

**DETECCIÓN DE SIGNOS DE RETINOPATÍA HIPERTENSIVA USANDO  
ALGORITMOS DE APRENDIZAJE AUTOMÁTICO EN IMÁGENES DIGITALES DE  
FONDO DE OJO**



**CHRISTIAN CAMILO DELGADO MOLINA**

**UNIVERSIDAD DEL CAUCA  
FACULTAD DE CIENCIAS NATURALES, EXACTAS Y DE LA EDUCACIÓN  
MAESTRÍA EN INGENIERÍA FÍSICA  
POPAYÁN  
2023**

**DETECCIÓN DE SIGNOS DE RETINOPATÍA HIPERTENSIVA USANDO  
ALGORITMOS DE APRENDIZAJE AUTOMÁTICO EN IMÁGENES DIGITALES DE  
FONDO DE OJO**



**CHRISTIAN CAMILO DELGADO MOLINA**

**Ingeniero Físico**

Trabajo de Grado presentado como requisito parcial para optar al título de  
Magíster en Ingeniería Física

Director:

**PhD. Rubiel Vargas Cañas**

Grupo de Investigación en Sistemas Dinámicos, Instrumentación y Control  
(SIDICO)

Codirectores:

**MSc. Jairo Alfonso Vásquez López**

Maestría Ciencias Básicas Médicas, énfasis Morfología

**Esp. Leonardo Africany Villamil**

*Optómetra, Universidad de la Salle. Especialista en cuidado primario, Pontificia Universidad  
Javeriana*

**UNIVERSIDAD DEL CAUCA  
FACULTAD DE CIENCIAS NATURALES, EXACTAS Y DE LA EDUCACIÓN  
MAestrÍA EN INGENIERÍA FÍSICA  
POPAYÁN  
2023**

## NOTA DE ACEPTACIÓN

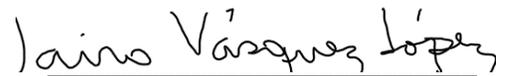
Aprobado por el comité de grado en cumplimiento de los requisitos exigidos por la Universidad del Cauca para optar al título de Magister en Ingeniería Física.



---

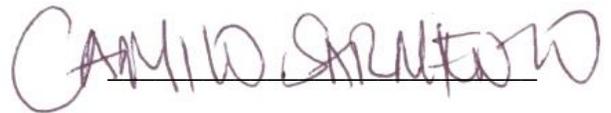
PhD. Rubiel Vargas Cañas

DIRECTOR



---

MSc. Jairo Alfonso Vásquez López  
.CODIRECTOR



---

MSc. Camilo Sarmiento Torres

JURADO



---

MSc. Martin Alonso Ruiz Orozco

JURADO

Fecha de sustentación: 02 de Febrero 2023

## **AGRADECIMIENTOS**

Primero que todo agradecer a Dios por permitir cumplir con la metas propuestas y soñadas, segundo expresar el más sincero y profundo agradecimiento a todas las personas que contribuyeron al desarrollo de este trabajo, en especial a mi familia, a mi Madre Luz Amparo Molina Arango, a mi Padre Alfredo Delgado García que está en el cielo, a mi esposa Deicy Clarena Otero Hernández, amigos y compañeros de trabajo; al Dr. Rubiel Vargas Cañas, director y orientador, por sus fundamentales aportes, seguimiento y dedicación. Al MSc Jairo Alfonso Vásquez por sus aportes, seguimiento y dedicación; al especialista Leonardo Africany Villamil, por brindarnos el conocimiento y dedicación y asesorías brindadas.

¡Muchas Gracias!

## Tabla de Contenido

<b>1. Introducción.....</b>	<b>1</b>
<b>1.1. Justificación.....</b>	<b>4</b>
<b>1.2. Objetivos .....</b>	<b>4</b>
<b>1.2.1 Objetivo General .....</b>	<b>4</b>
<b>1.2.2 Objetivos Específicos.....</b>	<b>4</b>
<b>1.3. Descripción metodologica.....</b>	<b>5</b>
<b>1.3.1 Comprensión del problema .....</b>	<b>5</b>
<b>1.3.2 Comprensión de los datos .....</b>	<b>6</b>
<b>1.3.3 Preparación de los datos.....</b>	<b>6</b>
<b>1.3.4 Modelado.....</b>	<b>6</b>
<b>1.3.5 Evaluación.....</b>	<b>6</b>
<b>1.4 Contribuciones.....</b>	<b>6</b>
<b>1.5 Estructura del documento.....</b>	<b>7</b>
<b>2. Retinopatía hipertensiva y sus signos.....</b>	<b>7</b>
<b>2.1 Marco conceptual.....</b>	<b>8</b>
<b>2.1.1 Anatomía del ojo.....</b>	<b>8</b>
<b>2.1.2 Macula.....</b>	<b>8</b>
<b>2.1.3 Nervio óptico.....</b>	<b>9</b>
<b>2.1.4 Retina.....</b>	<b>10</b>
<b>2.1.5 Hipertensión Arterial.....</b>	<b>11</b>
<b>2.1.6 Retinopatía Hipertensiva.....</b>	<b>11</b>

2.1.7 Fundoscopia.....	13
2.1.8 Retinopatía Hipertensiva Aguda y crónica.....	14
2.1.9 Signos y síntomas de retinopatía hipertensiva.....	14
2.2.0 Modelos de aprendizaje automático.....	16
2.2.1 Redes neuronales convolucionales.....	16
2.2.2 YOLO (You only look once).....	18
2.2.3 Los beneficios que presenta YOLO.....	18
2.2.4 Arquitectura de la Red.....	20
2.2.5 YOLOv2.....	21
2.2.6 YOLOv3.....	24
2.2.7 YOLOv4.....	26
2.2.8 Transfer Learning.....	27
2.3 Estado de la técnica.....	27
2.3.1 Brechas de investigación.....	33
3. Metodología.....	34
3.1 Comprensión del problema.....	34
3.2 Comprensión de los datos.....	35
3.2.1 Recolección de los datos.....	35
3.2.2 Anotación de los datos.....	35
3.3 Preparación de los datos.....	36
3.3.1 Aumento de datos.....	37
3.3.2 Etiquetado.....	38
3.4 Modelado.....	39
3.4.1 YOLO.....	39
3.4.2 Arquitectura YOLO.....	40

3.4.3 Arquitectura YOLOv3.....	41
3.4.4 Procesamiento YOLOv3.....	42
3.4.5 Arquitectura YOLOv4.....	44
3.5 Evaluación.....	45
3.5.1 Medida de la precisión en la detección de objetos.....	46
3.5.2 Matriz de confusión teórica.....	46
4. Resultados y discusión.....	48
4.1 Conjunto de datos.....	48
4.2 Etiquetado.....	49
4.3 Aumento de datos.....	50
4.3.1 Herramienta CLODSA.....	50
4.4 Modelado.....	52
4.4.1 Configuración de archivos y datos del modelo.....	53
4.4.2 Entrenamiento del modelo.....	63
4.4.2.1 Entradas.....	63
4.5 Detección del modelo entrenado.....	66
4.6 Evaluación del modelado.....	69
4.7 Discusión.....	73
5. Conclusiones y trabajos futuros.....	74
5.1 Conclusiones.....	74
5.2 Recomendaciones.....	75
5.3 Trabajos futuros.....	75
6. Bibliografía.....	76

## Índice de tablas

Tabla 1. Relación Arterio-venosa. ....	12
Tabla 2. Estado de la técnica.....	32
Tabla 3. Matriz de confusión teorica. ....	46
Tabla 4. Cantidad de imágenes organizadas por resolución para H y MA.....	48
Tabla 5. Cantidad de imágenes organizadas por resolución para EX y DR. ....	49
Tabla.6. Cantidad de imágenes aplicando aumento de datos.....	51
Tabla 7. Parametros usados para el entrenamiento de YOLO v3 y YOLO v4. ....	53
Tabla 8. Resultados del maP del experimento con YOLO v3 .....	61
Tabla 9. Resultados del maP del experimento con YOLOv4 .....	62
Tabla 10. Distribución de entrenamiento, validación y test.....	64
Tabla 11.Herramientas utilizadas para el entrenamiento.....	64
Tabla 12 Resultado del entrenamiento de los modelos con diferentes iteraciones.....	65

## ÍNDICE DE FIGURAS

Figura 1. Metodología CRISP-DM .....	5
Figura 2. Anatomía del ojo Humano.....	8
Figura 3. Macula, anatomía del ojo humano.....	9
Figura 4. Nervio óptico, anatomía del ojo humano. ....	9
Figura 5. Retina, anatomía del ojo humano. ....	10
Figura 6. Relación Arteria- Vena.....	12
Figura 7. Retina normal y con retinopatía . ....	13
Figura 8. Imagen del oftalmoscopio. ....	13
Figura 9. Imagen de la retina con su patología. ....	15
Figura 10. Diagrama de una red convolucional.....	16
Figura 11. Ejemplo de convolución.....	17
Figura 12. Ejemplo de 96 filtros extraídos en la primera capa de una CNN.....	17
Figura 13. Ejemplo de max Pooling. ....	18
Figura 14. Ejemplo del cálculo de coordenada de una imagen.....	19
Figura 15. Descripción gráfica de la intersección sobre la unión .....	20
Figura 16. Esquema de la arquitectura de una red neuronal .....	21
Figura 17. Ilustración del funcionamiento del algoritmo K-means. ....	22
Figura 18. Predicción de los cuadros delimitadores. ....	23
Figura 19. Explicación intuitiva de un cuadro delimitador.....	23
Figura 20. Procesamiento YOLOv3. ....	25
Figura 21. Arquitectura YOLOv4.....	26
Figura 22. Ejemplo de imagen de base de datos E-Ophta (EX y DR).....	36

Figura 23. Ejemplo de imagen de base de datos E-Ophta (H y MA). .....	36
Figura 24. Carpeta con n imágenes de signos de H y MA.....	37
Figura 25. Ejemplo de aumento de datos.....	38
Figura 26. Etiquetado con herramienta LabelImage.....	38
Figura 27. Etiquetado y formato de etiqueta YOLO.....	39
Figura 28. Detección de objeyos en YOLO.....	40
Figura 29. Arquitectura Darknet 53 y YOLO_tiny.....	41
Figura 30. Procesamiento YOLOv3. ....	43
Figura 31. Arquitectura YOLOv4.....	45
Figura 32. Metricas de evaluación para detección de objetos. ....	46
Figura 33. Etiqueta de H y MA.....	46
Figura 34. Etiqueta de EX y DR. ....	50
Figura 35. Aplicación CLODSA.....	51
Figura 36. Ilustración funcionamiento YOLO.....	53
Figura 37. Configuración del primer experimento con la versión pequeña de YOLOv3_tiny.	54
Figura 38. Resultado del maP del prmier expermiento.....	55
Figura 39. Configuración del segundo experimento con YOLOv3_tiny.....	56
Figura 40. Resultados del maP del segundo experimento. ....	57
Figura 41. Configuración del experimento tres con YOLOv3.....	58
Figura 42. Resultados del tercer Experimento.....	59
Figura 43. Configuración del cuarto experimento con YOLOv3. ....	60
Figura 44. Resultados del cuarto experimento con YOLOv3.....	61
Figura 45. Resultados de los cuatro experimentos YOLOv3 en barras .....	62
Figura 46. Resultado de los cuatro experimentos YOLOv4 en barras .....	63
Figura 47. maP de Hemorragias y Micro-aneurisma.....	63

Figura 48. maP de Exudados y drusas .....	66
Figura 49. Detección de hemorragias con el algoritmo propuesto. ....	66
Figura 50. Detección de Micro-Aneurisma cn el algoritmo propuesto. ....	67
Figura 51. Detección de Hemorragias y micro-Aneurisma. ....	67
Figura 52. Detección sin hemorragias ni micro-aneurisma. ....	68
Figura 53. Detección de Exudados. ....	68
Figura 54. Detección de Drusas. ....	69
Figura 55. Matriz de confusión para el modelo 1. ....	70
Figura 56. Metricas de evaluación del modelo 1. ....	70
Figura 57. Curva ROC del modelo1. ....	71
Figura 58. Matriz de Confusion del modelo 2. ....	72
Figura 59. Metricas de evaluación del modelo 2. ....	72
Figura 60. Curva ROC para el modelo 2. ....	73

## Lista de acrónimos

### LISTA DE ACRÓNIMOS

Acc	Exactitud
H	Hemorragia
MA	Micro-aneurisma
EX	Exudado
AUC	Área bajo la curva ROC
DR	Drusas
CNN	Red Neuronal Convolutiva
CRISP-DM	Cross Industry Standard Process of Data Mining
FN	Falso Negativo
FP	Falso Positivo
K-NN	K-Vecinos más cercanos
NB	Clasificador bayesiano Ingenuo
PCA	Análisis de Componentes Principales
Pr	Precisión
SVM	Máquina de soporte vectorial
TN	Verdadero Negativo
TP	Verdadero Positivo
HR	Retinopatía hipertensiva

# CAPÍTULO 1

## 1. Introducción

Hoy en día a nivel mundial la enfermedad cardiovascular es la primera causa de muerte, tan solo en Colombia, según el Departamento Administrativo Nacional de Estadística (Dane), los padecimientos cardíacos causaron el 17,1 % de los fallecimientos registrados en hombres y el 18% en mujeres en el 2022. Dado que su principal factor de riesgo es la hipertensión arterial (HTA), enfermedad que constituye un problema de salud por sus complicaciones y por su elevada trascendencia sanitaria, económica y social. La HTA frecuentemente suele ser asintomática y causa complicaciones que pueden llegar a ser graves cuando no se tratan a tiempo [1].

La Diabetes Mellitus (DM) y la HTA son dos enfermedades crónicas de gran prevalencia que, de alguna u otra forma afectan a la circulación sanguínea y generan numerosas complicaciones, entre las que se encuentran las enfermedades oculares, que se manifiestan en forma de retinopatías. La Retinopatía es cualquier proceso patológico que suponga alteración de la retina; desde la perspectiva anatómica, la retina es la capa más interna de las tres que conforman el globo ocular [2]. En este estudio se hablará de la retinopatía hipertensiva, la cual es un daño en la retina ocasionado por la elevación abrupta de la presión arterial sea por causa primaria o secundaria, con disminución de la agudeza visual que puede avanzar hasta la ceguera. Diferentes estudios han demostrado que la retinopatía hipertensiva se asocia a la subida de las cifras tensionales sin embargo también influye factores como la arterioesclerosis, la enfermedad de las arterias carótidas y edad avanzada. Se espera que para el 2025 una cantidad de 1,56 billones de personas tengan problemas de hipertensión arterial [2-6].

Según la Organización Mundial de la Salud (OMS), a nivel mundial, se estima que aproximadamente 1.300 millones de personas viven con alguna forma de deficiencia visual [5], y además en el mundo hay 1.130 millones de personas con hipertensión, donde la mayoría de ellas (cerca de dos tercios) vive en países de ingresos bajos y medianos. Según el Ministerio de Salud y Protección Social de Colombia (Minsalud) en su proyecto “conoce tus números” publicado en junio del 2020 informa que aproximadamente 4 de cada 10 adultos en Colombia sufren de hipertensión arterial, donde el 60% de estos no lo saben aún, además estos factores aumentan con la edad, teniendo pacientes mayores de 50 años con un 55% de posibilidad de ser hipertensos. Solo en el 2018 en Colombia se reportaron 4’048.776 personas diagnosticadas con hipertensión arterial. Sin embargo, datos provenientes de estudios poblacionales muestran que

este número puede ser tres veces mayor. En general, los casos son mayores en el sexo femenino que en el masculino. [7]

En la retinopatía hipertensiva el diagnóstico se da mediante un examen de fondo de ojo, que representa para el médico una ventana exploratoria de carácter único gracias a la cual es posible complementar el estudio sistémico de muchos pacientes. De esta manera, puede evaluarse el árbol vascular de la retina, el nervio óptico y la retina en toda su extensión [8].

La red capilar retiniana se puede visualizar fácilmente mediante técnicas no invasivas, como la retinografía, que permite, en la actualidad, un registro fotográfico comparativo digitalizado de alta calidad de las características circulatorias y el aspecto de los capilares retinianos. Por lo tanto, los cambios microvasculares en la retina, son marcadores de la patología vascular en la circulación coronaria y cerebral y, en consecuencia, pueden predecir el riesgo de las principales enfermedades cardiovasculares [6].

Además de los métodos habitualmente utilizados en medicina para realizar el diagnóstico existen métodos de diagnóstico asistidos por computador que ofrecen confiabilidad, reproducibilidad y cuantificación de diferentes variables, lo cual facilita el diagnóstico clínico. Existen muchas enfermedades que se ayudan con esta técnica, en cambio otras, continúan diagnosticándose de manera tradicional [9]. Comparar gran cantidad de imágenes médicas y reconocer los patrones necesarios que puedan identificar cada enfermedad, puede inducir en los especialistas agotamiento y errores al comparar gran cantidad de datos y casos [1,8].

La detección de la Retinopatía Hipertensiva (RH), es una actividad que consume recursos importantes como lo son el tiempo y el dinero. Desafortunadamente este procedimiento se lo realiza actualmente de manera manual, donde el especialista lo que hace es identificar cruces entre venas, exudados, hemorragias y drusas que en ocasiones no son fáciles de ver en la imagen del fondo de ojo. Generalmente los especialistas utilizan diferentes equipos o dispositivos que ayudan a visualizar la retina, pero éstos solo sirven para poder tener una mejor visión del fondo de ojo humano. A través de ellos se pueden también obtener imágenes para luego diagnosticar la enfermedad a través de los signos. Para este diagnóstico se necesita de un médico especialista en esta rama, que evalúe los diferentes síntomas que conlleva esta enfermedad. Además uno de los grandes obstáculos es que son pocos los médicos capacitados en esta rama, según el MindSalud solo el 4,1% son especialistas en oftalmología en Colombia y comúnmente se los encuentran en las grandes ciudades. Tan solo en el municipio de Popayán como un valor estimado tenemos 20 especialistas en las diferentes EPS de la ciudad para una población de 318.059 personas. [4,7-9].

Se han desarrollado diversos estudios computacionales que están presentes en la literatura relacionados con la retinopatía hipertensiva (RH) que buscan identificar los diferentes signos que

indican presencia de presión arterial en un paciente. Uno de ellos es un sistema automatizado propuesto por Sarmad Khitran [22] para la detección de la RH utilizando la relación arteriovena (AV). El sistema consiste en un método para la clasificación de vasos, como arterias y venas utilizando un vector de características y un clasificador híbrido, donde se utilizan dos bases de datos de imágenes de fondo de ojo digitales disponibles públicamente, VICAVR y DRIVE.

La técnica propuesta por Irshad, S. [17] consiste en clasificar los vasos en arterias y venas, utilizando una máscara de vaso binario y una localización del centro del disco óptico (OD) para extraer la Región de interés (ROI) alrededor del OD logrando una precisión de 81,3%. La investigación realizada por Triwijoyo, B. K [19]. Consiste en un sistema de detección precoz de la enfermedad para ello se procede a utilizar la imagen del fondo de ojo como una entrada de la red neuronal convolucional (CNN) para determinar si hay algún síntoma de HR o no, el conjunto de datos de imágenes utilizado es DRIVE y, se obtuvo una precisión de 98,6%. La investigación que propone Arasy, R. [25] es un sistema para la detección de la retinopatía hipertensiva mediante el análisis de componentes principales (PCA) y la red neuronal de retropropagación (BNN), donde las imágenes de la retina se tomaron de la base de datos STARE y mostraron una precisión del 86,36%.

Sin embargo, en todas las investigaciones ya mencionadas se concentran en identificar la relación Arterio-vena a través de redes neuronales, las cuales identifican si el paciente puede o no tener RH sin mostrar en que zona se encuentra el signo, y esto hace que el especialista deba verificar la veracidad del diagnóstico recorriendo toda la imagen, por esta razón es importante identificar el cuadrante (temporal superior e inferior, nasal superior e inferior) en donde se encuentran los signos para que el especialista se centre en la zona y sea más fácil corroborar el diagnóstico dado por el algoritmo.

En relación con los enfoques mencionados anteriormente se plantea la siguiente pregunta de investigación: ¿Cómo detectar signos de retinopatía hipertensiva mediante algoritmos de aprendizaje automático?

Para responder a esta pregunta de investigación se propone la siguiente hipótesis: La implementación de las técnicas y algoritmos basados en aprendizaje automático pueden permitir identificar las características y signos presentes en la retinopatía hipertensiva HR en los pacientes hipertensos. Que luego implementado en una herramienta computacional puede contribuir a disminuir el tiempo de análisis de un examen de fondo de ojo en un entorno clínico.

## **1.1 Justificación**

La detección de la retinopatía hipertensiva a través de las imágenes de fondo de ojo, es un procedimiento que demanda tiempo en el observador debido a que se realiza de manera manual, lo cual puede conducir a una mayor variabilidad intra e inter observador. En relación con el problema anteriormente mencionado, surge la necesidad de una herramienta que pueda asistir a al especialista, para reducir el tiempo invertido y brindar atención oportuna, eficiente y eficaz que permita el tratamiento oportuno a esta complicación [6-8,10].

El modelo propuesto se fundamenta en técnicas de clasificación de imágenes y aprendizaje automático requeridos para el apoyo de la toma de decisiones necesarias para la detección de signos de esta complicación ocular. Con esto se espera que beneficien tanto los médicos especialistas como los enfermos con HR [8-9].

Con esta herramienta computacional se pretende aumentar la eficiencia y reducir las barreras de acceso al diagnóstico y así contribuir al avance científico en este campo [8]. El desarrollo tecnológico está siendo cada vez más importantes en las diferentes áreas de las ciencias, logrando así que la optimización de estos sistemas beneficie al área de la salud y el desarrollo sostenible del país. Además se abre línea de investigación en patologías oculares en la Maestría Ingeniería Física de la Universidad del Cauca.

## **1.2 Objetivos**

### **1.2.1 Objetivo General**

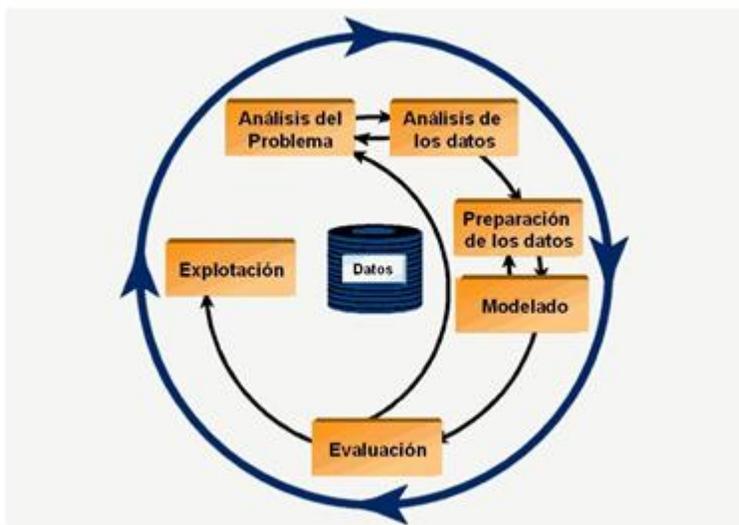
Detectar signos de retinopatía hipertensiva por medio de algoritmos de aprendizaje automático en imágenes digitales de fondo de ojo en pacientes hipertensos para estadificar esta complicación entre (leves, moderado y severo).

### **1.2.2 Objetivos Específicos**

- Construir un conjunto de datos con imágenes digitales del fondo de ojo de pacientes con diagnóstico de retinopatía hipertensiva, a través de bases de datos libres.
- Implementar técnicas y algoritmos de aprendizaje automático para la detección de signos de retinopatía hipertensiva en imágenes digitales del fondo de ojo.
- Evaluar el desempeño del modelo de detección automática de signos de retinopatía hipertensiva en imágenes digitales del fondo de ojo.

### 1.3 Descripción de la Metodología

La metodología adoptada para este trabajo es tipo CRISP-DM (Cross Industry Standard Process of Data Mining), que es el método más utilizado en proyectos de minería de datos y consta de seis etapas: comprensión del problema, comprensión de los datos, preparación de los datos, modelado, evaluación y despliegue. Estas etapas se siguen de manera exhaustiva, pero se realizan adaptaciones donde sea necesario.



**Fig. 1 Metodología CRIP-DM [30]**

El ciclo consta de seis etapas (Figura 1), en donde las fases no son de carácter rígidas y se presenta un constante movimiento de ida y vuelta donde el resultado de una fase determina que fase o tarea se tiene que realizar sucesivamente.

#### 1.3.1 Comprensión del problema

Consiste en identificar los requerimientos y problemáticas asociadas a la clasificación automática de las características encontradas en un examen de fondo de ojo de pacientes con retinopatía hipertensiva para convertirlos en objetivos técnicos y así poder elaborar un plan de trabajo.

#### 1.3.2 Comprensión de los datos

En esta fase se realiza la recopilación inicial de datos, donde se construye el conjunto de datos a partir de exámenes de fondo de ojo, adicionalmente se describen y exploran las características para poder identificar patrones propios para finalmente realizar las anotaciones y descripción de los datos.

### **1.3.3. Preparación de los datos**

Posteriormente a la recolección de los datos, se procede a adaptarlos para el proceso de aprendizaje automático, donde se establecen los criterios y la estandarización, con esto se espera la implementación de un proceso de extracción de características con el fin de obtener propiedades de las imágenes digitales de fondo de ojo y finalmente se realiza un análisis de selección de características.

### **1.3.4. Modelado**

En esta etapa se seleccionan las técnicas y algoritmos de clasificación, en donde se realiza la construcción y entrenamiento de los diferentes modelos. Adicionalmente se evalúan los algoritmos de detección en términos de su exactitud con el conjunto de datos de entrenamiento.

### **1.3.5. Evaluación**

Evaluar el modelo en relación a su desempeño teniendo como métricas la matriz de confusión y los indicadores que se derivan de ella: Especificidad, sensibilidad, área bajo la curva ROC, donde el diagnóstico del especialista es considerado como el referente final (Gold standar), para contrastar los datos aportados por el algoritmo.

## **1.4 Contribuciones**

Las contribuciones realizadas durante la realización de este trabajo son las siguientes:

- La principal contribución de este trabajo es detectar signos de retinopatía hipertensiva (Hemorragias, Micro-aneurismas, Exudados y Drusas) en imágenes de fondo de ojo, cabe resaltar que la arquitectura utilizada en esta investigación permite obtener un modelo de detección rápido y eficiente.
- La segunda contribución es la presentación y socialización de resultados en el evento científico titulado: Encuentro Internacional de Ciencia para la Paz y el Desarrollo, el cual se realizará entre los días 28, 29 y 30 de noviembre del presente año en el Aula Máxima de la Facultad de Ciencias Naturales, Exactas y de la Educación de la Universidad del Cauca. Participando en calidad de ponente. Título: “Detección De Signos De Retinopatía Hipertensiva Usando Algoritmos De Aprendizaje Automático En Imágenes Digitales De Fondo De Ojo”. Autores: Christian Camilo Delgado Molina, Jairo Alfonso Vásquez, Leonardo Africany Villamil y Rubiel Vargas.
- Sometimiento de un artículo de investigación en un volumen especial de la revista Journal of Physics: Conference Series (SJR index: 0,21), homologado en categoría B en

Publindex Título: “Detección De Signos De Retinopatía Hipertensiva Usando Algoritmos De Aprendizaje Automático En Imágenes Digitales De Fondo De Ojo”. Autores: Christian Camilo Delgado Molina, Jairo Alfonso Vásquez, Leonardo Africany Villamil y Rubiel Vargas.

### 1.5 Estructura del documento

A continuación, se muestran los capítulos en los cuales se ha dividido el resto del documento, así como un breve resumen del contenido de cada uno de ellos.

**Capítulo 2. Retinopatía Hipertensiva (HR) y detección de signos.** Este capítulo presenta y describe las generalidades y conceptos que giran en torno al contexto de investigación: Hipertensión, retinopatía hipertensiva, hemorragias, exudados, drusas, el aprendizaje automático y el estado de la técnica.

**Capítulo 3. Metodología.** Este capítulo introduce la descripción matemática, técnicas, herramientas y recursos utilizados durante la realización de esta investigación.

**Capítulo 4. Resultados y discusión.** Este capítulo presenta y discute los principales resultados obtenidos de la investigación realizada. Donde se exponen y discuten los resultados de la aplicación de las técnicas para la identificación automática de signos que conllevan a una retinopatía hipertensiva.

**Capítulo 5. Conclusiones y trabajo futuro.** Este capítulo presenta las contribuciones realizadas por esta investigación, las conclusiones obtenidas tras su realización, las cuales permiten verificar la hipótesis de investigación planteada, y una serie de líneas futuras que permiten continuar la investigación.

## CAPITULO 2

### 2. RETINOPATIA HIPERTENSIVA Y SUS SIGNOS

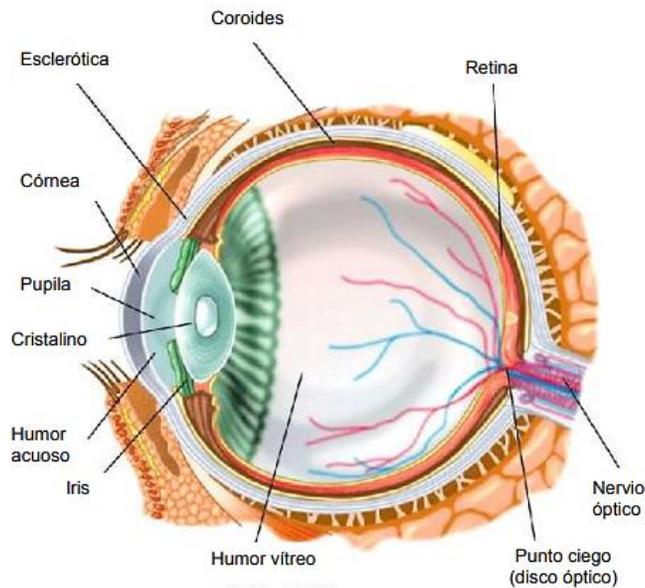
La retinopatía hipertensiva es el daño de los vasos retinianos causado por hipertensión. Habitualmente los signos se desarrollan cuando la enfermedad está avanzada, ahora bien, la oftalmoscopia permite ver contracción arteriolar, muescas arteriovenosas, cambios en la pared vascular, hemorragias en llama, focos blandos algodinosos, exudados duros amarillentos. Cuanto más alta sea la presión arterial y mayor sea el tiempo que esta haya estado elevada, más probable será que el daño sea grave, teniendo una pérdida de la visión. [17]

En este capítulo se introducen los principales conceptos relacionados con la retinopatía hipertensiva y la detección automática de los signos que presenta esta patología, además de los conceptos técnicos y algoritmos de aprendizaje automático.

## 2.1 Marco Conceptual

### 2.1.1 Anatomía del Ojo

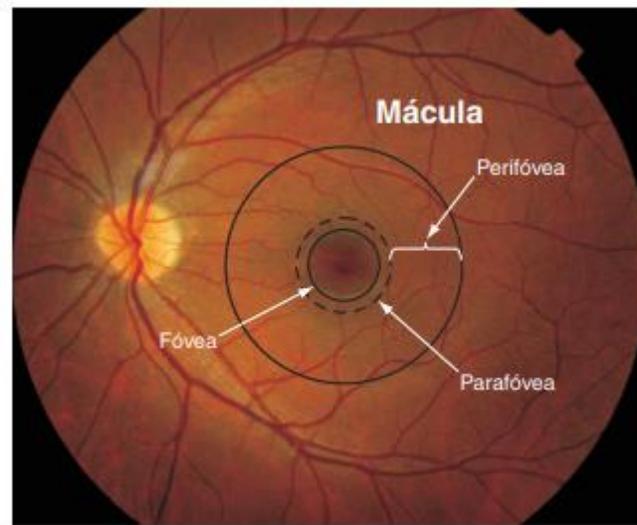
La visión es importante para las persona pues es uno de los sentidos más evolucionados, ya que nos permite captar e interpretar información visual sobre luz, color, forma, distancia, posición o movimiento. Su órgano receptor es el globo ocular (ojo) y se encuentra alojado en las cavidades orbitarias. [11] Los componentes más importantes de este órgano se pueden ver en la figura 2.



**Fig. 2. Anatomía del ojo [11]**

### 2.1.2 Mácula

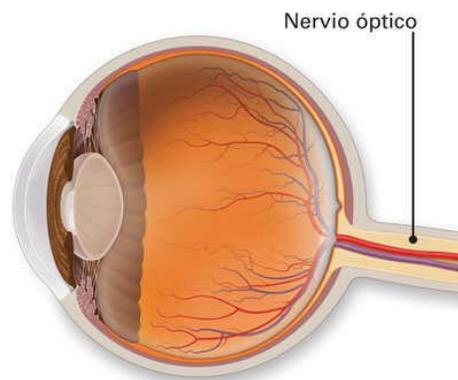
Se encuentra dentro de la retina y permite que el ojo alcance a desarrollar la visión en los detalles más pequeños, es la encargada de la visión central y permite actividades que requieren mucha precisión como leer o enhebrar una aguja. Su centro se denomina fovea [13]. En la figura 2 se aprecia una imagen de la mácula.



**Fig. 3. Mácula- Anatomía del ojo [15]**

### 2.1.3 Nervio óptico

El nervio óptico se encarga de enviar señales e información adquiridas por el ojo al cerebro que son procesadas por la corteza visual, el hipotálamo y el lóbulo occipital. En la figura 4 se observa imagen del nervio óptico. [16]

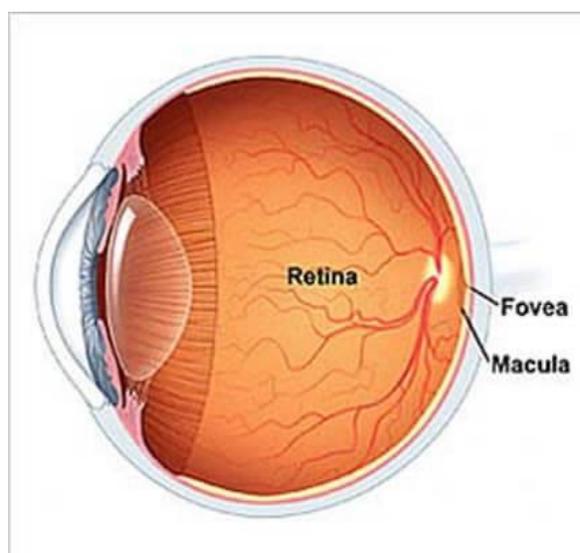


**Fig. 4. Nervio Óptico- Anatomía del ojo [16]**

### 2.1.4 Retina

Es la capa más interna y principal en el órgano del ojo, debido a que convierte los rayos de luz en impulsos nerviosos. “Para ello cuenta con foto-receptores: un tipo especial de células para detectar la luz” [13]. Es la estructura más sensible a la luz ubicada en la superficie interior del globo ocular, similar a una pantalla donde se proyectan las imágenes, además de células nerviosas contiene los vasos sanguíneos y tejidos conectivos del ojo. Está formada por células sensibles llamadas bastones y conos, quienes transforman la información en impulsos eléctricos y la transportan por intermedio del nervio óptico al cerebro en forma cruzada, es decir, el ojo izquierdo lleva la información al lado derecho del cerebro y así mismo el ojo derecho va hacia el lado izquierdo del cerebro. Los bastones ayudan a percibir la luz de baja intensidad y los conos son los responsables de la percepción de los colores y las formas que se observan en los objetos. [11]

La retina es una estructura ocular compleja que está formada por muchas capas histológicas, que pueden afectarse por diferentes patologías, provocando daños en la visión de forma parcial o total. Las enfermedades que más afectan a la retina son las de RD y RHTA. Estas complicaciones pueden ser prevenidas con exámenes rutinarios de fondo de ojo cada 6 meses como mínimo. En la figura 4 se aprecia una imagen de la retina. [12]



**Fig. 5. Retina- Anatomía del ojo [11]**

### **2.1.5. Hipertensión Arterial**

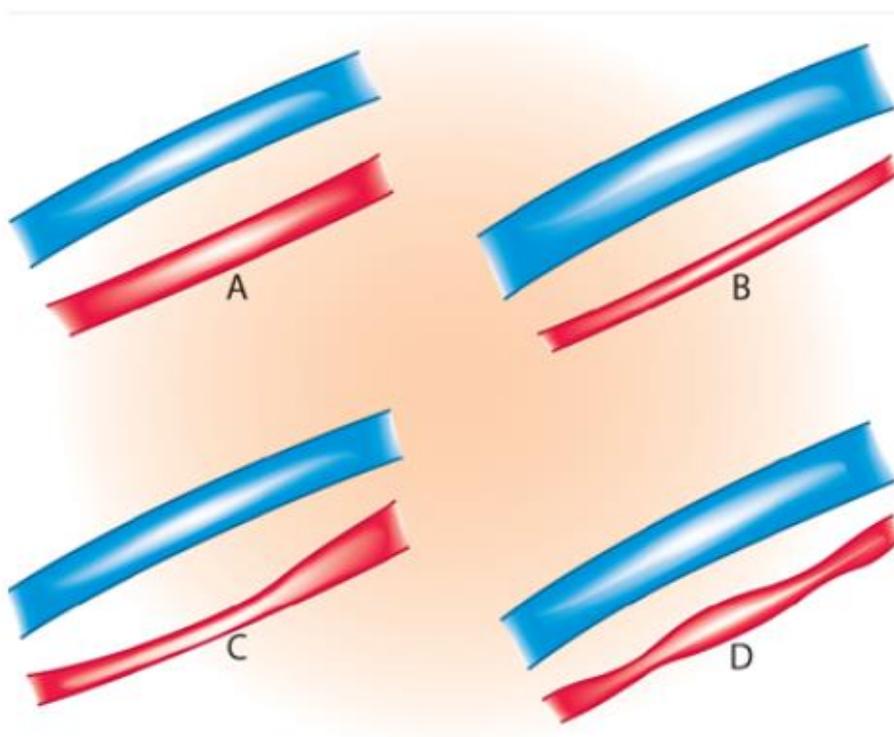
La hipertensión arterial es una patología crónica caracterizada por un aumento de la presión en el interior de los vasos sanguíneos (arterias), que trae como consecuencia daños progresivos en los vasos sanguíneos, desarrollándose complicaciones cardiovasculares (Ictus, infarto de miocardio, e insuficiencia cardíaca), el daño renal, y complicaciones oculares. Comúnmente, afecta al 35% de los adultos en nuestro país y al 68% de los mayores de 65 años. La hipertensión arterial (HTA) se conoce como la asesina silenciosa ya que en etapas iniciales el paciente no presenta ninguna sintomatología lo cual puede retrasar el tratamiento. Por lo general el paciente consulta cuando ya tiene sintomatología y la hipertensión arterial está muy avanzada y ha producido daños en el organismo. [21]

Cuando la HTA está demasiado alta produce cefalea (dolor de cabeza), náuseas, confusión, sangrados, derrames e incluso la muerte. Además de los antecedentes familiares y la diabetes mellitus los factores de riesgo asociados son: obesidad, estrés, ansiedad, consumo de alcohol, tabaquismo y el consumo exagerado de sal de cocina en sus comidas. Durante el examen médico de un paciente hipertenso, no solo es necesaria la toma de la presión arterial sino también es necesario realizar un examen de fondo de ojo que permite observar con más detenimiento el daño sufrido por los vasos sanguíneos retinianos. [21]

Pero, ¿Cómo se puede ver la HTA a través de estudios de fondo de ojo? Su respuesta se basa en que la HTA puede inducir hipertensión ocular y a la vez generar la retinopatía hipertensiva. Mediante los estudios de fondo de ojo se puede observar de forma más detenida y minuciosa los daños que puede ocasionar esta patología a nivel de la retina.

### **2.1.6 Retinopatía Hipertensiva**

Es una complicación en la retina consecuencia de la aparición y el control inadecuado de la hipertensión arterial. En el ojo humano y especialmente en la retina existen multitud de arterias y venas que recorren su superficie. Cuando la presión arterial aumenta, los vasos sanguíneos y los tejidos que los rodean sufren cambios: aumento de la permeabilidad vascular, que provoca el escape de líquido y sustancias del plasma hacia la retina. Además algunos vasos sufren una importante contracción, es decir relación arteria-vena como se puede ver en la (figura 6 y tabla 1), disminuyéndose el flujo sanguíneo hacia algunas regiones de la retina, que presentaran daños por disminución o falta de aporte de oxígeno y nutrientes. En ocasiones pueden observarse hemorragias debidas a este daño vascular y en casos más graves ocasionando edema macular y edema de papila.[21]



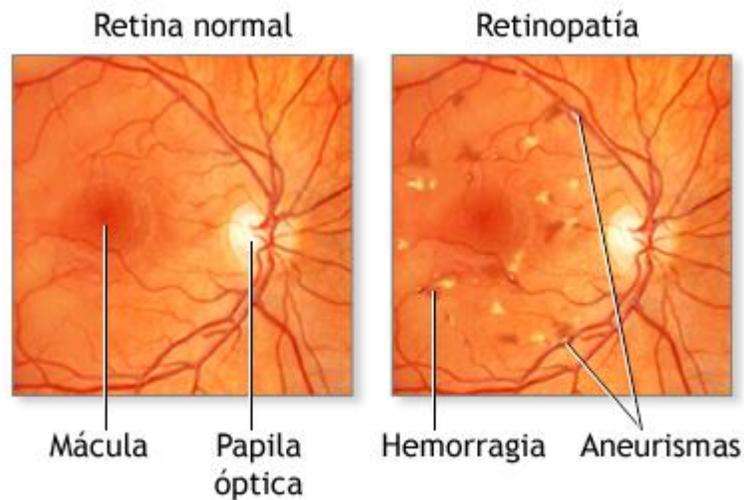
**Fig. 6** Relación arteria-vena. A. Relación A/V normal, 3/4. B. Relación A/V anormal, 1/4, angiotonía acentuada. C. Angioespasmo localizado en una relación A/V por otro lado normal. D. Angioespasmos localizados en una relación A/V de 2/4. [17]

**Tabla 1. Relación arteriovenosa (A/V) [10]**

Clasificación de la relación arteriovenosa (A/V)		
Grado	Descripción	Relación A/V
Normal	-	3/4 a 2/3
I. Ligero	+	2/3 - 1/2
II. Moderado	++	1/2 - 1/3
III. Intenso	+++	< 1/3
IV. Muy grave	++++	Arterias Filiformes

### 2.1.7 Fundoscopia (Examen fondo de ojo)

El examen de fondo de ojo (también llamado fundoscopia) consiste en la observación detallada de la retina y sus vasos por medio de un oftalmoscopio (figura 8). Es una prueba sencilla e indolora que se realiza en pocos minutos, la retina puede observarse alterada no solo por RHTA sino también por RD, arteriosclerosis o la edad avanzada. La RHTA se puede presentar como hemorragias en flama, depósitos anormales, conocidos como exudados, drusas o bien como edema macular como se muestra en la figura 7. [18]



**Fig. 7. Retina normal y con retinopatía [18]**



**Fig. 8 Oftalmoscopio [18]**

### 2.1.8 Retinopatía Hipertensiva Aguda

HRTA Maligna o transitorias, aparece en caso de aumento repentino de la presión arterial. Se caracteriza por tener síntomas como la visión borrosa o ceguera. Esta sintomatología puede revertirse al controlarse la enfermedad de base. [18]

### 2.1.8 Retinopatía Hipertensiva Crónica

No revela síntomas durante mucho tiempo y suele diagnosticarse en los exámenes de fondo de ojo. Este tipo de patología aumenta el riesgo de oclusiones vasculares, pudiendo afectar la visión. Puede aparecer visión borrosa o disminución de la agudeza visual y en algunos casos pérdida significativa de la visión, sobre todo en casos asociados a diabetes o arteriosclerosis. Se puede acompañar de cefalea (dolor de cabeza), y en el fondo de ojo se aprecia una disminución de los vasos, en los casos más graves también se pueden apreciar hemorragias, exudados y edema de papila. [18]

### 2.1.9 Signos y síntomas de la retinopatía hipertensiva

Habitualmente, no se desarrollan síntomas hasta que la enfermedad está avanzada e incluyen visión borrosa o defectos del campo visual. En las fases iniciales, la oftalmoscopia identifica contracción arteriolar, con una disminución en el cociente del calibre arteriolar/venular.

La hipertensión crónica mal controlada produce:

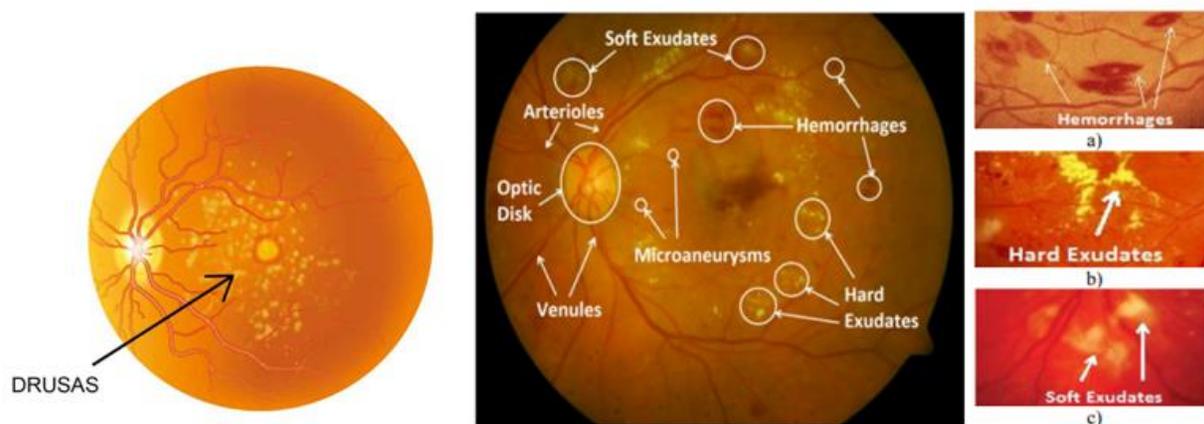
- Estrechamiento arterial permanente
- Anomalías en los cruces arteriovenosos (muescas arteriovenosas)
- Arterioesclerosis con cambios moderados de la pared vascular ("en hilo de cobre") hasta hiperplasia y engrosamiento vascular más grave ("en hilo de plata")
- A veces, ocurre una oclusión vascular total. Las muescas arteriovenosas representan un factor predisponente de importancia para el desarrollo de la oclusión de una rama de la arteria retiniana.

Si la enfermedad aguda es grave, pueden desarrollarse (ver figura 9):

- **Microaneurismas:** son deformaciones en las paredes de los vasos sanguíneos. Son difíciles de detectar con la oftalmoscopia convencional si no se tiene experiencia, pero aparecen como pequeñas manchas rojas con límites nítidos. [19]
- **Hemorragias:** son producto del colapso de los microaneurismas y son visualmente apreciables como manchas de sangre sobre la superficie de la retina, de tamaño muy superior a las de los microaneurismas. [20]
- **Exudados duros:** son objetos de color amarillo brillante o blanco en la retina. Estos objetos tienen una apariencia cerosa y bordes afilados contra el fondo de los vasos

sanguíneos. Los exudados duros se desarrollan debido a la pérdida de sangre de las venas y los exudados tienen forma circular alrededor de los vasos. [21]

- **Exudados Blandos:** también llamados manchas de algodón, son infartos de la capa de fibras nerviosas; tienen un aspecto más blanquecino. [22]
- **Drusas:** Las drusas son depósitos amarillos debajo de la retina. Están formadas por lípidos y proteínas. Es probable que las drusas no causen degeneración macular relacionada con la edad (DMRE), pero tener drusas aumenta el riesgo de una persona de desarrollar DMRE, y pueden ser una señal de tener DMRE [22].



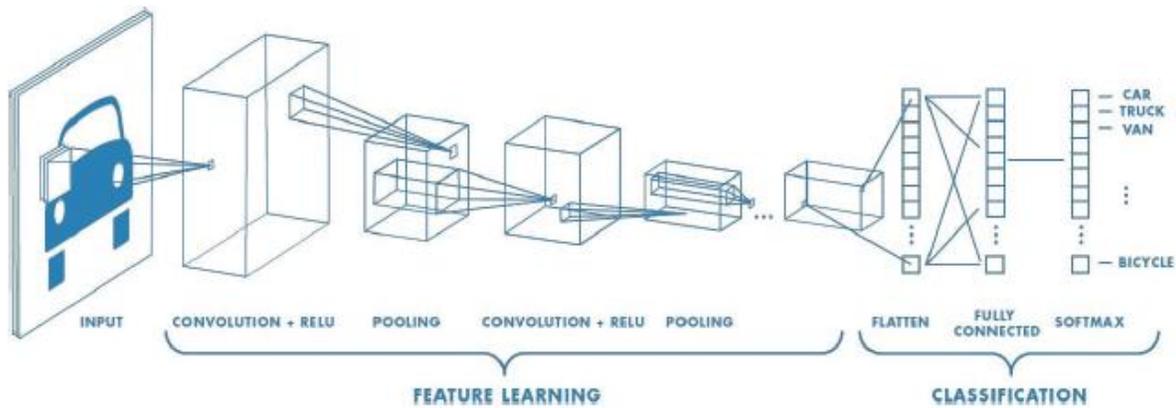
**Fig 9. (a) Imagen de la retina con sus patologías. (b) Hemorragias. (c) Exudados duros. (d) Exudados Blandos. [23]**

### 2.2.0 Modelos De aprendizaje automático (Deep Learning)

Las neuronas artificiales se basan en las que se tienen en el organismo, es decir, un conjunto de neuronas conectadas entre sí pero sin ninguna tarea asignada a cada una. Con entrenamiento las neuronas van realizando conexiones para aprender a realizar una tarea. En este proyecto se quiere detectar signos que evidencien Retinopatía hipertensiva (HR), a través de aprendizaje automático. A continuación se describen aspectos generales de las arquitecturas CNN, YOLO y sus ventajas.

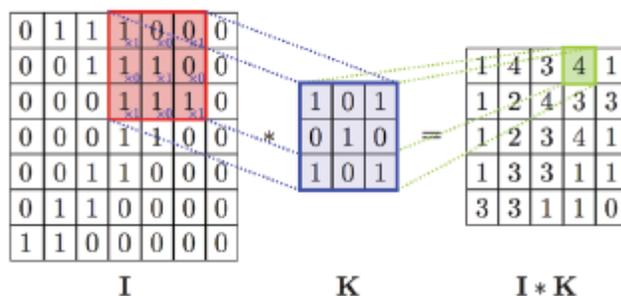
### 2.2.1 Redes Neuronales Convolucionales (CNN)

Las redes convolucionales tienen gran importancia en el campo del aprendizaje automático, ya que estas fueron los primeros modelos utilizados que obtuvieron excelentes resultados en el campo de reconocimiento de imágenes, como por ejemplo en los años 90 en el grupo AT&T para reconocimiento de cheques, en los años 2000 en el grupo Microsoft para reconocimiento de dígitos manuscritos, etc. estas redes pretenden extraer en cada capa características de la imagen para finalmente clasificarlas. De esta forma la información que se pasa a la capa siguiente es menor que la anterior.



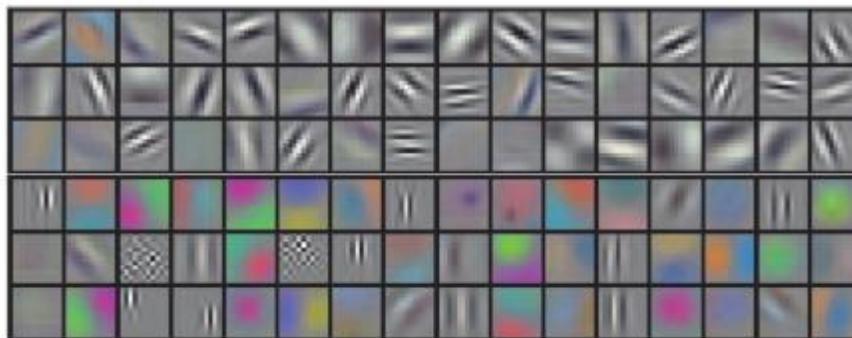
**Fig 10. Diagrama de una red convolucional [24]**

En la figura 10 podemos observar la arquitectura más común de estas redes. Como se puede observar, se realizan dos etapas, el aprendizaje de características y su clasificación. En la etapa de extracción se extraen los bordes, texturas, formas y diversos patrones. Esta extracción se lleva a cabo mediante convoluciones, una serie de operaciones matriciales entre la imagen y una matriz mucho más pequeña conocida como kernel. La operación convolución es realizada desplazando el kernel por toda la imagen como se puede ver en la figura 11.



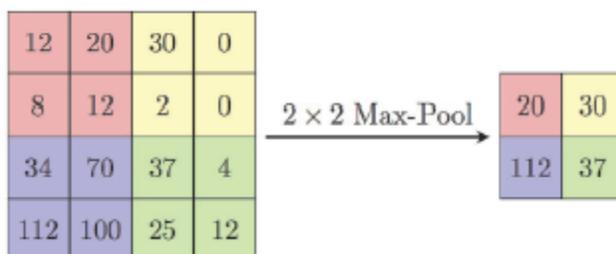
**Fig 11 Ejemplo de convolución [25]**

Con esta operación solo se pueden detectar bordes y texturas, pero lo interesante es que extraiga patrones y formas. La manera en que extrae estos patrones es mediante la composición de varias operaciones de convolución. La tarea de una CNN por lo tanto será de determinar los Kernels para que aprenda la red a extraer esas características, es decir, las primeras capas de la red extraerán bordes y texturas (Figura 12) pero a medida que se aplican operaciones convolucionales sobre la primera capa se consigue extraer características y patrones más complejos que es el objetivo de la red. Convencionalmente, cada una de las imágenes que se generan al aplicar la convolución dentro de la red se llama mapa de características.



**Fig. 12 Ejemplo de 96 filtros extraídos en la primera capa de una CNN entrenando con una dataset de imageNet [25]**

Al aplicar una convolución reducimos la dimensionalidad de los datos. Para reducir la dimensionalidad, aún, más y que la red aprenda mejor, normalmente se realiza un pooling. El pooling, lo que pretende es reducir el tamaño de las imágenes convolucionadas manteniendo las características detectadas. Normalmente la técnica que se utiliza es el Max Pooling ilustrada en la figura 13. En el ejemplo se ilustra como la imagen original de tamaño 4x4 se crea una división de sectores de 2x2 debido a que se le quiere aplicar un max pooling de 2x2. A continuación en cada sector se escoge el pixel de valor máximo para ser parte de la nueva imagen reducida.



**Fig 13 Ejemplo de max pooling [25]**

### 2.2.2 Yolo (You Only Look Once)

Se trata de una arquitectura de detección de objetos muy rápida y precisa, que fue creada por Joseph Redmond como principal autor [26]. El hecho de que sea una arquitectura muy rápida, la hace idónea para que se use en la detección en video en tiempo real. YOLO consiste en una red neuronal convolucional que predice simultáneamente múltiples cuadros delimitadores y las probabilidades de la clase de objeto que delimitan dichos cuadros delimitadores.

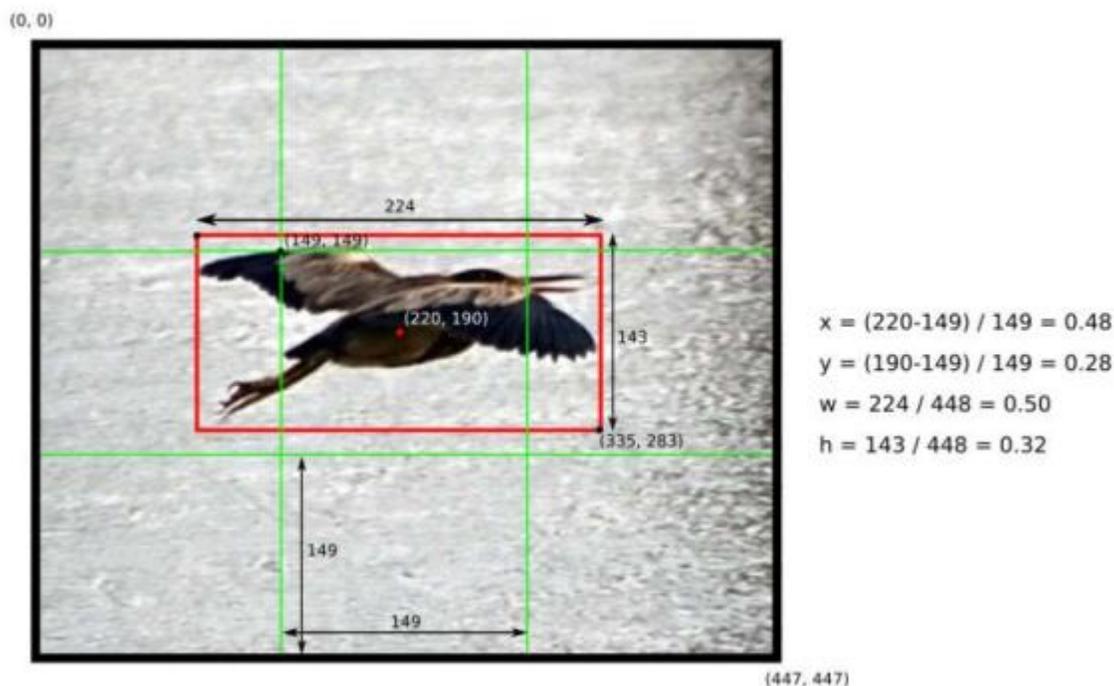
### 2.2.3 Los beneficios que presenta YOLO son [26]:

- Es un sistema muy rápido, debido a que se reduce la detección a un problema de regresión lineal y esto conlleva a que no se requiera un pipeline complejo.
- Simplemente se ejecuta la red neuronal en una imagen en la prueba para hacer predicciones.
- Para hacer las predicciones, a diferencia de los métodos anteriores, este sistema consume la imagen completa, en lugar de regiones de la misma. Esto hace que se limiten los errores a la hora de reconocer las clases de objetos que hay en la imagen.
- Aprende representaciones generalizables de objetos, lo que hace que, si se introducen datos de entrada nuevos, tenga menos probabilidades de fallo que las técnicas descritas anteriormente.

El funcionamiento se basa en una red neuronal convolucional, como se comentó previamente, en la que usa características de la imagen al completo para trazar los cuadros delimitadores (Bounding Box). Con ello se consigue que la red neuronal reaccione a todos los objetos que hay en una imagen. La imagen se divide en una rejilla de tamaño  $S \times S$  de forma que, si un objeto cae en una celda de la cuadrícula, esa celda es responsable de detectar ese objeto, como podemos ver en la Figura 14, que en este caso es la casilla central. Por otra parte, cada celda de la cuadrícula predice  $B$  cuadros delimitadores con sus puntuaciones confianza (score). Estas puntuaciones de confianza reflejan cómo de fiable es el modelo si esa celda contiene un objeto y también cómo de preciso es al predecirlo [26]. Formalmente definimos la confianza como  $Pr = (\text{Objeto}) * IOU$ . Si no hay un objeto en esa celda, las puntuaciones de confianza deben ser cero. De lo contrario,

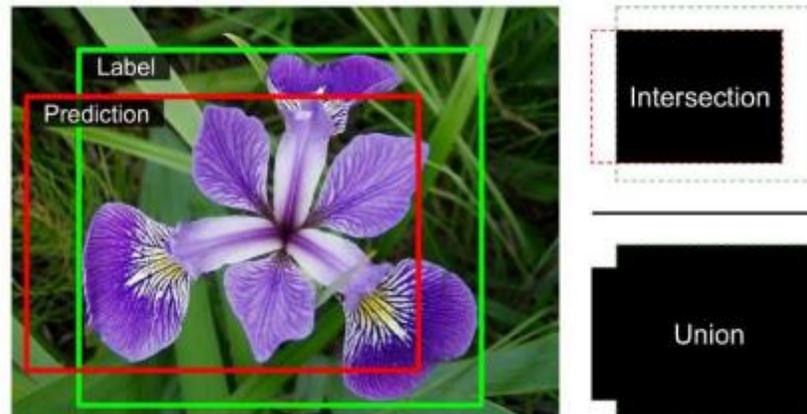
si hay un objeto en dicha celda, queremos que la puntuación de confianza sea igual a la intersección sobre unión (IOU). Cada cuadro delimitador consta de 5 predicciones: x, y, w, h, y la confianza. Las coordenadas (x; y) representan el centro del cuadro en relación con los límites de la celda de la cuadrícula. Como podemos ver en la Figura 13, esto se ve en el punto central marcado de color rojo con coordenadas (220,190). Como tenemos que calcularlo en referencia a la celda vemos que, si las celdas tienen un tamaño de 149×149, lo que hacemos es calcular la normalización respecto a la celda, con lo que el valor queda entre 0 y 1 como podemos observar en los cálculos. El caso de la coordenada y, es exactamente el mismo principio [26].

La anchura y la altura se toman en relación con la imagen completa, es decir, que en este caso la normalización no se toma en relación a la celda sino al tamaño de la imagen. Por tanto, lo que hacemos es tomar el ancho de la detección del objeto (cuadro de color rojo en la Figura 14), y la altura, y la normalizamos respecto al tamaño de la imagen que en este caso es 448×448.



**Fig 14: Ejemplo de cómo se calculan las coordenadas del cuadro de una imagen. El tamaño es 448x448 píxeles y S=3. [27]**

Finalmente, para obtener la puntuación de confianza de la detección, necesitamos varios parámetros. En primer lugar, se necesita la intersección sobre la unión IOU del cuadro delimitador y la anotación realizada sobre la imagen (Groundtruth), que podemos ver en la Figura 15. Como podemos ver, calculamos la IOU (intersección sobre la unión) como el cociente de la intersección entre la unión. Esto nos dará como resultado, la intersección.



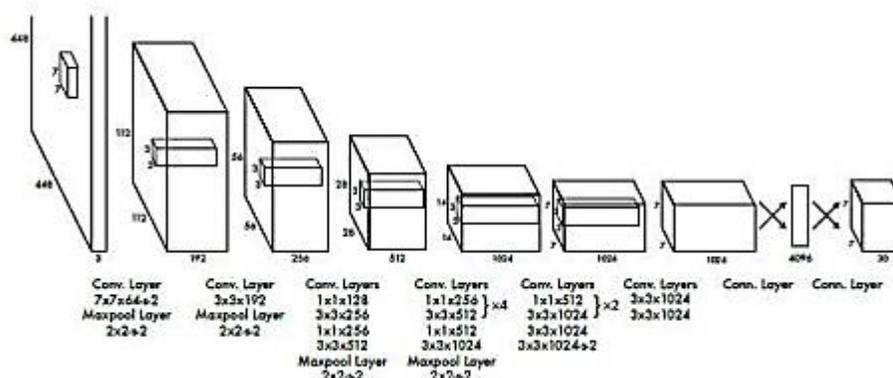
**Fig. 15: Descripción gráfica de la intersección sobre la unión. [27]**

Posteriormente, se obtiene la puntuación de confianza, teniendo en cuenta una serie de probabilidades. Cada celda de la cuadrícula también predice  $C$  probabilidades de clases condicionales,  $Pr(\text{Class}|\text{Object})$ . Estas probabilidades están condicionadas en la celda de la cuadrícula que contiene un objeto. Además, predice un solo objeto independientemente del número de cuadros delimitadores  $B$ . En el momento de la detección, se multiplica la probabilidad condicional de la clase por la probabilidad de un cuadro individual, lo que da las puntuaciones de confianza específicas de la clase para cada cuadro delimitador. Estas puntuaciones de confianza nos dan la probabilidad de que el objeto este en dicho cuadro y cómo de bien se ajusta el cuadro predicho al objeto [27].

#### 2.2.4 Arquitectura de red

El modelo se implementa en una red neuronal convolucional, ya que solo contiene capas convolucionales, a menudo se le suele denominar FCN o fully convolutional network. En las capas iniciales se extraen las características de la imagen y las capas completamente conectadas predicen las probabilidades de salida y las coordenadas. Esta red está inspirada en el modelo GoogleNet usado en la clasificación de imágenes.

Concretamente esta red se compone de 24 capas convolucionales y 2 capas completamente conectadas. Para reducir el número de capas los creadores usan una convolución  $1 \times 1$ , lo que permite reducir la profundidad de los mapas de características. A esto le sigue una capa de convolución  $3 \times 3$ . Estas capas  $1 \times 1$  y  $3 \times 3$  se van alternando como podemos apreciar en la Figura 16. Además, la última capa de convolución, genera un tensor con forma  $(7, 7, 1024)$ . Finalmente, el tensor se reduce, aplicando 2 capas completamente conectadas, y da como salida un tensor de tamaño  $7 \times 7 \times 30$ , como podemos apreciar en la Figura 16 [26] [28].



**Fig. 16** Esquema de la arquitectura de la red neuronal [26]

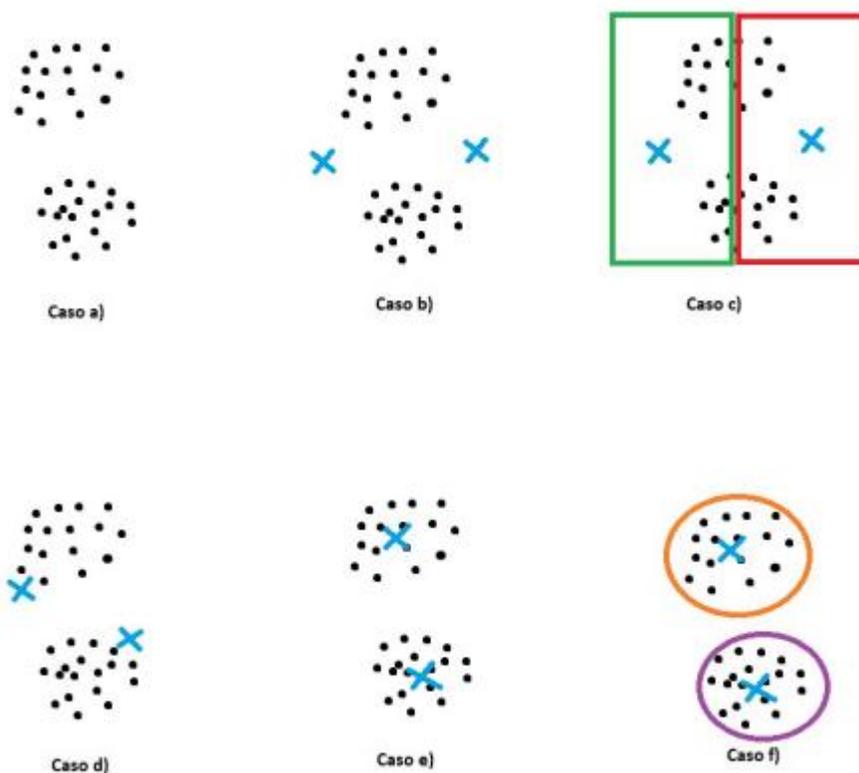
Hoy en día, se ha tenido una evolución de YOLO desde el 2015, permitiendo así tener cada vez una mejor precisión en la detección de los objetos, entre ellos tenemos en la actualidad 7 versiones desde YOLOv1 hasta YOLOv7. Concretamente se han obtenido mejores resultados con YOLOv3 y YOLOv4 de los cuales hablaremos los aspectos más generales.

### 2.2.5 YOLOv2

Esta versión, tiene como objetivo mejorar la precisión de manera notable y hacer una detección aún más rápida. En cuanto a las mejoras de precisión, se añade la normalización por lotes en las capas de convolución [26] [28]. La normalización por lotes o Batch Normalization se trata de una técnica que consiste en añadir operaciones en el modelo, antes o después de la función de activación de cada capa oculta. A continuación, se normaliza cada entrada, y se realiza un centrado en cero. Finalmente se escala y se cambia el resultado usando dos nuevos vectores de parámetros por capa. Esta operación permite que el modelo aprenda la escala y la media óptimas de cada una de las entradas de la capa [28]. Otro aspecto para destacar es que mejora el clasificador de forma que en la primera versión se entrenaba con imágenes 224x224 y luego se realizaba la detección con imágenes de tamaño 448x448. En esta segunda versión el clasificador también se entrena con imágenes de este último tamaño, mejorando la precisión media (mAP).

Otro de los aspectos que ha mejorado es el hecho de que ahora se usan cajas de anclaje, lo que permite que el sistema pueda detectar más de un objeto por celda, cuando hasta ahora era solamente un objeto por celda. Esto antes era un problema puesto que, con la idea anterior, si en una celda había dos o más objetos, no los podía detectar todos. Con la idea de los cuadros de anclaje, YOLO puede detectar un gran número de cuadros delimitadores. El cuadro de anclaje no es más que ancho y un alto determinados, con el que se puede predecir el cuadro delimitador de la detección. Esto consigue la detección en relación con el mencionado cuadro de anclaje, en lugar de predecir el cuadro delimitador respecto a la imagen completa. Las formas de las cajas de anclaje no son seleccionadas a mano, sino que las selecciona YOLO de forma que facilite a la red el que aprenda a detectar objetos. El tamaño lo selecciona YOLO mediante el algoritmo de

aprendizaje no supervisado K-Means [28]. Se trata de un algoritmo que ayuda a organizar los datos según las características que tiene. Lo que hacemos es alimentar el sistema con todos los datos que tenemos y vamos a decir en cuántos grupos o “clusters” queremos que nos los organice. Para poder realizar una explicación más intuitiva, vamos a suponer que queremos realizar una organización de los datos en 2 grupos o “clusters”. Para ello en la Figura 17, en el caso a), podemos observar en primer lugar la cantidad de datos que tenemos.



**Fig. 17: Ilustración del funcionamiento del algoritmo K-Means [28]**

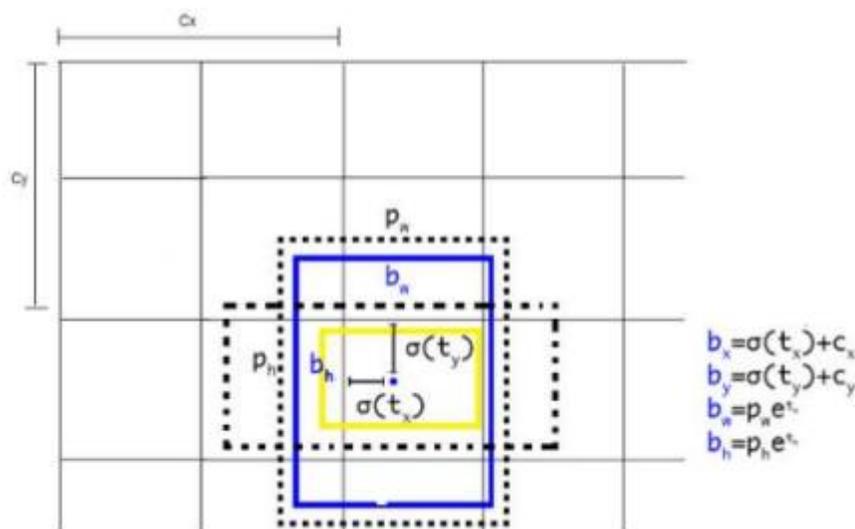
Para poder realizar la organización en dos grupos, el algoritmo crea dos puntos aleatorios llamados centroides, como podemos observar en la Figura 17 en el caso b), donde los encontramos marcados en azul. A continuación, lo que hará el algoritmo es sacar el promedio de los puntos más cercanos a cada centroide. Supongamos que en el caso c) se marcan que los puntos más cercanos a cada uno de los centroides son los que aparecen marcados por los cuadros verde y rojo, tanto para el centroide 1 y 2 respectivamente. Una vez el algoritmo saca el promedio de los puntos más cercanos a cada centroide, el centroide se moverá a dicha posición como podemos ver en el caso d). Esto se repetirá durante varias iteraciones hasta que se llegue a la situación del caso e), donde el algoritmo decidirá que los puntos más cercanos al primer centroide es el que pertenece al grupo número 1 (marcado en naranja), mientras que los puntos más cercanos al segundo centroide serán los que pertenezcan al grupo número 2 (marcado en morado), como podemos ver en el caso f). Esto es lo que se realiza con las cajas de anclaje, solo

que en este caso es ejecutar el algoritmo que acabamos de comentar ahora varios valores de grupos ( $k$ ), para generar el IOU promedio con el centroide más cercano.

Las coordenadas de la ubicación en esta versión de YOLO, se predice teniendo en cuenta las celdas de la cuadrícula. Esto implica que en los GroundTruth se tengan valores de entre 0 y 1. La red genera hasta 5 cuadros delimitadores para cada celda y predice 5 coordenadas para cada cuadro delimitador:  $t_x$ ,  $t_y$ ,  $t_w$ ,  $t_h$  y  $t_o$ . Si la celda está desplazada desde la esquina superior izquierda de la imagen por  $c_x$  y  $c_y$  y el cuadro de anclaje tiene un ancho y un alto dado por  $p_w$  y  $p_h$  respectivamente, entonces tenemos que las predicciones serán las que podemos ver en la Figura 18. Intuitivamente lo podemos ver con más claridad en la Figura 19

$$\begin{aligned} b_x &= \sigma(t_x) + c_x \\ b_y &= \sigma(t_y) + c_y \\ b_w &= p_w e^{t_w} \\ b_h &= p_h e^{t_h} \\ Pr(\text{object}) * IOU(b, \text{object}) &= \sigma(t_o) \end{aligned}$$

**Fig. 18: Predicciones de los cuadros delimitadores**



**Fig. 19: Explicación intuitiva de la posición de un cuadro delimitador**

En cuanto a la arquitectura de red, YOLOv2 es un sistema más preciso y más complejo debido a que se propone un modelo de clasificación nuevo, denominado Darknet19. Esta red consta de 19 capas convolucionales y 5 capas de máxima agrupación. El resultado es que alcanza una precisión muy notable, mejorando a la de YOLOv1.

### 2.2.6 YOLOv3

Es la versión de YOLO que hemos utilizado en el proyecto. En ella enumeraremos las evoluciones que ha manifestado respecto a anteriores versiones para poder apreciar cómo es de diferente el sistema en sí con el paso de las diferentes mejoras que se van aplicando. YOLOv3 [29] nace con el propósito de aumentar en precisión de detección de objetos. Esto hace que se gane en robustez, y esto se debe principalmente a la arquitectura de red usada, que es Darknet-53.

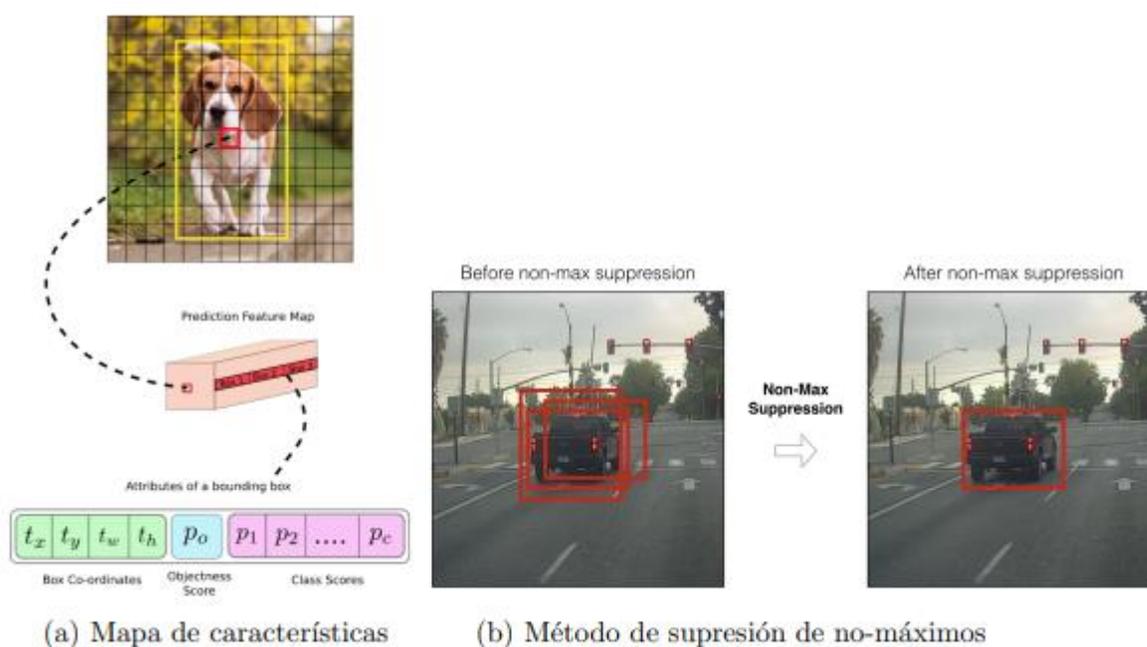
Esta red utilizada para la extracción de características es una red híbrida entre lo que ya se tenía de YOLOv2 y la red residual. El bloque residual, permite que tenga conexiones atajo ya que, en las redes neuronales tradicionales, cada capa se conecta a la siguiente dentro de la arquitectura de capas. Con los bloques residuales, se consigue que cada capa se conecte a la capa inmediatamente siguiente y a capas que están a unos saltos de distancia, lo que hace que este tipo de conexiones, se denominen conexiones atajo. La implicación de estas conexiones atajo la veremos un poco más adelante. Esta nueva red está compuesta por 53 capas completamente convolucionales, de ahí el nombre de Darknet-53. Como resultado se obtiene una red mucho más potente que Darknet-19 de YOLOv2, y el rendimiento es similar a otros clasificadores, pero con menos operaciones de coma flotante y mayor velocidad, ya que hace también un uso eficiente de la GPU, por lo que influye en el rendimiento de manera importante [29]

En esta versión, también se predice una puntuación de confianza para cada cuadro delimitador, que es uno si el cuadro delimitador se superpone a un objeto, es decir, que coincide con el groundtruth, más que cualquier otro cuadro delimitador. Si el cuadro delimitador no es el mejor, esto es que no tiene el IOU más alto, pero se superpone con otro groundtruth por encima de un cierto umbral, se ignora la predicción. Este sistema asigna solamente un cuadro delimitador por groundtruth. Si un cuadro delimitador no se asigna a ningún groundtruth, no incurre en ninguna pérdida de coordenadas o predicciones de clase [29].

Otro aspecto que mejora a sus antecesores es el hecho de la predicción de etiquetas múltiples. En algunos conjuntos de datos existen objetos que se pueden etiquetar de distinta forma, por ejemplo, un coche se puede etiquetar como tal pero también como “Berlina”. Esto hace que al usar Softmax para predecir, hace suponer que cada cuadro tiene una clase únicamente, y como hemos visto no es el caso, pues un mismo objeto puede tener varias etiquetas. Para poder utilizar el sistema con múltiples etiquetas, YOLOv3 no usa Softmax, sino que usa clasificadores logísticos independientes para cualquier clase. Con este tipo de clasificadores un coche se puede detectar como “Coche” y como “Berlina” al mismo tiempo. Para las predicciones de clase usa la pérdida de entropía cruzada binaria [29]. YOLOv3, también predice cuadros delimitadores en varias escalas distintas. En las versiones 1 y 2 de YOLO, la salida se predice en la última capa,

pero YOLOv3, predice cuadros delimitadores en 3 escalas diferentes. Para cada escala se usan 3 cuadros de anclaje y predice 3 cuadros para la celda de la cuadrícula en cuestión [29].

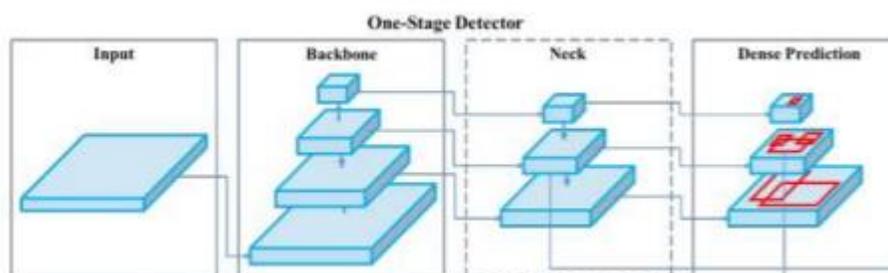
Otra característica importante es el uso de conexiones atajo. Estas permiten obtener un mejor rendimiento en cuanto a la detección de objetos pequeños, puesto que consigue una información más detallada del mapa de características. Un mapa de características es el resultado de aplicar un filtro a la capa anterior. Con ello se consigue que al pasar el filtro por la capa píxel a píxel, se consigue que se vayan activando las neuronas y el resultado de pasar el filtro por toda la imagen se recoge en el llamado mapa de características [29]. En la figura 20 veremos un procesamiento YOLOv3.



**Fig.20: Procesamiento de YOLOv3 [29].**

### 2.2.7 YOLOv4

Mientras se desarrollaba este proyecto, se publicó la actualización de YOLOv3, que es YOLOv4. Se trata de una versión en la que el principal artífice del concepto YOLO, Joseph Redmon ya no es autor, ya que al finalizar la versión 3, cesó su investigación por el posible mal uso que se podía dar a su tecnología, con fines militares o de terceros con nefastos propósitos.



**Fig.21: Arquitectura de la red YOLOv4**

En esta versión, su motivo principal es optimizar el detector para realizar cálculos en paralelo. Se compone de 3 etapas que podemos observar en la Figura 21: - Backbone: CSPDarknet53, una red que aumenta la capacidad de aprendizaje y que gracias a la unión del módulo de agrupación de pirámide espacial permite mejorar el campo receptivo y distinguir características importantes. - Neck: Módulo de agrupación de pirámides espaciales y agregación de ruta PANet, principalmente. PANet se implementa como sustitución de las redes piramidales de características empleadas para la detección en YOLOv3. - Head: YOLOv3

También encontramos dos conceptos que son: Bag of Specials (bolsa de especiales) y Bag of Freebees (Bolsa de regalos). Bag of specials se refiere a las mejoras que se realizan en el proceso de capacitación como la inferencia de datos, función de coste... pero que no presentan un impacto en la velocidad de inferencia. Mientras que Bag of specials se refiere al impacto marginal que tienen en el tiempo de inferencia y el buen rendimiento de las mejoras anteriores. Además, esto mejora el aprendizaje de las características [26] [29].

Las demás versiones aún presentan inconvenientes en la precisión por tal razón están cada vez realizando mejoras. En este proyecto se utilizó la versión YOLO V3 ya que se obtuvieron buenos resultados.

### 2.2.8 Transfer Learning

Consiste en aprovechar las características aprendidas de un problema resuelto. Por ejemplo, un modelo que reconoce las características para detectar un gato, puede ser utilizado para detectar lince Ibéricos. Esta técnica se utiliza cuando el conjunto de datos del que se dispone es muy reducido.

Para utilizar esta técnica normalmente se copia una arquitectura que ya haya sido entrenada con algún propósito, y luego en vez de utilizar un entrenamiento pesos aleatorios se cargan los pesos del modelo pre-entrenado. Normalmente esta técnica requiere utilizar una arquitectura que no se adapte al problema que se desea resolver, con lo cual se necesita utilizar una capa final de la arquitectura para adaptar el modelo pre-entrenado [29]. En el proyecto se utiliza una arquitectura ya entrenada que es capaz de predecir 80 clases distintas y es el Darknet\_53 de YOLO y se modifican las últimas tres capas para que sea capaz de identificar una nueva clase que es el signo presente en la retinopatía hipertensiva.

## 2.3. Estado de la técnica

Existen muy pocos estudios que realizan clasificación de forma automática de las características observadas en los exámenes de fondo de ojo, para detectar signos de retinopatía hipertensiva (Tabla 2). Por lo anterior en este apartado se pretende realizar una descripción detallada de algunos avances relacionados con técnicas de procesamiento de imágenes y aprendizaje automático versus tipos de retinopatía hipertensiva que hayan realizado clasificación y evaluación de desempeño del algoritmo. Para lo anterior se realizó una búsqueda en los diferentes bases de datos como Google Scholar, Scopus y Science Direct incluyendo los siguientes descriptores: “Retinopatía hipertensiva”, “Retinopatía hipertensiva y aprendizaje automático” y “Retinopatía, procesamiento de imágenes y aprendizaje profundo” de las cuales los criterios de inclusión que se tuvieron en cuenta: la actualidad del estudio partir de 2014; estudios realizados en humanos y el desempeño de la metodología propuesta en cada artículo.

### **“Automated System for the Detection of Hypertensive Retinopathy” [31]**

En este trabajo se propone un sistema automatizado para la detección de retinopatía hipertensiva (HR) utilizando la relación arteriovenosa (AV). El sistema propuesto consiste en un método novedoso para la clasificación de vasos como arterias y venas utilizando un nuevo vector de características y un clasificador híbrido. Este documento también presenta un nuevo método para calcular el ancho del vaso que es útil para medir la relación AV. El sistema detecta si la imagen del fondo de ojo contiene HR o no utilizando la relación AV ya calculada. Se utilizan dos bases de datos de imágenes de fondo de ojo digitales disponibles públicamente, es decir, VICAVR y

DRIVE, para probar el algoritmo propuesto. Los resultados experimentales muestran la validez del algoritmo propuesto.

### **“Classification of Retinal Vessels into Arteries and Veins for detection of Hypertensive Retinopathy” [32]**

Este artículo propone una técnica novedosa para clasificar los vasos en arterias y venas. El sistema propuesto utiliza una máscara de vaso binario y una localización del centro del disco óptico (OD) para extraer la Región de interés (ROI) alrededor del OD. Los módulos de extracción de características extraen una serie de características basadas en la intensidad seguidas de una clasificación de arterias y venas. Las características se extraen centrándose en la diferencia visual entre las arterias y las venas en diferentes espacios de color. El sistema propuesto se prueba en un conjunto de datos de imágenes de fondo de ojo de recopilación local y se consigue una precisión de 81,3%.

### **“Detection of hypertension retinopathy using deep learning and Boltzmann machines” [33]**

Esta investigación tiene como objetivo desarrollar un sistema para la detección temprana de la etapa de retinopatía hipertensiva. El objetivo propuesto es determinar las características relacionadas con las arterias y venas, relación de diámetro (AVR), así como los cambios de posición con el disco óptico (OD) en las imágenes de la retina, para realizar la clasificación de la retinopatía hipertensiva utilizando el enfoque de redes neuronales profundas (DNN) y máquinas de Boltzmann.

### **“The Classification of Hypertensive Retinopathy using Convolutional Neural Network” [34]**

Este artículo propone un sistema de detección precoz de retinopatía hipertensiva. Se procede a utilizar la imagen del fondo de ojo como una entrada de la red neuronal convolucional (CNN) para determinar si hay algún síntoma de HR o no. El sistema propuesto se prueba mediante el conjunto de datos de imágenes DRIVE y, según el experimento, la precisión del método propuesto es del 98,6%.

### **“Textural and intensity feature based retinal vessels classification for the identification of hypertensive retinopathy” [35]**

Este artículo presenta una técnica para la clasificación automática de los vasos sanguíneos de la retina utilizando diferentes clasificadores y el rendimiento de cada clasificador se compara en el mismo conjunto de características. Se utiliza una nueva combinación de características para la clasificación de vasos, que es un paso esencial para el cálculo de la relación arteriovenosa y, posteriormente, la detección de la retinopatía hipertensiva. Los resultados que se logran con el conjunto de funciones propuesto muestran una precisión del 89%.

**“Análisis y diseño de un sistema para identificar signos de retinopatía hipertensiva a través de imágenes de retina, aplicando la tecnología de deep learning” [36]**

Este proyecto plantea un prototipo de inteligencia artificial mediante la creación de un algoritmo para el limitante tecnológico en el área de la oftalmología, que es la detección manual de la retinopatía hipertensiva. Su finalidad es predecir mediante imágenes si una persona tiene o no la enfermedad. Para la creación de este sistema se decidió por la metodología de prototipado ya que tiene etapas funcionales que sirven para la creación del mismo. Se trabajó con la metodología de aprendizaje supervisado mediante redes neuronales convolucionales. Se utilizó la base de datos de kaggle para la obtención y estudio de imágenes de retinas. Además de esto, el proyecto guarda las imágenes para llevar un control de los pacientes para una futura prevención de la enfermedad.

**“Colour Image Segmentation of Fundus Blood Vessels for the Detection of Hypertensive Retinopathy” [37]**

La retina es la parte principal del ojo humano. La mayoría de los problemas diabéticos e hipertensivos afectarán inicialmente a los vasos sanguíneos de la retina del ojo. Este artículo Usa técnicas de segmentación normales, en donde puede identificar solo los vasos sanguíneos. Por lo tanto, la segmentación de la imagen en color es la mejor manera de identificar los problemas de la retina, porque mediante la segmentación de la imagen en color podemos diferenciar las venas y arterias de la retina. Se utilizó técnicas de enmascaramiento para segmentar las imágenes de arterias y venas. La técnica propuesta se evalúa con datos de diez pacientes diagnosticados de retinopatía hipertensiva. Para este proceso, la ecualización del histograma se realiza en la imagen del plano verde complementado. Del resultado, la región óptica se elimina mediante una operación morfológica, después de lo cual se convierte en una imagen binaria. Esta salida se multiplica con los canales R, G y B de datos de entrada y se combina para obtener la región segmentada requerida de los vasos sanguíneos de la retina. Toda esta extracción de vasos se ha realizado mediante técnicas de procesamiento de imágenes.

**“Artery and vein classification for hypertensive retinopathy” [38]**

En este trabajo, se ha propuesto un sistema de soporte automatizado para la clasificación de arterias y venas (AV) para la detección de HR. El marco propuesto clasifica AV utilizando diferentes vectores de características obtenidos a través del algoritmo de seguimiento de vasos Radón. Las características se extraen del conjunto de datos estándar DRIVE disponible públicamente. Una de las principales ventajas del seguimiento de vasos es que se puede utilizar más para detectar la relación arteriovenosa (AVR). Los resultados de la clasificación se obtienen utilizando tres clasificadores diferentes, a saber, SVM (Máquina de Vector de soportes), NN

(Red Neuronal) y CNN (Red Neuronal convolucional). Los resultados experimentales muestran que CNN supera a NN y SVM en la clasificación AV para la detección de HR.

**“A Comprehensive Study of Retinal Vessel Classification Methods in Fundus Images for Detection of Hypertensive Retinopathy and Cardiovascular Diseases” [39]**

En este artículo la retinopatía hipertensiva se puede detectar en una etapa más temprana mediante el análisis de la imagen retiniana. Hoy en día, es obvio que existe una relación entre los cambios en la estructura de los vasos retinianos y las enfermedades más comunes como la hipertensión, el ictus, las enfermedades cardiovasculares, que pueden ser detectadas por la imagen no invasiva del fondo de ojo de la retina. Esto podría sugerir la posibilidad de utilizar este modelo de imagen de fondo de ojo en enfoques de clasificación. Finalmente, los atributos de clasificación seleccionados se aplican a través del algoritmo genético con red neuronal de función de base radial para el diagnóstico de la enfermedad con el fin de mejorar la precisión de la clasificación con un menor costo computacional.

**“Detection of hypertensive retinopathy using principal component analysis (PCA) and backpropagation neural network methods” [40]**

Este artículo propuso un sistema para la detección de la retinopatía hipertensiva mediante el análisis de componentes principales (PCA) y la red neuronal de retropropagación (BNN). La imagen de la retina se tomó de la base de datos STARE que se separó en datos de aprendizaje y de prueba con una proporción de 7 a 3. El PCA se utiliza como método de reducción de dimensión de imagen de fondo de ojo lo que permite reducir así la carga de cálculo para la red neuronal. Este artículo presentó Backpropagation Neural Network (BNN) como el algoritmo de clasificación principal que se realiza estableciendo sus parámetros, aprendiendo y probando los datos. Entonces, el modelo podría clasificar la imagen de la retina en una de dos clases, a saber, la retina normal y la retina con presión arterial alta, según el resultado de salida de BNN. El resultado del modelo propuesto mostró una precisión del 86,36%.

**“DenseHyper: an automatic recognition system for detection of hypertensive retinopathy using dense features transform and deep-residual learning” [41]**

En este artículo se desarrolla un sistema computarizado de retinopatía hipertensiva (DenseHyper) para detectar la HR basada en el aprendizaje residual profundo (DRL) multicapa con estrategia de entrenamiento de características desde cero. El sistema DenseHyper consta de una arquitectura multicapa diferente de la capa de características entrenadas (TF-L) por modelo CNN y una capa de transformación de características densa (DFT-L) por DRL para extraer características de las imágenes del fondo de ojo retiniano y clasificarlas en HR y no HR enfermedad.

**“Pre-diagnosis of Diabetic Retinopathy using Blob Detection” [42]**

El método propuesto en este artículo presenta una técnica de extracción de características eficaz basada en la detección de manchas seguida de la clasificación de las diferentes etapas de la retinopatía diabética mediante la técnica de aprendizaje automático. Esta técnica de extracción de características podría ayudar a la caracterización automática de las imágenes de la retina para la retinopatía diabética con una precisión del 83% con el algoritmo de clasificación de aprendizaje automático más eficiente. Esta técnica se basa en la extracción de características eficiente, simple y precisa mediante el pre-procesamiento de imágenes y la detección de manchas en las imágenes de la retina capturadas mediante fotografía de fondo de ojo.

**“Automatic detection and severity classification of diabetic retinopathy” [43]**

El objetivo de este trabajo de investigación fue desarrollar un método para la detección automática de la retinopatía diabética y proponer un modelo para decidir la progresión / gravedad mediante imágenes de fondo de ojo. El método fue desarrollado para que la RD pueda ser detectada de manera efectiva y eficiente antes de causar daño al ojo, sin la presencia de un oftalmólogo. El cribado manual requiere la presencia de un oftalmólogo y el recurso del tiempo. La detección de exudados es importante para el diagnóstico de RD. El rendimiento del método propuesto ha sido evaluado y aprobado por un oftalmólogo. Este trabajo contribuye al campo de la detección automática de estructuras anómalas y su gravedad.

A partir del estado de la técnica se presenta una tabla resumen de los trabajos relacionados donde se utilizan las siguientes abreviaturas: **NN**: Red Neuronal ; **CNN**: Red Neuronal convolucional ; **SVM**: Máquina de Vector de soportes ; **Naive Bayes**: Ingenuo Bayes ; **DNN**: Red Neuronal Profunda ; **Bagged Trees**: Árbol de decisión **BNN**: Red Neuronal Binarizada.

Tabla 2. Estado de la técnica

Ref.	Año	Tipo de retinopatía	Tipo de clasificación	Base de Datos	Evaluación
[31]	2014	Hipertensiva(Relación A/V)	SVM y Naive Bayes	VICAVR y DRIVE	Exactitud: 0,97 y 0,98
[32]	2015	Hipertensiva (Relación A/V)	SVM	DRIVE y LOCAL DATABASE OF 25 IMAGES	Precisión: 75% y 81,3%
[33]	2017	Hipertensiva (Relación A/V y cambios de posición del disco óptico)	DNN	100 imágenes locales de Eye Hospital	No especifica
[34]	2017	Hipertensiva(características de HR)	CNN	DRIVE	Precisión: 98,6%
[35]	2018	Hipertensiva (Relación A/V)	SVM y Bagged Trees	locales AFIO, Pakistán	Precisión: 89 %
[36]	2018	Hipertensiva (características de HR)	CNN	kaggle	No especifica
[37]	2018	Hipertensiva(Segmentación de vasos (AV))	Procesamiento de imágenes	Locales	No especifica
[38]	2019	Hipertensiva (características de HR)	NN; CNN y SVM	DRIVE	Precisión: 92,55%
[39]	2019	Hipertensiva (Relación A/V)	CNN	Pima Indian Diabetes Dataset	No especifica

[40]	2019	Hipertensiva (Relación A/V)	BNN	Locales	Precisión 86,36%
[41]	2020	Hipertensiva (Relación A/V)	CNN	VICAVR	Precisión 95%
[42]	2020	Diabética (Detección de manchas)	CNN y Naive Bayes	MESSIDOR	Precisión 83%
[43]	2020	Diabética (Detección de exudados y microaneurismas manchas)	SVM	MESSIDOR	No especifica

### 2.3.1 Brechas de investigación

Como se evidencia en la tabla 2 se encuentran diversos avances que se han realizados y algunos todavía en fase teórica para la detección de la retinopatía hipertensiva a través de sistemas automáticos y bases de datos. Además de los trabajos relacionados en la tabla 2, se identificaron las siguientes debilidades que se traducen en brechas de investigación.

- A. La mayoría de los estudios se concentran en identificar la relación Arterio-Vena (AV), sin embargo, síntomas como cruces, exudados y demás que son incluidos en el diagnóstico dado por el especialista no son detectados por estos métodos.
- B. La mayoría de las metodologías estudiadas no identifican la zona (cuadrante: superior, inferior, temporal y nasal) en la que se encuentran los signos.
- C. No se han realizado pruebas en contexto clínico, es decir con pacientes o conjunto de imágenes diferente a las bases de datos de dominio público.
- D. La mayoría de trabajos no muestran el conteo de afectaciones, es decir, ¿cuántos cruces?, ¿Cuántos exudados?, etc, esto con el fin de poder hacerle un seguimiento y pronóstico al paciente.
- E. La mayoría de trabajos se enfocan en un solo síntoma de detección temprana, además son algoritmos aislados que no se integran a una plataforma que despliegue el resultado al especialista.

Esta propuesta busca aportar a las brechas A y C mencionadas anteriormente.

## CAPÍTULO 3

### 3. METODOLOGÍA

La metodología seguida en este trabajo fue de tipo CRISP-DM (Cross Industry Standard Process of Data Mining) a la cual se le realizaron algunas adaptaciones. Esta metodología está dividida en las siguientes fases: Comprensión del problema, comprensión de los datos, preparación de los datos, modelado y evaluación.

#### 3.1 Comprensión del problema

La formulación del problema y los objetivos de investigación son planteados en el Capítulo I y la revisión sobre el estado de la técnica en el Capítulo II. Donde se determinó el problema central de investigación: poder detectar signos de la retinopatía hipertensiva en imágenes de fondo de ojo mediante aprendizaje de máquina, además se identificaron las brechas de investigación: Gran cantidad de enfoques solo se centran en la relación arteria vena, más no en otro tipo de signo como las hemorragias, microaneurisma, drusas, exudados y cruces; La mayoría de los trabajos realizan una clasificación binaria, sin embargo, es necesario realizar una detección en la imagen que permita al especialista identificar rápidamente las lesiones además de observar el cuadrante en el que se encuentra para poder estadificar la lesión.

#### 3.2 Comprensión de los datos

##### 3.2.1 Recolección de datos

Las imágenes para la identificación de los signos se obtuvieron de bases de datos libre que se encuentran en internet y son muy utilizadas en investigaciones y trabajos relacionados con la retinopatía hipertensiva y diabética. A continuación se realiza una breve descripción de tres de estas bases de datos que se utilizaron para alimentar nuestro conjunto de datos para el entrenamiento de nuestro propio modelo a través de transfer learning.

**E-Ophta:** este conjunto de datos se divide en 2 categorías: una llamada E-Ophta MA y la otra E-Ophta EX. E-Ophta MA posee 148 imágenes con microaneurismas y pequeñas hemorragias, y 233 imágenes sin lesiones; en cambio E-Ophta EX posee un conjunto de 47 imágenes de fondo de ojo con una segmentación de exudados y 35 imágenes sin ningún tipo de lesión y que se encuentran etiquetadas como imágenes normales. [44].

**Kaggle DR Dataset:** esta base de datos es proporcionada por EyePACS, (n.d.) para colaborar en la investigación y desarrollo de trabajos relacionados a la detección y diagnóstico de la RD. En

total posee 88,702 imágenes, de las cuales 35,126 son para tareas de entrenamiento y 53,576 son para tareas de pruebas. [45]

**ROC (Retinopathy Online Challenge):** contiene 100 imágenes digitales a color de fondo de ojo con microaneurismas en todas las imágenes. Estas imágenes se encuentran aleatoriamente distribuidas y se dividen en 50 imágenes para entrenamiento de la red y 50 imágenes para pruebas. [46]

Estas bases de datos permitieron tener la lista de imágenes que se usaran para los respectivos entrenamientos que permitirán reconocer signos de retinopatía hipertensiva. Estas imágenes se organizaron en carpetas, cada una con dos signos distinto, es decir, una contendrá hemorragias y microaneurismas, otra exudados y drusas.

### 3.2.2 Anotación de los datos

Las imágenes que brindan las bases de datos mencionados anteriormente, fueron diagnosticadas por especialistas en oftalmología, y posteriormente realizan anotaciones que permitieran identificar en la imagen del examen de fondo de ojo alguno de los signos presentes en el parénquima que evidencian la presencia de hipertensión arterial en el paciente. Estas anotaciones se pueden evidenciar en la figura 22 y 23, donde están identificando los signos presentes en la

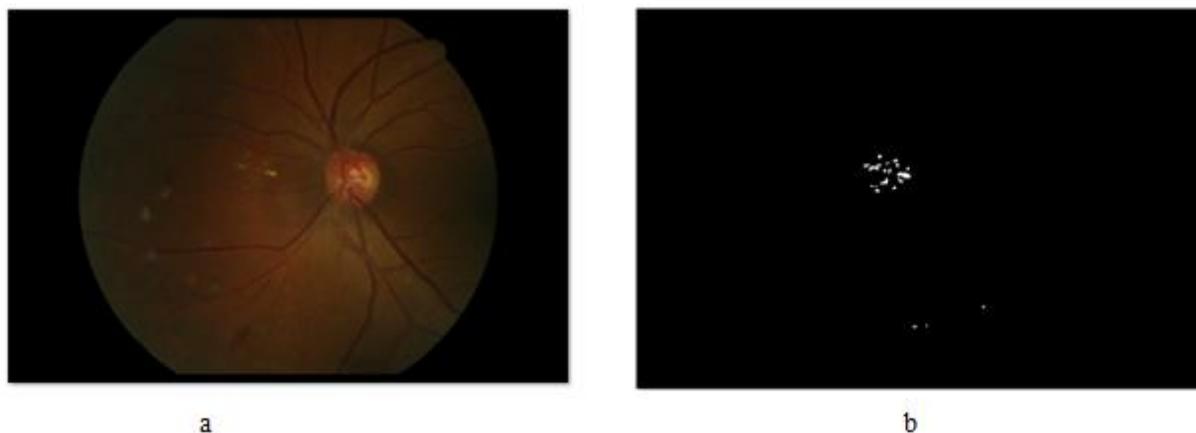
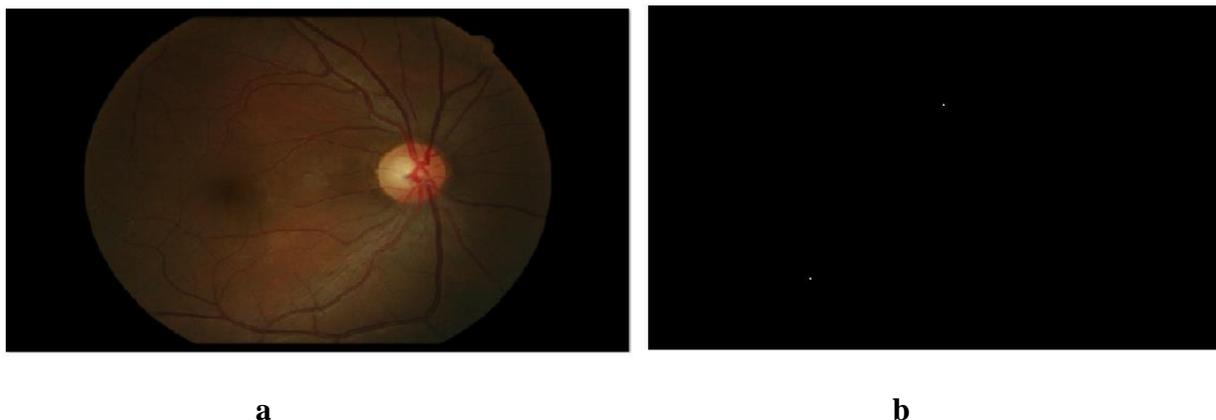


imagen de fondo de ojo. Estas anotaciones permiten evaluar el sistema y mostrar su rendimiento.

**Fig.22** Ejemplo de imágenes de la base de datos E-Ophta EX. Imagen con exudados (a); Anotaciones hechas por los especialistas con respecto a exudados (b)

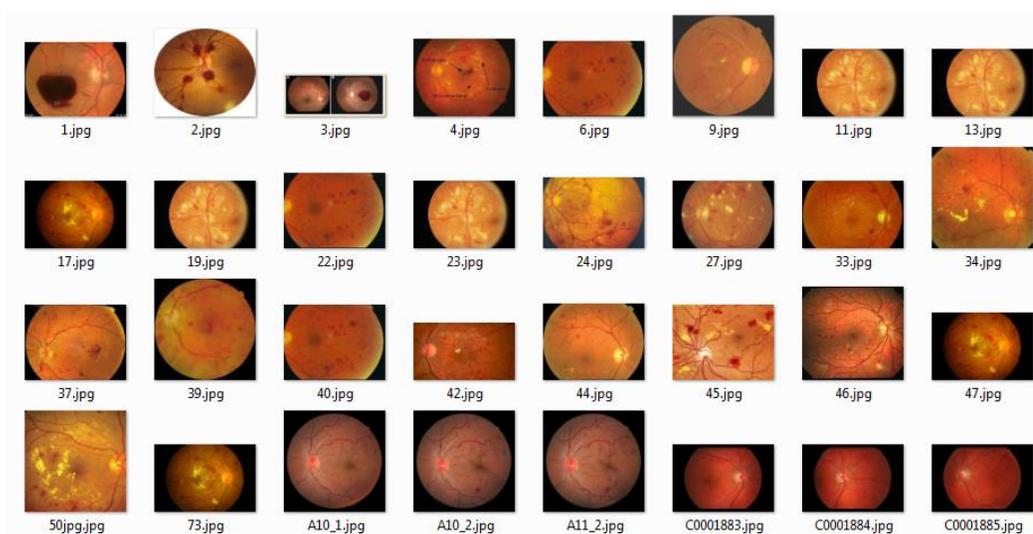


**Fig.23** Ejemplo de imágenes de la base de datos E-Ophta MA. Imagen con microaneurisma (a); Anotaciones hechas por los especialistas con respecto al microaneurisma (b)

Para este trabajo se utilizó un total de 200 imágenes que presentan microaneurismas y hemorragias; 150 imágenes con exudados y drusas (Estas imágenes son de la base de datos libres mencionadas anteriormente).

### 3.3 Preparación de los datos

El propósito principal de esta etapa es recolectar las imágenes para cada una de nuestras bases de datos y prepararla para su posterior entrenamiento en el modelo que se utilizará. Cada una de las imágenes obtenida de las bases de datos fue separada en diferentes carpetas, cada carpeta contiene dos signos importantes en la retinopatía hipertensiva, por ejemplo una de ellas contiene hemorragias y micro-aneurismas (figura 24), y así con los diferentes signos presentes en la retinopatía hipertensiva. Esto con el fin de obtener por cada dos signos un conjunto de datos que permita preparar las imágenes para el respectivo etiquetamiento y posterior entrenamiento.

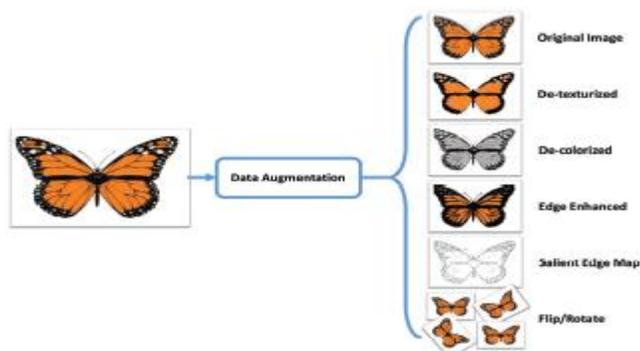


**Fig. 24.** Carpeta con N imágenes de los signos de hemorragias y micro-aneurismas.

Cada carpeta contiene imágenes para cada signo en formato Jpg y Jpeg con tamaños diferentes, por otro lado la base de datos nos suministra una cantidad de imágenes limitadas y teniendo en cuenta que las redes neuronales que se van utilizar en este proyecto se benefician de grandes cantidades de datos para el entrenamiento, se hizo necesario ampliar nuestra base de datos y una práctica común para ampliar artificialmente el conjunto de datos es mediante un proceso llamado aumento de datos o data augmentation [47]. Esta técnica permite crear imágenes sintéticas a partir de las imágenes disponibles. No obstante, se si eligen transformaciones que generan imágenes irreales, el entrenamiento se vería afectado negativamente, provocando que la red no aprendiese las características de las imágenes reales [47]. En general, la elección de la transformación utilizada para el aumento de datos depende del conjunto de los mismos, pero hay algunas estrategias comunes para el aumento de datos en el campo de visión artificial [48].

### 3.3.1 Aumento de datos (data augmentation)

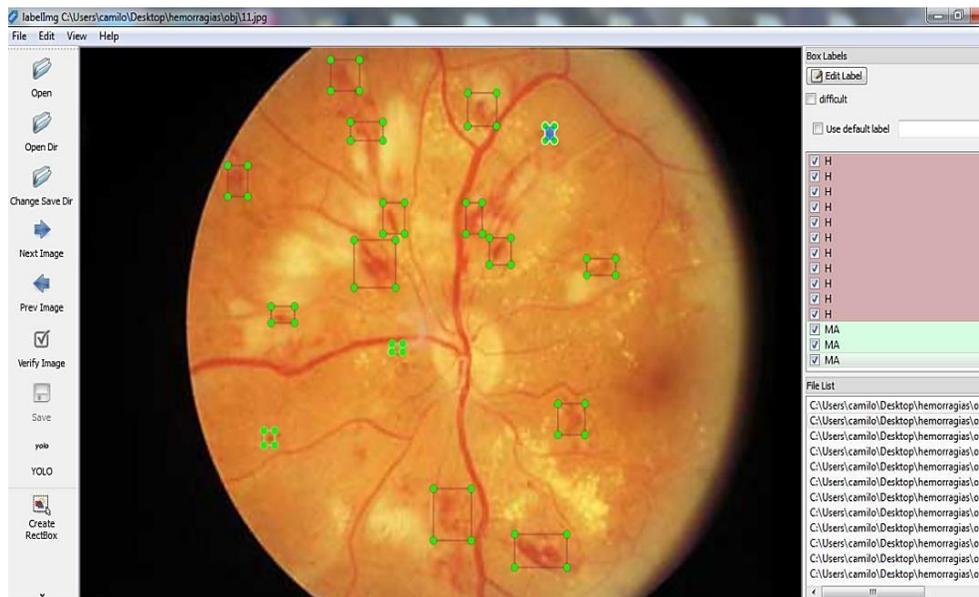
La data augmetation se utiliza para generar nuevos datos a partir de los originales. En el caso de las imágenes es mucho más sencillo realizar data augmentation. Para ello se realizan transformaciones geométricas, simétricas respecto a diferentes ejes, modificación del color, etc. Un ejemplo de data augmetation es el que se observa en la figura 25, en donde se utilizan diversas técnicas como giros, rotaciones, realce de bordes, etc. [49]



**Fig. 25.** Ejemplo de data augmetation [49]

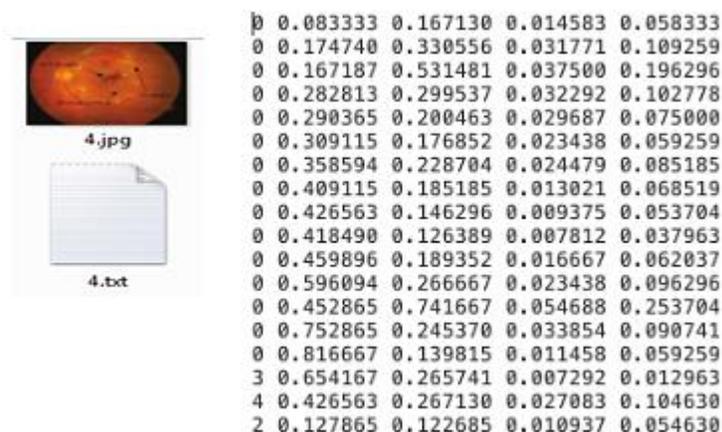
### 3.3.2 Etiquetado

Una vez se hallan seleccionado las imágenes se utiliza una herramienta gratuita de código abierto para etiquetar imágenes gráficamente, la herramienta está escrito en Python y usa QT para su interfaz gráfica y responde al nombre de labelImg. Teniendo en cuenta la importancia del conjunto de datos para poder entrenar el modelo, se hace uso de LabelImg donde las etiquetas se utilizan para ayudar a identificar los componentes de los datos que se desea que el modelo identifique en conjuntos de datos, La herramienta se muestra en figura 26. Este proceso se debe repetir con cada imagen del conjunto de datos que se tenga.



**Fig. 26** etiquetado de las imágenes mediante el editor labelImg.

LabelImg genera para cada imagen un archivo marcado como Yolo, por ejemplo la imagen 4.jpg tendrá el archivo 4.txt correspondiente en la misma carpeta como se observa en la figura 27



**Fig. 27** etiquetado y formato de etiqueta YOLO.

El formato de etiqueta YOLO que tiene la herramienta LabelMag guarda en el archivo txt que presenta la siguiente información ver (Figura 27):

- Yolo object-id: representa el número correspondiente a la categoría de objetos que enumeré en el archivo 'classes.txt',
- centro\_x , centro\_y: representan el punto central del rectángulo de selección
- width, height: ancho y alto representan el ancho y alto del rectángulo.

### 3.4 Modelado

En esta etapa se seleccionan las técnicas y algoritmos de detección, en donde se realiza la construcción y aprendizaje de los diferentes modelos. En este caso se trabajará con dos versiones del modelo YOLO (**You Only Look Once**), la cual es una arquitectura de detección de objetos muy rápida y precisa en algunas versiones, esta arquitectura fue creada por Joseph Redmond como principal autor [26].

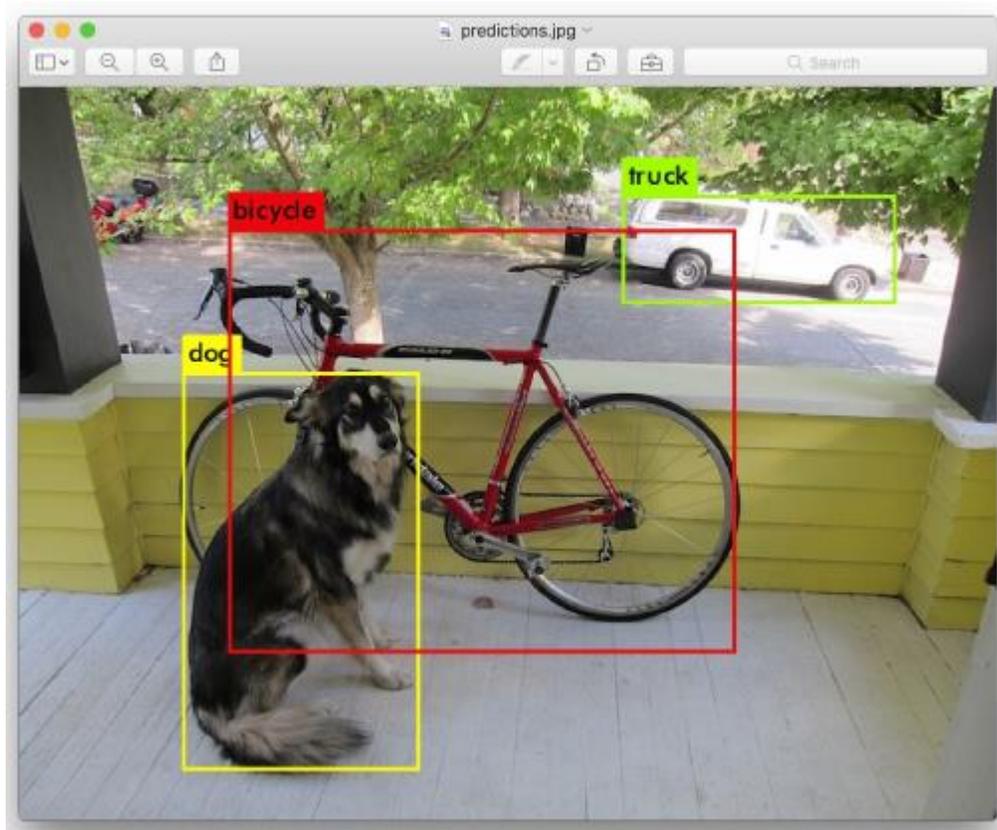
#### 3.4.1 YOLO

Hoy en día las únicas redes que detectan regiones de interés son las R-CNN o sus versiones mejoradas Fast R-CNN. El problema de estas redes son su lentitud para dar una respuesta, en contraste YOLO es mucho más rápidas aunque menos precisas en comparación con R-CNN y Fast R-CNN, YOLO ha mostrado muy buenos resultados en la detección de objetos, por otro lado este modelo ha estado evolucionado mejorando su precisión siendo así que hasta la actualidad tenemos la versión YOLOV7. En este trabajo se ha elegido el modelo YOLOV3 y YOLOv4 en su versión pequeña es decir YOLOV3\_Tiny y YOLOV4\_Tiny y su versión grande

YOLOV3 y YOLOV4 , ya que entre todas las versiones son más rápida de entrenar, presenta buena precisión y se puede vincular con las bibliotecas Opencv y Tensorflow de Python, mientras que las demás versiones trabajan con las bibliotecas Pytorch y presenta conflictos con la biblioteca opencv.

### 3.4.2 Algoritmo YOLO

YOLO es un sistema, que utiliza una red neuronal convolucional para la detección de objetos en tiempo real [29]. El sistema aplica una única red neuronal a la imagen completa, razón por la cual es muy rápido. Esta red divide la imagen en regiones y predice múltiples cajas delimitadoras (en inglés, bounding box), y la probabilidad de detección de las clases de entrenamiento para cada caja delimitadora. Por último, utiliza un método de supresión de no-máximos para eliminar múltiples detecciones del mismo objeto. Como resultado, el sistema imprime los objetos que detectó, su confianza y el tiempo de ejecución. En la Figura 26 se muestra una ejecución del sistema sobre una imagen de entrada.



**Fig 28.** Detección de objetos con YOLO [29].

### 3.4.3 Arquitectura Neuronal YOLOV3

La arquitectura de YOLOv3 está compuesta por 53 capas convolucionales, razón por la cual recibe el nombre de Darknet-53 Ver Fig.29a. Cada capa convolucional es seguida de una normalización de lote (en inglés, batch normalización) y la función de activación Leaky ReLU. No se utiliza ninguna capa de reducción, en su lugar, se utilizan capas convolucionales con paso=2. Con esto, se reduce la dimensionalidad del mapa de características, evitando perder las características de bajo nivel que se le atribuyen a las capas de reducción-máxima.

Por otro lado, se plantea otra arquitectura sin capas residuales y con menos capas convolucionales (13) llamada YOLOv3 tiny. Como resultado se tiene una red neuronal más rápida, pero con menos confianza en la detección y clasificación. En la Figura 29b se muestra la arquitectura de YOLOv3 tiny.

Type	Filters	Size	Output
Convolutional	32	3 × 3	256 × 256
Convolutional	64	3 × 3 / 2	128 × 128
1x	Convolutional	32	1 × 1
	Convolutional	64	3 × 3
Residual			128 × 128
Convolutional	128	3 × 3 / 2	64 × 64
2x	Convolutional	64	1 × 1
	Convolutional	128	3 × 3
Residual			64 × 64
Convolutional	256	3 × 3 / 2	32 × 32
8x	Convolutional	128	1 × 1
	Convolutional	256	3 × 3
Residual			32 × 32
Convolutional	512	3 × 3 / 2	16 × 16
8x	Convolutional	256	1 × 1
	Convolutional	512	3 × 3
Residual			16 × 16
Convolutional	1024	3 × 3 / 2	8 × 8
4x	Convolutional	512	1 × 1
	Convolutional	1024	3 × 3
Residual			8 × 8
Avgpool		Global	
Connected		1000	
Softmax			

**a**

Layer	Type	Filters	Size/Stride	Input	Output
0	Convolutional	16	3 × 3/1	416 × 416 × 3	416 × 416 × 16
1	Maxpool		2 × 2/2	416 × 416 × 16	208 × 208 × 16
2	Convolutional	32	3 × 3/1	208 × 208 × 16	208 × 208 × 32
3	Maxpool		2 × 2/2	208 × 208 × 32	104 × 104 × 32
4	Convolutional	64	3 × 3/1	104 × 104 × 32	104 × 104 × 64
5	Maxpool		2 × 2/2	104 × 104 × 64	52 × 52 × 64
6	Convolutional	128	3 × 3/1	52 × 52 × 64	52 × 52 × 128
7	Maxpool		2 × 2/2	52 × 52 × 128	26 × 26 × 128
8	Convolutional	256	3 × 3/1	26 × 26 × 128	26 × 26 × 256
9	Maxpool		2 × 2/2	26 × 26 × 256	13 × 13 × 256
10	Convolutional	512	3 × 3/1	13 × 13 × 256	13 × 13 × 512
11	Maxpool		2 × 2/1	13 × 13 × 512	13 × 13 × 512
12	Convolutional	1024	3 × 3/1	13 × 13 × 512	13 × 13 × 1024
13	Convolutional	256	1 × 1/1	13 × 13 × 1024	13 × 13 × 256
14	Convolutional	512	3 × 3/1	13 × 13 × 256	13 × 13 × 512
15	Convolutional	255	1 × 1/1	13 × 13 × 512	13 × 13 × 255
16	YOLO				
17	Route 13				
18	Convolutional	128	1 × 1/1	13 × 13 × 256	13 × 13 × 128
19	Up-sampling		2 × 2/1	13 × 13 × 128	26 × 26 × 128
20	Route 19 8				
21	Convolutional	256	3 × 3/1	13 × 13 × 384	13 × 13 × 256
22	Convolutional	255	1 × 1/1	13 × 13 × 256	13 × 13 × 256
23	YOLO				

**b**

**Fig 29.** Arquitecturas (a) Darknet-53 (b) YOLOV3\_Tiny [29].

### 3.4.4 Procesamiento YOLOv3

La imagen de entrada del sistema YOLOv3 se puede configurar de diferentes dimensiones en cada proceso de entrenamiento, como único requisito es que sean imágenes cuadradas de un tamaño múltiplo de 32, por ejemplo 608x608 o 416x416. De lo contrario, Darknet redimensiona las imágenes al tamaño establecido en el archivo de configuración.

La entrada viene dada por:

$$\text{Entrada (m, h, w, d)} \quad (3.0)$$

Dónde:

**m** es el tamaño del lote de entrenamiento (en inglés, batch).

**h** es la altura de la imagen (en inglés, height).

**w** es el ancho de la imagen (en inglés, width).

**d** son los canales de la imagen de entrada (en inglés, depth).

Dependiendo de la tarjeta gráfica, el tamaño del lote de entrenamiento puede ser subdividido. De esta manera, si se tiene un lote de tamaño 64 (64 imágenes) y subdivisiones de 16, entonces se procesarán 4 imágenes en paralelo en cada iteración de la red neuronal convolucional. Siguiendo este ejemplo y utilizando una imagen de 3 canales (RGB) de 416x416, a la entrada de YOLOv3 tendremos:

$$\text{Entrada (64; 416; 416; 3)} \quad (3.1)$$

YOLOv3 hace el proceso de detección en tres diferentes escalas, redimensionando la imagen de entrada por un factor de 32, 16 y 8 respectivamente. Por ejemplo, si se tiene como entrada una imagen de 416x416 y reduce por un factor de 32, el tamaño del mapa de características será de 13x13 (Fig 30a). A cada elemento del mapa se le conoce como celda [12]. En el mapa de características, cada celda predice un número fijo de cajas delimitadoras. En cada celda de un mapa de características se tienen los siguientes atributos:  $B \times (5+C)$ .  $B$  representa el número de cuadros delimitadores que cada celda puede predecir. Según [11], cada uno de estos cuadros delimitadores  $B$  puede especializarse en la detección de cierto tipo de objeto. Cada uno de los cuadros delimitadores tiene atributos  $5+C$ , que describen las coordenadas del centro, las dimensiones, la puntuación de objetividad y las confianzas de clase  $C$  para cada cuadro delimitador. En este trabajo de tesis, YOLOv3 predice 3 cuadros delimitadores para cada celda.

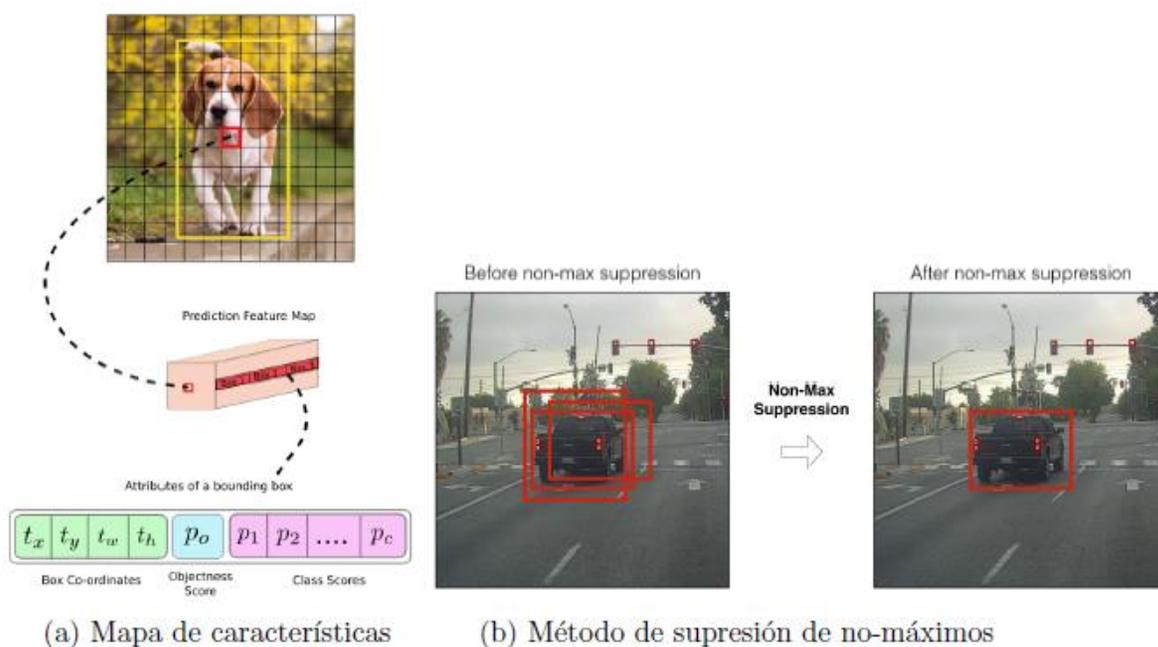
Por ejemplo, si en la arquitectura principal de YOLOv3 (DarkNet-53) se tiene una imagen de entrada de 416x416, se tendrán como resultado 3 mapas de características con 10,647 cajas delimitadoras predichas:

$$((52 * 52) + (26 * 26) + (13 * 13)) * 3 = 10,647 \quad (3.2)$$

Para reducir este número significativo de cajas delimitadoras se utilizan dos filtros.

- Se utiliza un umbral mínimo en la probabilidad de detección de un objeto en cada caja delimitadora, si no se supera dicho umbral, la caja delimitadora será descartada.
- Se utiliza un método de supresión de no-máximos con el objetivo de evitar múltiples detecciones del mismo objeto. Para esto se utiliza la métrica IoU (siglas del inglés Intersection Over Union) de la siguiente manera:

Se elige la caja delimitadora con mayor probabilidad de detección y se compara con las demás cajas delimitadoras. Si el resultado de IoU es superior a un umbral, la caja delimitadora con menor probabilidad de detección es descartada (Fig 30b).



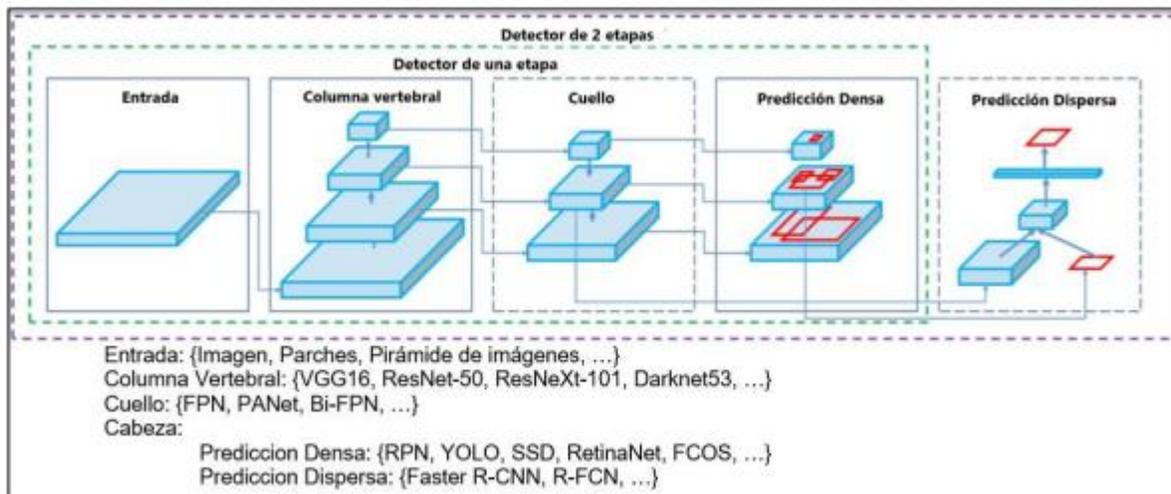
**Fig 30.** Procesamiento YOLOV3 [50].

### 3.4.5 Arquitectura Neuronal YOLOv4 y Procesamiento

En esta arquitectura se presenta una gran diferencia con respecto a YOLOV3 y es con sus autores, puesto que los autores de las versiones anteriores, Joseph Redmon y Ali Farhadi se retiraron del proyecto, Joseph anunció en su cuenta de Twitter que fue debido al uso militar que se le estaba dando a su tecnología, junto con todas las implicaciones sociales que había observado, en cambio Ali Farhadi fundó una compañía llamada Xnor.ai que fue adquirida por Apple tiempo después. Fue entonces en 2020 cuando surgió YOLO v4, un algoritmo diseñado por Alexey Bochkovskiy, Chien-Yao Wang y Hong-Yuan Mark Liao, que mostró una notable mejora en su desempeño, tanto en velocidad como en precisión comparado con otros métodos como YOLO v3 [51].

Un cambio en YOLO v4 es la “Data Augmentation” o aumento de datos, es la interpretación de la misma información desde diferentes puntos de vista, este fue una de las mejoras respecto a YOLO v3 en su entrenamiento, se basa en modificaciones pixel por pixel en las imágenes de entrenamiento, cambios de color, textura, parches negros o blancos, cortes y demás modificaciones sobre la imagen que ayudan al algoritmo a aumentar su precisión y flexibilidad, pero sin afectar su rendimiento en términos de velocidad. Este cambio de entrenamiento los autores lo denominan “Bag of freebies”, bolsa de regalos en español, pues implica un aumento en el costo de entrenamiento, pero aumentando la precisión sin costos en hardware, de forma que obtenemos un mejor desempeño gratis [51].

Para su arquitectura se basaron en distintas etapas del algoritmo (Ver Figura 31), que a su vez pueden tener varias subestructuras en su interior, por ello los autores realizaron pruebas para determinar los modelos más adecuados, en primer lugar tenemos la columna vertebral que es la estructura principal para extraer el mapa de características de la imagen, en este caso observaron el mejor rendimiento con CSPDarknet53 para el set de datos de MS COCO, este es una modificación de Darknet53 usado en YOLO v3, que implementa la base del modelo CSPNet basado en conexiones parciales cruzadas y permite obtener un mapa de características más significativo o elaborado, sin cambiar la velocidad del algoritmo. Para el cuello utilizan path-aggregation network o PAN, que similar a la FPN de YOLO v3 busca crear diferentes niveles de mapas de características desde los cuales observar objetos a diferentes escalas. Y finalmente la cabeza, en este caso se mantiene la cabeza de YOLO v3 con algunos ajustes de optimización [51].



**Fig 31.** Arquitecturas YOLOV4 [51]

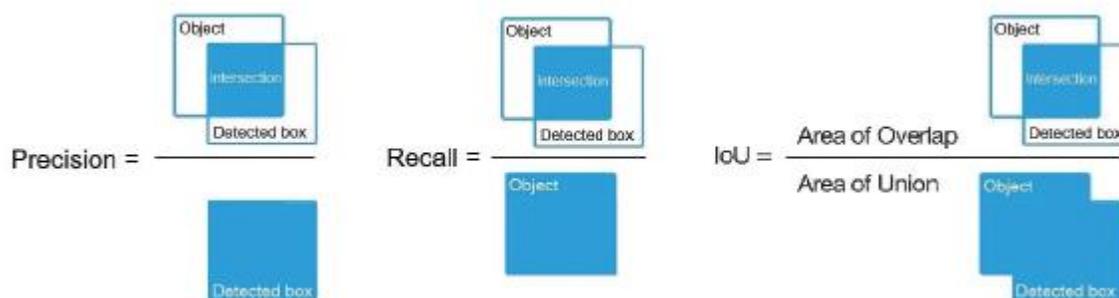
YOLOv4 es la actualización de YOLOv3 y es una red que aumenta la capacidad de aprendizaje y que gracias a la unión del módulo de agrupación de pirámide espacial y agregación de ruta PANet, que se implementó como sustitución de las redes piramidales de características empleadas para la detección en YOLOv3, permite mejorar el campo receptivo y distinguir características importantes.

### 3.5 Evaluación

En esta etapa se evalúa el modelo en relación a la detección de los diferentes signos presentes en la retinopatía hipertensiva, adicionalmente se realiza una prueba del rendimiento del algoritmo con imágenes de fondo de ojo que no fueron utilizados para la construcción del modelo.

#### 3.5.1 Medida de precisión en la detección de objetos

Para medir la precisión de la red, es decir, que un bounding box sea más preciso que otro que pueda encontrar la red, se necesita definir el término de buen resultado y mal resultado. Para ellos se usa la función IoU (Intersection over union), es decir, computa una división entre el área de intersección del verdadero bounding box y el detectado, entre la union de ambos como se puede observar en la Figura. 32. Por consenso, en el test de visión por computador se dice que el resultado es correcto si  $\text{IoU} \geq \text{umbral}$ . El umbral que se establece para que se detecte un objeto. Generalmente 50%, 75%, o 95%, en conclusión entre mayor sea el IoU más preciso es el resultado.



**Fig 32.** Métrica de medida IoU para detección de objetos [52]

Otras de las métricas utilizadas para reflejar el rendimiento del modelo es mAP (precisión media). Para ellos se requiere de la precisión y Recall. Para ello es necesario obtener la matriz de confusión de los modelos.

### 3.5.2 Matriz de confusión

Una matriz de confusión es una representación matricial de los resultados de las predicciones de un modelo. Los diferentes índices que se obtienen de la matriz de confusión, se encuentran en la tabla 3, esta matriz muestra el rendimiento de los algoritmos donde cada columna representa el número de predicciones de cada clase, mientras que cada fila representa una instancia de clase verdadera.

Tabla 3. Matrix de Confusión Teórica.

		Clase Predicha		
		Positivo	Negativo	
Clase Real	Positivo	TP (Verdadero Positivo)	FN (Falso Negativo)	$N^+ = TP + FN$
	Negativo	FP (Falso Positivo)	TN (Verdadero Negativo)	$N^- = FP + TN$
		$\hat{N}^+ = TP + FP$	$\hat{N}^- = FN + TN$	

- **Exactitud (Accuracy):** número de segmentos correctamente identificados dividido número total de segmentos.

$$Acc = \frac{TP + TN}{TP + FN + FP + TN} \quad (3.3)$$

- **Sensibilidad (Recall):** mide la proporción de positivos reales que se identifican correctamente como tales.

$$Se = \frac{TP}{TP + FN} \quad (3.4)$$

- **Precisión (Precisión):** Número de instancias correctamente clasificadas dividido el número de resultados devueltos.

$$Pr = \frac{TP}{TP + FP} \quad (3.5)$$

- **Puntaje F1 (F1-Score):** Es la media armónica de la precisión y sensibilidad, donde una puntuación F1 alcanza su mejor valor en 1 (precisión y sensibilidad perfecta) y peor en 0.

$$F1 = \frac{2 \times Pr \times Se}{Pr + Se} \quad (3.6)$$

- **Área bajo la curva ROC (AUC):** El enfoque balanceado más popular para medir la sensibilidad y la especificidad es el análisis ROC (análisis de la característica operativas del receptor) que involucra una representación gráfica de la sensibilidad frente a la tasa de falsos positivos. La información sobre el rendimiento de clasificación en la curva ROC se puede resumir en una puntuación conocida como el área bajo la curva ROC (AUC).

$$AUC = \int_0^1 Pr[TP](v)dv \quad (3.7)$$

Donde  $Pr[TP]$  es función de  $v = Pr[FP]$ .

Donde el verdadero positivo (TP) representa el número total de segmentos normales detectados correctamente y verdadero negativo (TN) representa el número total de segmentos anormales detectados correctamente. Falso positivo (FP) y Falso negativo (FN) representa el número total de errores positivos y errores negativos.

## CAPÍTULO 4

### 4. RESULTADOS Y DISCUSIÓN

#### 4.1 Conjunto de datos

Se recolectaron un total de 200 imágenes de fondo de ojo que presentan micro-aneurismas y hemorragias; 150 imágenes con exudados y drusas de acuerdo a las bases de datos libres mencionadas en la sección 3.2.

Las imágenes de fondo de ojo se clasificaron de acuerdo a su dimensión y formato, para el signo de hemorragias y micro-aneurisma como puede verse en la tabla 4 y para exudados y drusas en la tabla 5.

Tabla 4. Cantidad de imágenes de cada resolución en el conjunto de datos utilizado para hemorragias y miro-aneurismas.

Dimensión	Formato	Total de imágenes
1440x960	JPEG	80
2912x2912	JPEG	30
1504x1000	JPEG	20
1500x1152	JPG	30
2544x1696	JPG	40

Tabla 5. Cantidad de imágenes de cada resolución en el conjunto de datos utilizado para exudados y drusas.

Dimensión	Formato	Total de imágenes
2544x1696	JPEG	80
776x582	JPEG	30
1440x960	JPEG	40

Posterior a esta clasificación de las imágenes de fondo de ojo de acuerdo al signo presente y sus dimensiones, se agrupan con la idea de que cada modelo pueda detectar 2 signos presentes en la retinopatía hipertensiva. Luego se hace necesario etiquetar mediante la herramienta LabelImg mencionada en la sección 3.3.2 cada una de las imágenes de fondo de ojo para preparar la información para su respectivo entrenamiento en el modelo elegido.

## 4.2 Etiquetado

En esta etapa se procede a realizar el respectivo etiquetamiento de cada imagen de fondo de ojo para obtener el rectángulo delimitador y su respectivo archivo txt en formato YOLO como se muestra en la Figura. 33 y 34.

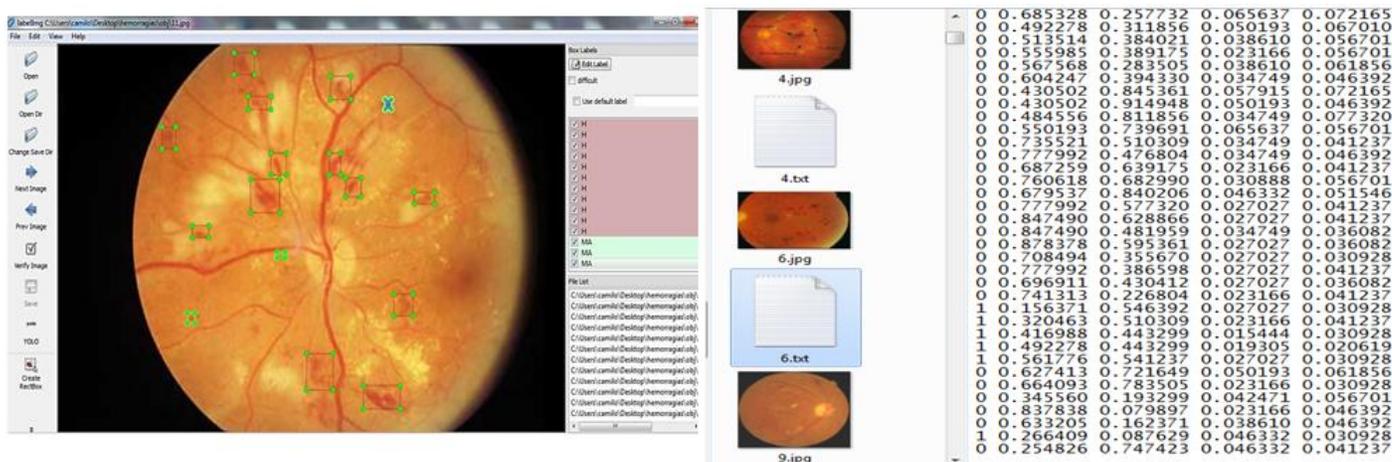
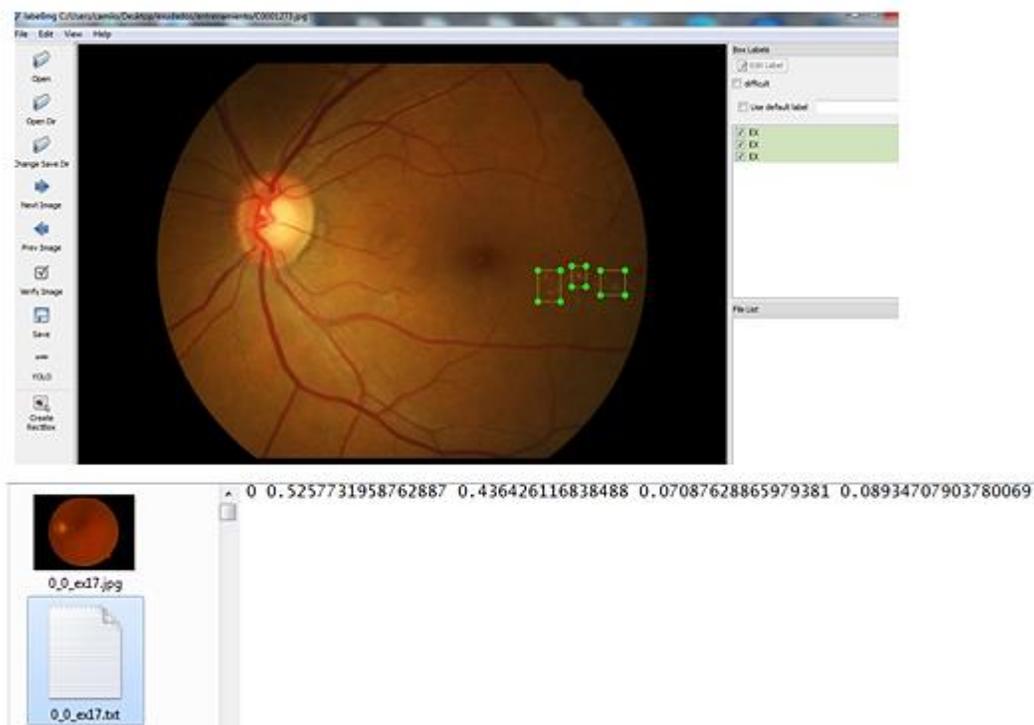


Fig. 33. Etiquetas de hemorragias y micro-aneurismas mediante LabelImg.



**Fig. 34.** Etiquetas de exudados y drusas mediante LabelImg.

### 4.3 Aumento de datos (Data Aumentación)

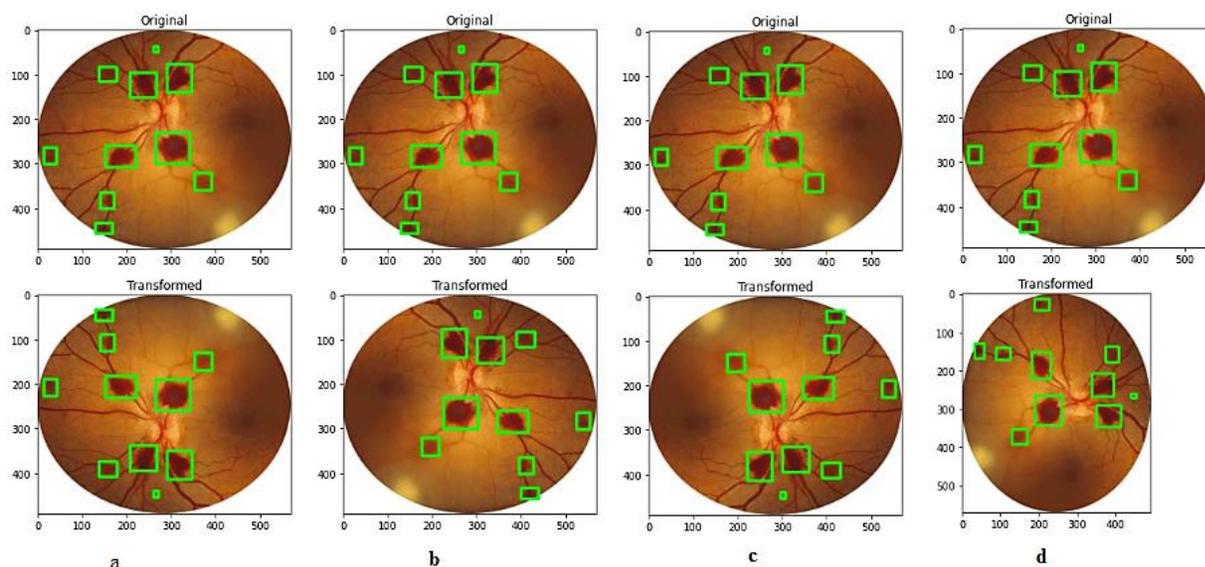
El método de aumento de datos presentado en este documento ha sido implementado en el paquete de código abierto CLODSA. Hemos empleado esta biblioteca para mejorar la precisión de los modelos para la detección de los signos presentes en una imagen de fondo de ojo en la retinopatía hipertensiva.

#### 4.3.1 CLODSA

El procedimiento de aumento de CLODSA [53] es flexible para adaptarse a diferentes necesidades y se basa en seis parámetros: el conjunto de datos de imágenes, el tipo de problema, el modo de anotación de entrada, el modo de anotación de salida, el modo de generación y las técnicas a aplicar.

El conjunto de datos de imágenes viene dado por la ruta donde se encuentran las imágenes; y el tipo de problema. En el caso de la localización de objetos, el primer paso para generar automáticamente la etiqueta a partir de una imagen anotada consiste en generar una máscara a partir del cuadro delimitador anotado. Posteriormente, se aplica la técnica de transformación tanto a la imagen original como a la máscara generada. Luego, a partir de la máscara

transformada, la región blanca simplemente se ubica utilizando las propiedades básicas de los contornos y se obtiene el cuadro delimitador de la región; algunas transformaciones pueden generar un cuadro delimitador muy pequeño o producir una imagen sin ningún cuadro delimitador, ya que estar ubicado fuera de los límites de la imagen; para evitar ese problema, se requiere un porcentaje mínimo para mantener la imagen, es decir, que se mantiene el rectángulo mínimo que contiene el objeto; de lo contrario, la imagen se descarta. Finalmente, la imagen transformada se combina con el cuadro delimitador resultante para obtener la nueva imagen anotada. Este proceso se representa en la Figura. 35.



**Fig. 35** Aplicación de CLODSA; (a) transformación Volteo vertical; (b) transformación volteo horizontal; (c) Transformación volteo vertical y horizontal; (d) transformación Rotación 90°.

Para este proyecto se utilizó aumento de datos mediante la aplicación ya mencionada CLODSA, Por tanto, el número de imágenes con la que se va realizar el entrenamiento tuvo un aumento considerado. Las transformaciones que se aplicaron sobre las 200 imágenes originales para el signo de hemorragias y micro-aneurismas fueron volteos horizontales y verticales aleatorios y rotaciones de 90° como se observa en la fig. 33, con esto se logró un aumento de datos en un total de 708 imágenes para realizar el entrenamiento. Esta misma operación se realizó sobre las 150 imágenes originales con el signo de exudados y drusas, para obtener un total de 800 imágenes para el entrenamiento, como se muestran en tabla 6

Tabla 6. Cantidad de imágenes con aumento de datos para el entrenamiento

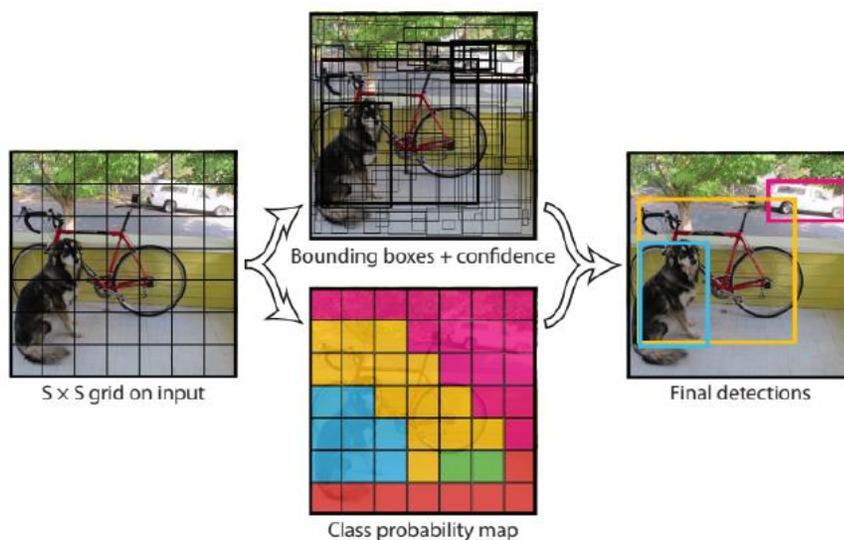
Modelo	Imágenes	Aumento de datos
Signo hemorragias y micro-aneurismas	200	708
Signo exudados y drusas	150	800

#### 4.4 Modelado

El modelo seleccionado para esta investigación es YOLO, ya que los sistemas de detección actuales reutilizan clasificadores para realizar la detección. Para detectar un objeto, estos sistemas toman un clasificador para ese objeto y lo evalúan a lo largo de la imagen en varios lugares y escalas. Enfoques más recientes como el R-CNN utilizan métodos como propuesta de regiones para generar primero las potenciales cajas delimitadoras en una imagen y luego ejecutar un clasificador en cada una de estas cajas propuestas. Después de la clasificación, el post-procesamiento se utiliza para refinar las cajas delimitadoras, eliminar las detecciones duplicadas, y volver a clasificar las cajas en base a otros objetos de la imagen. Estos sistemas son lentos y difíciles de optimizar porque cada componente individual debe ser entrenado por separado.

YOLO, por el contrario, replantea la detección de objetos como un único problema de regresión, directamente de los píxeles de la imagen a las coordenadas de la caja delimitadora y las probabilidades de clase. De esta manera una sola red convolucional predice simultáneamente múltiples cajas delimitadoras y probabilidades de clase para esas cajas. Podemos ver una ilustración de este funcionamiento en la figura 36. YOLO se entrena en imágenes completas y optimiza directamente el rendimiento de la detección. Este modelo unificado tiene varios beneficios sobre los métodos tradicionales de detección de objetos:

- YOLO es extremadamente rápido. Al enfocar la detección como un problema de regresión, no se necesitan sistemas complejos. Simplemente se ejecuta la red neuronal en una nueva imagen para predecir las detecciones. Además, YOLO alcanza más del doble de la precisión media de otros sistemas en tiempo real.
- YOLO trabaja globalmente sobre la imagen. A diferencia de las técnicas de ventana deslizante y propuestas de regiones, YOLO ve toda la imagen durante el tiempo de entrenamiento y prueba, de manera que implícitamente codifica la información contextual sobre las clases, así como su apariencia.



**Fig. 36** Ilustración del funcionamiento de YOLO

De acuerdo con lo anterior en la investigación se propuso trabajar con dos modelos YOLO; YOLOv3 y YOLOv4. Con estos modelos se propusieron 4 experimentos variando su configuración y obteniendo el valor de mAp (Es la media de los AP calculados para todas las clases) para seleccionar el modelo de mejor rendimiento (modelo que es más estable y consistente) de todos los pesos entrenados a través de iteraciones/épocas. Con el valor del mAp se puede comprender si el modelo debe entrenarse o ajustarse más o no.

#### 4.4.1 Configuración de archivos y datos del modelo

Todos los ajustes de configuración que analizamos se basan en una GPU Tesla P100 de 16 GB disponible en Google Colab. Por tanto podemos ajustar las configuraciones según su disponibilidad, pero los experimentos y resultados que discutimos aquí se basan en la configuración según el hardware mencionado. Para los experimentos se tuvieron en cuenta los parámetros que se ven en la tabla 7.

**Tabla 7.** Parámetros usados para el entrenamiento del modelo de red neuronal YOLOv3 y YOLOv4.

<b>img</b>	Define el tamaño de la imagen de entrada, para el entrenamiento. Debe ser múltiplo de 32
<b>bath</b>	Se determina el tamaño del lote de imágenes.
<b>max_batches</b>	Define el número de épocas o iteraciones de entrenamiento. Como mínimo debe cumplir con (# de clases) * 2000.
<b>Steps (pasos)</b>	Se definen a partir del max_batch, es decir, (80% of max_batches), (90% of max_batches).

<b>Filter (filtros)</b>	El número de filtros que se debe aplicar depende del número de clases y se define de la siguiente manera: $\text{filters} = (\# \text{ de clases} + 5) * 3$
<b>data</b>	Establece la ruta al archivo data y name donde esta las rutas y clases.
<b>cfg:</b>	Especifica la configuración del modelo seleccionado.
<b>Random</b>	Este permite ahorrar memoria en el entrenamiento; si es igual a 1 el variará el tamaño de los lotes durante el entrenamiento, lo que hace que demore el entrenamiento, mientras que si es cero tendremos un lote de entrada fija y esto reduce memoria y acelera el entrenamiento.

Para nuestros primeros 4 experimentos se trabaja con la arquitectura YOLOv3. En el archivo de configuración Darknet53 (cfg), se cambió el lote de 64 a 32, se estableció max\_batches en 7000 y los pasos en 5600, 6300. Básicamente, entrenaremos el modelo para 7000 iteraciones con un tamaño de lote de 32. Las tasas de aprendizaje se programarán para reducirse en los pasos 5600 y 6300. Luego procede a determinar el número de filtros dependiendo de la cantidad de clases que en este caso serán 2 ya que cada modelo detectará dos signos. Yolo presenta capas convolucionales que contiene parámetros para los filtros, en este caso el número de filtros se determina mediante  $\text{Filtros} = (\text{num\_classes} + 5) * 3$ , que Para nuestro proyecto que se utilizaran 2 clases tendremos 21 filtros como se indica en la Fig. 37

```

1 [net]
2 # Testing
3 #batch=1
4 #subdivisions=1
5 # Training
6 batch=32
7 subdivisions=16
8 width=416
9 height=416
0 channels=3
1 momentum=0.9
2 decay=0.0005
3 angle=0
4 saturation = 1.5
5 exposure = 1.5
6 hue=.1
7
8 learning_rate=0.001
9 burn_in=1000
0 max_batches = 7000
1 policy=steps
2 steps=5600,6300
3 scales=.1,.1

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

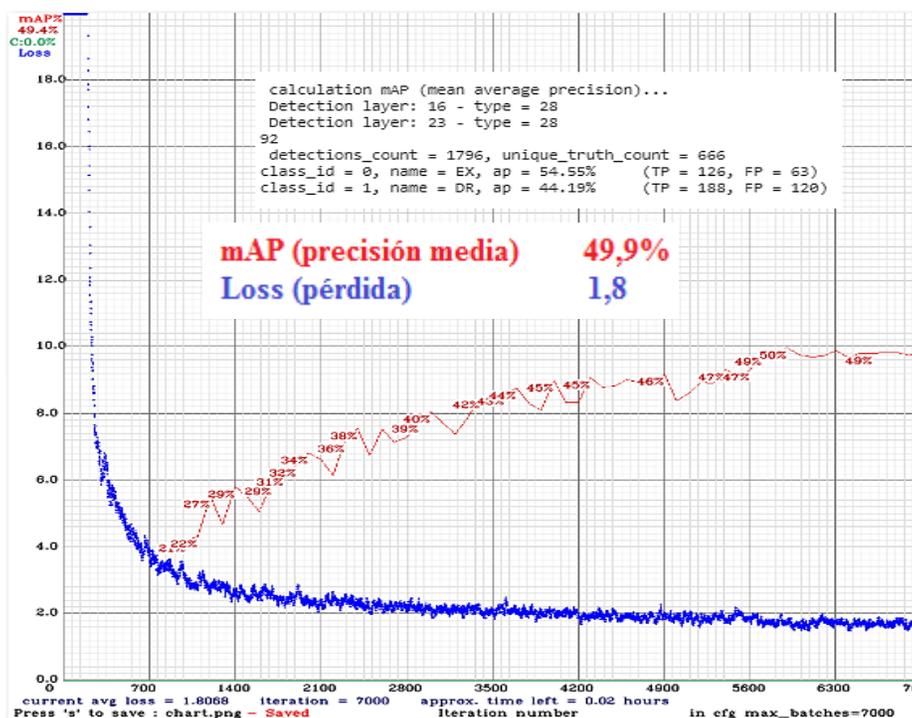
[convolutional]
size=1
stride=1
pad=1
filters=21
activation=linear

[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

```

**Fig. 37** configuraciones para el primer experimento con la versión pequeña de Yolo (yolov3\_tiny) en resolución fija 416x416.

El entrenamiento del experimento 1 vemos que nos arroja un mAp de 49,9% durante 7000 iteraciones y una pérdida de 1,8 como se muestra en la figura 38.



**Fig. 38** Resultado del mAp durante 7000 iteraciones para el primer experimento con la versión pequeña de Yolo (yolov3\_tiny) en resolución fija 416x416.

Para el experimento dos tendremos una resolución múltiple, para este caso Darknet proporciona un parámetro aleatorio, entonces en los archivos de configuración del modelo pequeño de yolov3, el valor predeterminado es de random es 0, lo que indica que no se utilizará ninguna resolución aleatoria (o resolución múltiple) durante el entrenamiento. En este experimento cambiaremos ese valor para que exista la resolución aleatoria, es decir cambiaremos el valor de 0 a 1 en el archivo de configuración por random=1. El resto de configuraciones y parámetros serán los mismos que en el entrenamiento anterior, es decir, entrenamiento del modelo YOLOv3-Tiny de resolución fija como se observa en la Fig.39

```

[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

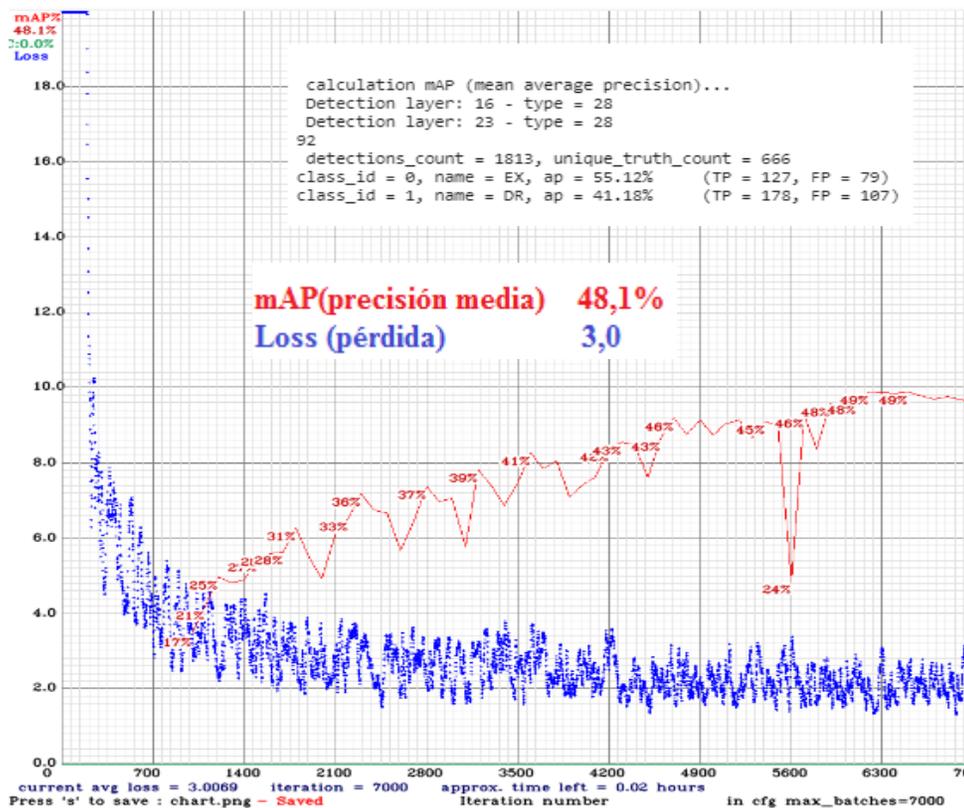
[convolutional]
size=1
stride=1
pad=1
filters=21
activation=linear

[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

```

**Fig. 39** configuraciones para el segundo experimento con la versión pequeña de Yolo (yolov3\_tiny) en multiresolución.

El entrenamiento del experimento 2 vemos que nos arroja un mAp de 48,1% durante 7000 iteraciones, al final del entrenamiento, la pérdida de 3, que es mucho mayor que en el experimento1 con el entrenamiento de resolución única. Ahora, esto es de esperar ya que los datos de entrenamiento se vuelven difíciles cuando se entrena en imágenes más pequeñas. Pero al mismo tiempo, el modelo vio muchos escenarios variados, pero teniendo un mAp cercano al del experimento 1 como podemos ver en la figura 40.



**Fig. 40** Resultado del mAp durante 7000 iteraciones para el segundo experimento con la versión pequeña de Yolo (yolov3\_tiny) en multiresolución.

Para el experimento tres, algunas de las configuraciones del experimento 2 serán igual, lo que se cambia es que ya no se utiliza la versión pequeña de YOLOv3\_tiny, sino la versión grande, es decir YOLOV3. El ancho y la altura se configuran en 608, es decir tendremos una resolución base en  $608 \times 608$ , el lote y la subdivisión serán de 32, el max\_batches será de 7000 y los pasos son 5600, 6300. Estos siguen siendo los mismos que en el caso del entrenamiento de los modelos pequeños, lo demás sigue igual al experimento 2 como muestra la Fig 41.

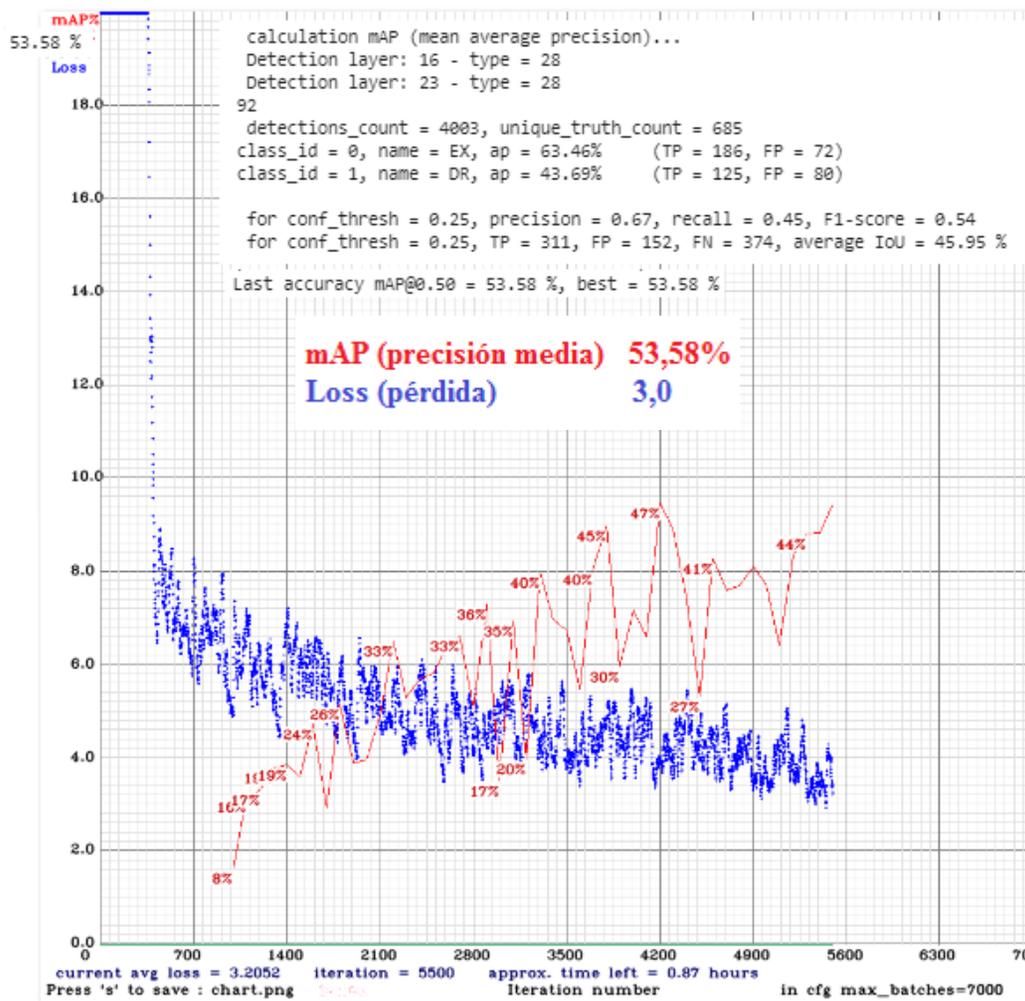
```

2 # Testing
3 #batch=1
4 #subdivisions=1
5 # Training
6 batch=32
7 subdivisions=32
8 width=608
9 height=608
10 channels=3
11 momentum=0.9
12 decay=0.0005
13 angle=0
14 saturation = 1.5
15 exposure = 1.5
16 hue=.1
17
18 learning_rate=0.001
19 burn_in=1000
20 max_batches = 7000
21 policy=steps
22 steps=5600,6300
23 scales=.1,.1
24
25 [convolutional]
26 batch_normalize=1
27 filters=16
28
29 [convolutional]
30 batch_normalize=1
31 filters=256
32 size=3
33 stride=1
34 pad=1
35 activation=leaky
36
37 [convolutional]
38 size=1
39 stride=1
40 pad=1
41 filters=21
42 activation=linear
43
44 [yolo]
45 mask = 1,2,3
46 anchors = 10,14, 23,27, 37,58, 81,82, 135,169, 344,319
47 classes=2
48 num=6
49 jitter=.3
50 ignore_thresh = .7
51 truth_thresh = 1
52 random=1

```

**Fig. 41** configuraciones para el tercer experimento con la versión grande de YOLOv3 en multiresolución.

El entrenamiento del experimento 3 vemos que nos arroja un mAp de 53,58 % durante 7000 iteraciones, al final del entrenamiento, la pérdida de 3, similar al del experimento 2, que sigue siendo mayor que en el experimento 1 con el entrenamiento de resolución única. Ahora, esto es de esperar ya que los datos de entrenamiento se vuelven difíciles cuando se entrena en imágenes más pequeñas. Pero al mismo tiempo, el modelo vio muchos escenarios variados, pero teniendo un mAp mejor que los experimentos 1 y 2, además de que la versión YOLOv3 grande presenta 53 capas mientras que la versión pequeña YOLOv3\_tiny solo utiliza 13 de estas capas, es de esperar que el resultado del mAp sea mucho mejor. Esto se puede ver en la figura 42.



**Fig. 42** Resultado del mAp durante 7000 iteraciones para el tercer experimento con la versión grande de YOLOv3 en multiresolución.

En el experimento 4 será de resolución fija, lo único que cambiara en su configuración será que el valor de random sea cero, es decir no será aleatorio y tendremos un lote fijo como muestra la figura 43.

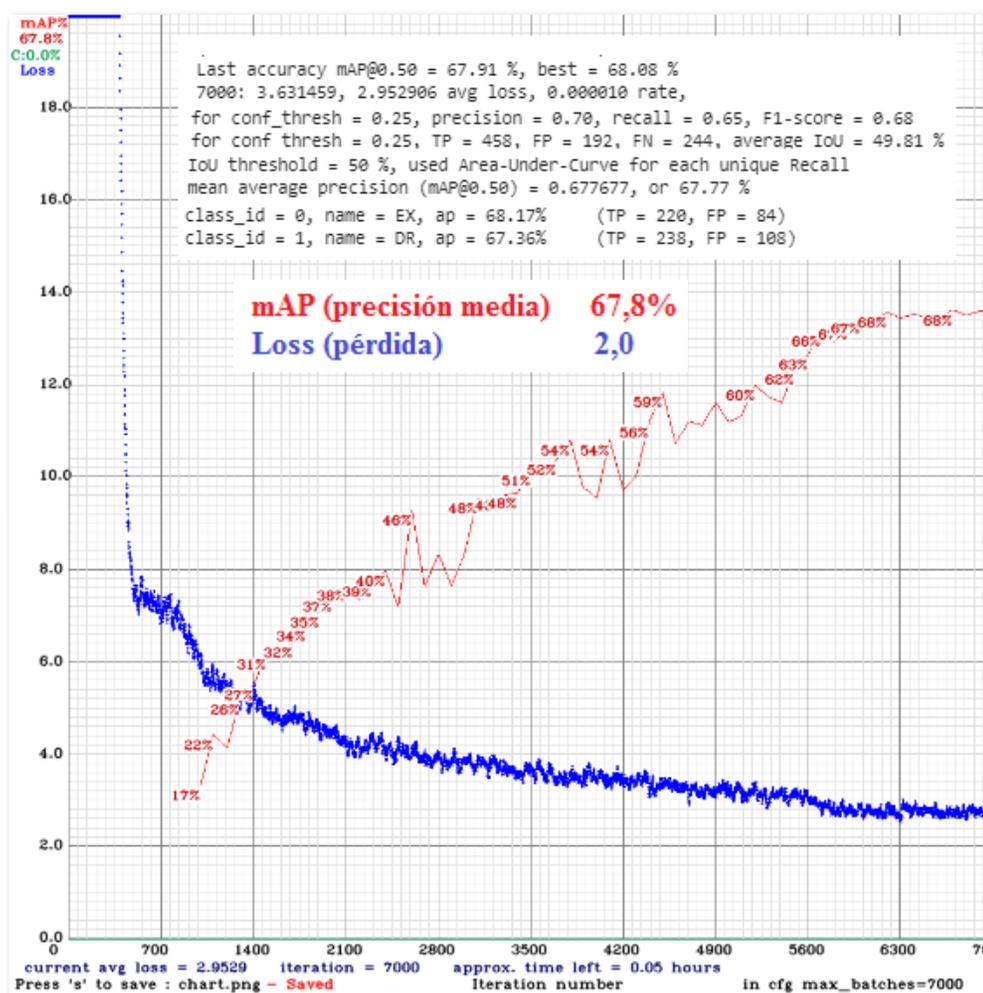
```

2 # Testing
3 #batch=1
4 #subdivisions=1
5 # Training
6 batch=32
7 subdivisions=32
8 width=608
9 height=608
0 channels=3
1 momentum=0.9
2 decay=0.0005
3 angle=0
4 saturation = 1.5
5 exposure = 1.5
6 hue=.1
7
8 learning_rate=0.001
9 burn_in=1000
0 max_batches = 7000
1 policy=steps
2 steps=5600,6300
3 scales=.1,.1
4
5 [convolutional]
6 batch_normalize=1
7 filters=16
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59 [convolutional]
60 batch_normalize=1
61 filters=256
62 size=3
63 stride=1
64 pad=1
65 activation=leaky
66
67 [convolutional]
68 size=1
69 stride=1
70 pad=1
71 filters=21
72 activation=linear
73
74 [yolo]
75 mask = 1,2,3
76 anchors = 10,14, 23,27, 37,58, 81,82, 135,169, 344,319
77 classes=2
78 num=6
79 jitter=.3
80 ignore_thresh = .7
81 truth_thresh = 1
82 random=0

```

**Fig. 43** configuraciones para el cuarto experimento con la versión grande de Yolov3 en resolución fija 608x608.

El entrenamiento del experimento 4 vemos que nos arroja un mAp de 67,08 % durante 7000 iteraciones, al final del entrenamiento, la pérdida de 2, que es mucho mejor que el experimento 2 y 3, El mAp es mucho mejor que los experimentos 1,2 y 3, esto nos permite ver que si se prolongan más iteraciones con esta configuración se puede obtener muy buenos resultados. Esto se puede ver en la figura 44.

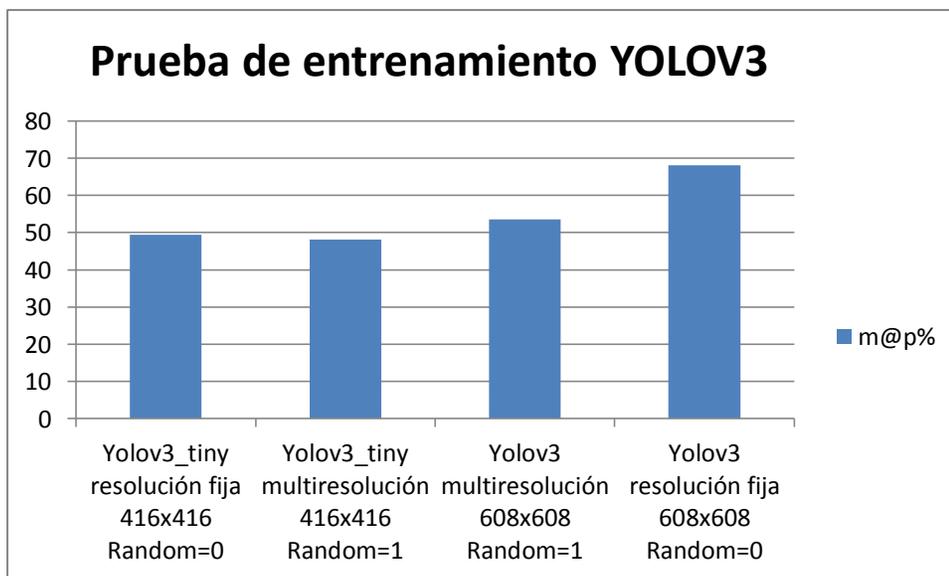


**Fig. 44** Resultado del mAp durante 7000 iteraciones para el cuarto experimento con la versión grande de Yolov3 en resolución fija 608x608.

Estos resultados con YOLOv3 se organizan en una tabla para poder visualizar mejor como se ve en la tabla 8 y en la figura 45.

**Tabla 8. Resultados de experimentos con YOLOv3.**

Versión Yolo	Yolov3_tiny resolución fija 416x416 Random=0	Yolov3_tiny multiresolución 416x416 Random=1	Yolov3 multiresolución 608x608 Random=1	Yolov3 resolución fija 608x608 Random=0
mAp%	49,4	48,1	53,58	68,08

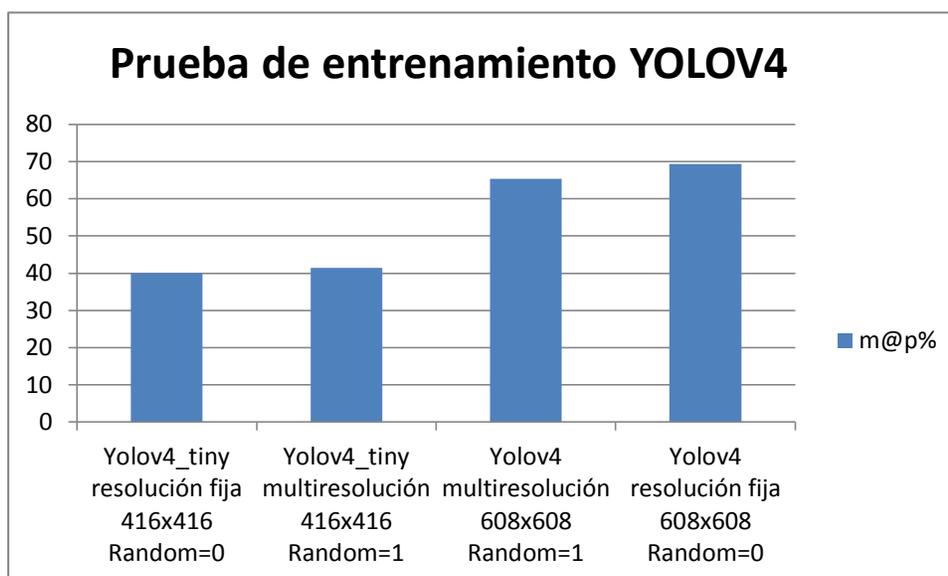


**Fig. 45** Resultado de los experimentos con YOLOv3.

Con la arquitectura YOLOv4 que es la versión actualizada de YOLOv3 se realizaron los mismos experimentos con el fin de observar que tanto mejora el mAP con respecto a YOLOv3. Estos resultados se muestran en la tabla 9 y la figura 46.

**Tabla 9. Resultados de experimentos con YOLOv4.**

Versión Yolo	Yolov4_tiny resolución fija 416x416 Random=0	Yolov4_tiny multiresolución 416x416 Random=1	Yolov4 multiresolución 608x608 Random=1	Yolov4 resolución fija 608x608 Random=0
mAP%	40.02	41.50	65.32	69.34



**Fig. 46 Resultado de los experimentos con YOLOv4.**

Estos experimentos se realizaron para seleccionar el modelo de mejor rendimiento (modelo que es más estable y consistente), después de ver resultados en este proyectos se utilizará la configuración del experimento 4 con YOLOv4 que fue el que mejor rendimiento tuvo.

#### 4.4.2 Entrenamiento del modelo

En esta etapa se explica que el entrenamiento es el proceso por el que el sistema va consumiendo los datos de entrada, para que pueda ir obteniendo de cada entrada los objetos que se desean detectar mediante la descomposición que hemos comentado antes de las imágenes. Con ello, se permitirá que el sistema aprenda el objeto que deseamos que detecte y que así se pueda usar para reconocimiento del mismo de manera automática.

##### 4.4.2.1 Entradas

El primer paso que se debe tener es la preparación de los datos que se habló en la sección 3.3 En este punto hay que destacar que algunas redes utilizadas en la detección de objetos, como es el caso del sistema YOLO, funcionan con un tamaño de imagen determinado (tamaño múltiplo de 32). Esto es que de manera óptima las imágenes de entrada se deben de adaptar al tamaño que requiere el sistema por ahorro de costo computacional. Para la recolección de los datos se utilizaron bases de datos libres mencionadas en la sección 3.2.1 y se organizan en carpetas por cada 2 signos que se quieren detectar como se habla en la sección 4.1.

El siguiente paso es etiquetar la imagen puesto que en la detección de objetos es necesario indicar al sistema dónde está localizado el objeto en la imagen, para que este sea capaz de aprender dicho objeto. Esto se menciona en la sección 4.2, y posterior a ello se realiza aumento de datos ya que el modelo requiere una mayor cantidad de imágenes para el entrenamiento, esto se discute en la sección 4.3.

Tomando ya las entradas como hemos explicado anteriormente, lo que se hace es en primer lugar dividir el conjunto de datos en entrenamiento, validación y test para los dos modelos como se muestra en la tabla 10. Para los modelo de Hemorragias (H); Micro-aneurismas (MA); Drusas (DR) y Exudados (EX)

**Tabla 10.** Distribución de entrenamiento, validación y test

Modelo	Entrenamiento	Validación	Test
Signo hemorragias y micro-aneurismas	602	71	35
Signo exudados y drusas	680	80	40

El modelo elegido en el proyecto es YOLOv4 en su versión grande por las razones mencionadas en la sección 4.3.1. Posteriormente se configura YOLOv4 de acuerdo a la sección 4.3.1 en el cuarto experimento que fue donde tuvo mejores resultados. Esta configuración se puede ver en la figura 44.

Las herramientas utilizadas para el entrenamiento y posterior visualización de los resultados se pueden ver en la tabla 11.

**Tabla 11.** Herramientas utilizadas para el entrenamiento

GOOGLE COLAB	LABELIMAGE	YOLO	PYTHON
Permite ejecutar y programar en Python sin configuración	Para la distribución de las etiquetas de las imágenes.	Es un algoritmo de detección de objetos.	Lenguaje de programación utilizado en el proyecto

Cabe resaltar que en este proyecto se utiliza transfer learning que es un método de aprendizaje de maquina donde un modelo diseñado para una tarea es reusado como el punto de partida para otra tarea de visión computacional y procesamiento de lenguaje natural. Dicho lo anterior se hace uso de esta técnica para el entrenamiento de nuestro modelo. El modelo utilizado es yolov4.conv.137 el cual es capaz de predecir 80 clases en una imagen o video, este modelo pre-entrenado nos servirá como punto partida para entrenar nuestro propio modelo.

Los modelos se entrenaron durante diferentes iteraciones, obteniendo una precisión media del 80% y 88,45%, donde los modelos se estabilizaron durante el entrenamiento, como se evidencia en la tabla 12 y la figuras 47 y 48

**Tabla 12. Resultados del entrenamiento de los modelos en diferentes iteraciones**

Modelo	Iteraciones	%mAP
Hemorragias (H) y micro-aneurismas (MA)	14700	80%
Exudados (EX) y drusas (DR)	20000	89%

```
(next mAP calculation at 14700 iterations)
Last accuracy mAP@0.50 = 74.61 %, best = 79.98 %
  calculation mAP (mean average precision)...
  Detection layer: 16 - type = 28
  Detection layer: 23 - type = 28
72
detections_count = 657, unique_truth_count = 532
class_id = 0, name = H, ap = 93.96%      (TP = 104, FP = 4)
class_id = 1, name = MA, ap = 60.79%    (TP = 150, FP = 7)

for conf_thresh = 0.25, precision = 0.96, recall = 0.48, F1-score = 0.64
for conf_thresh = 0.25, TP = 254, FP = 11, FN = 278, average IoU = 71.01 %
```

**Fig. 47 % mAP Hemorragias (H) y Micro-Aneurismas (MA)**

```

calculation mAP (mean average precision)...
Detection layer: 16 - type = 28
Detection layer: 23 - type = 28
92
detections_count = 1293, unique_truth_count = 816
class_id = 0, name = EX, ap = 85.35%    (TP = 288, FP = 96)
class_id = 1, name = DR, ap = 91.63%    (TP = 412, FP = 92)

for conf_thresh = 0.25, precision = 0.79, recall = 0.86, F1-score = 0.82
for conf_thresh = 0.25, TP = 700, FP = 188, FN = 116, average IoU = 60.69 %

IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
mean average precision (mAP@0.50) = 0.884906, or 88.49 %
Total Detection Time: 4 Seconds

```

**Fig. 48** % mAP Exudados (EX) y Drusas (DR)

#### 4.5 Detección con el modelo entrenado

Con el modelo entrenado se realizaron pruebas en el algoritmo, donde se le pasa la configuración y los pesos entrenados para visualizar las detecciones que se esperan de cada modelo. De acuerdo a la cantidad de signos presentes en la imagen de fondo de ojo (ver tabla 13 ) se le asocia una categoría de Retinopatía leve, moderada y severa como muestra las figuras que se describen a continuación: En la figura 49 tenemos la detección de Hemorragias; En la figura 50 tenemos la detección de los Micro-aneurismas; en la figura 51 tenemos la detección de Hemorragias y micro-Aneurismas; en la figura 52 no tenemos detecciones; en la figura 53 vemos la detección de Exudados y en la figura 54 tenemos la detección de Drusas, cada detección con sus respectivas confianzas.

Tabla 13. Estatificación de retinopatía hipertensiva de acuerdo a cantidad de signos detectados

Equivalencia	Clasificación internacional	Clasificación de la SERV y de la SEO.
Solo MA o solo EX	Retinopatía Leve	Mayor e igual a 5 MA y H
MA y H	Retinopatía Moderada	Menos de 20 MA y H
MA, H, EX y DR	Retinopatía Severa	Mayor e igual de 20 MA, H, EX o DR

SERV: (Sociedad Española de Retina y Vítreo); SEO: (Sociedad Española de Oftalmología). [1]

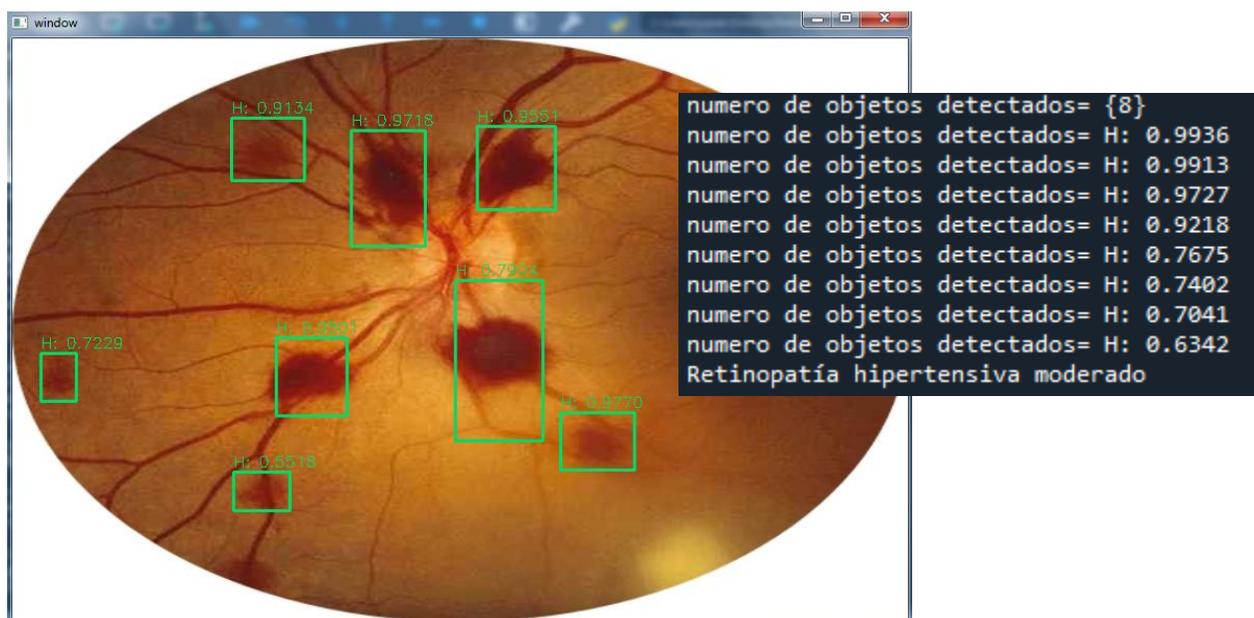


Fig 49. Detección de Hemorragias con el algoritmo propuesto en Python



Fig 50. Detección de Micro-Aneurismas con el algoritmo en Python



Fig 51. Detección de Hemorragias y Micro-aneurismas



Fig 52. Sin detecciones de hemorragias ni micro-aneurismas



Fig 53. Detección de Exudados con el algoritmo en Python

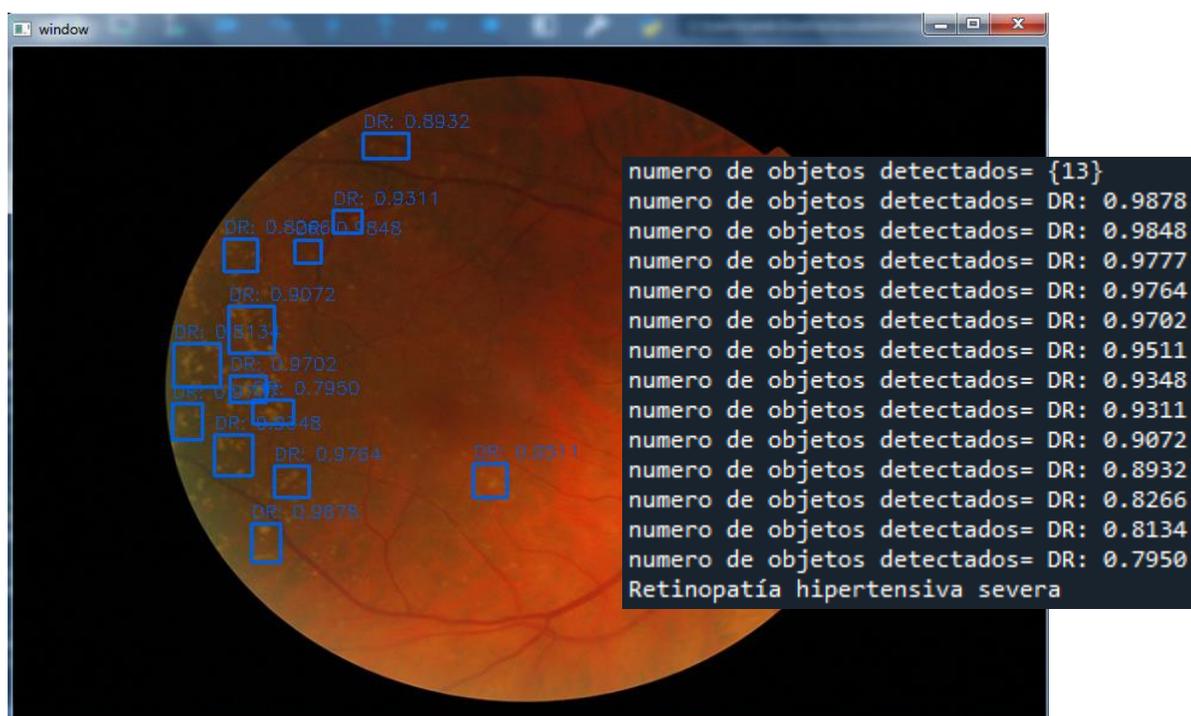


Fig 54. Detección de Drusas con el algoritmo en Python

Además, se realizó la prueba en términos del tiempo de interpretación del examen de fondo de ojo por parte del oftalmólogo con y sin ayuda del algoritmo propuesto, como se puede ver en la tabla 14. Donde el algoritmo fue capaz de reducir el tiempo promedio de interpretación en la prueba realizada aproximadamente un 25%, es decir en 2 minutos.

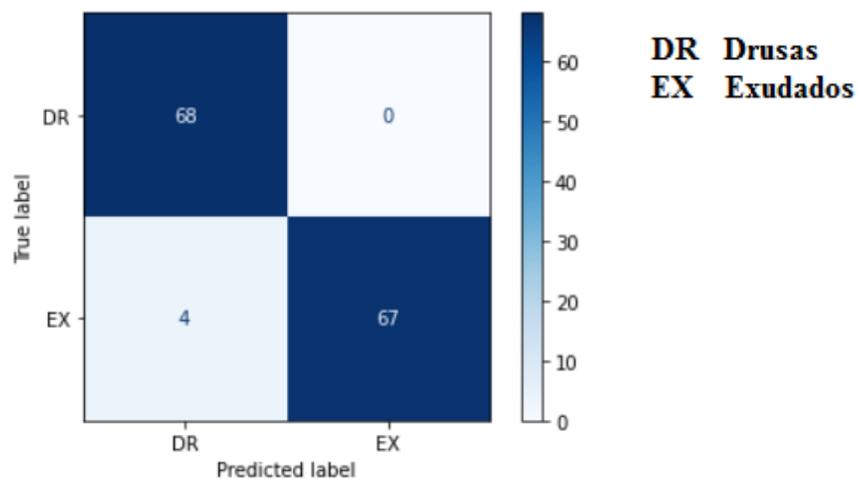
**Tabla 14.** Evaluación del tiempo de análisis de un examen de fondo de ojo

No	Tiempo oftalmólogo (Minutos)	Tiempo del algoritmo (Minutos)
1	2	0.007
2	3	1
3	4	1
4	4	1
5	4	1
6	3	0.005
7	3	1
8	4	1
9	3	0.5
10	4	0.5
promedio	3	0,70

#### 4.6 Evaluación del modelado

De acuerdo a los resultados mencionados en la sección 4.4 se procede a evaluar los modelos mediante las métricas correspondientes a la detección de objetos. Finalmente, recordando que en aprendizaje de maquina el mayor desafío es obtener una buena precisión en un conjunto de datos no visto por el modelo, se utilizó otro set de testeo para calcular la curva ROC utilizando 525 imágenes de la base de datos mencionados en la sección 3.2.

Para el modelo 1 que es detección de exudados y drusas tenemos la matriz de confusión en la figura 55, las métricas de evaluación en la figura 56 y la curva ROC en la figura 57

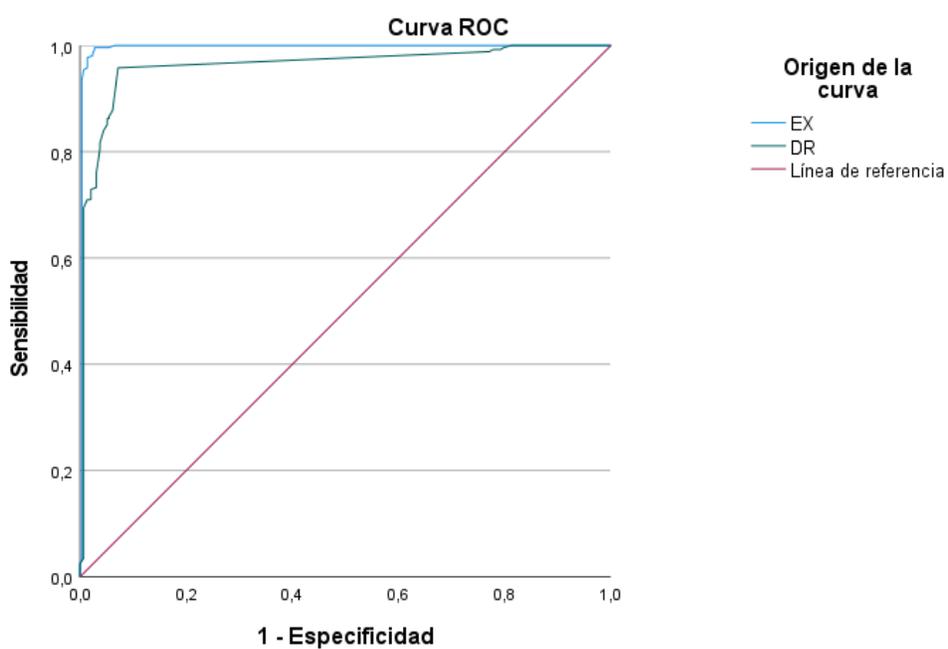


**Fig.55** Matriz de confusión del modelo 1

Analizando modelo #1

	precision	recall	f1-score	support
DR	0.94	1.00	0.97	68
EX	1.00	0.94	0.97	71
accuracy			0.97	139
macro avg	0.97	0.97	0.97	139
weighted avg	0.97	0.97	0.97	139

**Fig.56** Métricas de evaluación del modelo 1

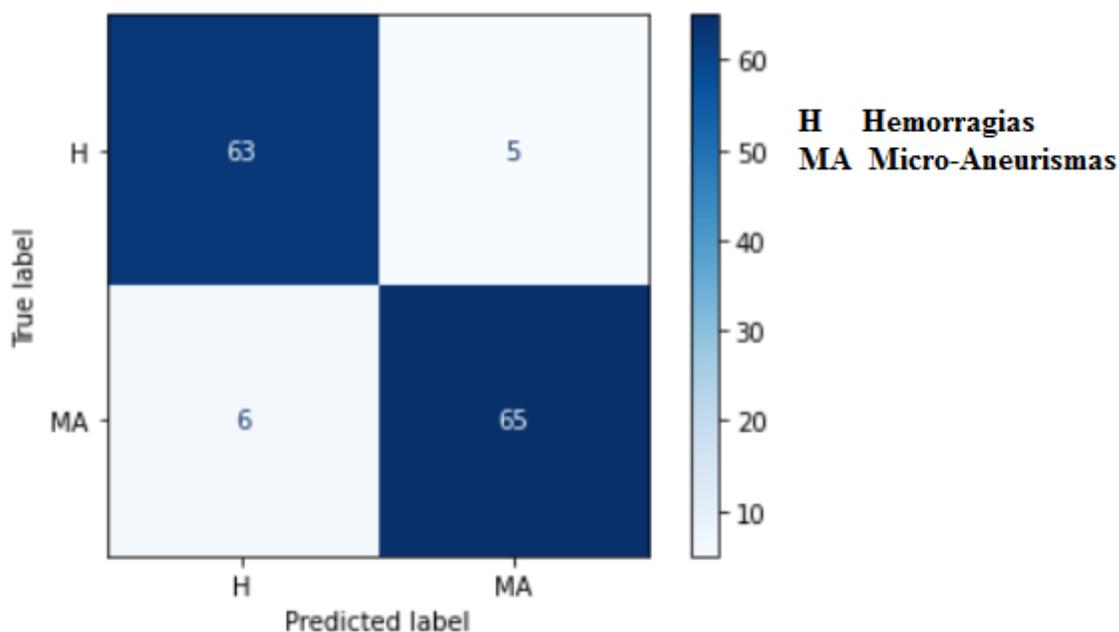


**Área bajo la curva**

Variables de resultado de prueba	Área	Desv. Error <sup>a</sup>	Significación asintótica <sup>b</sup>	95% de intervalo de confianza asintótico	
				Límite inferior	Límite superior
EX	,996	,003	,000	,989	1,000
DR	,962	,009	,000	,944	,979

**Fig.57** Curva ROC del modelo 1

Para el modelo 2 que es detección de Hemorragias y micro-Aneurisma tenemos la matriz de confusión en la figura 58, las métricas de evaluación en la figura 59 y la curva ROC en la figura 60.

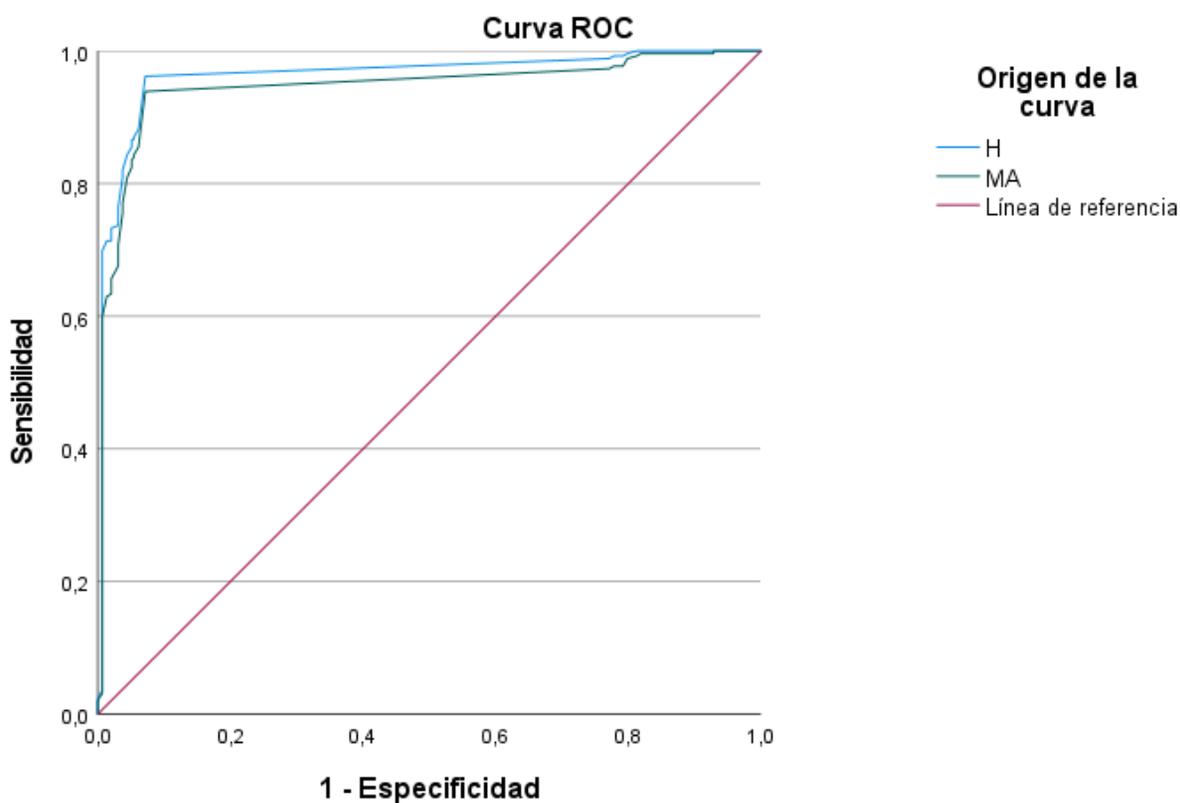


**Fig.58** Matriz de confusión del modelo 2

Analizando modelo #1

	precision	recall	f1-score	support
H	0.91	0.93	0.92	68
MA	0.93	0.92	0.92	71
accuracy			0.92	139
macro avg	0.92	0.92	0.92	139
weighted avg	0.92	0.92	0.92	139

**Fig.59** Métricas de evaluación del modelo 2



**Área bajo la curva**

Variables de resultado de prueba	Área	Desv. Error <sup>a</sup>	Significación asintótica <sup>b</sup>	95% de intervalo de confianza asintótico	
				Límite inferior	Límite superior
H	,963	,009	,000	,946	,980
MA	,946	,011	,000	,924	,967

**Fig.60** Curva ROC del modelo 2

## 4.7 Discusión

Luego de validar los dos modelos de detección y haber revisado diferentes métricas y evaluado su comportamiento en los datos de entrenamiento, validación y testeo, es claro que para la arquitectura utilizada YOLOv4 y la configuración de Resolución fija que se evidenció en el experimento 4 en la sección 4.4.1 y mostrada en la Fig.46 se obtuvieron mejores resultados comparado con la versión de YOLOv3.

Para medir la efectividad de ambos modelos se utilizó la matriz de confusión como se observan en las figuras 55 y 58, que son medida del rendimiento para la detección de objetos en aprendizaje de máquina. Para el modelo 1, el cual realiza detección de drusas y exudados, tenemos que los verdaderos negativos son 67 y los verdaderos positivos 68, es decir, de los 135 casos (diagonal) de un total de 139 imágenes tomadas del testeo, predijo correctamente las categorías de cada clase. Por otra parte en el modelo 2, el cual realiza detección de hemorragias y micro-aneurismas tenemos que los verdaderos negativos son 65 y los verdaderos positivos 63, es decir, de los 128 casos (diagonal) de un total de 139 tomadas del testeo, predijo correctamente las categorías de cada clase.

Evaluando los resultados de ambos modelos en la matriz de confusión se calcula las métricas de precisión, sensibilidad y F1-score, como se observa en la figuras 56 y 59. Para el modelo 1, se tiene un recall alto para drusas de un valor 1.0 y para exudados de 0.94 y una precisión de 0.94 para drusas y 1.0 para exudados, esto significa que el modelo 1 detecta bien las clases y un F1-score del 97% que corrobora el buen rendimiento del modelo 1. Por otra parte, el modelo 2 cuenta con resultados altos en precisión y recall como se observa en la figura 59, indicando que el modelo en la mayoría de las veces maneja bien la detección de las clases y un F1-score del 92% que respalda un buen rendimiento del modelo.

Evaluando los resultados del AUC en la curva ROC como se muestra en las figuras 57 y 60, se puede observar que el modelo 1 presento un AUC de 0,99 para EX y 0,96 para DR, mientras que el modelo 2 presento un AUC de 0,96 para H y 0,94 para MA, obteniendo estos dos modelos los mejores resultados permitiendo así poder detectar de manera eficiente los signos para los cuales fueron entrenados. Esto es sinónimo de robustez de los modelos, al contar con un gran número de imágenes y un modelo previamente preentrenado.

## CAPÍTULO 5

### 5 CONCLUSIONES Y TRABAJO FUTURO

#### 5.1 Conclusiones

Los resultados obtenidos en este trabajo muestran un panorama alentador en el propósito de detectar signos de retinopatía hipertensiva, además, el algoritmo de aprendizaje automático implementado muestra resultados óptimos que para el especialista pueden ser como usados como una alternativa en la toma de decisiones frente a la patología ocular mencionada en esta investigación. La investigación permitió extraer las siguientes conclusiones:

- El algoritmo de aprendizaje automático implementados en conjunto con el método propuesto de procesamiento de las imágenes de fondo de ojo fue capaz de detectar 4 signos presentes en la retinopatía hipertensiva los cuales son Hemorragias, Micro-Aneurismas, Exudados y drusas con AUC del 0,99 y 0,96. Permitiendo así ser una alternativa en la asistencia del diagnóstico al especialista.
- De acuerdo con los resultados de la investigación, el algoritmo de acuerdo a la cantidad de detecciones realizadas es capaz de identificar la patología ocular de manera leve, moderada y severa. (ver sección 4.4)
- La recolección del conjunto de datos es una de las etapas más importantes y se debe hacer énfasis en la correcta, recolección de las imágenes y de su respectiva anotación. Se debe hacer énfasis en el preprocesamiento de las imágenes puesto que es fundamental para obtener buenos resultados en consecuencia las fases de comprensión de los datos y preparación de los datos llevabas a cabo en este trabajo por medio de la extracción de características fueron exitosas ya que se obtuvieron buenos resultados en la evaluación de la clasificación de la detección de signos de retinopatía hipertensiva.
- Se implementó dos versiones del algoritmo de detección YOLO; YOLOv3 y YOLOv4 con diferentes configuraciones las cuales permitieron elegir la arquitectura YOLOv4 para esta investigación. donde el enfoque presentó valores elevados en todos los índices: sensibilidad, exactitud, precisión, puntaje F1 y AUC.

## 5.2 Recomendaciones

El trabajo realizado deja una serie de recomendaciones que se despliegan a continuación

- Se recomienda utilizar otros tipos de arquitecturas de detección que permitan comparar los resultados en las métricas de evaluación y determinar cuánto difieren.
- Se recomienda que en la recolección de las imágenes, sean de entorno clínico, además que con ayuda del especialista puedan ser etiquetadas, con el fin de minimizar el margen de error.
- Se pueden utilizar en la evaluación bases de datos públicos como Drive y Store que vienen con anotaciones que permiten realizar comparaciones entorno a un diagnóstico dado por un especialista.

## 5.3 Trabajo Futuro

El trabajo realizado deja una serie de líneas abiertas para seguir investigando:

- Se puede implementar el modelo en un servidor web que permita mediante una plataforma poder interactuar con el usuario, ya que se espera que el método propuesto tenga un gran impacto en el apoyo al diagnóstico al especialista. y, por lo tanto, aumentar la posibilidad de que los pacientes tengan un diagnóstico oportuno en la patología ocular llamada Retinopatía Hipertensiva
- Incluir imágenes de pacientes de un entorno clínico que permita alimentar la base de datos para el posterior entrenamiento del algoritmo de detección.
- Desarrollar los entrenamientos en una GPU de alto rendimiento y que no estén limitados en recursos de uso.
- Realizar el entrenamiento con modelos más recientes de YOLO, como YOLOv5 o YOLOv6, para mejorar el porcentaje de la precisión media mAP en esta investigación.

## 6. Bibliografía

- [1] Romero-Aroca, R.S. “la retinopatía diabética e hipertensiva”, Colegio oficial de médicos de córdoba (comcordoba) AMF-2018
- [2] Daniela Gasca Cuello. P. Manifestaciones de la retinopatía hipertensiva y de la retinopatía diabética en población adulta. *Scientific & Education Medical Journal / Vol. 1, N° 1, 2021, 9.*
- [3] Tsukikawa M, W Stacey A. A Review of Hypertensive Retinopathy and Chorioretinopathy; 12: 67-73. 2020
- [4] 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. 2020; 17 (1): 435-458*
- [5] Isabella Torres Revelo, U. A. Interpretabilidad De Un Modelo Basado En Aprendizaje Profundo Para El Diagnóstico De Retinopatía Diabética. Cali. (2020).
- [6] OMS, Organización mundial de la salud : ``hipertensión`` 17 de mayo del 2021: online: <https://www.who.int/es/news-room/fact-sheets/detail/hypertension>
- [7] MinSalud, Ministerio de Salud y protección Social de Colombia: “Conoce tus números” 14 de Junio del 2020: online: <https://www.minsalud.gov.co/Paginas/Conoce-tus-numeros-para-prevenir-la-hipertension-arterial.aspx>
- [8] ANORO, A. T. Retinopatía hipertensiva. *Medicina General II*, Revisión. 2020; 524-564
- [9] 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. Guayaquil. (2018).
- [10] 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. 2020; 17 (1): 435-458*
- [11] Transferida del libro: Quiroz G., Anatomía Humana, Editorial Porrua, 2010
- [12] Fuentes, S. M. Estudio de métodos para identificar signos de retinopatía diabética en imágenes de fondo del ojo. (Maestría en Ciencias en Ciencias de la Computación Proyecto), Centro de Investigación Científica; (2018).

- [13] Murray, T. G. Anatomía Del Ojo ONCOLOGÍA OCULAR (Vol. 2018). Florida, Estados Unidos. (2012).
- [14] Alcon Cusí, S. A. El ojo humano, nuestra conexión con el mundo. Gente con Vista. Scientific & Education Medical Journal / Vol. 1, N° 1, 2021.
- [15] Spaide RF, Miller-Rivero NE. Anatomy. In Spaide RF, ed. Diseases of the Retina and Vitreous. Philadelphia: Saunders; 1999.
- [16] American academy of ophthalmology. El Nervio óptico. *Ojos sanos*. (2016). Online: <https://www.aaof.org/salud-ocular/anatomia/nervio-optico>
- [17] Enrique Gradue Weicher, Oftalmología en la práctica de la medicina general, Online: <https://accessmedicina.mhmedical.com/content.aspx?bookid=1498&sectionid=99962415>
- [18] Ocularis, proyecto divulgativo sobre la visión, Online: <https://ocularis.es/retinopatia-diabetica-i-conceptos/>
- [19] Mookiah, M. R. K., Acharya, U. R., Chua, C. K., Lim, C. M., Ng, E. Y. K., & Laude, A. (2013). Computer-aided diagnosis of diabetic retinopathy: A review. *Computers in Biology and Medicine*, 43(12), 2136–2155. <https://doi.org/https://doi.org/10.1016/j.combiomed.2013.10.007>
- [20] Rudas, J., & Torres, G. S. (2013). Detección de patologías derivadas de las afecciones diabéticas: una revisión del análisis digital de imágenes de retina The detection of pathologies from diabetic disease: a review of digital retinal image analysis. *Ingeniería y Desarrollo. Universidad Del Norte.*, 3461.
- [21] Salamat, N., Missen, M. M. S., & Rashid, A. (2019). Diabetic retinopathy techniques in retinal images: A review. *Artificial Intelligence in Medicine*, 97(January 2018), 168–188. <https://doi.org/10.1016/j.artmed.2018.10.009>.
- [22] Mittal, K., & Rajam, V. M. A. (2020). Computerized retinal image analysis - a survey. In *Multimedia Tools and Applications* (Vol. 79, Issues 31–32). Multimedia Tools and Applications. <https://doi.org/10.1007/s11042-020-09041-y>
- [23] Badar, M., Haris, M., & Fatima, A. (2020). Application of deep learning for retinal image analysis: A review. *Computer Science Review*, 35, 100203. <https://doi.org/10.1016/j.cosrev.2019.100203>
- [24] A comprehensive guide convolutional to Neural Networks- the ELI5 way, <http://towardsdatascience.com/A-comprehensive-guide-convolutional>.
- [25] Implementing Kernel/filter/ convolutional in your own <http://laptrinhx.com/implementing-kernel-fliter-convolutional-in-your-own-1511826395/>

- [26] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Harhali. “You Only Look Once: Unified, Real-Time Object Detection.”, 2016. <https://arxiv.org/abs/1506.02640>
- [27] Mauricio Menegaz. “Understanding YOLO.” Hackernoon. <https://hackernoon.com/understanding-YOLO-f5a74bbc7967> . Fecha de última consulta: Septiembre 2022
- [28] Jonathan Hui. “Real-time Object Detection with YOLO, YOLOv2, and now YOLOv3.” Fecha de última consulta: Septiembre 2022
- [29] Joseph Redmon, Ali Rarhadi. “YOLOv3: An incremental improvement.” 2018, <https://arxiv.org/abs/1804.02767>
- [30] Modelo de proceso CRISP–DM ([CRISP-DM, 2000]).
- [31] Sarmad Khitran, M. U. Automated System for the Detection of Hypertensive. *Image Processing Theory, Tools and Applications*, 6. (2014).
- [32] Irshad, S. Classification of Retinal Vessels into Arteries and Veins for detection of Hypertensive Retinopathy. *Cairo International Biomedical Engineering Conference*, 4. (2015).
- [33] Pradipto, B. K. Detection of Hypertension Retinopathy Using Deep Learning and Boltzmann Machines. *Journal of Physics*, 7. (2017).
- [34] Triwijoyo, B. K. The Classification of Hypertensive Retinopathy using Convolutional Neural Network. *ScienceDirect*, 8. (2017).
- [35] Faiza Ahmad, A. Y. Textural and Intensity Feature Based Retinal Vessels Classification for the Identification of Hypertensive Retinopathy. *IEEE*, 4. (2018).
- [36] 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. Guayaquil. (2018).
- [37] M, A. L. Colour Image Segmentation of Fundus Blood Vessels for the Detection of Hypertensive Retinopathy. *International Conference on Biosignals, Images and Instrumentation*, 4. (2018).
- [38] M, K. Artery and Vein classification for hypertensive retinopathy. *Proceedings of the Third International Conference on Trends in Electronics and Informatics*, 5. (2019).
- [39] Kalavathy, J. A. A Comprehensive Study of Retinal Vessel Classification Methods in Fundus Images for Detection of Hypertensive Retinopathy and Cardiovascular Diseases. Springer Nature Switzerland, 11. (2019).

- [40] Arasy, R. Detection of hypertensive retinopathy using principal component analysis (PCA) and backpropagation neural network methods. *AIP Conference Proceedings 2092*, 9. (2019).
- [41] Abbas, Q. DenseHyper: an automatic recognition system for detection of hypertensive retinopathy using dense features transform and deep-residual learning. *Multimedia Tools and Applications*, 29. (2020).
- [42] Dharmana, M. M. Pre-diagnosis of Diabetic etinopathy using Blob Detection. *Proceedings of the Second International Conference on Inventive Research in Computing Applications*, 4. (2020).
- [43] Saman, G. Automatic detection and severity classification of diabetic retinopathy. *Multimedia Tools and Applications*, 15. (2020).
- [44] E. Decencière et al., 2013, El enlace para acceder a este dataset es: <http://www.adcis.net/en/third-party/e-ophta/>
- [45] El enlace para acceder a este dataset es: <https://www.kaggle.com/c/diabetic-retinopathy-detection/data>
- [46] Niemeijer et al., 2010 El enlace para acceder a esta base de datos de imágenes es <http://webeye.ophth.uiowa.edu/ROC/>
- [47] Mikołajczyk, A., & Grochowski, M. (Mayo de 2018). Data augmentation for improving deep learning in image classification problem. En *2018 international interdisciplinary PhD workshop (IIPhDW)* (págs. 117-122). IEEE
- [48] Zhong, Z., Zheng, L., Kang, G., Li, S., & Yang, Y. (Abril de 2020). Random erasing data augmentation. En *Proceedings of the AAAI conference on artificial intelligence* (Vol. 34, págs. 13001-13008).
- [49] 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\\_hand-drawn\\_partially\\_colored\\_sketches\\_for\\_visual\\_search](https://www.researchdate.net/publication/319413978_Data_augmentation--assisted_Deep_learning_of_hand-drawn_partially_colored_sketches_for_visual_search)
- [50] Python Lessons. YOLOv3 theory explained. 2018. url: <https://medium.com/@pythonlessons0/yolo-v3-theory-explained> (visitado 08-08-2019).
- [51] A. Bochkovskiy, C.-Y. Wang y H. Y. Mark Liao, «YOLOv4: Optimal Speed and Accuracy of Object Detection,» Taiwan, 2020

- [52] Metrics to use to evaluate Deep learning object detector  
<https://www.kdnuggets.com/2020/08/metrics-evaluate-deep-learning-object-detectors.html>
- [53] 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).