representing the response class (0: Mild Pterygium, 1: Normal, 2: Severe Pterygium)

Backend. The backend has the responsibility of being the intermediary between the front-end, database and model web service components. To do this, a Rest API made in Spring Boot using Java 17 was deployed in the AWS Lambda service. In this way, for the application to interact with the trained model, it will have to create a revision using the backend endpoint. Table 7 details the main endpoints exposed by the backend:

Table 7. List of main endpoints exposed by the backend component

Endpoint	Method Type	Input Data	Output Data
/reviews	POST	• Text string representing the base64 encoded ASPI • Patient ID to whom the review belongs	Review data which contains id, image, result and date.
/patients/{patientId}/reviews	GET	• Patient ID to whom the reviews belong	Lis of Reviews that belong to the patient specified.
/specialists/createPatient/{specialistId}	POST	• Patient data • Specialist ID to whom the patient belongs	Patient data
/specialists/register	POST	• New specialist data	Specialist data
/specialists/login	POST	• Login data	JWT token

Database. In the AWS RDS service, a PostgreSQL relational database schema was deployed that will be responsible for storing all the information about patients, doctors and analyses, each in a table.

5.2 Pterygium classification flow using the application

Specialist registration. The first step consists of registering and logging in a specialist, who will be in charge of carrying out the reviews. As detailed in Figure 12a, the information requested consists of names, DNI (or some other identification document), password, hospital and position. Once the specialist is stored, you can enter the platform using your ID and password as shown in Figure 12b. To prevent unauthorized access to sensitive information, JWT is being used as an authentication method. If the credentials are correct, the specialist will be able to view the home screen, where the patients created and the results of their latest check-ups will be shown to further monitor the progress of the disease as shown in Figure 12c.

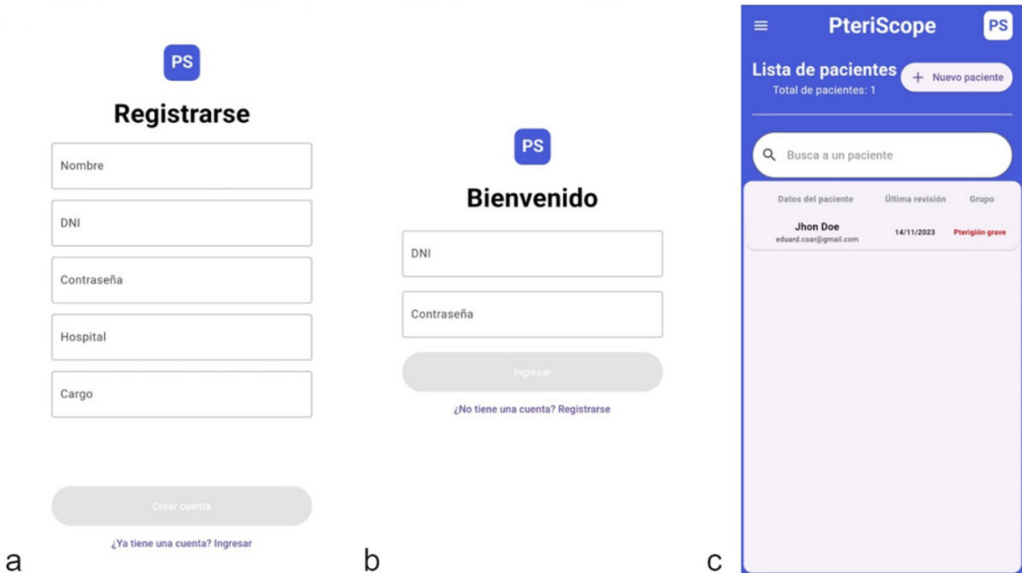

Fig. 12. Application screens. (a) Register view (b) Login view (c) Home view with an already registered patient

User creation. To create the patient, the following information described in Figure 13a will be required: name, surname, ID, age and email. The latter will serve to send the results to the patient's email. When the patient has just been created, the information from their last review will not be displayed in the Home view as it appears in Figure 13b.

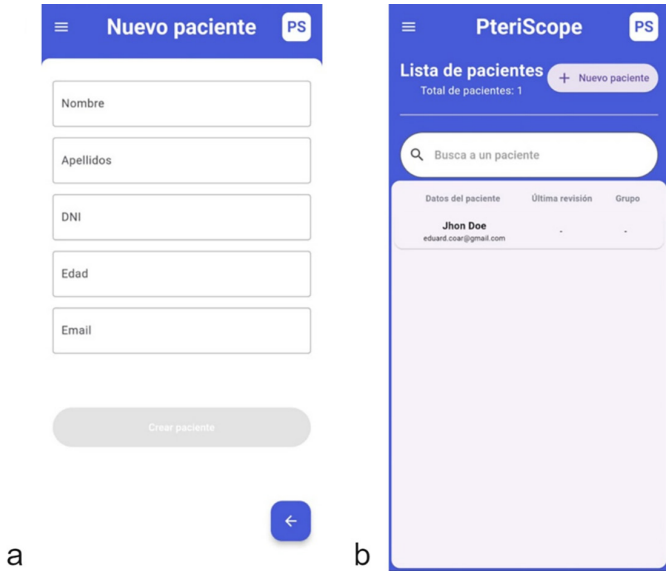

Fig. 13. Application screens. (a) New Patient view (b) Home view with a new patient registered

Revision creation. Finally, to create a review, permissions will be required to activate the device's camera to capture the ASPI. Based on the suggestion of the contacted specialists, the device should be approximately 15 cm from the eye at the time of capture. All revisions will be shown in the view of Figure 14a with relevant information for each one. Likewise, the application has the option to activate the flash as shown in Figure 14b. This functionality helps improve the visibility of details, which

can be crucial in detection, ensures consistency in lighting conditions, reduces shadows, etc. After the ASPI is processed by the model, the result is displayed in the view of Figure 14c with the details of the patient, the result and the reviewer.

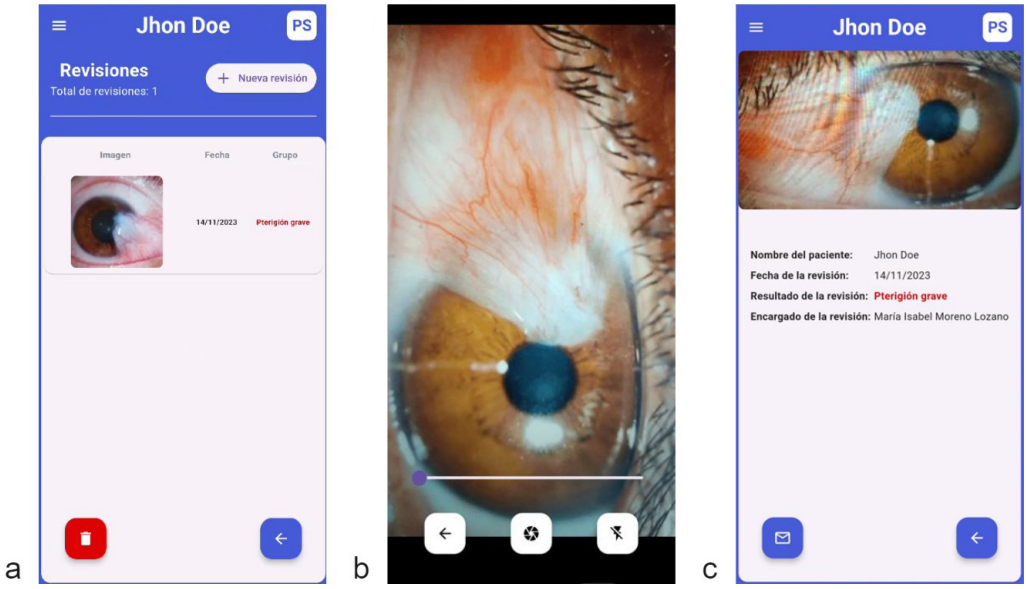

Fig. 14. Application screens. (a) View of the list of reviews for each patient (b) View of the camera to capture the ASPI (c) View of the detail of a revision

It is important to emphasize that this tool does not intend to replace the clinical judgment of ophthalmologists. Instead, the application is conceived as a complement to enhance current clinical practice, providing a second opinion and facilitating quick and well-founded decision-making. The application could be particularly useful in scenarios where access to specialists and ophthalmological tools are limited and it's more common to have a cell phone. Unlike having a computer or some other expensive tool that includes the model, a cell phone is more usable because it has a camera and flash to capture the ASPI.

6 DISCUSSION

The results presented highlight the superiority of the model trained with ResNext50, which achieved scores of 0.88, 0.86, 0.90, and 0.95 in F1 Score, Precision, Recall, and Specificity, respectively, as shown in Tables 2, 3, 4, and 5. ResNext50 also demonstrated good performance in distinguishing between classes, as evidenced in the ROC curve, where it equals DenseNet201 with an AUC of 0.98. Similarly, it had better result in the evolution of loss and accuracy during the training process compared to the other models. These results demonstrate that ResNext50 was also superior to the other evaluated architectures in the field of classification of ophthalmic diseases, which aligns with the findings observed in the literature review [8], [23], [24], [25].

However, it is also important to highlight the errors made by the four models during the testing phase due to the limited number of ASPIs in the dataset. ResNet50, for instance, misclassified 10 cases of mild pterygium as severe. On the other hand, ResNext50 classified 5 cases as mild pterygium that belonged to the "No pterygium" class. This could be due to similarities between the photographs, issues with

generalizing features, and a lack of more ASPI's during the training phase. Although these numbers are small, they may imply unnecessary or inappropriate treatment, delays in correct treatment, a burden on the healthcare system, a reduction in confidence in medical care, among other issues. For this reason, it is important that future research has a broader and more diverse dataset for training and ensuring the generalization of the model. Since there is currently no public dataset large enough for the proposed task, it is recommended to collect data from hospitals and clinics, as well as consult authors who have conducted research related to the disease. This second option can provide more diverse data since there are some investigations that collected ASPIs using smartphones, which adds diversity to the dataset and is ideal taking into account that the images will be captured using a cell phone camera in the final solution.

Although the model has promising results up to the training phase, it is important to rigorously validate the integrated system in real clinical settings before its widespread implementation in ophthalmic medical centers. This validation ensures that the system can function effectively and accurately in real-world situations, where you interact with real patients and healthcare professionals. Additionally, it provides us with an opportunity to identify and address potential challenges in system adoption, such as integration with existing clinical workflows, acceptance by healthcare professionals, and appropriate interpretation of results by clinicians and ophthalmological doctors. By discussing these challenges, we can offer a more complete view of the practical implementation of our mobile solution for pterygium detection in ophthalmic medical centers, strengthening confidence in its usefulness and effectiveness in the treatment and management of this ocular disease.

On the other hand, the lack of interpretability in deep learning models, such as convolutional neural networks (CNN), is a significant limitation that affects their acceptance in medical environments due to their "black box" nature by preventing people from understanding what it is like that a certain result was reached regardless of its precision. As mentioned in [29], if users do not trust a model, it is unlikely that they will use it for decision making, even more so when talking about medical issues, where predictions cannot be acted on with blind faith, since the consequences can be catastrophic. Although our pterygium classification model has shown promising results in the testing phase, the lack of a clear explanation of how and why a specific prediction is made limits its clinical utility. Interpretation of model decisions is essential for health professionals to trust the predictions and make informed decisions in diagnosis and treatment, so it is important to address

In [30], various available CNN explainability methods were collected. A very interesting one consists of deconvolutional neural networks, which take the features extracted by a CNN and reconstruct them to provide a visual explanation of the model's decision. These networks densify the explanation, offering an activation map closer to the original image, which facilitates the understanding and verification of the model's decisions, thus improving its interpretability. Another proposal is LIME, or Local Interpretable Model-Independent Explanations, which provides local explanations for the classifications made by a machine learning model, regardless of their complexity. The end result is the generation of understandable explanations, such as patches in images or words in text, that correspond to the model's classification decisions. This improves interpretability by allowing users to understand which features or parts of the data influence the model's predictions, increasing confidence in its performance.

Likewise, more conventional methods can be used too, such as visualization of layer activations, or integrating the model with another focused-on segmentation, which would highlight the affected region. In any of the mentioned cases, the detailed exploration of these approaches is left for future research, where it is expected to delve into their applicability and benefits to improve the interpretability of the proposed model.

7 CONCLUSION

In this article, we provided a deep transfer learning technique based on convolutional neural networks for categorizing patients with pterygium. To investigate the proposed deep transfer learning technique, four pretrained convolutional neural network models were employed. It was observed that fine-tuning pretrained models can be effectively used on a multi-class dataset. Among the models used, the ResNeXt-50 architecture achieved a maximum accuracy of 86%, a maximum F1 score of 88%, a maximum recall of 90%, a maximum specificity of 95%, and an overall accuracy of 89% for our dataset. This resulted in model sensitivity, specificity, and precision exceeding 85%. Our high-precision findings can help doctors and researchers make clinical judgments due to the ease and usefulness of having this model on a smartphone with a camera and flash. However, our work still includes some limitations that can be addressed in future research, such as the evident lack of a larger and more varied dataset that allows generalizing the characteristics of the disease in various situations and conditions. Likewise, it is essential to achieve the acceptance and trust of medical institutions, exploring methods to improve the interpretability of the model, as well as proving that the final solution has optimal results in a real environment. By applying such improvements, the proposed method can be used in other medical problems, such as cancer and tumors, and at the same time be applicable in other computer vision industries, such as energy, agriculture, and transportation.

8 ACKNOWLEDGMENT

The authors are grateful to the Dirección de Investigación de la Universidad Peruana de Ciencias Aplicadas for the support provided for this research work through the incentive UPC-A-103-2023, as well as the health personnel who participated in the survey.

9 REFERENCES

[1] S. R. Abdani, M. A. Zulkifley, and N. H. Zulkifley, “Group and shuffle convolutional neural networks with pyramid pooling module for automated pterygium segmentation,” *Diagnostics*, vol. 11, no. 6, p. 1104, 2021. https://doi.org/10.3390/diagnostics11061104

[2] W. M. D. Wan Zaki, M. Mat Daud, S. R. Abdani, A. Hussain, and H. A. Mutalib, “Automated pterygium detection method of anterior segment photographed images,” *Computer Methods and Programs in Biomedicine*, Elsevier Ireland Ltd, vol. 154. no. 1, pp. 71–78, 2018. https://doi.org/10.1016/j.cmpb.2017.10.026

[3] L. J. Coan *et al.*, "Automatic detection of glaucoma via fundus imaging and artificial intelligence: A review," *Surv Ophthalmol*, vol. 68, no. 1, pp. 17–41, 2023. https://doi.org/10.1016/j.survophthal.2022.08.005

[4] N. S. M. Zamani, W. M. D. W. Zaki, A. B. Huddin, A. Hussain, H. A. Mutalib, and A. Ali, "Automated pterygium detection using deep neural network," *IEEE Access*, vol. 8, pp. 191659–191672, 2020. https://doi.org/10.1109/ACCESS.2020.3030787

[5] B. Sowmya *et al.*, "A visual computing unified application using deep learning and computer vision techniques," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 18, no. 1, pp. 59–74, 2024. https://doi.org/10.3991/ijim.v18i01.42673

[6] Y. Liu *et al.*, "Accurate detection and grading of pterygium through smartphone by a fusion training model," *British Journal of Ophthalmology*, vol. 108, no. 3, pp. 336–342, 2023. https://doi.org/10.1136/bjo-2022-322552

[7] X. Fang *et al.*, "Deep learning algorithms for automatic detection of pterygium using anterior segment photographs from slit-lamp and hand-held cameras," *British Journal of Ophthalmology*, vol. 106, no. 12, pp. 1642–1647, 2021. https://doi.org/10.1136/bjophthalmol-2021-318866

[8] H. Tariq, M. Rashid, A. Javed, E. Zafar, S. S. Alotaibi, and M. Y. I. Zia, "Performance analysis of deep-neural-network-based automatic diagnosis of diabetic retinopathy," *Sensors*, vol. 22, no. 1, 2022. https://doi.org/10.3390/s22010205

[9] "Centro Peruano del Pterigion – Somos el primer Centro Especializado en el tratamiento integral del Pterigion en Perú," [Accessed: Jan. 08, 2024]. [Online]. Available: https://carnosidadperu.com/

[10] A. H. Saad, N. S. M. Zamani, W. M. D. W. Zaki, A. B. Huddin, and A. Hussain, "Automated pterygium detection in anterior segment photographed images using deep convolutional neural network," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 8, no. 1.6, pp. 225–232, 2019. https://doi.org/10.30534/ijatcse/2019/3481.62019

[11] S. N. A. Ahmad, W. M. D. W. Zaki, and N. S. M. Zamani, "Sistem saringan penyakit pterigium untuk imej mata terangkum hadapan," *Jurnal Kejuruteraan*, vol. 31, no. 1, pp. 99–105, 2019. https://doi.org/10.17576/jkukm-2019-31(1)-12

[12] N. S. M. Zamani, L. A. Ramlan, W. M. D. W. Zaki, A. Hussain, and H. A. Mutalib, "Mobile screening framework of anterior segment photographed images," *International Journal of Engineering and Technology (UAE)*, vol. 7, no. 4, pp. 85–89, 2018. https://doi.org/10.14419/ijet.v7i4.11.20780

[13] Y. Sun, W. Huang, Z. Wang, X. Xu, M. Wen, and P. Wu, "Smart teaching systems: A hybrid framework of reinforced learning and deep learning," *International Journal of Emerging Technologies in Learning (iJET)*, vol. 18, no. 20, pp. 37–50, 2023. https://doi.org/10.3991/ijet.v18i20.44217

[14] T. Schlippe and K. Bothmer, "Skill scanner: An AI-Based recommendation system for employers, job seekers and educational institutions," *International Journal of Advanced Corporate Learning (iJAC)*, vol. 16, no. 1, pp. 55–64, 2023. https://doi.org/10.3991/ijac.v16i1.34779

[15] K. H. Hung *et al.*, "Application of a deep learning system in pterygium grading and further prediction of recurrence with slit lamp photographs," *Diagnostics*, vol. 12, no. 4, 2022. https://doi.org/10.3390/diagnostics12040888

[16] S. Zhu *et al.*, "Pterygium screening and lesion area segmentation based on deep learning," *J Healthc Eng*, vol. 2022, 2022. https://doi.org/10.1155/2022/3942110

[17] R. Reguant, S. Brunak, and S. Saha, "Understanding inherent image features in CNN-based assessment of diabetic retinopathy," *Scientific Reports*, vol. 11, no. 1, pp. 1–12, 2021. https://doi.org/10.1038/s41598-021-89225-0

[18] A. Dixit, J. Yohannan, and M. V. Boland, "Assessing glaucoma progression using machine learning trained on longitudinal visual field and clinical data," *Ophthalmology*, vol. 128, no. 7, pp. 1016–1026, 2021. https://doi.org/10.1016/j.ophtha.2020.12.020

[19] T. C. Yeh *et al.*, "Prediction of treatment outcome in neovascular age-related macular degeneration using a novel convolutional neural network," *Scientific Reports*, vol. 12, no. 1, pp. 1–8, 2022. https://doi.org/10.1038/s41598-022-09642-7

[20] F. Gan, W. Y. Chen, H. Liu, and Y. L. Zhong, "Application of artificial intelligence models for detecting the pterygium that requires surgical treatment based on anterior segment images," *Front Neurosci*, vol. 16, 2022. https://doi.org/10.3389/fnins.2022.1084118

[21] B. Zheng *et al.*, "Research on an intelligent lightweight-assisted pterygium diagnosis model based on anterior segment images," *Dis Markers*, vol. 2021, 2021. https://doi.org/10.1155/2021/7651462

[22] W. Xu, L. Jin, P. Z. Zhu, K. He, W. H. Yang, and M. N. Wu, "Implementation and application of an intelligent pterygium diagnosis system based on deep learning," *Front Psychol*, vol. 12, p. 4922, 2021. https://doi.org/10.3389/fpsyg.2021.759229

[23] A. Gaur, G. Pant, and A. S. Jalal, "Comparative assessment of artificial intelligence (AI)-based algorithms for detection of harmful bloom-forming algae: An eco-environmental approach toward sustainability," *Appl Water Sci*, vol. 13, no. 5, pp. 1–11, 2023. https://doi.org/10.1007/s13201-023-01919-0

[24] S. Hira, A. Bai, and S. Hira, "An automatic approach based on CNN architecture to detect Covid-19 disease from chest X-ray images," *Applied Intelligence*, vol. 51, no. 5, pp. 2864–2889, 2021. https://doi.org/10.1007/s10489-020-02010-w

[25] M. Kim, Y. Gil, Y. Kim, and J. Kim, "Deep-learning-based scalp image analysis using limited data," *Electronics*, vol. 12, no. 6, p. 1380, 2023. https://doi.org/10.3390/electronics12061380

[26] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings – 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, Honolulu, HI, USA, 2017, pp. 5987–5995. https://doi.org/10.1109/CVPR.2017.634

[27] W.-H. Yang *et al.*, "Guidelines on clinical research evaluation of artificial intelligence in ophthalmology," *Int J Ophthalmol*, vol. 16, no. 9, pp. 1361–1372, 2023. https://doi.org/10.18240/ijo.2023.09.02

[28] G. Tebes, D. Peppino, P. Becker, and L. Olsina, "Especificación del Modelo de Proceso para una Revisión Sistemática de Literatura," *La Paz*, 2019. [Online]. Available: https://www.researchgate.net/publication/333855959.

[29] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should I Trust You?': Explaining the Predictions of any classifier," in *NAACL-HLT 2016 – 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Demonstrations Session*, 2016, pp. 97–101. https://doi.org/10.18653/v1/N16-3020

[30] L. V. Haar, T. Elvira, and O. Ochoa, "An analysis of explainability methods for convolutional neural networks," *Eng Appl Artif Intell*, vol. 117, p. 105606, 2023. https://doi.org/10.1016/j.engappai.2022.105606

10 AUTHORS

Edward Jordy Ticlavilca-Inche is a Software Engineering student at the Peruvian University of Applied Sciences in Lima, Peru (E-mail: U201923961@upc.edu.pe).

Maria Isabel Moreno-Lozano is a Systems Information Engineering student at the Peruvian University of Applied Sciences in Lima, Peru (E-mail: U201924630@upc.edu.pe).

Pedro Castañeda has a PhD in Systems Engineering – UNMSM, a Master's Degree in Management and Information Technology Management – UNMSM and a Master's Degree in Business Administration (MBA) – ESAN. He currently leads projects of electronic brokering, software development and process improvement, using agile and traditional methodologies. He has the following certifications: Project Management Professional (PMP), Scrum Certified Developer (CSD), IBM Certified Professional in Rational Unified Process, ORACLE Certifications. Teacher of Project Management Methodologies and Software Development in public and private universities. Areas of Interest: Software Productivity, Business Intelligence, Machine Learning, Software Engineering (E-mail: pcsipcas@upc.edu.pe; ORCID: https://orcid.org/0000-0003-1865-1293).

Sandra Wong-Durand has a master's degree in Artificial Intelligence, a master's degree in Business Administration from ESAN with mention in Advanced Project Management, Systems Engineer from UNIFE, with specialization studies in Innovation and Leadership at the Escuela Superior de Administración y Dirección de Empresas (ESADE) – Spain, Process Improvement Management with CMMI at the Software Engineering Institute, Software Quality at UNIFE, Strategic Project Management at PM Certifica, SOA Architectures at IBM and Oracle (E-mail: pcsiswon@upc.edu.pe).

Alejandra Oñate-Andino She holds a degree in Computer Systems Engineering from Escuela Superior Politécnica de Chimborazo (Ecuador), a Master in Network Interconnectivity from Escuela Superior Politécnica de Chimborazo (Ecuador), and a PhD in Systems Engineering and Computer Science from Universidad Mayor de San Marcos (Peru). Currently she is the Coordinator of the Software Career at the Escuela Superior Politécnica de Chimborazo (Ecuador). In addition, she is a Research Professor, with more than 15 years of experience, leading teaching, research and management processes. She has directed and participated in several research and community outreach projects. Author of several scientific articles in the area of Information Technology Governance, Business Intelligence, Information Technology Management, among others (E-mail: monate@espoch.edu.ec).