INVESTIGACIÓN e INNOVACIÓN en INGENIERÍAS
ISSN2344-8652

Detecting hypertensive retinopathy symptoms in digital fundus images using machine learning algorithms

Detección de signos de retinopatía hipertensiva usando algoritmos de aprendizaje automático en imágenes digitales de fondo de ojo

Christian Delgado Molina **Jairo Vásquez López**

Leonardo Africany Villamil **Rubiel Vargas Cañas**

Universidad del Cauca, Colombia

OPEN ACCESS

Received: **01/20/2023**

Accepted: **04/10/2023**

Published: **05/15/2023**

Author Correspondence:
ccdelgado@unicauca.edu.co

Abstract

Objective: This study aims to apply detection architecture as a strategy for identifying and classifying symptoms of hypertensive retinopathy (HR) in digital fundus images.
Methodology: The proposed model is based on the CRISP-machine learning (ML) methodology, which is the most widely used method in ML projects, and consists of six stages. These stages are followed exhaustively; however, adaptations are made where necessary.
Results: The findings from this study reveal that the proposed system can accurately detect symptoms of HR in a fundus image with 96% accuracy.
Conclusions: Image classification and ML techniques are presented as viable alternatives to support the decision-making process required for detecting symptoms of this ocular complication. These methods are expected to benefit specialized physicians as well as patients with HR.

Keywords: hypertensive retinopathy, machine learning, detection, accuracy, YOLO

Abstract

Objetivo: Aplicar una arquitectura de detección como estrategia para identificar y estadificar signos de retinopatía hipertensiva en imágenes digitales de fondo de ojo.
Metodología: El modelo propuesto se fundamenta en la metodología CRISP-ML que es el método más utilizado en proyectos de aprendizaje automático y consta de seis etapas. Estas etapas se siguen de manera exhaustiva, pero se realizan adaptaciones donde sea necesario.
Resultados: Los resultados obtenidos en este trabajo muestran que el sistema propuesto es capaz de detectar signos de retinopatía hipertensiva en una imagen de fondo de ojo con una precisión del 96%.
Conclusiones: las técnicas de clasificación de imágenes y aprendizaje automático se presentan como una alternativa para el apoyo de la toma de decisiones necesarias en la detección de signos de esta complicación ocular. Con esto se espera que beneficien tanto los médicos especialistas como los pacientes con retinopatía hipertensiva.

Palabras clave: Retinopatía hipertensiva, aprendizaje automático, detección, precisión, YOLO

How to Cite (IEEE): C. Delgado-Molina., L. Africany-Villamil., J. Vásquez-López, and R. Vargas-Cañas, "Detecting hypertensive retinopathy symptoms in digital fundus images using machine learning algorithms," *Investigación e Innovación en Ingenierías*, vol. 11, no. 1, 104–124, 2023, https://doi.org/10.17081/invinno.11.1.6364.

Introduction

Cardiovascular disease is the foremost global cause of mortality. In Colombia, cardiac conditions alone contributed to 17.1% of documented male deaths and 18% of female deaths in 2022 [1]. Here, high blood pressure (HBP) emerges as the primary risk factor, capable of causing severe complications if not promptly treated [2]. The World Health Organization (WHO) [3] reported a global prevalence of 1.13 billion people with HBP, with the majority, almost two-thirds, residing in low- and middle-income countries. According to the Ministry of Health and Social Protection of Colombia (Minsalud) [4], ~4 out of 10 adults in the country grapple with HBP, with 60% of them being unaware of their condition. Moreover, these prevalence rates increase with age, with a higher incidence among females than males [5, 6, 7, 8]. Hypertensive retinopathy (HR) damages the retina due to the sudden surge in blood pressure, whether of primary or secondary origin, leading to diminished visual acuity and potential progression to blindness. Multiple studies have established a correlation between HR and elevated blood pressure levels. The World Health Organization estimates that ~1.3 billion people worldwide live with some form of visual impairment [9].

The diagnosis of HR involves a fundus examination that supplements the systemic evaluation of several patients. This examination can unveil vascular pathology markers in coronary and cerebral circulation, encompassing circulatory characteristics, the appearance of retinal capillaries, and microvascular changes in the retina. Consequently, a comprehensive assessment of the retinal vascular tree, optic nerve, and retina proves highly beneficial in predicting the risk of major cardiovascular diseases [10]. Detecting HR is a labor-intensive undertaking in terms of time and financial resources. Unfortunately, this procedure is currently performed manually, with specialists using different equipment or devices to visualize the retina and identify symptoms of the disease, such as crossings between veins, exudates, hemorrhages, and drusen, which are sometimes challenging to discern in fundus images. In addition, the comparison of extensive medical data and images and the recognition of necessary patterns for identifying each disease may contribute to fatigue and errors among specialists [10]. Another challenge is the limited number of doctors specialized in this field; according to Minsalud, only 4.1% of physicians specialize in ophthalmology, and they are predominantly located in larger cities [11].

In addition to the methods usually used in medicine to diagnosis, there are computer-assisted diagnostic methods that offer reliability, reproducibility, and quantification of different variables, which facilitate clinical diagnosis [11]. Consequently, several studies related to HR have been documented in the literature, aiming to identify diverse symptoms indicating the presence of hypertension in patients. One of these studies features an automated HR detection system proposed by Sarmad Khitran [12] based on the retinal arteriovenous ratio (AVR). The system encompasses a method for vessel classification, distinguishing between arteries and veins using a feature vector and a hybrid classifier. For these purposes, two publicly available databases of digital fundus images, VICAVR and DRIVE, were employed. The technique proposed by Irshad [13] focuses on classifying vessels into arteries and veins using a binary vessel mask and the localization of the optic disc (OD) center to extract the region of interest around the OD, achieving an accuracy of 81.3%. The research conducted by Triwijoyo [14] centers on an early HR detection system. It utilizes the fundus image as input to a convolutional neural network to determine the presence of HR symptoms. Triwijoyo used DRIVE as the dataset, and an accuracy of 98.6% was attained. The research proposed by Arasy [15] introduces a system for HR detection through principal component analysis (PCA) and a backpropagation neural network. Retina images were sourced from the STARE database, demonstrating an accuracy of 86.36%. While these studies identify the arteriovenous ratio (AVR), they fail to consider other symptoms crucial for diagnosis, such as significant crossings and exudates. Moreover, they do not quantify the symptoms or their locations, which are essential for classifying complications as mild, moderate, or severe as well as for patient monitoring and prognosis.

This study discusses the implementation of algorithms based on machine learning (ML) techniques capable of identifying characteristics and symptoms present in patients with HR. When integrated into a computational tool, it has the potential to streamline the analysis time of a fundus examination in a clinical environment.

Method and Procedures

This project was developed using the CRISP-ML methodology, where the initial four stages aided in the detection of HR symptoms. To achieve this goal, an image dataset was created to identify the pathology's specific symptoms. During the modeling stage, two phases were considered: training and testing. Ultimately, in the testing phase, the performance was assessed using a separate test dataset, ensuring that the data had not been utilized in the training process.

Database Configuration

The images used for symptom identification were sourced from commonly used public databases in research and projects associated with this pathology.

E-ophta [16]. This dataset is divided into two categories: E-Ophta MA and E-Ophta EX. E-Ophta MA includes 148 images featuring microaneurysms and small hemorrhages, along with 233 images showing no lesions. Further, E-Ophta EX comprises 47 fundus images with exudate segmentation and 35 images without any form of lesion, designated as normal images.

Kaggle DR dataset [17]. This dataset is provided by EyePACS (n.d.) to contribute to the research and development of projects related to the detection and diagnosis of diabetic retinopathy. In total, it comprises 88,702 images, wherein 35,126 are used for training tasks and 53,576 are used for testing.

Retinopathy online challenge [18]. ROC contains 100 color fundus images, all of which display microaneurysms. These images are randomly distributed, with 50 images assigned for training and 50 images for testing.

A total of 200 fundus images presenting microaneurysms and hemorrhages were collected, along with 150 images featuring exudates and drusen.

Data Preparation

Labeling

After selecting the images, graphical labeling is performed using a free and open-source tool. This tool, termed LabelImg [19], which was developed in Python, employs QT for its graphical interface and is crucial for recognizing the dataset's importance in training the model. LabelImg is selected to utilize labels that help identify components within the data, aiding the model in dataset recognition. For each image, LabelImg generates a YOLO-formatted marked file, storing information in a .txt file. In this file, the first column indicates the class of the selected object. The second and third columns denote the central point in the x- and y-directions of the bounding rectangle, and the fourth and fifth columns represent the width and height of the rectangle.

Data Augmentation [20]

In this project, data augmentation was implemented through the open-source CLODSA package [21], designed for image transformations. The selection of this library was based on its adaptability to various requirements and its effectiveness in improving detection model accuracy. Transformations applied to the original images depicting hemorrhages and microaneurysms involved random horizontal and vertical flips, along with 90° rotations. This approach substantially increased the number of images available for training. Similar operations were conducted on the original images showcasing exudates and drusen.

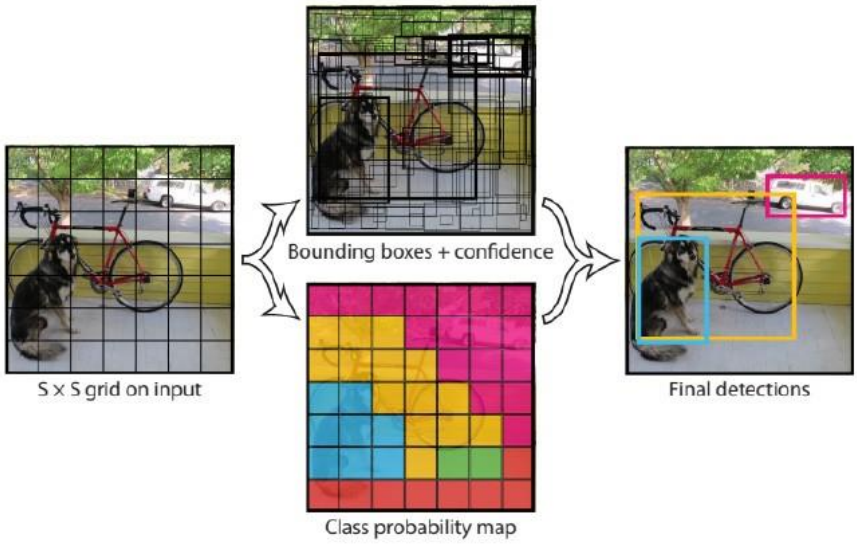
Modeling

The model selected for this study was YOLO [22], an architecture used for object detection. In this approach, potential bounding boxes are initially generated in an image, followed by the classification of each of these boxes [23]. Subsequently, a postprocessing stage refines the bounding boxes, eliminates duplicate detections, and reclassifies the boxes based on other objects in the image. These systems are slow and challenging to optimize due to the separate training of each individual component. On the contrary, YOLO utilizes a single convolutional network that predicts multiple bounding boxes and class probabilities for those boxes simultaneously (Figure 1). It undergoes training on complete images, directly optimizing detection performance. This unified model provides several advantages over traditional object detection methods [23]:

YOLO is exceptionally fast, with the neural network simply running on a new image to predict detections. YOLO achieves more than double the average precision of other real-time systems.

YOLO works globally across the image. Unlike sliding window techniques and region proposal methods, YOLO considers the entire image during both training and testing, implicitly incorporating contextual information about classes and their appearance.

Figure 1. Illustration of how YOLO works



Source: [23]

File and model data configuration: All the configuration settings analyzed (Table 1) are based on a Tesla P100 GPU with 16 GB available in Google Colab.

Table 1. Parameters used for training the YOLO neural network model

img	Defines the size of the input image for training. It should be a multiple of 32
batch	Determines the size of the image batch
max_batches	Defines the number of training epochs or iterations. At a minimum, it must comply with (# of classes) × 2000
Steps	Defined from the max_batch, i.e., (80% of max_batches), (90% of max_batches)
Filter	The number of filters to be applied depends on the number of classes and is defined as filters = (# of classes + 5) × 3
data	Defines the path to the data and name file where the paths and classes are
cfg:	Specifies the configuration of the selected model
Random	This allows to save memory during training; if it is set to 1, the batch size will vary during training, slowing down the process, whereas if it is set to 0, a fixed input batch is used, reducing memory usage and speeding up training

Source: Adapted from [23]

Training

In this study, we worked with two YOLO models: YOLOv3 and YOLOv4, both in their larger versions and smaller versions (YOLOv3_tiny and YOLOv4_tiny). Four experiments were conducted with these models by varying their configuration. In the Darknet-53 configuration file (cfg), the batch size was changed from 64 to 32, max_batches were set to 7,000, and the steps were set to 5,600 and 6,300. The model was trained for 7,000 iterations with a batch size of 32. Learning rates were scheduled to decrease at steps 5,600 and 6,300. Next, the number of filters was determined based on the number of classes (Figure 2), which in this case were two since each model detects two symptoms.

Figure 2. Configurations of the first experiment with the small version of YOLO (YOLOv3_tiny) at a fixed resolution of 416 × 416

```
[net]
# Testing
#batch=1
#subdivisions=1
# Training
batch=32
subdivisions=16
width=416
height=416
channels=3
momentum=0.9
decay=0.0005
angle=0
saturation = 1.5
exposure = 1.5
hue=.1

learning_rate=0.001
burn_in=1000
max_batches = 7000
policy=steps
steps=5600,6300
scales=.1,.1
```

```
[convolutional]
batch_normalize=1
filters=256
size=3
stride=1
pad=1
activation=leaky

[convolutional]
size=1
stride=1
pad=1
filters=21
activation=linear

[yolo]
mask = 1,2,3
anchors = 10,14,  23,27,  37,58,  81,82,  135,169,  344,319
classes=2
num=6
jitter=.3
ignore_thresh = .7
truth_thresh = 1
random=0
```

Source: Prepared by the authors

Assessment

The model assessment relies on its performance, employing metrics, such as the confusion matrix, and derived indicators, such as specificity, sensitivity, area under the ROC curve, and precision. The calculated mean average precision (mAP) value for all classes is then determined to choose the best-performing model based on stability and consistency.

Results

Datasets

For this study, 200 fundus images featuring microaneurysms and hemorrhages were gathered. In addition, 150 images, including exudates and drusen, were sourced from the databases mentioned in Section 1.1. These fundus images were classified based on their dimensions and format, and they were organized to facilitate the detection of two symptoms associated with HR by each model. Specifically, the symptoms considered were hemorrhages and microaneurysms (Table 2) as well as exudates and drusen (Table 3).

Table 2. Number of images from each resolution in the dataset used for hemorrhages and microaneurysms

Dimension	Format	Total number of images
1440 × 960	JPEG	80
2912 × 2912	JPEG	30
1504 × 1000	JPEG	20
1500 × 1152	JPG	30
2544 × 1696	JPG	40

Source: Prepared by the authors

Table 3. Number of images from each resolution in the dataset used for exudates and drusen

Dimension	Format	Total number of images
2544 × 1696	JPEG	80
776 × 582	JPEG	30
1440 × 960	JPEG	40

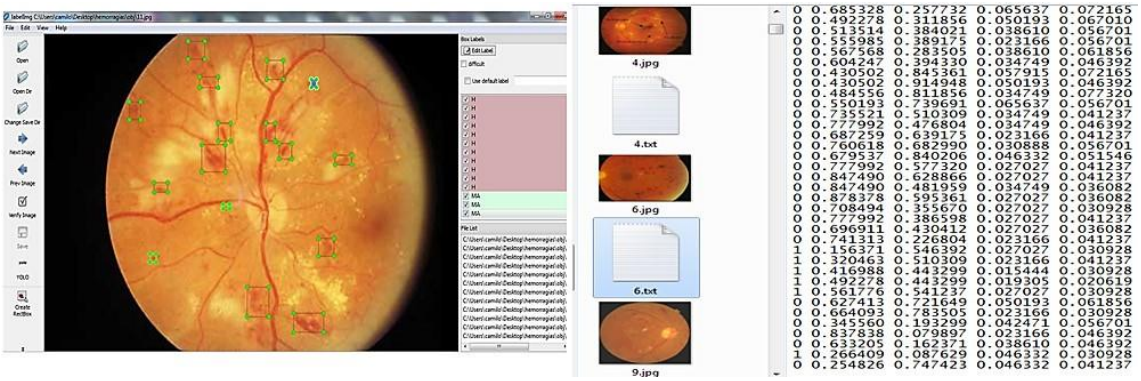
Source: Prepared by the authors

Data Adequacy

Labeling

In this stage, the respective labeling of each image was conducted to obtain the delimiting rectangle and its respective .txt file in YOLO format (Figure 3). This same process was conducted for the model that detects exudates and drusen.

Figure 3. Hemorrhage and microaneurysm labeling using LabelImg

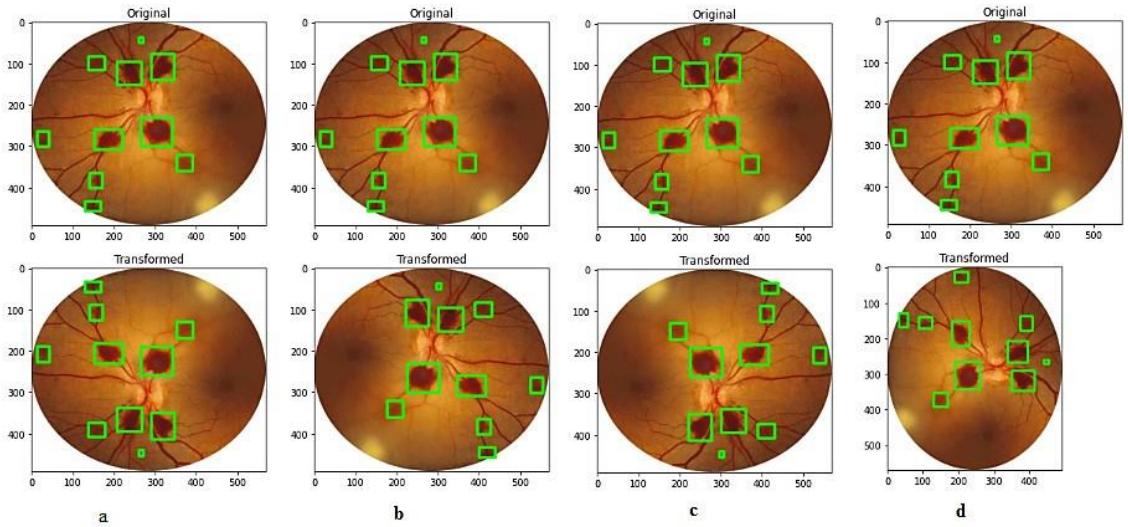


Source: Prepared by the authors

Data Augmentation

Data augmentation was used to improve the precision of the models in identifying symptoms present in a fundus image in HR. This involved considerably increasing the number of training images by transforming the original images (Figure 4), resulting in a total data augmentation of 708 images for training. The same process was applied to the 150 original images displaying symptoms of exudates and drusen, generating a total of 800 images for training (Table 4).

Figure 4. CLODSA application: (a) vertical flip transformation; (b) horizontal flip transformation; (c) vertical and horizontal flip transformation; (d) 90° rotation transformation



Source: Prepared by the authors

Table 4. Number of images augmented for training

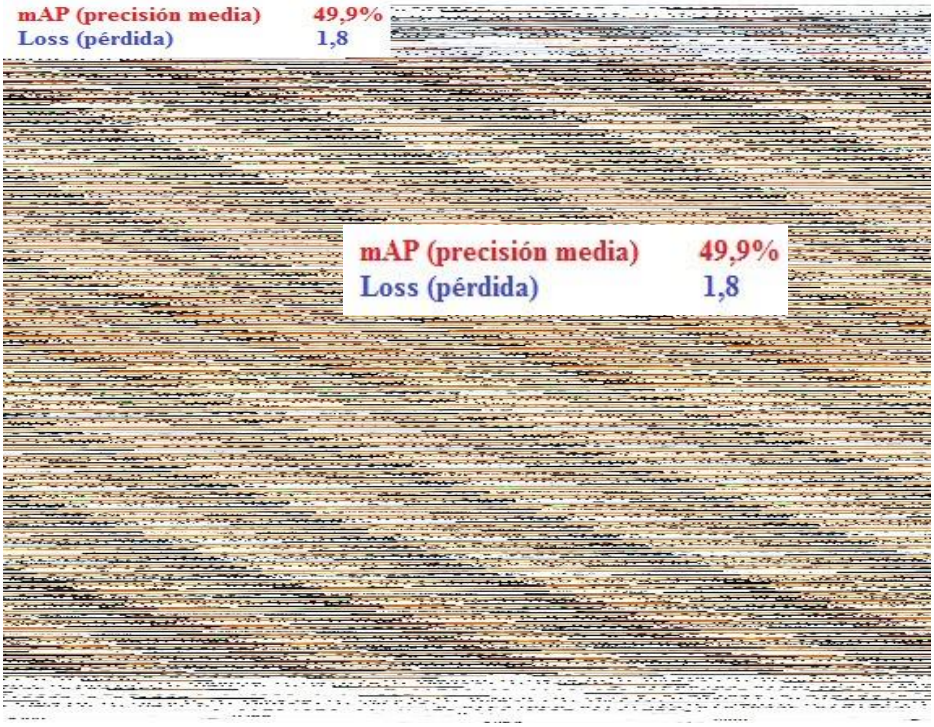
Model	Images	Data augmentation
Symptoms of hemorrhages and microaneurysms	200	708
Symptoms of exudates and drusen	150	800

Source: Prepared by the authors

Modeling

For this stage, four experiments were conducted based on the YOLOv3, YOLOv3_tiny, YOLOv4, and YOLOv4_tiny architectures. Based on the first experiment, a mAP of 49.9% was achieved over 7,000 iterations with a loss of 1.8 (Figure 5).

Figure 5. Results of mAP over 7000 iterations for the first experiment using the yolov3_tiny version of YOLO at a fixed resolution of 416 × 416



Source: Prepared by the authors

Based on the second experiment, a multiple resolution setup was used. In this case, Darknet [23] was used to obtain a random parameter. In the configuration files of the YOLOv3 small model, the default value of random was 0, indicating that no random resolution (or multiple resolution) was used during training. In this experiment, that value was changed to introduce random resolution. The remaining configurations and parameters remained constant (Figure 6).

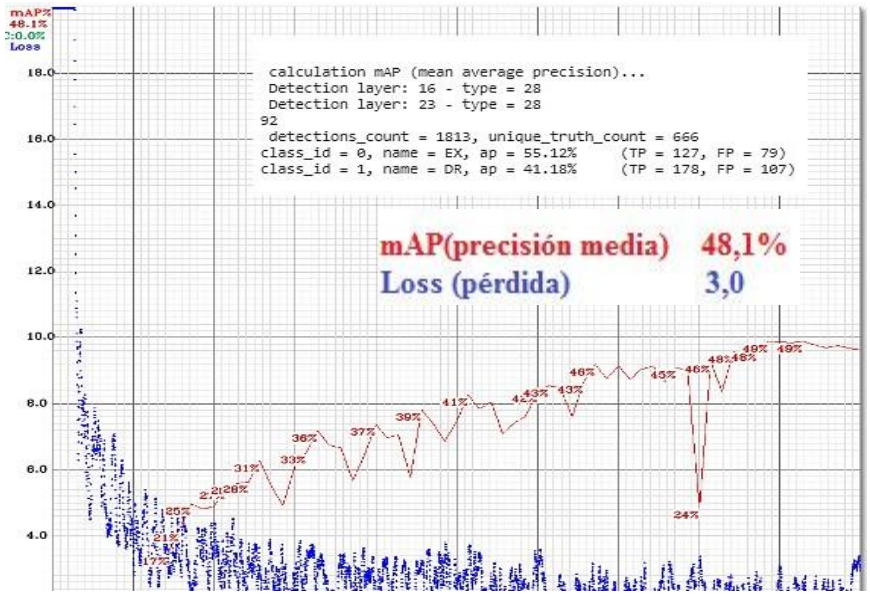
Figure 6. Configurations for the second experiment using the yolov3_tiny version of YOLO at several resolutions

```
[net]
# Testing
#batch=1
#subdivisions=1
# Training
batch=32
subdivisions=16
width=416
height=416
channels=3
momentum=0.9
decay=0.0005
angle=0
saturation = 1.5
exposure = 1.5
hue=.1

learning_rate=0.001
burn_in=1000
max_batches = 7000
policy=steps
steps=5600,6300
scales=.1,.1

[convolutional]
batch_normalize=1
filters=256
size=3
stride=1
pad=1
activation=leaky

[convolutional]
size=1
stride=1
pad=1
filters=21
activation=linear

[yolo]
mask = 1,2,3
anchors = 10,14,  23,27,  37,58,  81,82,  135,169,  344,319
classes=2
num=6
jitter=.3
ignore_thresh = .7
truth_thresh = 1
random=1
```

Source: Prepared by the authors

In the second experiment, training demonstrated an mAP of 48.1% and loss of 3.0 over 7,000 iterations. This is considerably higher than that observed in the first experiment with fixed-resolution training. This outcome highlights that training on smaller images makes the training data more challenging. Nevertheless, the model encountered numerous diverse scenarios, leading to an mAP that closely approached that observed in the first experiment (Figure 7).

Figure 7. Results of mAP over 7000 iterations for the second experiment using the YOLOv3_tiny version of YOLO at several resolutions



Source: Prepared by the authors

In the third experiment, we used the configuration used in the second experiment but transitioned to the larger version of YOLOv3. The width and height were adjusted to 608, resulting in a resolution of 608 × 608. The batch and subdivision were both set to 32, with max_batches at 7,000 and steps at 5,600 and 6,300. These parameters remained consistent with those used in training the smaller models (Figure 8).

Figure 8. Configurations for the third experiment using the large version of YOLOv3 at several resolutions

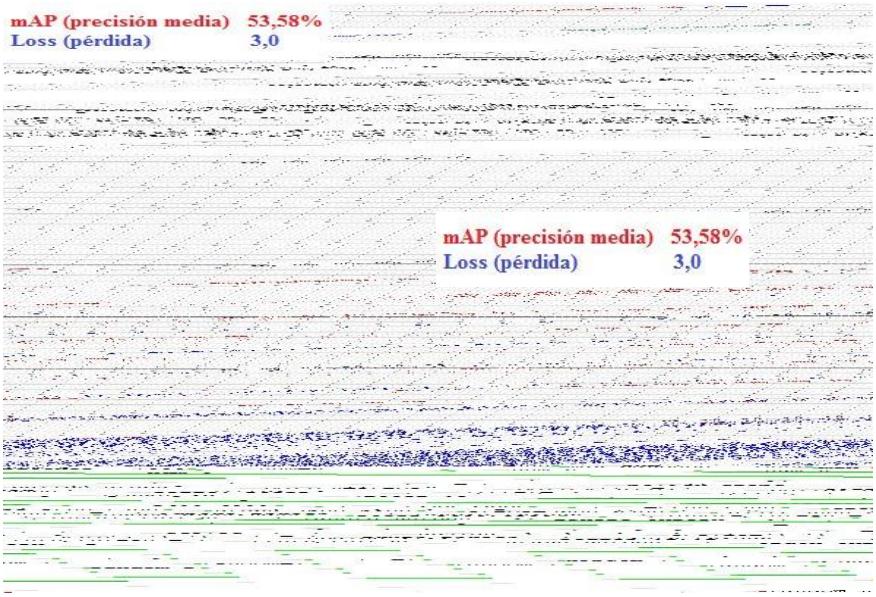
```
# Testing
#batch=1
#subdivisions=1
# Training
batch=32
subdivisions=32
width=608
height=608
channels=3
momentum=0.9
decay=0.0005
angle=0
saturation = 1.5
exposure = 1.5
hue=.1

learning_rate=0.001
burn_in=1000
max_batches = 7000
policy=steps
steps=5600,6300
scales=.1,.1

[convolutional]
batch_normalize=1
filters=16
```

```
[convolutional]
batch_normalize=1
filters=256
size=3
stride=1
pad=1
activation=leaky

[convolutional]
size=1
stride=1
pad=1
filters=21
activation=linear

[yolo]
mask = 1,2,3
anchors = 10,14,  23,27,  37,58,  81,82,  135,169,  344,319
classes=2
num=6
jitter=.3
ignore_thresh = .7
truth_thresh = 1
random=1
```

Source: Prepared by the authors

In the third experiment, training demonstrated an mAP of 53.58% over 7,000 iterations at the end of the training, with a loss of 3.0, which is similar to that of the second experiment, which is still higher than that observed in the first experiment with fixed-resolution training. This observation was also accepted because training the data becomes challenging when training on smaller images. Moreover, the model encountered several diverse scenarios, achieving a better mAP than the first and second experiments. Furthermore, considering that the large YOLOv3 version has 53 layers while the small YOLOv3_tiny version only uses 13 of these layers, it is expected that the mAP result would be much better (Figure 9).

Figure 9. Results of mAP over 7000 iterations for the third experiment using the large version of Yolov3 at several resolutions



Source: Prepared by the authors

In the fourth experiment, we reverted to a fixed resolution (Figure 10) and the other parameters remained constant.

Figure 10. Configurations for the fourth experiment using the large version of Yolov3 at a fixed resolution of 608 x 608

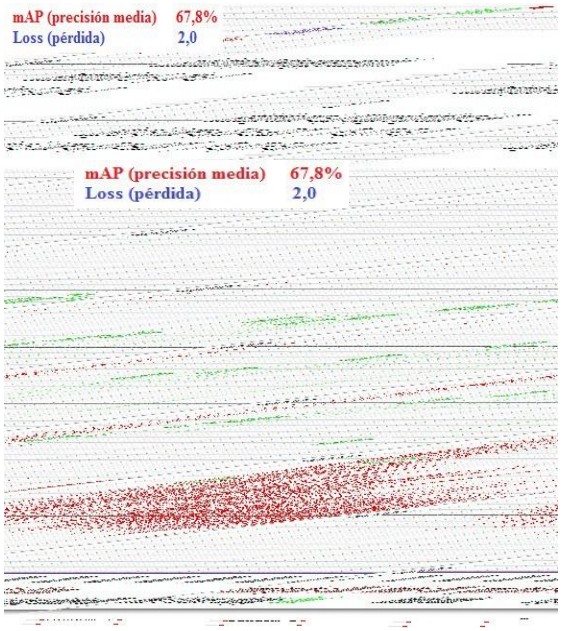
```
# Testing
#batch=1
#subdivisions=1
# Training
batch=32
subdivisions=32
width=608
height=608
channels=3
momentum=0.9
decay=0.0005
angle=0
saturation = 1.5
exposure = 1.5
hue=.1

learning_rate=0.001
burn_in=1000
max_batches = 7000
policy=steps
steps=5600,6300
scales=.1,.1

[convolutional]
batch_normalize=1
filters=16
```

```

[convolutional]
batch_normalize=1
filters=256
size=3
stride=1
pad=1
activation=leaky

[convolutional]
size=1
stride=1
pad=1
filters=21
activation=linear

[yolo]
mask = 1,2,3
anchors = 10,14,  23,27,  37,58,  81,82,  135,169,  344,319
classes=2
num=6
jitter=.3
ignore_thresh = .7
truth_thresh = 1
random=0
```

Source: Prepared by the authors

In the fourth experiment, training demonstrated an mAP of 67.08% achieved over 7,000 iterations at the end of the training, with a loss of 2.0. This performance is notably superior to that of the second and third experiments. The achieved mAP surpasses that of the first, second, and third experiments, suggesting that extending iterations with this configuration could yield excellent results (Figure 11).

Figure 11. Results of mAP over 7000 iterations for the fourth experiment using the YOLOv3 version of YOLO at a fixed resolution of 608 × 608.



Source: Prepared by the authors

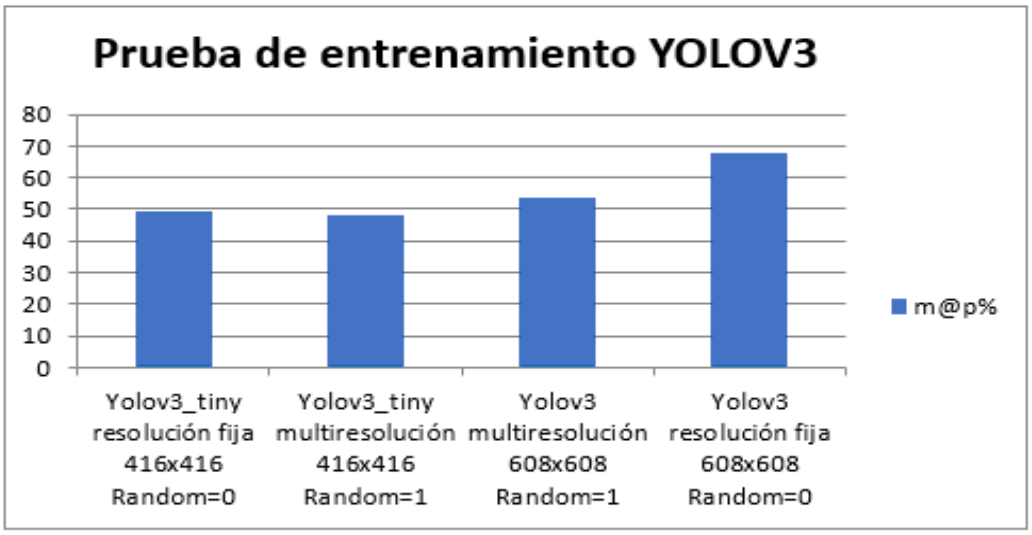
The comparison of the results obtained using different versions of YOLOv3 is provided in Table 5 and Figure 12.

Table 5. Results of experiments conducted using different versions of YOLOv3

YOLO version	Yolov3_tiny at a fixed resolution of 416 × 416 Random = 0	Yolov3_tiny at several resolutions of 416 × 416 Random = 1	Yolov3 at several resolutions of 608 × 608 Random = 1	Yolov3 at a fixed resolution of 608 × 608 Random = 0
mAP%	49.4	48.1	53.58	68.08

Source: Prepared by the authors

Figure 12. Results of experiments conducted using different versions of YOLOv3



Source: Prepared by the authors

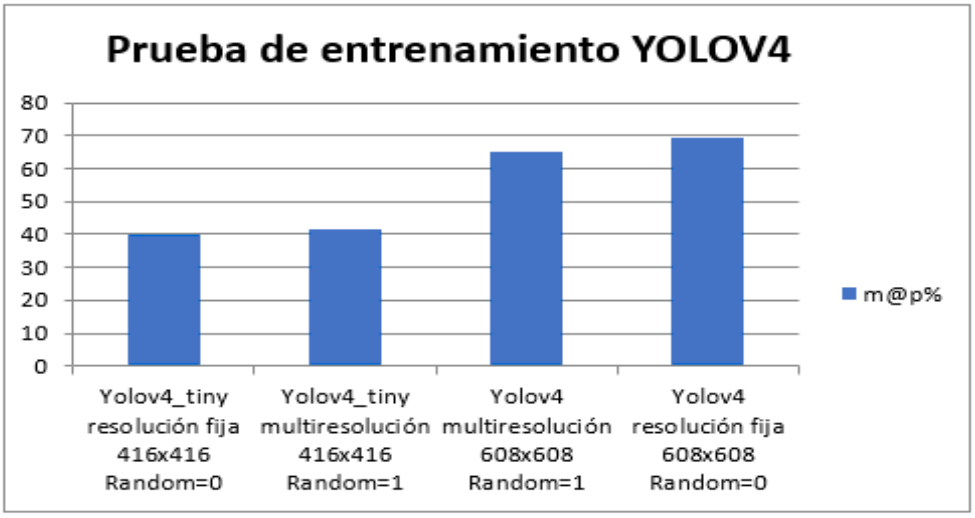
Using the YOLOv4 architecture, which is the updated version of YOLOv3, the same experiments were conducted to observe how much the mAP improves compared to YOLOv3 (Table 6 and Figure 13).

Table 6. Results of experiments conducted using different versions of YOLOv4

YOLO version	Yolov4_tiny at a fixed resolution of 416 × 416 Random = 0	Yolov4_tiny at several resolutions of 416 × 416 Random = 1	Yolov4 at several resolutions of 608 × 608 Random = 1	Yolov4 at a fixed resolution of 608 × 608 Random = 0
mAP%	40.02	41.50	65.32	69.34

Source: Prepared by the authors

Figure 13. Results of experiments conducted using different versions of YOLOv4



Source: Prepared by the authors

These experiments allow us to select the best-performing model (a model that is more stable and consistent), YOLOv4 [29], before proceeding with the next phases.

Training

Once the best-performing version is selected, the dataset is then divided into training, validation, and testing for both scenarios (Table 7).

Table 7. Training, validation, and testing distribution

Model	Training	Validation	Testing
Symptoms of hemorrhages and microaneurysms	602	71	35
Symptoms of exudates and drusen	680	80	40

Source: Prepared by the authors

Later, YOLOv4 is configured based on to the experiment through which the best results were achieved. The tools used for training and subsequent result visualization can be seen in Table 8.

Table 8. Tools used for training

Google Colab	LABELIMAGE	YOLO	Python
Enables execution and programming in Python without configuration	For the distribution of image labels	It is an object detection algorithm	Programming language used in the project

Source: Prepared by the authors

The transfer learning technique was used in this project. Transfer learning is an ML method where a model designed for one task is reused as the starting point for another task. In this particular case, this technique was employed to train the model starting from the yolov4.conv.137 model, which is capable of predicting 80 classes in an image or video. The models were trained for different iterations, achieving an average accuracy of 80% and 88.45% (Table 9).

Table 9. Model training results obtained through different iterations

Model	Iterations	mAP %
Hemorrhages and microaneurysms	14,700	80%
Exudates and drusen	20,000	88.49%

Source: Prepared by the authors

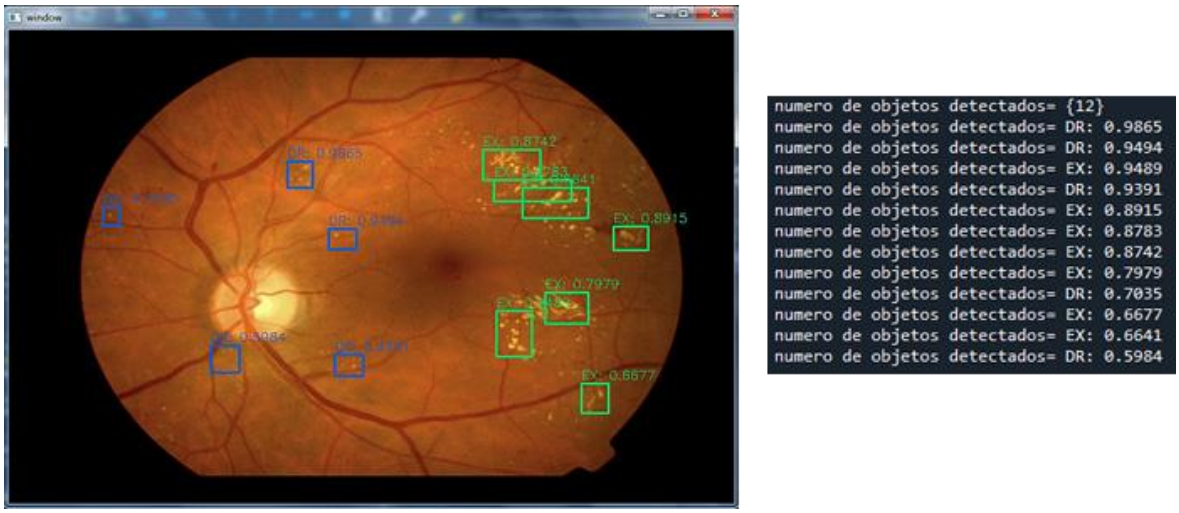
Detection Using the Trained Model

Using the trained model, tests were conducted on a new dataset, maintaining the configuration and weights (Figures 14 and 15).

Figure 14. Detection of hemorrhages and microaneurysms

Source: Prepared by the authors

Figure 15. Detection of exudates and drusen



Source: Prepared by the authors

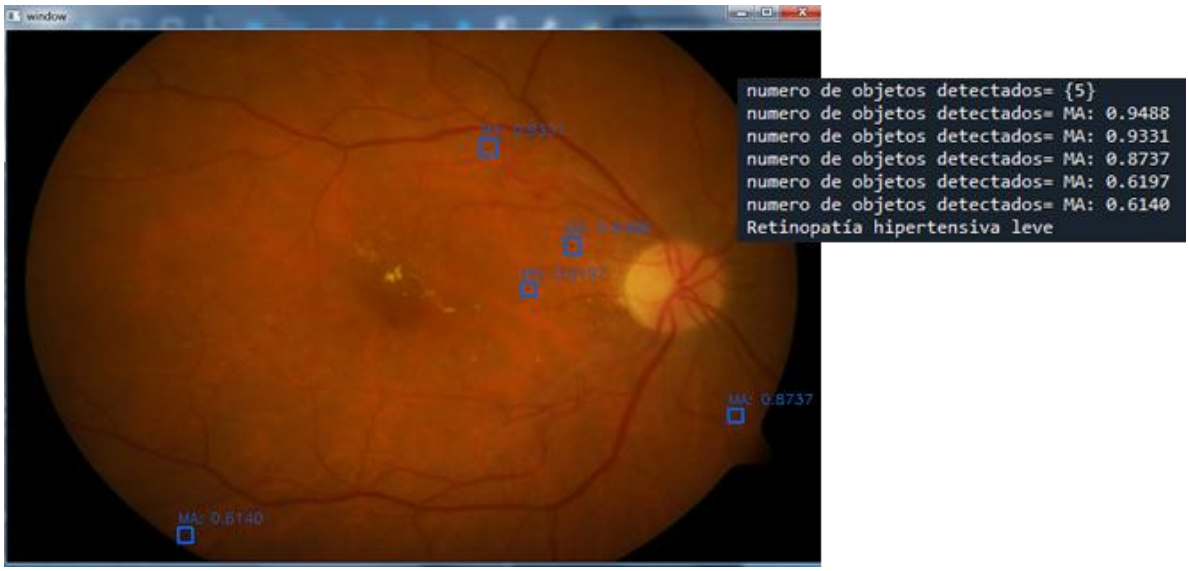
According to the number of symptoms present in the fundus image, a category of HR was associated: mild, moderate, or severe, referring to the grading of the Spanish Society of Retina and Vitreous (SERV) and the Spanish Society of Ophthalmology (SEO) (Table 10 and Figures 16 and 17).

Table 10. Stages of hypertensive retinopathy according to the number of symptoms detected

Equivalence	International classification	SERV and SEO classification
Only microaneurysms or only exudates	Mild retinopathy	Less than 5 or equal to 5
Microaneurysms, hemorrhages, and exudates	Moderate retinopathy	Less than 20 and greater than 5
Microaneurysms, hemorrhages, exudates, and drusen	Severe retinopathy	Greater than 20 or equal to 20

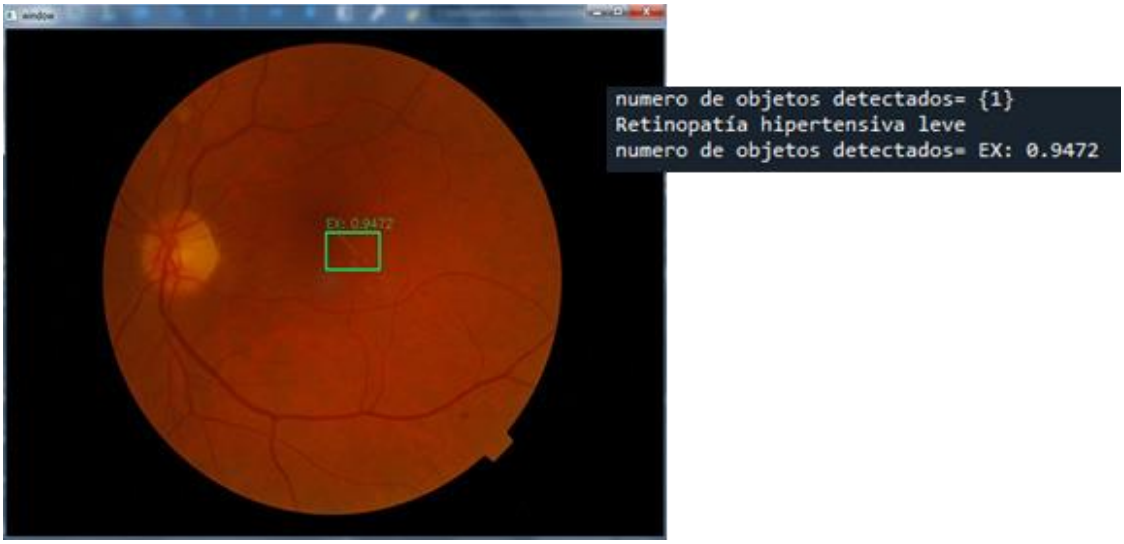
Source: Prepared by the authors

Figure 16. Hypertensive retinopathy (HR) staging using the proposed algorithm in Python



Source: Prepared by the authors

Figure 17. HR staging using the proposed algorithm in Python

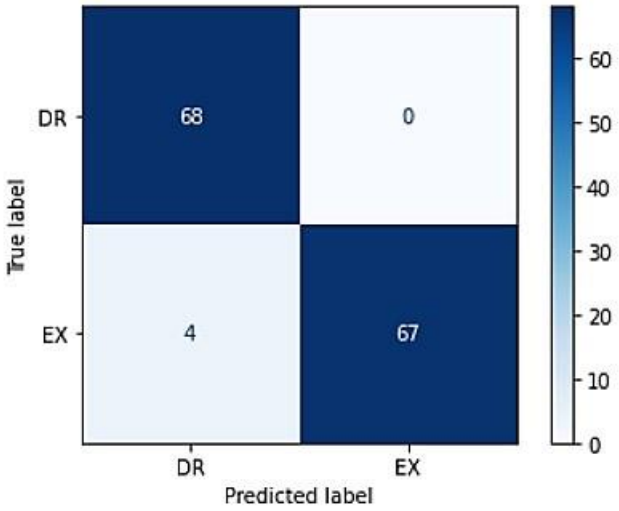


Source: Prepared by the authors

Assessment

Two scenarios were considered during the assessment. When assessing the different models using metrics associated with object detection using an unseen dataset for scenario one, we utilized 525 images to calculate the confusion matrix (Figure 18) and derived evaluation metrics (Table 11), along with the ROC curve (Figure 19).

Figure 18. Model 1 confusion matrix



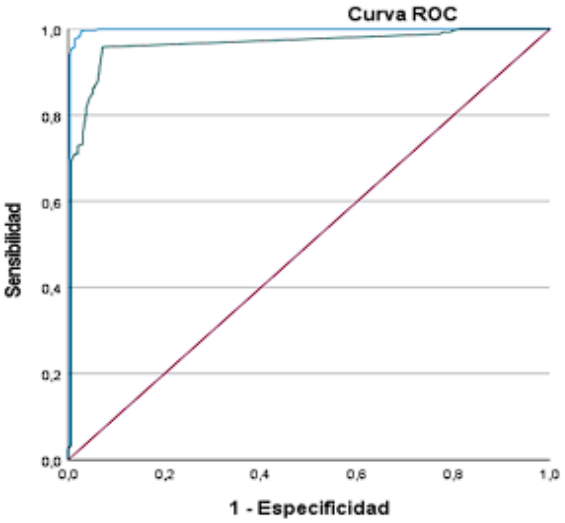
Source: Prepared by the authors

Table 11. Model 1 assessment metrics

	Accuracy	Recall	F1 score
Drusen	0.94	1.00	0.97
Exudates	1.00	0.94	0.97

Source: Prepared by the authors

Figure 19. ROC curve of Model 1



Origen de la curva
EX
DR
Línea de referencia

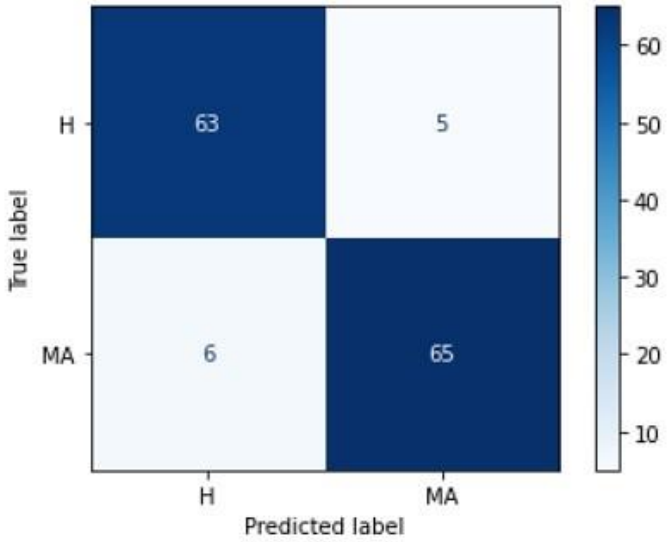
Área bajo la curva

Variables de resultado de prueba	Área
EX	0.996
DR	0.962

Source: Prepared by the authors

In scenario two, hemorrhages and microaneurysms were detected by calculating the confusion matrix (Figure 20), assessment metrics (Table 12), and ROC curve (Figure 21).

Figure 20. Model 2 confusion matrix



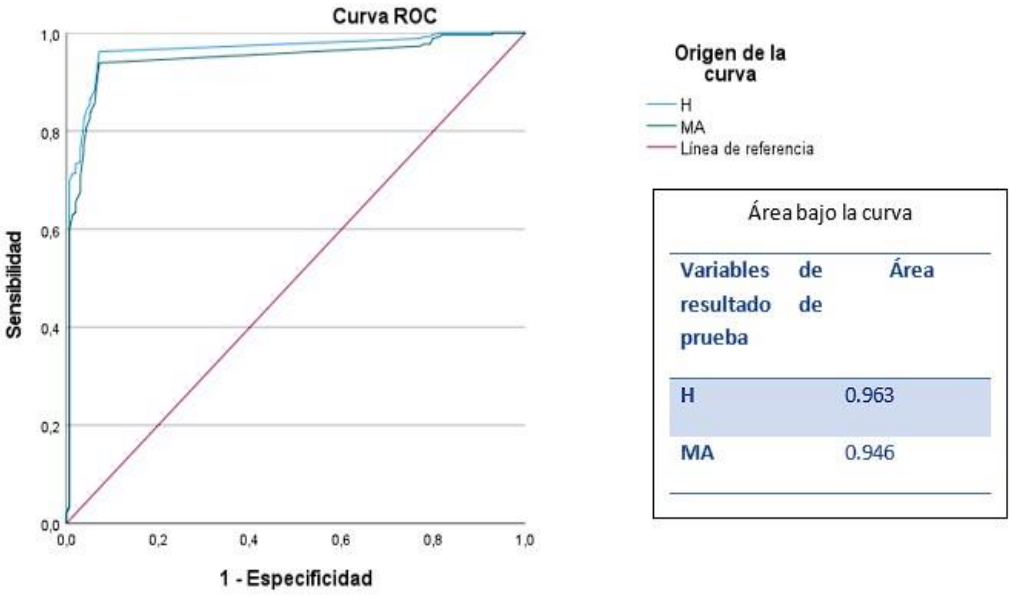
Source: Prepared by the authors

Table 12. Model 2 assessment metrics

	Accuracy	Recall	F1-Score
Hemorrhages	0,91	0,93	0,92
Microaneurysms	0,93	0,92	0,92

Source: Prepared by the authors

Figure 21. ROC curve of Model 2



Source: Prepared by the authors

Discussion

Following the validation of both detection models and the assessment of various metrics across training, validation, and testing datasets, it becomes evident that the YOLOv4 architecture, particularly in the fixed-resolution configuration highlighted in the fourth experiment, outperforms the YOLOv3 architecture. To measure the effectiveness of both models, the confusion matrix was utilized, revealing that for model one, which detects drusen and exudates, there were 67 true negatives and 68 true positives. In particular, out of the 135 cases (diagonal) from a total of 139 images taken from the testing dataset, the category of each class was accurately predicted. Similarly, for model two, which was designed for detecting hemorrhages and microaneurysms, it yielded 65 true negatives and 63 true positives. This indicates that out of the 128 cases (diagonal) from a total of 139 during testing, the category of each class was accurately predicted.

Upon evaluating the results using the confusion matrix, precision, sensitivity, and F1-score metrics were computed. Model one demonstrated high recall for drusen with a value of 1.0, and for exudates, it achieved 0.94 in recall, with precision reaching 0.94 and 1.0 for drusen and exudates, respectively. These figures suggest that model one effectively detects the classes, and the F1 score of 97% further affirms its robust performance. On the contrary, model two exhibited high precision and recall, indicating its effective handling of class detection in the majority of instances. It achieved an F1 score of 92%, reinforcing its overall good performance. Furthermore, when analyzing the AUC results on the ROC curve, model one displayed an AUC of 0.99 for exudates and 0.96 for drusen and model two showcased an AUC of 0.96 for hemorrhages and 0.94 for microaneurysms. These outcomes establish that these two models attained the best results, enabling efficient detection of the symptoms for which they were trained. This underscores the robustness of the models, bolstered by a substantial number of images and a pretrained model.

The presentation of the data collected aligns with the proposed objectives, showcasing clear and precise results in accordance with the established method. The relevant tables and figures required for presenting the results are included in this section.

Conclusions

The results obtained in this study provide an encouraging outlook for detecting symptoms of HR. Moreover, the implemented ML algorithm demonstrates optimal results, which can serve as supportive information for specialists in decision-making regarding the ocular pathology discussed in this study. The ML algorithm, employed in conjunction with the proposed method for processing fundus images, successfully detected four symptoms present in HR: hemorrhages, microaneurysms, exudates, and drusen, achieving AUC values of 0.99 and 0.96. This positions it as a viable alternative for aiding specialists in making diagnoses.

As indicated by the study findings, the algorithm, based on the quantity of detections, can categorize the ocular pathology as mild, moderate, or severe. The data collection phase is crucial, and emphasis should be placed on the accurate gathering of images and their respective annotations. In addition, emphasis on image preprocessing is essential for obtaining good results. The phases of data understanding and data preparation performed in this study, including feature extraction, were successful, yielding positive outcomes in the evaluation of HR symptom detection [24].

Two iterations of the YOLO detection algorithm, YOLOv3 and YOLOv4, were implemented with distinct configurations. The outcomes favored the adoption of the YOLOv4 architecture for this study, showcasing elevated values across all metrics, including sensitivity, accuracy, precision, F1 score, and AUC. This study paves the way for further exploration.

- The model can be deployed on a web server, enabling interaction with users through a platform. This is expected to have a notable impact on supporting specialists in diagnosis, thereby increasing the likelihood of timely diagnoses for patients with HR.
- Inclusion of images from patients in a clinical setting can help enrich the database for subsequent

algorithm training.

- Execution of training on a high-performance GPU can be performed to eliminate resource limitations.
- Investigation of training with more recent YOLO models, such as YOLOv5 or YOLOv6, can be conducted to improve the mAP value.

References

1. EEVV, DANE vital statistics, “difuciones fetales y no fetales” [Fetal and non-fetal diffusions], 05/28/2022, online: https://www.dane.gov.co/files/investigaciones/poblacion/bt_estadisticasvitales_defunciones_Itrim_2022pr.pdf.
2. WHO, World Health Organization, “¿de que morimos?” [What do we die of?] March 18, 2019, online: https://www.who.int/es/news-room/fact-sheets/detail/hypertension.
3. WHO, World Health Organization, “Hypertension”, May 17, 2021, online: https://www.who.int/es/news-room/fact-sheets/detail/hypertension.
4. MinSalud, Ministerio de Salud y Protección Social de Colombia, “Conoce tus números” [Know your numbers], June 14, 2020, online: https://www.minsalud.gov.co/Paginas/Conoce-tus-numeros-para-prevenir-la-hipertension-arterial.aspx.
5. Daniela Gasca Cuello. P. Manifestaciones de la retinopatía hipertensiva y de la retinopatía diabetica en población adulta [Manifestations of hypertensive retinopathy and diabetic retinopathy in adult population]. Scientific & Education Medical Journal / Vol. 1, N° 1, 2021, 9.
6. Tsukikawa M, W Stacey A. A Review of Hypertensive Retinopathy and Chorioretinopathy; 12: 67-73. 2020, doi: 10.2147/OPTO.S183492.
7. Zhang Y, Zhao L, Li H, Wang Y. Risk factors for hypertensive retinopathy in a Chinese population with hypertension: The Beijing Eye study. Experimental and therapeutic medicine. Experimental and therapeutic medicine. 2020; 17 (1): 435-458, doi: 10.3892/etm.2018.6967.
8. Isabella Torres Revelo, U. A. Interpretabilidad De Un Modelo Basado En Aprendizaje Profundo Para El Diagnóstico De Retinopatía Diabética [Interpretability of a Model Based on Deep Learning for the Diagnosis of Diabetic Retinopathy]. Cali. (2020).
9. Vision atlas “numero de personas afectadas porperdida de visión” [number of people affected by vision loss], 2020, online: https://www.iapb.org/es/learn/vision-atlas/.
10. ANORO, A. T. Hypertensive Retinopathy. Medicina General 11, Review. 2020; 524-564.
11. Herrera, M. D. Análisis y Diseño de un Sistema para Identificar Signos de Retinopatía Hipertensiva a través de Imágenes de Retina, Aplicando la Tecnología de Deep Learning [Analysis and design of a system to identify symptoms of hypertensive retinopathy through retinal images, applying deep learning technology]. Guayaquil, 2018.
12. Sarmad Khitran, M. U. Automated system for the detection of hypertensive. Image Processing Theory, Tools and Applications, 6, 2014, DOI: 10.1109/IPTA.2014.7001984.
13. Irshad, S. Classification of retinal vessels into arteries and veins for detection of hypertensive retinopathy. Cairo International Biomedical Engineering Conference, 4, 2015.
14. Triwijoyo, B. K. The classification of hypertensive retinopathy using convolutional neural network. ScienceDirect, 8, 2017. DOI: 10.1109/CIBEC.2014.7020937.
15. [Arasy, R. Detection of hypertensive retinopathy using principal component analysis (PCA) and backpropagation neural network methods. AIP Conference Proceedings 2092, 9, 2019, https://doi.org/10.1063/1.5096735.
16. E. Decencière et al., 2013, http://dx.doi.org/10.1016/j.irbm.2013.01.010, The link to access this dataset is:

http://www.adcis.net/en/third-party/e-ophtha/,

17. Cuello Navarro, J., Barraza Peña, C. & Escorcia-Gutierrez, J. (2020). Una revisión de los métodos de deep learning aplicados a la detección automatizada de la retinopatía diabética [A review of deep learning methods applied to the automated detection of diabetic retinopathy]. Sextant Magazine, 23, pp. 12 - 27, 2020, The link to access this dataset is: https://www.kaggle.com/c/diabetic- retinopathy-detection/data
18. Niemeijer et al., 2010 The link to access this image database is http://webeye.ophth.uiowa.edu/ROC/.
19. Mauricio Menegaz. “Understanding YOLO.” Hackernoon. https://hackernoon.com/understanding-YOLO-f5a74bbc7967. Last accessed: September 2022.
20. Data augmentation-assisted Deep learning of hand-drawn partially colored sketches for visual, search: https://www.researchdate.net/publication/319413978_Data_augmentation-assisted_Deep_ learning_of_handfrawn_artially_colored_sktches_for_visual_search,DOI: https://doi.org/10.1371/ journal.pone.0183838.
21. Asado-García, Á., Domínguez, C., García-Domínguez, M. et al. CLoDSA: A tool for augmentation in classification, localization, detection, semantic segmentation and instance segmentation tasks. BMC Bioinformatics 20, 323, 2019.
22. Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Harhali. “You only look once: Unified, real-time object detection.”, 2016. https://arxiv.org/abs/1506.02640.
23. Jonathan Hui. “Real-time object detection with YOLO, YOLOv2, and now YOLOv3.” Last accessed: September 2022.
24. C. A. Ruíz Ramírez, D. M. Montoya Quintero, and J. A. Jimenez Builes, “Un Ambiente visual integrado de desarrollo para el aprendizaje de programación en robótica” [An integrated visual development environment for learning programming in robotics], Investigación e Innovación en Ingenierías, vol. 9, No. 1, pp. 7–21, 2021. DOI: https://doi.org/10.17081/invinno.9.1.3957.