



UNIVERSIDAD DE GUANAJUATO

**DIVISIÓN DE CIENCIAS E
INGENIERÍAS CAMPUS
LEÓN**

**DESARROLLO DE UN
SISTEMA DE DETECCIÓN Y
CLASIFICACIÓN DE
RETINOPATÍA DIABÉTICA
UTILIZANDO YOLOv8**

TESIS PROFESIONAL

**QUE PARA OBTENER EL GRADO DE:
MAESTRO EN CIENCIAS APLICADAS**

PRESENTA:

JOSÉ ANDRÉS ECHEVESTE VÁZQUEZ

ASESORES:

DR. CARLOS VILLASEÑOR MORA

DR. JULIO CESAR ARMAS PÉREZ

LEÓN, GUANAJUATO

SEPTIEMBRE, 2024

"Al final todo estará bien,
y si no está bien, no hemos llegado al final"

-Anónimo

Agradecimientos

Este trabajo de tesis está dedicado principalmente a mis padres, Mariano y Laura, quienes me han apoyado incondicionalmente a lo largo de mi trayectoria académica. Me han brindado su amor, cariño, consejos, paciencia y esfuerzo, incluso frente a las dificultades que surgieron. Gracias a ellos, he podido alcanzar mis metas sin importar los obstáculos. Les agradezco por inculcarme valores que me inspiran a ser una mejor persona cada día y a luchar por mis sueños. Hoy puedo decir que he logrado una de muchas metas. Infinitas gracias por todo, los llevo siempre conmigo.

A mi novia Camila, quien siempre ha estado a mi lado y ha demostrado ser una de las mejores personas que he conocido. Su lealtad e incondicional apoyo me han impulsado a alcanzar mis objetivos y me han ayudado a superar momentos difíciles a lo largo de esta travesía.

A mi amigo Carlos Daniel Moreno "El cubo", quien me apoyó incondicionalmente durante todo mi proyecto de tesis, ayudándome a superar diversas dificultades. También agradezco los días en los que me invitaba a desayunar en el laboratorio. Además, extendiendo mi agradecimiento a todos esos amigos verdaderos que me brindaron apoyo moral y me animaron a no desistir.

Quiero hacer una mención honorífica a mis queridas mascotas: mis dos gatitas Luna y Piri, mi perrita Canelita, mi fiel perrito no tan pequeño Goku, y mi gatita apadrinada "Botas". Les tengo mucho cariño y agradecimiento, ya que, de cierta forma, fueron un motor para seguir adelante y no desistir en mis objetivos. Su mera presencia mientras trabajaba me motivaba enormemente.

Finalmente, agradezco al Dr. Carlos Villaseñor Mora, al Dr. Julio César Armas Pérez y a la M. en C. Georgina Soto Cruz, por haber confiado en mí y en mis habilidades. Agradezco su paciencia cuando abundaban mis dudas durante el proyecto, los recursos que me brindaron para desarrollar mi trabajo, sus consejos y todo su apoyo en general.

“Siempre los llevaré en mi corazón”

Capítulos

1.	Introducción.....	1
2.	Trabajos Relacionados.....	3
3.	Problemática y Retinopatía Diabética	8
4.	Inteligencia Artificial y Técnicas.....	21
5.	Inteligencia Artificial Aplicada - Arquitectura YOLO	28
6.	Materiales y Métodos	33
7.	Resultados.....	42
8.	Discusión	60
9.	Conclusiones.....	63
10.	Declaratoria de Autenticidad	65
11.	Anexos.....	66
12.	<i>Referencias</i>	78

1. Introducción

Las personas con diabetes pueden tener una enfermedad ocular llamada retinopatía diabética. Esta enfermedad ocurre porque los altos niveles de azúcar en la sangre causan daño a los vasos sanguíneos y nervios de la retina. Los vasos sanguíneos pueden hincharse y tener fugas o bloquearse e impedir que la sangre fluya. En ocasiones, se generan nuevos vasos sanguíneos anormales en la retina, condición conocida como retinopatía proliferativa. Todas estas alteraciones pueden causar una pérdida parcial o total de la visión [1].

En las últimas etapas de la enfermedad los daños en la retina suelen ser irreversibles; todas las personas diabéticas corren el riesgo de desarrollar retinopatía diabética (RD) debido a una hiperglucemia mal controlada, hipertensión arterial y colesterol elevado que aumentan la probabilidad de pérdida de visión por RD, aproximadamente 1 de cada 3 personas que viven con diabetes tienen algún grado de retinopatía diabética y 1 de cada 10 desarrollará una forma de la enfermedad que amenace su visión, junto con el riesgo de trastornos visuales asociados, como cataratas o glaucoma [1].

La Comisión de Salud Mundial de Lancet, comentó sobre el cuidado de la RD, mencionando que éste depende de una estrecha colaboración con los servicios médicos para poder diagnosticar a tiempo a las personas con diabetes y así adelantarse a la aparición de sus complicaciones. El diagnóstico y tratamiento de la RD se está transformando gracias a la creciente disponibilidad de cámaras de retina de bajo costo, que adquieren imágenes para su clasificación incluso a distancia [2].

Bajo este contexto, el análisis automatizado de imágenes de fondo de ojo mediante técnicas de inteligencia artificial (aprendizaje profundo), se ha convertido en una herramienta de alto valor para el diagnóstico y seguimiento de la RD.

Esta tesis se centró en el desarrollo de un sistema de clasificación de grados de RD y detección de lesiones utilizando YOLOv8, un algoritmo de detección de objetos de última generación. La razón de utilizar este algoritmo fue debido a que presenta ventajas significativas en cuanto a velocidad y precisión, lo que lo hace ideal para aplicaciones en tiempo real. Cabe destacar que a lo largo de los años se han desarrollado distintas metodologías de aprendizaje profundo enfocadas en esta problemática, sin embargo, la implementación de YOLOv8 en esta tesis se centró en la búsqueda de mejoras e innovación en este campo.

Los objetivos de esta investigación fueron los siguientes:

- Implementación del modelo YOLOv8 para la clasificación de imágenes de fondo de ojo en diferentes grados de la RD: normal, leve, moderada, severa, proliferativa y no graduable.
- Detección y localización de lesiones asociadas a la RD: microaneurismas, hemorragias, exudados duros y exudados suaves.
- Evaluación del rendimiento del modelo en términos de las métricas de rendimiento más comunes para clasificadores y detectores.
- Análisis de las limitaciones del modelo y potenciales propuestas de mejora.

Algunas, las justificaciones para el desarrollo de esta investigación fueron:

- Aumentar la eficiencia del proceso de diagnóstico: apoyando a los profesionales de la salud en la posible detección de anomalías en la retina.

- Mejorar el acceso a la detección temprana: proporcionando una herramienta que permita la detección de anomalías, que los especialistas validaran si corresponden o no a alguna condición de retinopatía.
- Reducir la subjetividad en la interpretación de imágenes: con el fin de proporcionar un diagnóstico más objetivo y consistente.
- Facilitar el seguimiento de la progresión de la enfermedad: esto podría permitir una intervención más temprana y personalizada.

La estructura del trabajo de tesis se basa en abordar en distintos capítulos la información más relevante para contextualizarse en este tema, primeramente, se abordará un capítulo de trabajos relacionados o estado del arte; posteriormente se presenta el marco teórico en donde se aborda todo lo relacionado a la RD, además de la teoría, técnicas y funcionamiento de las tecnologías utilizadas; continuando con la metodología de la investigación mostrando la estructura de seguimiento para desarrollar este trabajo; después los resultados muestran los datos obtenidos durante la investigación y su interpretación; por último discusión y conclusiones, en donde se resaltan los hallazgos en relación con los objetivos y la literatura revisada, respuestas a las preguntas de la investigación, los obstáculos que se presentaron, las limitaciones de estudio y recomendaciones futuras.

En general se espera que esta tesis contribuya al desarrollo de herramientas automatizadas para el diagnóstico y seguimiento de la RD, con el potencial de mejorar la calidad de vida de los pacientes con este padecimiento y además buscar la reducción del impacto de la enfermedad en la salud pública.

2. Trabajos Relacionados

Numerosos investigadores han dedicado esfuerzos significativos a la exploración de soluciones para el tratamiento de la retinopatía diabética (RD). A continuación, se presenta una síntesis de algunos de estos trabajos, destacando sus contribuciones más relevantes y los resultados obtenidos. Algunos de estos incluyen detalles sobre los conjuntos de datos empleados, los cuales se examinan en el capítulo de materiales y métodos, centrándonos exclusivamente en aquellos utilizados en esta investigación.

Lalitha y Ajith [3], (2021), propusieron un sistema que clasifica los datos de imagen en clases definidas mediante el empleo de la arquitectura YOLO-RF. Este sistema fue sometido a una comparación exhaustiva con diversos clasificadores de aprendizaje automático tradicionales, tales como Máquinas de Soporte Vectorial (SVM), Árbol de Decisión (DT) y Bosque Aleatorio (RF), así como con un modelo de Deep Learning como YOLO. Para llevar a cabo este estudio, utilizaron conjuntos de datos de imágenes de fondo de ojo provenientes de las plataformas KAGGLE e IDRID. Los resultados obtenidos evidenciaron un rendimiento destacado del modelo propuesto YOLO-RF, alcanzando una precisión del 99.3%, una puntuación de precisión del 97.2% y un recall del 99.1%.

Santos et al. [4], (2022), presentaron una nueva forma de detectar lesiones en el fondo del ojo, utilizando técnicas de procesamiento de imágenes y una red neuronal profunda, basada en la arquitectura YOLO de última generación (YOLOv7, para ese momento). Emplearon dos conjuntos de datos públicos de imágenes de RD para entrenar y evaluar la precisión del enfoque propuesto: DDR (Dataset for Diabetic Retinopathy Classification) (Ver la sección Datasets) e IDRiD (Indian Diabetic Retinopathy Image Dataset). Solo se utilizaron imágenes con lesiones anotadas en los conjuntos de datos para llevar a cabo el entrenamiento y la evaluación del enfoque propuesto. Estos conjuntos de datos se dividieron en conjuntos de entrenamiento, validación y prueba en una proporción de 50:20:30, respectivamente. Los mejores resultados se obtuvieron en el conjunto de datos DDR utilizando el optimizador Adam y el método Tiling, alcanzando en la etapa de validación un mAP de 0.2630 para el límite de IoU de 0.5 y un puntaje F1 de 0.3485, y en la etapa de prueba un mAP de 0.1540 para el límite de IoU de 0.5 y un puntaje F1 de 0.2521. (Para más información acerca de estas métricas ver la sección: **Métricas de rendimiento**).

Chetoui & Akhloufi [5], (2020), desarrollaron un algoritmo de Deep Learning (DL) capaz de detectar signos de RD en imágenes del fondo de ojo. Utilizaron los conjuntos de datos EyePACS, MESSIDOR, MESSIDOR-2, DIARETDB0, DIARETDB1, STARE, IDRID, E-ophtha y UoA-DR y más de 90,000 imágenes para evaluar la eficiencia del algoritmo. Además, implementaron un algoritmo de explicabilidad basado en el mapeo de activación de clases ponderado por gradiente, para mostrar visualmente los signos seleccionados por el modelo y para clasificar las imágenes de retina como positivos para RD. Los resultados obtenidos fueron: área bajo la curva (AUC) de 0.986, sensibilidad de 0.958 y especificidad de 0.971 para EyePACS. Para MESSIDOR, MESSIDOR-2, DIARETDB0, DIARETDB1, STARE, IDRID, E-ophtha y UoA-DR, el AUC fue de 0.963, 0.979, 0.986, 0.988, 0.964, 0.957, 0.984 y 0.990, respectivamente.

Merino Hernandez et al. [6], (2019), desarrollaron un algoritmo basado en métodos avanzados de clasificación DL y SVM, que identifica automáticamente las imágenes patológicas de forma eficaz y segura. El software discrimina automáticamente retinografías normales y patológicas. Los resultados obtenidos fueron sensibilidad de 94% y especificidad del 96%.

P.L. y Kumar [7], (2023), presentaron un sistema computarizado para el análisis y evaluación de RD basado en fotografías del fondo de ojo, con el objetivo de mejorar la fiabilidad del diagnóstico de la RD, utilizaron YOLOv3 para reconocer y clasificar RD a partir de imágenes retinianas. La RD se clasificó en cinco etapas principales: normal, leve, moderada, grave y proliferativa. Se evaluó el rendimiento del algoritmo YOLOv3 basado en imágenes de fondo de ojo en color y se calculó la precisión media promedio (mAP) en la detección de lesiones de RD. Se logró una alta precisión y

sensibilidad en los datos de entrenamiento y prueba para la clasificación de la RD, Los resultados indican que el modelo sugerido distingue todas las fases de la RD y tiene un mejor rendimiento que los modelos existentes en términos de precisión y tiempo de implementación.

Wahab Sait, A.R. [8], (2023), desarrolló un modelo de clasificación de gravedad de la RD de varias clases utilizando DL. El modelo propuesto integró el preprocesamiento de imágenes (para mejorar la calidad de las imágenes) usando YOLOv7, QMPA y MobileNet V3-Small. Los conjuntos de datos de imágenes de fondo de ojo que utilizó estaban altamente desbalanceados, contenían ruido y artefactos. Los sesgos del conjunto de datos se abordaron utilizando el proceso de aumento de datos. Para el proceso de extracción de características aplicó YOLOv7 y aplicó QMPA con la estrategia de mutación Cauchy-Gaussiana para seleccionar las características críticas relacionadas con la gravedad de la RD y también MobileNet V3 para clasificar las imágenes según los niveles de gravedad. Se utilizaron APTOS y EyePacs como conjuntos de datos de referencia. El modelo propuesto ofrece la oportunidad de desarrollar una aplicación móvil para tratar a los pacientes con RD, sin embargo, se encontraron limitaciones en la clasificación de las imágenes de fondo de ojo en aquellas imágenes donde la gravedad de la RD es mínima reduciendo la precisión de predicción del modelo propuesto. Se requiere un preprocesamiento efectivo de imágenes para mejorar la calidad de las imágenes en tiempo real. Wahab Sait menciona que seguirá trabajando en el problema y que en un futuro ampliará la investigación para resolver las deficiencias del modelo propuesto.

Kobat et al. [9], (2022), proponen un modelo de detección de RD utilizando DenseNet pre-entrenado, emplean la división de parches horizontales y verticales en la extracción de características y obtienen una precisión promedio del 84.9% y 86.7% en los conjuntos de datos APTOS y EyePacs.

Modi y Kumar [10], (2023), usan BA-DeepForest para la detección de RD. El modelo propuesto combina el escaneo de múltiples granos, un método de extracción de características basado en el algoritmo de murciélago y la técnica de cascada de bosques profundos. El escaneo de múltiples granos se adopta para extraer las características de las imágenes de fondo de ojo. La eficacia del modelo propuesto BA-Deep Forest se evalúa utilizando dos conjuntos de datos de RD. Los resultados de la simulación se comparan con varios modelos existentes como KNN, ANN, SVM, VGG16, VGG19, InceptionV3 y Deep Forest y mostraron que el modelo propuesto BA-DeepForest logra tasas de precisión más altas (92.9% y 94.6%), tasas de puntuación F1 (95.3% y 96.5%), tasas de sensibilidad (95.7% y 97.4%) y tasas de especificidad (95.0% y 98.0%) utilizando el método de validación cruzada de 10 pliegues.

Kumar y Priyadharsini [11], (2021), construyeron un sistema basado en una red neuronal convolucional (CNN). Utilizan la CNN debido a su precisión y se prefiere el agrupamiento soft max. Además, se implementa una capa de abandono antes de la capa completamente conectada para evitar problemas de sobreajuste. Este modelo abarca tres etapas de RD, no proliferativa (RDNP), proliferativa (RDP) y Retina Normal (RN). La arquitectura ResNet y YOLO se utiliza para una respuesta y clasificación rápida. Este método predice la RD con una precisión del 93.5% para RDP, 94.8% para RDNP y 93.5% para RN. El valor del error cuadrático medio también es bajo en este sistema. Se calculan también otros parámetros de aprendizaje profundo para resaltar la importancia de este modelo. El modelo propuesto garantiza un mínimo del 93.0% en la predicción y apoya ampliamente a la comunidad de investigación clínica al proporcionar soluciones oportunas.

Prayas et al. [12], (2020), se centra en la detección automática de lesiones rojas en imágenes de fondo de ojo utilizando YOLOv3, identifica todas las lesiones rojas presentes en una imagen y crea un cuadro delimitador alrededor de cada una. Para la detección, utiliza un modelo basado en CNN Darknet 53 y una función de pérdida de error cuadrado para identificar el cuadro delimitador, y regresión logística para encontrar la puntuación de confianza de un objeto, y por último utiliza supresión de no máximos para eliminar cuadros delimitadores que no son los mejores ajustes, obtuvieron una precisión media de 83.3% en sus datos de prueba.

Suvajit et al. [13], (2018), utilizaron imágenes de RD procesadas para entrenar CNN, considerando un modelo altamente calificado para la visión de imágenes (VGGnet), entrenaron el modelo con 1000 imágenes obteniendo una precisión de 72.5%. Para probar el modelo, se consideraron 300 imágenes, en cuanto a los resultados su modelo reporta que las imágenes con grados de retinopatía no proliferativa y proliferativa fueron predichos perfectamente.

Gangwar & Ravi [14], (2021), proponen un híbrido de DL para la detección automática de la RD. Este híbrido se compone de un bloque personalizado de capas CNN sobre el modelo de aprendizaje por transferencia de Inception-ResNet-v2 previamente entrenado. Usaron el conjunto de datos MESSIDOR y APTOS 2019 (conjunto de datos de Kaggle). Lograron una precisión de prueba del 72.3% y el 82.2% en el conjunto de datos MESSIDOR y APTOS, respectivamente.

Pozo [15], (2019), sugieren analizar e implementar modelos de redes neuronales artificiales supervisadas para la detección de estructuras vasculares retinianas de fondo de ojo. Aplicaron una red convolucional basada en la arquitectura U-Net, con ventanas de imagen de 16 x 16 píxeles. La red convolucional estaba compuesta por cuatro capas convolucionales y tres capas deconvolucionales. Obtuvieron resultados de sensibilidad del 74.2%, una especificidad del 90.7% y una exactitud del 88.2%.

Auccahuasi et al. [16], (2020), presentan una metodología para reconocer exudados duros, que es la primera manifestación de la RD, en imágenes de la retina. Usan una CNN con un conjunto de datos de 906 imágenes de exudados duros y 1,068 imágenes sin exudados duros, que se entrena con un grupo de 400 imágenes, con una prevalencia de 0.5 con respecto a la presencia de exudados duros. Después de esto se hacen dos comprobaciones al clasificador, utilizando grupos de imágenes similares, como el grupo de pruebas y validación, obteniendo valores de sensibilidad de 0.92 en ambos casos y valores de especificidad de 0.92 y 0.93 para los grupos de imágenes de prueba y validación, respectivamente. Se realizaron mediciones adicionales para calcular el Valor Predictivo Positivo y el Valor Predictivo Negativo, obteniéndose valores de 0.92 para el grupo de imágenes de prueba y 0.9% para el grupo de imágenes de validación. En el caso del Valor Predictivo Negativo se obtuvieron valores de 0.92 para el grupo de imágenes de prueba y 0.93 para el grupo de imágenes de validación.

Moreira et al. [17], (2020) formulan el desarrollo de CNNs para detectar la RD, y el impacto de la resolución y la red de imágenes en la precisión de la predicción. Obtienen resultados de un área bajo la curva característica operativa del receptor de 0.93 al aumentar la resolución de la arquitectura Inception-V3.

Long et al. [18], (2019), desarrollaron y evaluaron un algoritmo de procesamiento automático de imágenes de retina para la detección de exudados duros (ED), que utiliza umbral dinámico y agrupamiento difuso de C-means (FCM) seguido de máquina de soporte vectorial (SVM) para la clasificación. Este algoritmo consta de cuatro etapas principales: preprocesamiento de imágenes, localización del disco óptico (OD), determinación de candidatos ED mediante umbral dinámico en combinación con umbral global basado en FCM, y extracción de ocho características de textura de la región de candidatos HE, que luego se ingresan en un clasificador SVM para la clasificación automática de ED. El algoritmo propuesto fue entrenado y validado utilizando cross validation (10 veces) en una base de datos pública de e-ophtha. La cual cuenta con imágenes de exudados (EX) [47 imágenes] a nivel de píxel, logrando una sensibilidad promedio general, valor predictivo positivo (PPV) y puntaje F de 76.5%, 82.7% y 76.7%, respectivamente. Posteriormente, se probó en otra base de datos independiente, DIARETDB1 (89 imágenes), obteniendo una sensibilidad promedio general, especificidad y precisión del 97.5%, 97.8% y 97.7%, respectivamente.

Li et al. [19], (2019), desarrollaron una serie de algoritmos automatizados basados en el Deep Learning (DL) para la clasificación de RD, los cuales han logrado una alta sensibilidad y especificidad

(mayor al 90%). Sin embargo, estos modelos de DL no mostraron un buen desempeño en aplicaciones clínicas debido a las limitaciones de los conjuntos de datos de imágenes de fondo de ojo disponibles públicamente. Para abordar esta situación, recolectaron 13,673 imágenes de fondo de ojo de 9,598 pacientes y las dividieron en seis clases según la calidad de la imagen y el nivel de RD, gracias a la evaluación de siete graduadores. Además, seleccionaron 757 imágenes con RD para anotar cuatro tipos de lesiones relacionadas con esta enfermedad. Finalmente, evaluaron sus algoritmos de aprendizaje profundo en estas imágenes, realizando tareas como clasificación de imágenes, segmentación semántica y detección de objetos. Aunque obtuvieron una precisión del 0.82 en la clasificación de RD, observaron que los algoritmos no se desempeñaron de manera satisfactoria en la segmentación y detección de lesiones, lo que indica que estas tareas son bastante desafiantes. En resumen, crearon un nuevo conjunto de datos llamado DDR (Dataset for Diabetic Retinopathy Classification) para evaluar sus modelos de aprendizaje profundo y explorar aún más las aplicaciones clínicas, especialmente en el reconocimiento de lesiones.

Alyoubi et al. [20], (2021), presentaron sistemas de diagnóstico totalmente automáticos que superan las técnicas manuales para evitar diagnósticos erróneos, reduciendo tiempo, esfuerzo y costos. El sistema propuesto clasifica las imágenes de RD en cinco etapas (sin RD, leve, moderada, grave y proliferativa), además de localizar las lesiones afectadas en la superficie de retención. El sistema comprende dos modelos basados en DL. El primer modelo (CNN512) utilizó la imagen completa como entrada al modelo CNN para clasificarla en una de las cinco etapas de DR. Logró una precisión del 88.6 % y 84.1 % en los conjuntos de datos públicos DDR y APTOS Kaggle 2019, respectivamente. Simultáneamente, el segundo modelo utilizó un modelo YOLOv3 adoptado para detectar y localizar las lesiones de DR, logrando un mAP de 0.216 en la localización de lesiones en el conjunto de datos de DDR, lo que mejora los resultados actuales de última generación. Finalmente, las dos estructuras propuestas, CNN512 y YOLOv3, se fusionaron para clasificar imágenes de DR y localizar lesiones de DR, obteniendo una precisión del 89.0% con una sensibilidad del 89.0%, una especificidad del 97.3% que supera los resultados actuales del estado del arte.

Navaneethan y Devarajan [21], (2024), en este estudio se introduce el enfoque Modified Generative Adversarial-based Crossover Salp Grasshopper (MGA-CSG) para la detección temprana y clasificación precisa de enfermedades retinianas diabéticas a partir de imágenes de fondo de ojo. El método MGA-CSG combina aprendizaje profundo con algoritmos de optimización para alcanzar una alta precisión de clasificación. Primero, se preprocesan las imágenes para mejorar su calidad, seguido por la extracción de características mediante una Red Neuronal Convolutiva (CNN). Para mitigar la limitación de vectores de características generados y mejorar la incertidumbre en el aprendizaje, se utiliza un modelo de Red Generativa Adversarial (GAN) que genera imágenes adicionales, permitiendo entrenar el clasificador con menos muestras de entrada. Además, el enfoque utiliza tanto datos etiquetados como no etiquetados, optimizando el rendimiento del entrenamiento de GAN. El clasificador binario se reemplaza por un clasificador multiclase, diferenciando mejor las clases de enfermedades retinianas. La optimización del modelo se logra mediante los algoritmos Crossover Grasshopper Optimizer (GOA) y Salp Swarm Algorithm (SSA), mejorando la precisión en la clasificación. Los resultados experimentales muestran que MGA-CSG alcanza altos niveles de precisión, recall, especificidad y coeficiente Kappa en la predicción de glaucoma, retinopatía diabética, cataratas, edema macular y miopía, con una precisión del 98.8%.

El trabajo aquí presentado utiliza para entrenamiento la base de datos creada por Li et al. [19] (DDR-dataset) ya que es una base de datos que posee imágenes etiquetadas en 6 formas distintas: sano, leve, moderado, severo, proliferativo y ellos agregan la clase no graduable la cual representa una imagen que no se ha podido identificar el tipo de clase que pertenece, ya sea por falta de información o sea una imagen con defectos de cámara, tales como: fotografías mal tomadas, borrosas o desenfocadas. Además, esta base de datos cuenta con una cantidad significativa de imágenes con la segmentación y

detección de cuatro tipos de lesiones de la retinopatía diabética las cuales son: exudados duros, exudados suaves, microaneurismas y hemorragias.

La propuesta de trabajo a grandes rasgos fue utilizar la tecnología más actual (al momento de escribir este trabajo de investigación) de la serie de algoritmos de YOLO, en este caso YOLOv8, para poner a prueba su modelo clasificador de imágenes a la par de utilizar también su modelo estelar (detección de objetos) para la detección de los tipos de lesiones antes mencionados. Todo esto se entrenó con la base de datos DDR, y para su validación general se utilizó la misma base (considerando la parte de testeo) sumado a imágenes de otra serie de datasets relevantes que son: IDRiD y EyePACS (KAGGLE). Dichos datasets serán descritos más adelante.

3. Problemática y Retinopatía Diabética

3.1. *Introducción a la Retinopatía Diabética (RD)*

La retinopatía diabética es una afección del ojo que puede causar pérdida de visión y ceguera en personas con diabetes. Afecta, entre otras estructuras, los vasos sanguíneos de la retina (la capa de tejido sensible a la luz en la parte de atrás del ojo).

El daño a la retina comienza cuando se bloquean, rompen o se modifican los diminutos vasos sanguíneos de la retina, lo que puede provocar que goteen o sangren. Para compensar estos vasos sanguíneos bloqueados, existen mecanismos que provocan el crecimiento de nuevos vasos sanguíneos (retinopatía proliferativa), que son más frágiles. Estos nuevos vasos sanguíneos pueden gotear o sangrar fácilmente también [22].

3.1.1. *Factores de riesgo*

Cualquier persona con diabetes tipo 1, tipo 2 o gestacional (diabetes en el embarazo) puede presentar retinopatía diabética. Mientras más tiempo tenga diabetes, más altas son las probabilidades de que presente retinopatía diabética. Los siguientes factores también pueden aumentar el riesgo [23]:

- Los niveles de glucosa en la sangre, presión arterial y colesterol demasiado altos.
- El tabaquismo.
- La raza o etnia: las personas afroamericanas hispanas o latinas e indoamericanas o nativas de Alaska tienen mayor riesgo.

3.1.2. *Síntomas*

Las primeras etapas de la retinopatía diabética por lo general no presentan síntomas. Algunas personas notan cambios en la visión por lo general cuando el daño ya es muy avanzado, y en ocasiones pueden presentar dificultad para leer o ver objetos de lejos. Estos cambios podrían aparecer y desaparecer.

En las etapas avanzadas de la enfermedad, los vasos sanguíneos de la retina comienzan a sangrar en el vítreo o humor acuoso (fluido con la consistencia de gel que llena el ojo). Si esto sucede, es posible que vea manchas oscuras flotantes o líneas que parecen telarañas. Algunas veces, las manchas desaparecen por sí solas, pero es importante recibir tratamiento de inmediato. Sin tratamiento, el sangrado puede volver a ocurrir, empeorar o provocar cicatrices [22].

3.1.3. *Causas*

La retinopatía diabética es causada por una alta concentración de glucosa en la sangre como consecuencia de la diabetes. Con el tiempo, tener demasiada glucosa en la sangre puede dañar la retina, la parte del ojo que detecta la luz y envía señales al cerebro a través del nervio óptico en la parte de atrás o fondo del ojo [22].

3.1.4. *Diagnóstico*

La retinopatía diabética se diagnostica generalmente mirando desde fuera del ojo a través de la córnea, usando la ventana que permite la pupila, sin embargo, si ésta no tiene una apertura adecuada es difícil poder ver dentro del ojo, así, es mejor mediante un examen ocular con la pupila dilatada. Para este examen, se administran gotas que dilatan (abren más) la pupila para permitirle al oftalmólogo ver mejor dentro del ojo [24]. Durante el examen, el oftalmólogo buscará anomalías en la retina, como hemorragias, exudados, alteración en la red venosa, entre muchas otras.

Angiografía con fluoresceína:

Cuando el daño no es fácil de identificar se puede aprovechar que la pupila ya está dilatada e inyectar un agente de contraste en una vena del brazo, este agente de contraste circulará por todo el cuerpo, incluido las arterias y venas de la retina, lo que permitirá adquirir imágenes con sangre contrastada que fluórese a la luz que se está utilizando para ver dentro del ojo, estas imágenes permiten localizar vasos sanguíneos cerrados, rotos o con fugas.

Tomografía de coherencia óptica:

En presencia de anomalías que no son fáciles de ver desde fuera del ojo, se usa esta prueba, las fotografías proporcionan imágenes trasversales de la retina que muestran su grosor, por ejemplo. Esto ayudará a determinar cuánto líquido, si es que hay alguno, se ha filtrado en el tejido de la retina, detrás o delante de ella, la tomografía de coherencia óptica también se puede utilizar para controlar el avance de algún tratamiento.

3.1.5. Tratamiento

En las primeras etapas de la RD, el oftalmólogo probablemente solo hará un seguimiento del estado de los ojos. Algunas personas con RD podrían necesitar un examen completo de ojos con dilatación de las pupilas cada 2 a 4 meses, en personas que están controladas de sus niveles de glucosa la recomendación es un examen de retina cada 6 meses.

En las etapas avanzadas, es importante comenzar el tratamiento inmediato, especialmente si se experimentan cambios en la visión. Si bien no se puede deshacer el daño sufrido a la visión, el tratamiento puede evitar que la visión empeore. También es importante tomar medidas para controlar la diabetes, la presión arterial y el colesterol [22].

Entre los tratamientos más comunes están:

- **Inyecciones.** Los medicamentos conocidos como anti-VEGF pueden retardar o revertir la RD. Otros medicamentos, conocidos como corticoesteroides, también pueden ayudar.
- **Tratamiento con láser.** Para reducir la inflamación de la retina, los oftalmólogos pueden usar láseres para contraer los vasos sanguíneos y detener el goteo de sangre.
- **Cirugía del ojo.** Si su retina está sangrando mucho o usted tiene muchas cicatrices en el ojo, el oculista podría recomendarle un tipo de cirugía conocida como vitrectomía.

3.1.6. Prevención

La mejor forma de prevenir la RD es llevar un adecuado control de la diabetes para reducir el riesgo de desarrollarla, esto significa mantener los niveles de glucosa en la sangre lo más normal posible. Entre otras actividades para la prevención se encuentran la realización de exámenes de los ojos con dilatación, mantener los niveles de presión arterial y colesterol dentro de los rangos normales, dejar de fumar y realizar actividad física [23].

3.2. Fisiopatología de la Retinopatía Diabética

Basado en el artículo de la revista médica de Chile publicada en octubre de 2009 [25], el proceso bioquímico que desarrolla la RD es el siguiente:

La retinopatía diabética (RD) es una complicación ocular de la diabetes mellitus que se desarrolla lentamente a lo largo de años o décadas. El proceso se caracteriza por una serie de eventos que

conducen al deterioro de los vasos sanguíneos de la retina, la capa sensible a la luz en la parte posterior del ojo.

Fisiopatología de la RD resumido paso a paso:

Hiper glucemia y Glicosilación: [25]

- La hiper glucemia crónica, sello distintivo de la diabetes, desencadena la glicosilación no enzimática de proteínas.
- Los Productos de Glicosilación Avanzada (AGE) se acumulan en la membrana basal de los capilares retinales, alterando su estructura y función.

Vía de la Aldosa Reductasa y muerte de Pericitos: [25]

- La hiper glucemia activa la enzima aldosa reductasa, que convierte la glucosa en sorbitol, un alcohol de azúcar que se acumula dentro de las células, incluyendo los pericitos.
- Esta acumulación, junto con la toxicidad de los AGEs, daña y destruye los pericitos, células de soporte cruciales para la integridad de los capilares retinales.

Disfunción endotelial e isquemia: [25]

- La pérdida de pericitos debilita la pared capilar, haciéndola más permeable.
- La PKC (proteína kinasa C) activada por la hiper glucemia, aumenta la expresión de endotelina-1, una sustancia que constriñe los vasos sanguíneos, contribuyendo a la isquemia retinal.

Lesiones Retinales: [25]

- Exudados céreos: La mayor permeabilidad capilar permite la salida de proteínas y lípidos que forman depósitos amarillentos visibles en la retina.
- Microhemorragias: Los capilares debilitados se rompen fácilmente, causando pequeñas hemorragias.
- Microaneurismas: Se forman dilataciones en los capilares debilitados, propensos a la formación de microtrombos e isquemia.
- Exudados algodonosos: Son áreas de microinfartos de retina causados por la isquemia y la obstrucción de los vasos sanguíneos.

Neovascularización y Hemorragia Vítrea: [25]

- La isquemia retinal estimula la producción de VEGF, que promueve el crecimiento de nuevos vasos sanguíneos (neovascularización).
- Estos nuevos vasos son frágiles, proliferan hacia el vítreo (humor acuoso), y se rompen fácilmente, causando hemorragia vítrea, una complicación grave que puede conducir a la ceguera.

En general la RD, es un proceso complejo que inicia con la hiper glucemia y culmina con la pérdida de la visión. La glicosilación de proteínas, la vía de la aldosa reductasa, la disfunción endotelial, la isquemia retinal y la neovascularización son los pasos clave en su desarrollo.

Además, es necesario y fundamental que el control glucémico sea estricto al igual que la presión arterial, para prevenir y retrasar la progresión de la RD.

En la Figura 1 se muestra una representación gráfica de la evolución de la fisiopatología de la RD.

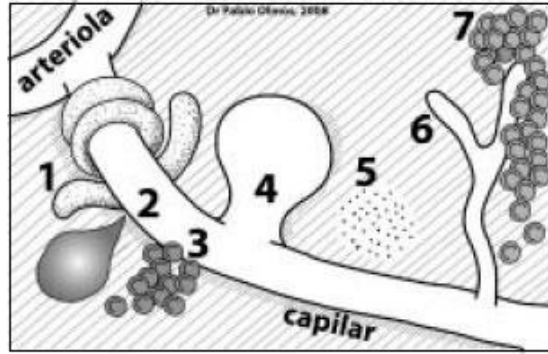


Figura 1. Retinopatía diabética. Secuencia de los eventos a nivel capilar retinal. Desde la arteriola emerge el vaso capilar; puede observarse dos pericitos envolviendo a este último, lo que es la situación normal. 1 = pericito en fase de destrucción mientras se desprende del vaso capilar; 2 = "exudado céreo". 3 = microhemorragia. 4 = microaneurisma. 5 = "exudado algodonoso". 6 = vaso de neoformación. 7 = hemorragia vítrea. Los tamaños son aproximados. El diámetro de un eritrocito es de aproximadamente 7 micrómetros. Las líneas paralelas en el plano de fondo corresponden a la capa de fibras nerviosas de la retina. Fuente: [25].

3.3. Clasificación de la RD

Existen dos etapas de la enfermedad ocular diabética, las cuales se clasifican como:

- **NPDR (retinopatía diabética no proliferativa)**

Esta es la etapa temprana de la enfermedad ocular diabética. Muchas personas con diabetes la tienen. Cuando tiene NPDR, muchos vasos sanguíneos pequeños sufren pérdidas y hacen que la retina se hinche. Cuando se hincha la mácula, se denomina edema macular. Esta es la razón más común por la que la gente con diabetes comienza a perder la visión.

Además, los vasos sanguíneos en la retina pueden cerrarse. Esto se llama isquemia macular. Cuando eso sucede, la sangre no puede llegar a la mácula. En algunos casos, se forman pequeñas partículas en la retina llamadas exudados. Estas partículas también pueden afectar la visión. Si tiene retinopatía diabética no proliferativa, su visión será borrosa [1].

- **PDR (retinopatía diabética proliferativa)**

Esta es la etapa más avanzada de la enfermedad ocular diabética. Se produce cuando la retina comienza a desarrollar nuevos vasos sanguíneos. Esto se denomina neovascularización.

Estos vasos nuevos frágiles a menudo sangran hacia el vítreo. Si sólo sangran un poco, quizá vea unas cuantas manchas oscuras flotando dentro del ojo. Si sangran mucho, puede que bloqueen toda la visión. Estos vasos sanguíneos nuevos pueden desarrollar cicatrices. El tejido cicatrizante puede causar problemas con la mácula o derivar en un desprendimiento de retina.

La retinopatía diabética proliferativa es muy grave y puede hacerle perder tanto la visión central como la periférica (lateral) [1].

Las principales patologías de la retinopatía diabética (DR) (ver Figura 2) son:

- **Microaneurismas (MA):** son deformaciones en las paredes de los vasos sanguíneos. Es una mancha roja de menos de 125 μm de tamaño y tiene márgenes definidos [26].
- **Hemorragias (HE):** ocurren debido a fugas de capilares débiles. Se define como una mancha roja con margen irregular y/o densidad desigual. Suele tener un tamaño superior a 125 μm [26].

- Exudados duros (EX): 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 [27].
- Exudados suaves (SE): también llamados manchas de algodón, son infartos de la capa de fibras nerviosas; tienen un aspecto más blanquecino que los exudados duros [28].

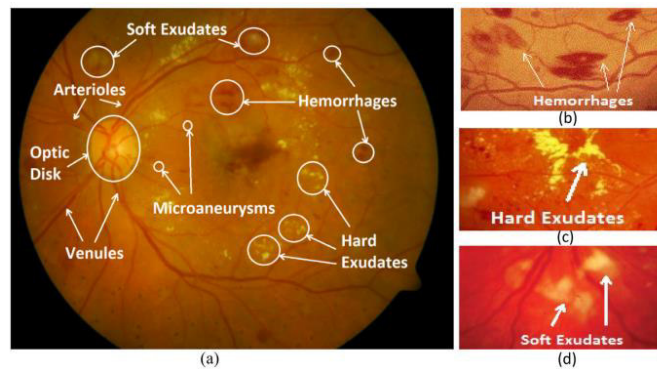


Figura 2. (a) Imagen de la retina con sus patologías. (b) Hemorragias. (c) Exudados duros. (d) Exudados suaves. Fuente: Tomado de [29].

3.3.1. Escalas de clasificación clínica

A pesar de que la RD se divide en 2 etapas principales (No proliferativa y proliferativa), existe un marco para la clasificación clínica de las subetapas hasta antes de llegar a la retinopatía diabética proliferativa, basado en la guía práctica clínica del IMSS [30] para la clasificación de las subetapas se obtiene la Tabla 1.

Tabla 1. Escala internacional de severidad de la retinopatía diabética.

Grados de Recomendación	Interpretación
No proliferativa Leve	Al menos un microaneurisma. No se reúnen los criterios para retinopatía no proliferativa moderada, severa, proliferativa o proliferativa de alto riesgo. (véase más adelante)
No proliferativa Moderada	Hemorragias y / o microaneurismas \geq que la fotografía estándar 2A*; y / o definitivamente presentes: exudados blandos, rosarios venosos (RV), o anomalidades microvasculares intrarretinianas (AMIR); No se reúnen los criterios para retinopatía no proliferativa severa, retinopatía proliferativa o retinopatía proliferativa de alto riesgo (véase más adelante)
No proliferativa Severa	Exudados blandos, RV y AMIR, todos sin duda presente en al menos dos de los campos de cuatro a siete; o dos de los tres lesiones precedentes presente en al menos dos de los campos a través de cuatro de siete y hemorragias y microaneurismas presentes en estos cuatro campos, igual o superior al 2A foto estándar en al menos uno de ellos; o AMIR presentes en cada uno de los campos de cuatro al siete y valor igual o superior 8A fotografía estándar en al menos dos de ellos; y la definición no se reunió para la retinopatía proliferativa temprana o retinopatía proliferativa de alto riesgo (véase más adelante)

Proliferativa temprana (es decir, la retinopatía proliferativa sin características de alto riesgo)	Neovasos; y la definición no se cumplen para la retinopatía proliferativa de alto riesgo (véase más adelante)
Proliferativa de alto riesgo de (con características de alto riesgo)	Los nuevos vasos en o dentro de un diámetro de disco del disco óptico (NVD) \geq norma fotografía 10A * (alrededor de un cuarto a la zona de un tercio del disco), con o sin hemorragia vítrea o prerretinal; o / o hemorragia prerretinal acompañado de nuevos buques, ya sea NVD $<$ 10A fotografía estándar o nuevos vasos en otros lugares (NVE) \geq área vítrea y un cuarto disco.

*Early Treatment Diabetic Retinopathy Study Research Group. Grading diabetic retinopathy from stereoscopic color fundus photographs--an extension of the modified Airlie House classification. ETDRS report number 10. Ophthalmology 1991; 98:786-806. Adapted from the Early Treatment Diabetic Retinopathy Study Research Group. Early Treatment Diabetic Retinopathy Study design and baseline patient characteristics. ETDRS report number 7. Ophthalmology 1991; 98:742. <http://one.aaopt.org/preferred-practice-pattern/diabetic-retinopathy-ppp--september-2008-4thprint#APPENDIX5.CLASSIFICATIONOFDIABETICRETINOPATHYINTHEEARLYTREATMENTOFDIABETICRETINOPATHYSTUDY>

Fuente: [30]. A continuación, en la Figura 3, se observan visualmente como lucen algunas etapas de la RD mencionados en la Tabla 1.

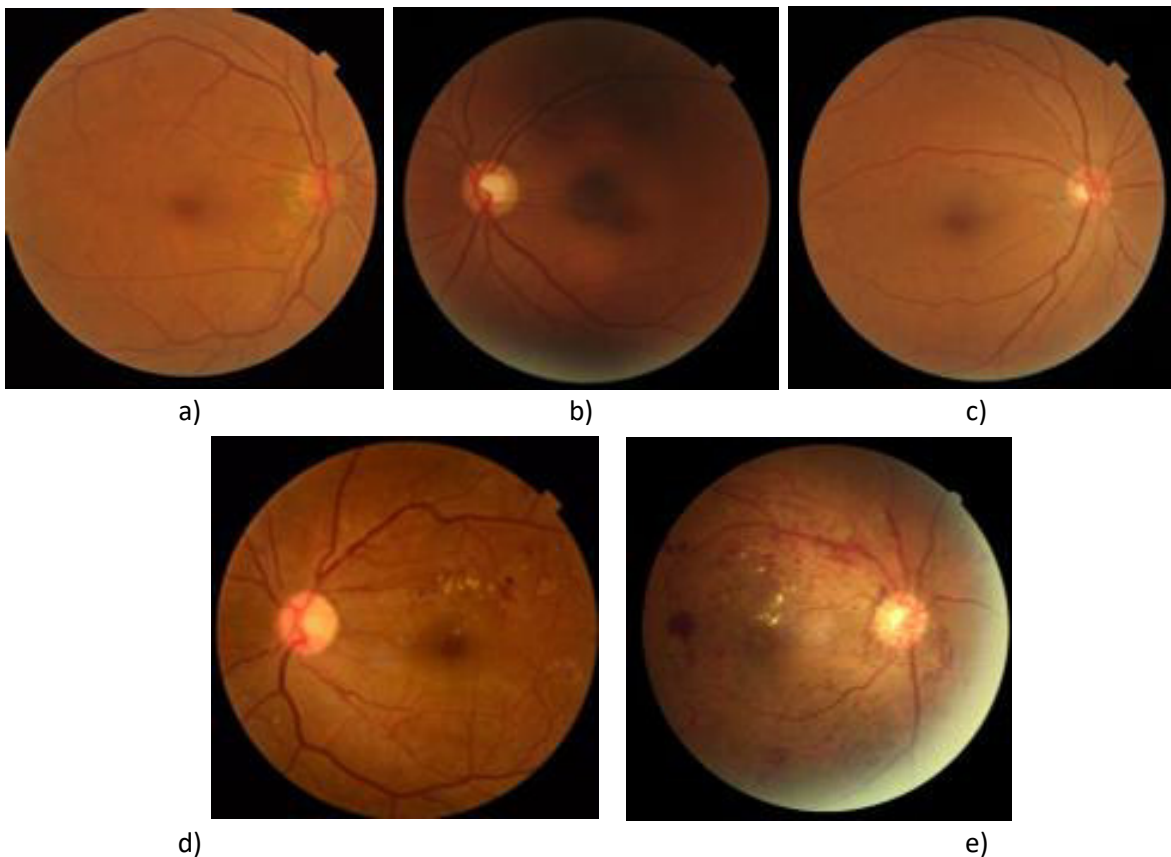




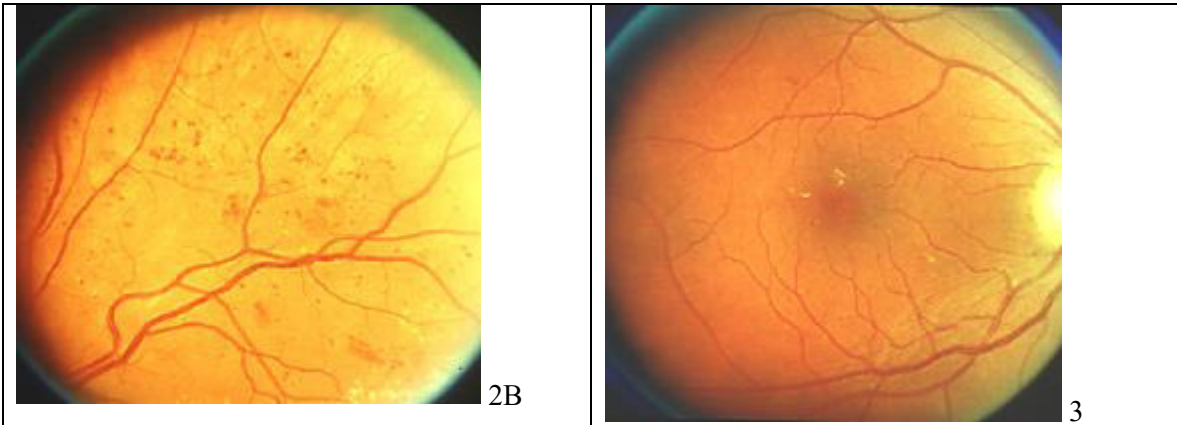
Figura 3. Etapa de la retinopatía diabética. a) Sano, b) Leve, c) Moderada, d) Severa, e) RD proliferativa. Se puede ver como avanza significativamente la enfermedad teniendo múltiples apariciones de las distintas lesiones existentes en esta patología.

3.3.2. Protocolo de graduación de la fotografía estándar

➤ Lesiones no proliferativas

Aunque las lesiones no proliferativas pueden variar desde muy leves hasta muy graves, generalmente no requieren tratamiento, a excepción del "Edema Macular Clínicamente Significativo (EMCS)". Determinar la gravedad de la retinopatía diabética es complicado, pero en general, se basa en la extensión o el área física que abarcan las lesiones. Las siguientes fotografías de retina mostradas a continuación (de la Fotografía 1 a la 15), son utilizadas en el Estudio de Tratamiento Temprano de la Retinopatía Diabética (ETDRS), considerado el estándar de oro para clasificar la gravedad de la enfermedad tanto en el ámbito clínico como en la investigación [30].

 <p>1</p>	 <p>2A</p>
<p>Fotografía estándar 1 de ETDRS. Muestra microaneurismas (Ma): puntos rojos más pequeños en el centro de la foto.</p> <p>Microaneurismas (Ma): Son las primeras lesiones que aparecen en la retinopatía diabética. Son dilataciones de los capilares. Estas lesiones aparecen como pequeños puntos rojos circulares que tienen márgenes bien definidos y no son más grandes que el grueso de los vasos sanguíneos en el margen del nervio óptico.</p>	<p>Fotografía estándar 2A de ETDRS. Muestra Ma y hemorragias (HE): puntos rojos más grandes dentro de la fotografía. Es posible observar algunos Ma y HE en esta fotografía. Estas lesiones frecuentemente ocurren juntas.</p> <p>Hemorragias de retina, microhemorragias (μHE) - Estas lesiones representan sangrado real dentro de la retina, y/o son el resultado de la ruptura de Ma o cuando los capilares se vuelven lo suficientemente permeables para permitir que la sangre de los vasos sanguíneos fluya hacia el exterior de éstos. Estas hemorragias pueden ser de una gran variedad de formas, incluyendo punto, mancha y en forma de flama. Generalmente más grandes que los Ma, con bordes irregulares / mal definidos.</p>



Fotografía estándar 4 ETDRS. Muestra lesiones en la mácula.



Los exudados duros: son lesiones blanco / amarillas por depósitos de colesterol que usualmente se originan por fugas de Ma. Estas lesiones son de forma irregular, afilados, varían en tamaño y tienen una apariencia "grasa o aceitosa". Con frecuencia están asociados con acumulación de líquido (edema retinal) dentro de la retina.

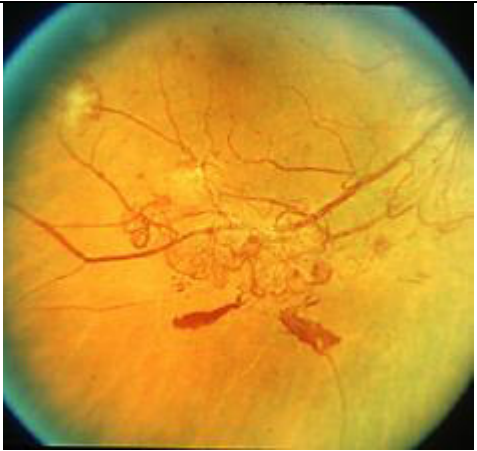
*Los exudados duros y el edema pueden ocurrir adyacente a la mácula. Cuando el edema se extiende en o muy cerca de la mácula (edema macular clínicamente significativo [EMCS]), la visión puede verse afectada y hará que la visión

Fotografía estándar 5 ETDRS. Muestra exudados blandos.

En esta fotografía se aprecian exudados blandos justo por encima del centro de la foto, a las 3 horas (a la derecha extrema de la foto) y a las 9:30 (justo por encima de la parte extrema izquierda de la foto).

Los exudados blandos, (también conocidos como "manchas algodonosas"): son formaciones redondeadas u ovales, blanco amarillentas, (a veces muy pequeños, discretamente grisáceas) de bordes imprecisos y localizados superficialmente. Son de tamaño variable y generalmente de mayor tamaño que los exudados duros.

<p>central se vuelva constantemente borrosa. El EMCS puede ocurrir en cualquier etapa de la retinopatía (no proliferativa o proliferativa), y por lo general se trata con láser.</p>	
 <p style="text-align: right;">6A</p>	 <p style="text-align: right;">6B</p>
<p>6A. Fotografía estándar 6A. Muestra rosarios venosos (RV). Define el límite inferior de la categoría no proliferativa moderada.</p>	<p>Fotografía estándar 6B ETDRS: Muestra anomalías microvasculares intrarretinianas (AMIR) y RV muy evidentes.</p> <p>AMIR- Se observan en el centro de la fotografía, pequeños vasos sanguíneos en forma de telaraña – AMIR tempranos-son muy sutiles, pero este es un ejemplo clásico y obvio.</p>
	<p>RV- Obsérvese el vaso principal a su paso por el centro de la foto y todo su trayecto izquierdo, (posición 9 del reloj).</p> <p>AMIR- ocurren en la retinopatía no proliferativa, éstos aparecen como vasos anormales (telarañas) que aparecen dentro de la retina. Por lo general con apariencia retorcida con esquinas agudas, a menudo cruzando sobre sí mismos.</p> <p>RV - También ocurren en las últimas etapas de la enfermedad no proliferativa, cuando las paredes de las venas pierden su alineación normal paralela y comienzan a parecer más como una fila de salchichas. Esta lesión es fuerte predictor de progresión a enfermedad proliferativa.</p>



7



8A

Fotografía estándar 7 ETDRS. Muestra neovasos.

Los nuevos vasos - también conocidos como neovasos (NV) o "neovascularización." La formación de neovasos define la fase proliferativa de la enfermedad. Estos neovasos nacen en la retina o en el nervio óptico y pueden extenderse sobre la retina o hacia el centro del ojo (vítreo). Representan la respuesta del ojo al proceso isquémico, son delicados, con alto riesgo de fuga y hemorragia. Pueden cruzar los principales vasos sanguíneos de la retina. A menudo parecen el botón de una flor con la parte exterior más dilatada que la interior (también llamados penachos). También pueden aparecer dentro de un área de fibrosis

Fotografía estándar 8A ETDRS. Muestra AMIR moderados. Define el límite inferior de la categoría moderada para AMIR. Los pacientes con retinopatía diabética no proliferativa severa tienen AMIR moderados en al menos un cuadrante.



8B


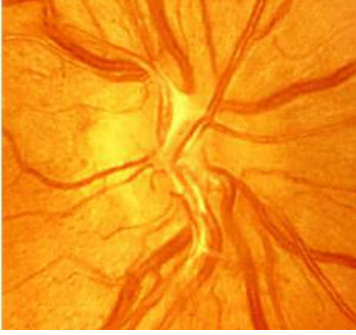

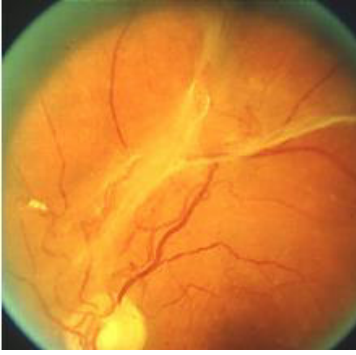





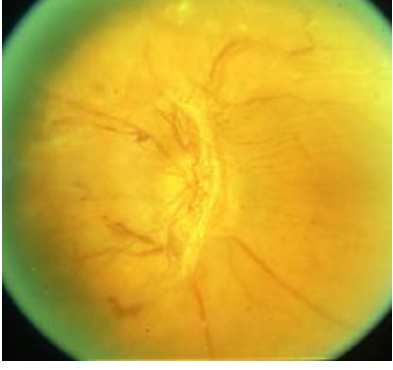
9

Fuente: [30]

➤ **Lesiones proliferativas**

Las lesiones proliferativas son signos avanzados de retinopatía diabética y en general, requieren tratamiento, así como un seguimiento cercano. El objetivo del tratamiento en este estado de la enfermedad es reducir la necesidad de oxígeno, lo que genera regresión de las lesiones proliferativas y evita la pérdida grave de la visión [30].

 <p>10A</p>	 <p>10B</p>
<p>Fotografía estándar 10A ETDRS. Define el límite inferior de la neovascularización moderada en el nervio óptico (NVD), los neovasos cubren aproximadamente un tercio del área del nervio óptico. Esta característica de NVD sitúa al paciente en retinopatía diabética proliferativa de alto riesgo.</p>	
 <p>10C</p>	 <p>11</p>
<p>Fotografía estándar 10C ETDRS. Muestra NVD</p>	<p>Fotografía estándar 11 ETDRS. Muestra proliferación fibrosa fuera del nervio óptico (PFE) asociada con neovasos fuera del nervio óptico (NVE)</p>
 <p>12</p>	 <p>13</p>

<p>Fotografía estándar 12 ETDRS. Muestra proliferación fibrosa (en el nervio óptico, FPD - o en otra parte, FPE) - Después de la aparición de NV, un tejido fibroso blanco crece adyacente o entrelazado con los nuevos vasos, la proliferación fibrosa (PF) tiende a adherirse a la retina y a los NV lo que les da la estructura para apoyo y crecimiento hacia la cavidad vítrea. Este tejido puede traccionar accidentalmente los NV y generar hemorragia o puede traccionar la retina y provocar un desprendimiento traccional.</p>	<p>Fotografía estándar 13 ETDRS. Muestra una hemorragia prerretinal. Nótese cómo el sangrado cubre los grandes vasos, lo que indica que se encuentra anterior a la retina, por lo tanto, prerretiniana. Hemorragia vítrea (HV) y/o prerretiniana (HPR) - Se produce cuando el NV se rompe y la sangre se filtra. Estas dos lesiones se separan en el nombre por su ubicación anatómica. Si esta sangre se filtra en el espacio potencial entre la retina y la membrana limitante interna, que recubre la superficie de la retina, se llama una hemorragia prerretinal. Si la sangre se filtra en el vítreo o en la cámara posterior, se llama hemorragia vítrea.</p>
 <p>14</p>	 <p>15</p>
	<p>Fotografía estándar 15 ETDRS. Muestra FPD con NVD asociada.</p>

Fuente: [30]

3.3.3. *Limitaciones de los métodos tradicionales de clasificación clínica*

La metodología aplicada en la actualidad posee una serie de limitantes a considerar:

Interpretación subjetiva: Los métodos tradicionales a menudo dependen de la interpretación subjetiva del médico, lo que puede llevar a variaciones en el diagnóstico [31].

Necesidad de expertos en oftalmología: Estos métodos requieren de especialistas altamente calificados, lo cual puede ser un desafío en áreas con escasez de estos profesionales.

Costo y tiempo: Los métodos clínicos actuales para la detección y clasificación de la retinopatía diabética son costosos y requieren mucho tiempo, lo que puede ser prohibitivo para la población con menos recursos [31].

Accesibilidad: Estos métodos requieren la presencia física tanto del paciente como del médico, aun y cuando la fotografía ya se adquirió, lo que puede ser inaccesible para quienes viven en áreas remotas o no pueden viajar.

Detección tardía: A menudo, la retinopatía diabética se detecta en etapas avanzadas cuando ya ha ocurrido un daño significativo. Los métodos de detección temprana pueden ser menos efectivos en personas que no presentan síntomas [32].

Estas limitaciones subrayan la necesidad de métodos de detección más eficientes, accesibles y objetivos, como los que podrían proporcionar las técnicas de inteligencia artificial.

3.4. Epidemiología de la retinopatía diabética (RD)

La prevalencia de la Diabetes está aumentando por la mayor sobrevivencia y el cambio en estilo de vida, llegando incluso a más del 10% en algunos países. Después de 20 años, el 90% de los casos de Diabetes tipo 1 y el 60% del tipo 2 sufrirán retinopatía y de ellas el 5% requerirá de tratamiento para evitar una ceguera irreversible [33].

La RD es la tercera causa de ceguera irreversible en el mundo, pero la primera en personas en edad productiva (16 a 64 años) en países en vías de desarrollo, generando grandes pérdidas económicas. Por esto es urgente desarrollar Programas Nacionales para la detección temprana de retinopatía. El riesgo de pérdida visual y ceguera se reduce con un control metabólico adecuado, una detección oportuna y tratamiento adecuado. Un examen periódico y el tratamiento de retinopatía no eliminan todos los casos de pérdida visual, pero reduce considerablemente el número de pacientes ciegos por esta dolencia.

La diabetes es causa principal de ceguera, aunque las personas con diabetes se hallan en mayor riesgo de desarrollar catarata o glaucoma, la retinopatía es el patrón más habitual de enfermedades en los ojos en estas personas [33].

La detección de la RD ha sido abordada por bastantes investigadores alrededor del mundo, haciendo uso de una gran variedad de conjuntos de imágenes con distintos grados de RD, con el fin de desarrollar herramientas que sean auxiliares en la detección temprana de la enfermedad.

El objetivo central de este trabajo es sumar al desarrollo de una herramienta que pueda ser eficiente para la tarea de detección temprana del grado de la RD, haciendo uso de algoritmos de clasificación potenciados con redes neuronales convolucionales (Inteligencia artificial), adicional a mostrar un algoritmo capaz de hacer detección de lesiones, tales como microaneurismas (MA), hemorragias (HE), exudados suaves (SE), exudados duros (EX); todas en una sola inferencia.

4. Inteligencia Artificial y Técnicas

4.1. *Consideraciones éticas del uso de IA*

La IA ya se ha utilizado en la atención médica en numerosas aplicaciones, aunque aún no se ha aprovechado todo su potencial. En esta etapa inicial, la integración ya ha generado una serie de beneficios. Por ejemplo, los sistemas impulsados por IA son capaces de analizar grandes conjuntos de datos en poco tiempo, incluidos registros médicos e imágenes de diagnóstico, lo que les permite ayudar a los profesionales de la salud a realizar diagnósticos más precisos en períodos de tiempo cortos.

Sin embargo, aún existen algunas preocupaciones éticas relacionadas a la atención médica impulsada por la IA, que se confía poco a poco se disipen.

4.1.1. *Abordar el sesgo algorítmico*

Los algoritmos de aprendizaje automático pueden verse afectados por sesgos inherentes a los datos de entrenamiento, lo que puede resultar en disparidades en el diagnóstico y tratamiento de pacientes de diferentes grupos demográficos, esta es una limitante que se reduce aumentando los datos de entrenamiento y la variabilidad de estos, incluso con el propio uso y autoaprendizaje que se va teniendo con el tiempo [34].

4.1.2. *Toma de decisiones informada*

Los pacientes deben ser informados sobre el uso de algoritmos de aprendizaje automático en su atención médica y tener la oportunidad de dar su consentimiento informado para su uso [34].

4.1.3. *Privacidad y seguridad de los datos*

Los datos médicos utilizados para entrenar y validar los algoritmos deben manejarse con cuidado para garantizar el resguardo seguro de la información de los pacientes.

4.1.4. *Supervisión humana*

Los profesionales de la salud deben garantizar que las decisiones de atención médica impulsadas por herramientas de inteligencia artificial se alineen con los intereses del paciente. Se debe mantener la supervisión humana de las herramientas de IA utilizadas en la atención médica y los profesionales de la salud deben evaluar críticamente las recomendaciones dadas por las herramientas de IA para garantizar que se alineen con la experiencia clínica. Además, los profesionales sanitarios deben ser transparentes sobre el uso de la IA en la asistencia sanitaria. Es importante que los pacientes comprendan cómo se aprovechan las herramientas para influir en su atención [34].

4.2. *Machine Learning*

En la detección de objetos y clasificación, se suelen emplear técnicas de aprendizaje automático o Machine Learning. Se trata de una disciplina informática que consiste en crear sistemas que pueden aprender por sí mismos. Todo ello, está relacionado con el desarrollo de inteligencia artificial. Todo esto tiene como beneficio, que muchas tareas complejas se puedan desarrollar de manera automática, reduciendo la intervención de personas en el proceso para el que esté destinado el sistema. El sistema aprende, es decir, es capaz de identificar una serie de patrones complejos a partir de los datos. Todo esto lo hace mediante un algoritmo de aprendizaje, ya que se introducen una serie de datos, y a partir de ellos, éste es capaz de generar una hipótesis que le permita realizar predicciones cuando se enfrenta a nuevas entradas [35].

Existen diversos tipos de aprendizaje automático:

- **Aprendizaje supervisado:** Se trata de un tipo de aprendizaje donde al sistema se le alimenta con datos de entrenamiento etiquetados. Una vez se ha completado el entrenamiento, se pueden introducir nuevos datos al sistema de manera que ya no están etiquetados, puesto que al estar entrenado el sistema es capaz de reconocer patrones para identificar los datos correspondientes. En este tipo de aprendizaje están los denominados problemas de clasificación y de regresión [35].
- **Aprendizaje no supervisado:** En este tipo de aprendizaje se alimenta el sistema sin datos etiquetados. Uno de los problemas más comunes de este tipo de aprendizaje automático es el de agrupamiento o clustering, en el que el sistema aprende a organizar los datos en distintos grupos, de modo que cada grupo contenga datos similares entre sí [35].
- **Aprendizaje por refuerzo:** Se trata de que el sistema aprenda a partir de la experiencia. Esto es que, el sistema al aprender, si se equivoca se le penaliza de forma que el sistema va aprendiendo con el objetivo de maximizar la recompensa. Es algo muy parecido al aprendizaje humano, cuando se enseña a los niños conceptos o conductas y si lo hacen correctamente se les recompensa y si no se les penaliza [36]. Es una idea muy similar

En el caso de esta investigación YOLOv8 es una tecnología de aprendizaje automático supervisado, debido a que se utilizan datos previamente etiquetados para su entrenamiento.

4.3. Deep Learning

El aprendizaje profundo o deep learning es un campo dentro del área del Machine Learning. Se trata de un paradigma de aprendizaje automático que intenta asemejarse a la forma del aprendizaje humano para poder aprender determinadas tareas [37]. Para el procesamiento de la información, usa redes neuronales artificiales, pero concatenando un gran número de capas formadas por unidades de procesamiento sencillas (neuronas), que van transformando progresivamente la información. Hay varios tipos de capas: la capa de entrada a la red, las capas ocultas, y la capa de salida.

La capa de entrada sirve como entrada de la información a la red neuronal, las capas ocultas permiten el procesamiento de la información, descomponiendo la información de entrada, y la capa de salida es la que realiza la toma de decisiones con base en el procesamiento realizado en las capas anteriores [37].

4.3.1. Problemas de clasificación

Uno de los problemas más habituales son los de clasificación, los cuales se tratan de problemáticas que a partir de datos concretos y en función de los parámetros con los que se configure el sistema, éste sea capaz de realizar una clasificación de forma adecuada, en función de los datos con los que fue entrenado.

En este trabajo utilizamos clasificación de imágenes de diferentes etapas de RD en fondo de ojo, para poder entrenar un algoritmo y este sea capaz de distinguir los tipos de categorías de la enfermedad.

4.3.2. Problemas de detección de objetos

Esta problemática del machine Learning es mucho más complejo que el de clasificación, debido a que consiste en no solo decir si un objeto se encuentra o no presente en una imagen y/o vídeo, sino que también debe delimitar la región de la imagen donde se encuentra el objeto de interés, generando lo que se conoce como un cuadro delimitador o bounding box.

En este proyecto la detección de objetos entra en juego a la hora de discriminar las distintas lesiones que aparecen en la RD, la dificultad de dicha tecnología radica en la eficacia que posee para detectar objetos grandes y pequeños.

4.4. Redes neuronales

Para entender la red neuronal es necesario explicar lo que es una TLU (threshold logic unit) o unidad lógica de umbral. Se trata de una neurona artificial, que se compone de una serie de entradas y de una salida. A cada entrada se le asignará un peso de forma que la TLU calcule la suma ponderada de sus entradas para obtener una salida, como podemos ver en la Figura 4 [38].

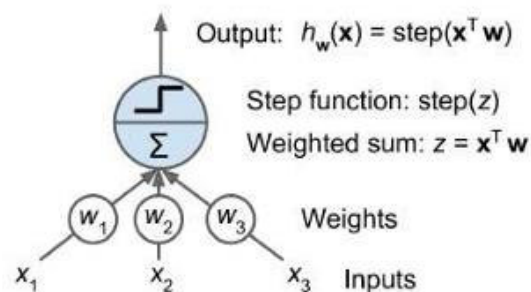


Figura 4. Estructura de un TLU o unidad lógica de umbral. En esta figura, se muestra cómo una neurona de una red neuronal recibe tres entradas (x_1 , x_2 , x_3), cada una con un peso asociado (w_1 , w_2 , w_3). La neurona suma estas entradas multiplicadas por sus pesos, obteniendo un valor total (z). Luego, la neurona aplica una función de umbral: si z es mayor o igual a un cierto valor, la salida es 1; de lo contrario, la salida es 0. Esta función de umbral ayuda a tomar decisiones simples, como clasificar datos en una de dos categorías. [39]

Un perceptrón, es una capa de TLUs con cada TLU conectada a todas las entradas. A continuación, en la Figura 5 apreciamos una serie de TLU que en este caso forman parte de la capa de salida, y además están completamente conectadas a sus entradas [38].

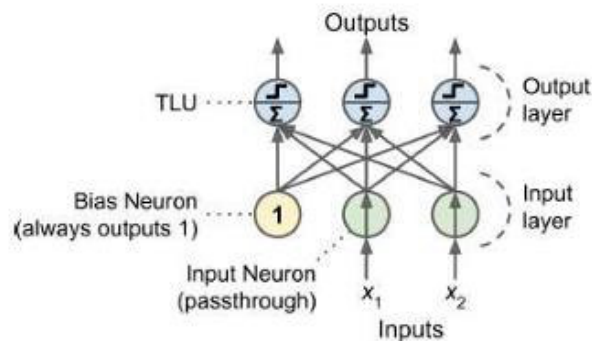


Figura 5. Composición de un perceptrón. El diagrama muestra una red neuronal simple con una Capa de entrada y una Capa de salida. En la capa de entrada, se incluyen neuronas de entrada (x_1 y x_2) que pasan los valores de entrada, y una Neurona de sesgo (Bias Neuron) que siempre tiene una salida de 1. Las neuronas en la capa de salida están conectadas a todas las neuronas de la capa de entrada, incluidas las de sesgo. Cada neurona de salida aplica una suma ponderada de sus entradas seguida de una función de activación lineal umbral (TLU). Las conexiones entre las capas tienen pesos que se ajustan durante el entrenamiento para minimizar el error de predicción. Este esquema básico es la base de redes neuronales más complejas. [39].

Cada TLU se activará en función de si la suma ponderada de las entradas tiene un valor mayor o menor que un umbral. De esta forma si la suma ponderada es mayor que el umbral, tendremos una

salida de valor 1, mientras que, si es menor que el umbral, la salida será nula o bien -1, todo dependiendo del tipo de función de activación [38]. Las funciones de activación dan un valor de salida según sea el valor que tienen a la entrada. Estos valores de salida pueden ser entre 0 y 1 o bien entre -1 y 1 [39].

Finalmente tenemos los MLP o multilayer perceptron, que contiene una capa de entrada, una o más capas ocultas y una capa de salida, donde las capas cercanas a la de entrada se llaman capas inferiores y las que están cerca de la salida se llaman superiores. Esto se ve con más detalle en la Figura 6 [38].

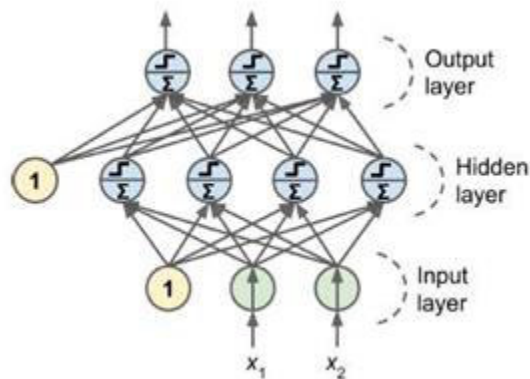


Figura 6. Estructura de un Perceptrón Multicapa (MLP). El diagrama muestra una red neuronal con tres capas: la Capa de entrada, la Capa oculta y la Capa de salida. Los nodos de la capa de entrada (x_1 y x_2) representan las características de entrada. La capa oculta contiene nodos que procesan estas entradas mediante sumas ponderadas y funciones de activación, generando características internas que no son visibles directamente. Finalmente, la capa de salida produce el resultado final de la red, que puede ser utilizado para tareas de clasificación o regresión. Cada conexión entre nodos tiene un peso asociado, que se ajusta durante el entrenamiento del modelo para minimizar el error en las predicciones. La estructura MLP es un ejemplo de red neuronal feedforward, donde la información fluye en una sola dirección, de la entrada a la salida. Fuente: [39].

En una red neuronal para detección de objetos, se utilizan redes neuronales convolucionales debido a su capacidad para procesar imágenes de manera eficiente. En este tipo de red, las entradas son los píxeles de cada imagen. A diferencia de las redes totalmente conectadas, en las redes convolucionales la primera capa no se conecta a todos los píxeles de la imagen de entrada. En cambio, cada neurona en esta capa está conectada solo a una pequeña porción de la imagen, lo que permite a la red captar características locales y patrones específicos dentro de la imagen, que podemos apreciar en la Figura 7 [38].

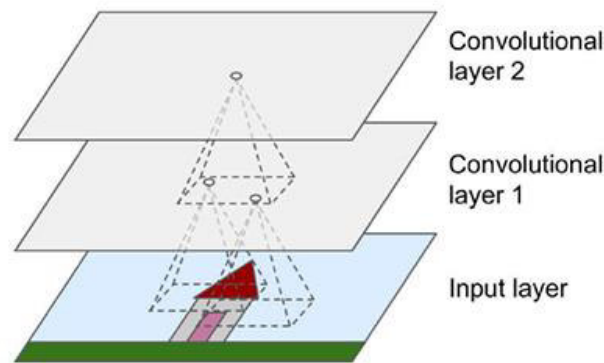


Figura 7. Esquema del funcionamiento de una red neuronal convolucional (CNN). La imagen muestra cómo las capas convolucionales procesan la Capa de entrada. La Capa convolucional 1 extrae características básicas, como bordes y texturas, de la imagen de entrada. Luego, la Capa convolucional 2 combina estas características para detectar patrones

más complejos, como formas y objetos. Cada capa convolucional en la red aplica filtros para identificar diferentes aspectos de la imagen, construyendo una representación jerárquica de los datos de entrada. Este proceso de extracción de características es fundamental en tareas de visión por computadora, como el reconocimiento de objetos y la clasificación de imágenes. Fuente: [39].

4.4.1. Redes neuronales convolucionales

Las redes neuronales convolucionales están basadas en la operación de convolución descrita en la Ecuación 1, donde X es la entrada y W es el filtro convolucional. Es posible ajustar el paso L de la convolución para abarcar un área más grande con el mismo filtro convolucional [40].

$$\text{conv}(X, W)_{x,y} = \sum_{i=0}^n \sum_{j=0}^n W_{i,j} X_{x+Li, y+Lj}$$

Ecuación 1. Operación de convolución.

Las redes neuronales convolucionales están compuestas por capas convolucionales, capas de reducción y dependiendo de la arquitectura, capas completamente conectadas. Cabe destacar que, a diferencia de una red neuronal completamente conectada como el perceptrón multicapa, una red neuronal convolucional procesa la información por regiones (ver Figura 8) disminuyendo el tiempo de ejecución en las primeras capas que usualmente son las más grandes [40].

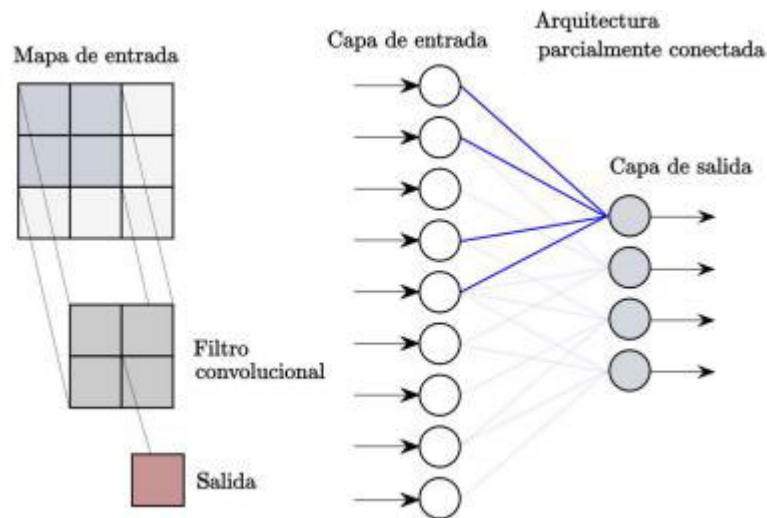


Figura 8. Conexiones parcialmente conectadas en una red neuronal convolucional (CNN). A la izquierda, se muestra un Mapa de entrada con un Filtro convolucional aplicado, resultando en una salida que representa una parte específica de la entrada original. A la derecha, se ilustra la Arquitectura parcialmente conectada, donde las neuronas de la Capa de entrada están conectadas sólo a un subconjunto de neuronas de la Capa de salida. Esto es característico de las CNNs, donde cada filtro convolucional aprende a detectar características locales específicas en la entrada, reduciendo la complejidad de la red y mejorando la eficiencia al evitar conexiones completas entre capas. Fuente: [40].

A los pesos sinápticos de una red convolucional se les denomina filtros ya que dada una entrada el resultado es una versión que puede resaltar bordes, difuminar áreas u otro tipo de filtro. A estas salidas procesadas por los filtros convolucionales se denominan mapas de características [40].

4.4.2. Características del entrenamiento de CNN

El entrenamiento de una red neuronal puede entenderse como un problema de optimización. El objetivo de este problema es encontrar los pesos y sesgos que minimicen la función de costo. Estos parámetros se ajustan durante el entrenamiento [41].

Función de costo: Indica qué tan cerca está la predicción obtenida de la solución real. Normalmente, las funciones de costo se calculan haciendo un promedio de los costos de los N ejemplos de entrenamiento [41].

- **Descenso del gradiente:** Es un método iterativo para encontrar los mínimos locales de una función. En el campo de las redes neuronales, el descenso del gradiente se usa para minimizar la función de costo actualizando los pesos de la red y disminuyendo la desviación de la salida. Para ello, es de vital importancia la tasa de aprendizaje (learning rate), que define cuánto afecta el gradiente a la actualización de los parámetros en cada iteración. Una tasa de aprendizaje muy baja resultará en una convergencia extremadamente lenta, mientras que una tasa muy alta impedirá que el algoritmo converja [41].
- **Propagación hacia atrás:** Las redes neuronales actualizan sus pesos a medida que aprenden durante el proceso de entrenamiento. Este proceso se denomina backpropagation. El objetivo es calcular las derivadas parciales de los pesos y los sesgos [41].
- **Regularización o dropout:** Uno de los problemas más comunes en las redes neuronales convolucionales (CNN) es el sobreajuste (overfitting). Ocurre cuando el modelo se ajusta tanto a los datos de entrenamiento que no puede generalizar bien para los datos de prueba. El objetivo de los modelos de aprendizaje automático es obtener patrones de los datos de entrenamiento que luego sean aplicables a nuevos datos. El overfitting sucede cuando un modelo aprende demasiado bien los detalles y ruidos de los datos de entrenamiento, de manera que no puede generalizar a datos nuevos. Es como si alguien estudiara para un examen memorizando todas las preguntas del año pasado: podría obtener una buena nota si las preguntas son iguales, pero fallaría si cambian. El underfitting, por el contrario, se produce cuando el modelo se entrena con pocos datos o con un número demasiado pequeño de parámetros. En otras palabras, el underfitting ocurre cuando el modelo no logra captar los patrones importantes en los datos de entrenamiento, lo que lo hace ineficaz tanto con los datos de entrenamiento como con los nuevos. Es como si alguien estudiara para un examen solo leyendo un resumen superficial: no entendería lo suficiente para responder bien, incluso si las preguntas fueran sencillas. Para detectar tanto el underfitting como el overfitting, se divide el conjunto de datos de entrada en dos subconjuntos: uno para entrenamiento y otro para prueba. La regularización o dropout consiste en suprimir aleatoriamente un porcentaje de neuronas y sus conexiones en la red neuronal durante el entrenamiento. Es decir, la red activa aleatoriamente algunas neuronas mientras que otras permanecen desactivadas para un batch durante las etapas de forward y backward propagation; luego se reactivan y se vuelven a desactivar aleatoriamente para el siguiente batch [41].

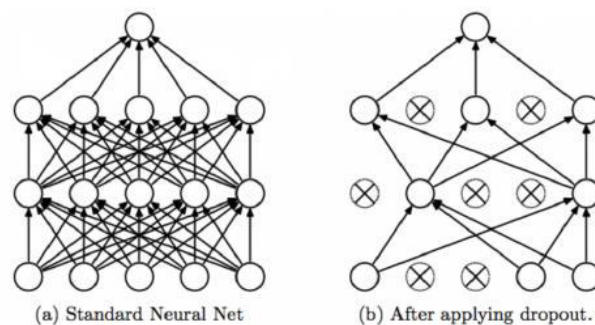


Figura 9. Comparación de una red neuronal antes y después de aplicar dropout. En la subfigura (a), se muestra una Red Neuronal Estándar con todas las conexiones completamente conectadas entre las capas. En la subfigura (b), se ilustra la red Después de aplicar Dropout, donde se han eliminado aleatoriamente algunas neuronas y sus conexiones (marcadas con un "X"), para prevenir el sobreajuste y mejorar la generalización del modelo. Dropout es una técnica de regularización que, durante el entrenamiento, desconecta temporalmente un porcentaje de neuronas, reduciendo la complejidad de la red y ayudando a evitar el aprendizaje de patrones espurios en los datos de entrenamiento [41].

- **Normalización del batch:** La normalización en lotes consiste en añadir un paso extra entre las neuronas y la función de activación para normalizar las activaciones de salida. La normalización se realiza utilizando la media y la varianza de todo el conjunto de entrenamiento, excepto cuando se emplea el descenso del gradiente estocástico, que utiliza la media y la varianza de cada mini-lote de entrada. Esta técnica ayuda a que la convergencia hacia el mínimo global se produzca más rápidamente [41].
- **Data Augmentation:** El aumento de datos consiste en aplicar transformaciones sobre las entradas originales, obteniendo muestras ligeramente diferentes [41].

4.4.3. Single Shot Detector (SSD)

YOLOv8 es una tecnología que utiliza esta clase de arquitectura, la cual consiste en una red neuronal convolucional de tipo piramidal, lo que le permite la detección de objetos grandes y pequeños. Concretamente utiliza múltiples capas de detección, con lo que consigue múltiples mapas de características, cada mapa para una escala diferente. La red neuronal convolucional va reduciendo la dimensión espacial gradualmente, y con ello, la resolución de los mapas también disminuye. Con ello, los mapas de menor resolución se utilizan para la detección de objetos de mayor escala, mientras que los de mayor resolución se usan para los objetos pequeños. La arquitectura de la red de SSD, se observa en la Figura 10, donde usa la red VGG16 para extraer características y mediante capas de convolución va realizando detecciones [39].

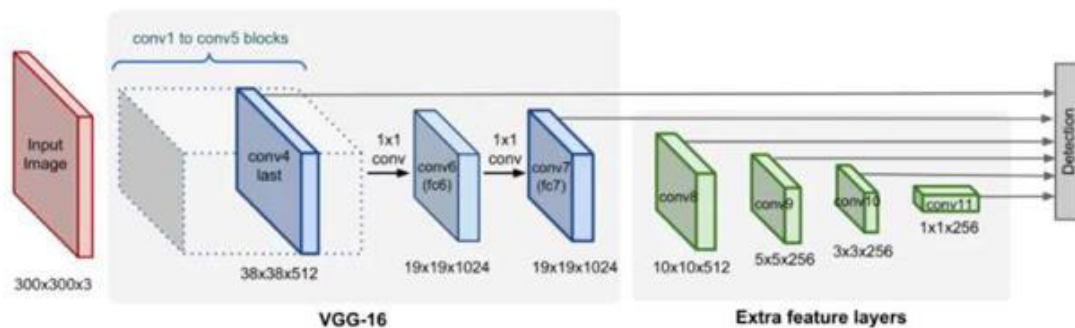


Figura 10. Arquitectura de la red SSD utilizando VGG-16 como extractor de características. La imagen de entrada con dimensiones 300x300x3 pasa a través de la red VGG-16, que incluye bloques convolucionales desde conv1 hasta conv5. Después del último bloque convolucional (conv4), las características extraídas pasan a través de una serie de capas convolucionales adicionales, incluyendo conv6 y conv7 (también conocidas como fc6 y fc7), que reducen la dimensionalidad a 19x19x1024. Luego, se agregan capas de características extra (Extra Feature Layers), que incluyen conv8, conv9, conv10 y conv11, con tamaños de filtro y números de canales variados. Estas capas adicionales permiten la detección de objetos a diferentes escalas. Finalmente, las salidas de estas capas se combinan para realizar la Detección de objetos, generando predicciones para la presencia y ubicación de objetos en la imagen. Fuente: [39].

Las predicciones se componen de un cuadro delimitador y de N+1 puntuaciones de clase, siendo N el número de clases que puede detectar, más una que no se corresponde a ningún objeto. Además, SSD no utiliza la propuesta de región como pasaba en el caso de R-CNN, sino que va obteniendo las puntuaciones y ubicaciones mediante pequeños filtros de convolución, después de extraer los mapas de características. En VGG16, se agregan 6 capas más, en las que 5 de ellas serán para la detección de objetos. Con estos mapas de características múltiples se logra una gran mejora de la precisión [39].

5. Inteligencia Artificial Aplicada - Arquitectura YOLO

5.1.1. YOLO

You Only Look Once (YOLO) es un algoritmo de detección de objetos introducido en 2015 por Joseph Redmon, Santosh Divvala, Ross Girshick y Ali Farhadi. La arquitectura YOLO fue una revolución significativa en el espacio de detección de objetos en tiempo real, superando a su predecesora: la red neuronal convolucional basada en regiones (R-CNN). YOLO es un algoritmo de disparo único, que clasifica directamente un objeto en un solo paso, haciendo que solo una red neuronal prediga cuadros delimitadores y probabilidades de clase utilizando una imagen completa como entrada. El modelo familiar YOLO está en continua evolución. Desde entonces, varios equipos de investigación han lanzado diferentes versiones de YOLO [42], siendo YOLOv8 la versión que aquí se eligió por ser la versión más reciente al momento de iniciar el desarrollo de este trabajo.

5.1.2. Funcionamiento básico de los detectores de objetos YOLO

La *Figura 11* muestra la mecánica esencial de un modelo de detección de objetos con YOLO:

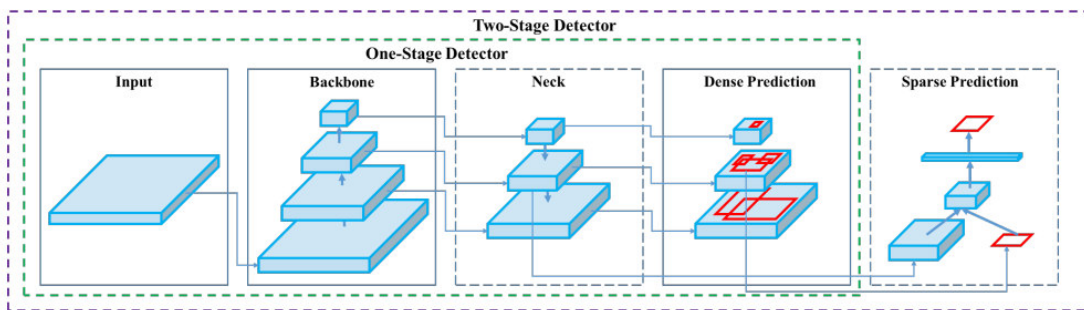


Figura 11. Diagrama ilustrativo del funcionamiento de un modelo para detección de objetos. El flujo de trabajo comienza con la Entrada (Input), que representa la imagen que se procesará. Luego, la imagen pasa por el Backbone, una serie de capas que extraen características importantes de la imagen. A continuación, las características se procesan en el Neck, que refina y organiza la información extraída. En la etapa de Predicción Densa (Dense Prediction), se generan predicciones a partir de un gran número de posibles ubicaciones de objetos en la imagen. Finalmente, en la etapa de Predicción Escasa (Sparse Prediction), se refinan y seleccionan las predicciones más relevantes, eliminando redundancias y ajustando las ubicaciones de los objetos. El diagrama distingue entre detectores de una y dos etapas, destacando las diferencias en el proceso de predicción y refinamiento. Fuente: [43].

Las arquitecturas de los modelos de YOLO tienen componentes similares que se describen a continuación:

Backbone: Es una red neuronal convolucional que acumula y produce características visuales con diferentes formas y tamaños. Los modelos de clasificación como ResNet, VGG y EfficientNet se utilizan como extractores de características [44].

Neck: Conjunto de capas que integran y combinan características antes de pasarlas a la capa de predicción, ejemplo: red piramidal de funciones (FPN), red de agregación de rutas (PAN) y Bi-FPN [44].

Head: Incorpora características del cuello junto con las predicciones del cuadro delimitador. Realiza clasificación junto con regresión de las características y coordenadas del cuadro delimitador para

completar el proceso de detección. Genera 4 valores, generalmente coordenadas x,y junto con el ancho y el alto [44].

5.1.3. YOLOv8

Ultralytics YOLOv8 es un modelo de vanguardia que se basa en el éxito de versiones anteriores de YOLO e introduce nuevas características y mejoras para impulsar aún más el rendimiento y la flexibilidad. YOLOv8 está diseñado para ser rápido, preciso y fácil de usar, lo que lo convierte en una excelente opción para una amplia gama de tareas de detección y seguimiento de objetos, segmentación de instancias, clasificación de imágenes y estimación de poses (puntos clave) [45].

Cada variante de la serie YOLOv8 está optimizada para una tarea específica, lo que garantiza un alto rendimiento y precisión. Además, estos modelos son compatibles con varios modos operativos, incluidos Inferencia, Validación, Entrenamiento y Exportación, lo que facilita su uso en diferentes etapas de implementación y desarrollo.

YOLOv8 (modelo mediano) tiene una puntuación mAP de 50.2 a 1.83 milisegundos en el conjunto de datos COCO y A100 TensorRT. YOLO v8 también presenta un paquete de Python y una implementación basada en CLI, lo que facilita su uso y desarrollo [42]. La arquitectura de YOLOv8 se presenta en la Figura 14. Dentro de las características que más destaca YOLOv8 son:

- **Aumento de mosaico**

La investigación en deep learning tiende a centrarse en la arquitectura del modelo, pero la rutina de entrenamiento en YOLOv5 y YOLOv8 es una parte esencial de su éxito. YOLOv8 aumenta las imágenes durante el entrenamiento en línea. En cada época, el modelo ve una variación ligeramente diferente de las imágenes que se le han proporcionado. Una de esas técnicas de aumento se llama aumento de mosaico. Esto implica unir cuatro imágenes, lo que obliga al modelo a aprender objetos en nuevas ubicaciones, en oclusión parcial y contra diferentes píxeles circundantes. Sin embargo, se ha demostrado empíricamente que este aumento degrada el rendimiento si se realiza durante toda la rutina de entrenamiento. Es ventajoso desactivarlo durante las últimas diez épocas de entrenamiento [46].

En la Figura 12 se observa una representación de como luce dicho aumento de datos de mosaico.

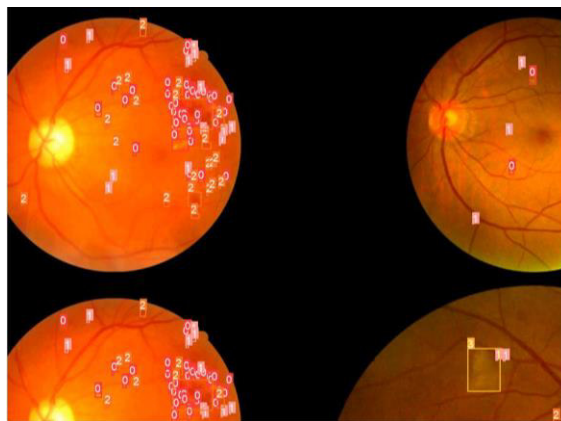


Figura 12. Aumento de mosaico en fotografías de RD durante el proceso de entrenamiento. (Detector de lesiones)

- **Anchor-Free Detection**

YOLOv8 es un modelo "anchor-free" (libre de anclas). Esto significa que predice directamente el centro de un objeto en lugar del desplazamiento desde una caja de ancla conocida. Las cajas de ancla

eran una parte notoriamente complicada de los modelos YOLO anteriores, ya que podían representar la distribución de las cajas del conjunto de datos de referencia, pero no la distribución del conjunto de datos personalizado. La detección sin anclas reduce el número de predicciones de cajas, lo que acelera la supresión no máxima (NMS), un complicado paso de posprocesamiento que filtra las detecciones candidatas después de la inferencia [46].

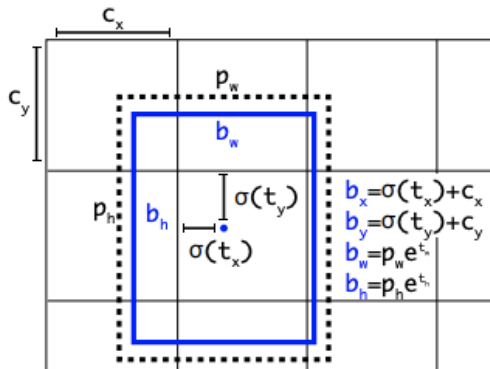


Figura 13. Visualización de una caja libre de anclas en YOLO. Fuente: [46].

- **C2f Module**

El backbone del modelo ahora consiste en un módulo C2f en lugar de uno C3. La diferencia entre los dos es que en C2f, el modelo concatena la salida de todos los módulos bottleneck. En contraste, en C3, el modelo utiliza la salida del último módulo bottleneck. Un módulo bottleneck consiste en bloques residuales de bottleneck que reducen los costos computacionales en las redes de deep learning. Esto acelera el proceso de entrenamiento y mejora el flujo del gradiente [42].

- **Cabeza desacoplada**

El diagrama de la *Figura 14* ilustra que la cabeza (head) ya no realiza la clasificación y la regresión juntas. En cambio, realiza las tareas por separado, lo que aumenta el rendimiento del modelo.

- **Loss (Pérdida)**

La desalineación es posible ya que la cabeza desacoplada separa las tareas de clasificación y regresión. Esto significa que el modelo puede localizar un objeto mientras clasifica otro. La solución es incluir una puntuación de alineación de tareas basada en la cual el modelo conoce una muestra positiva y negativa. La puntuación de alineación de tareas multiplica la puntuación de clasificación con la puntuación de Intersección sobre Unión (IoU). La puntuación IoU corresponde a la precisión de una predicción de cuadro delimitador. Con base en la puntuación de alineación, el modelo selecciona las k mejores muestras positivas y calcula una pérdida de clasificación utilizando BCE y una pérdida de regresión utilizando IoU Completa (CIoU) y Pérdida Focal Distribucional (DFL). La pérdida BCE simplemente mide la diferencia entre las etiquetas reales y las predichas. La pérdida CIoU considera cómo el cuadro delimitador predicho es relativo a la verdad fundamental en términos del punto central y la relación de aspecto. En contraste, la pérdida focal distribucional optimiza la distribución de los límites del cuadro delimitador al enfocarse más en las muestras que el modelo clasifica erróneamente como falsos negativos [42].

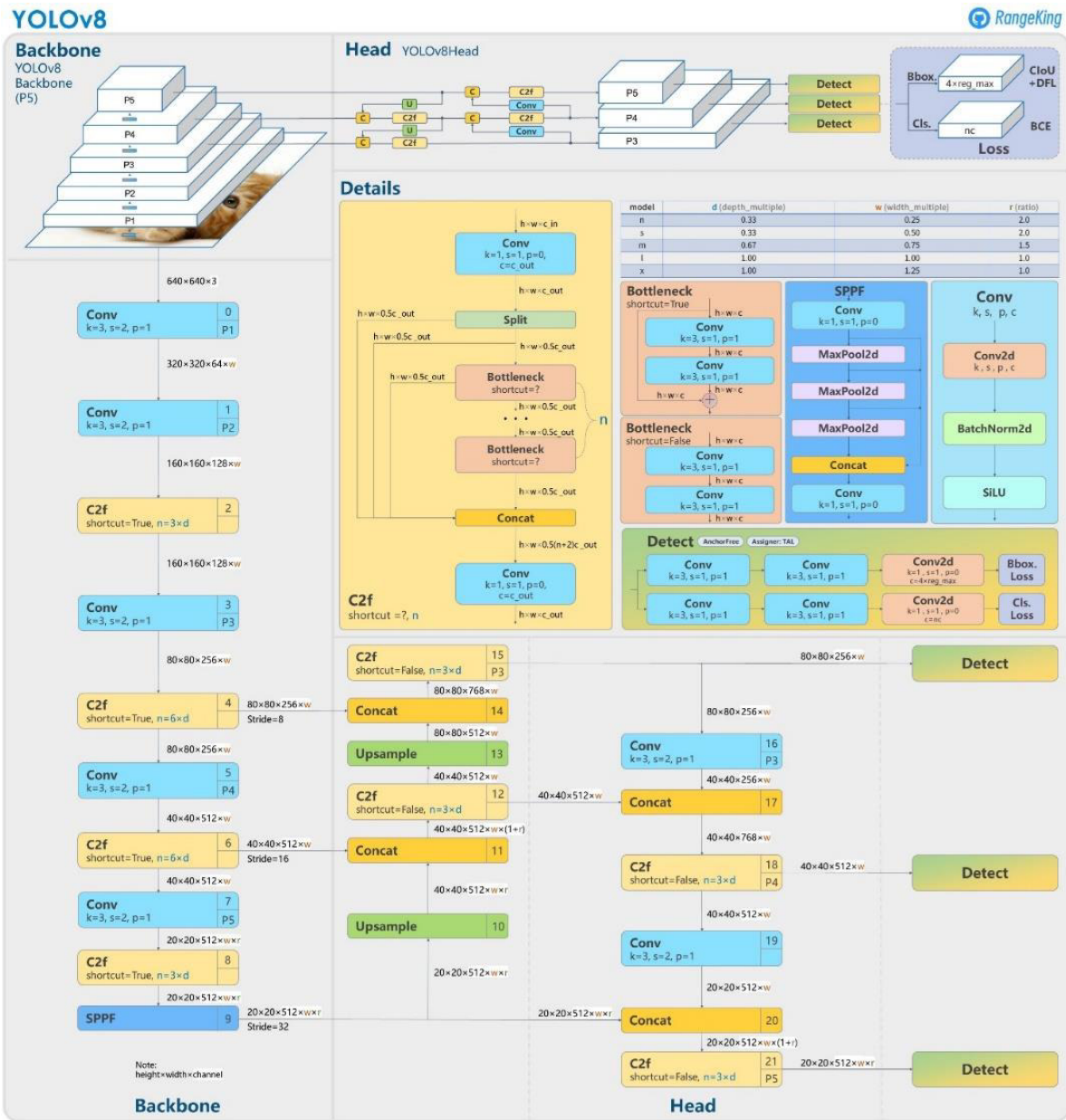


Figura 14. Arquitectura de YOLOv8, creada por el usuario de github RangeKing. Fuente: [47].

5.1.4. Proceso de detección de objetos de YOLOv8

División en cuadrícula: La imagen se divide en una cuadrícula de celdas. Cada celda es responsable de predecir la presencia de objetos dentro de su región asignada.

Extracción de características: Se extraen características de cada celda utilizando la red neuronal convolucional.

Predicción de cajas delimitadoras y confianzas: La red predice una caja delimitadora y una puntuación de confianza para cada objeto potencial dentro de cada celda. La puntuación de confianza indica la probabilidad de que exista un objeto en la caja delimitadora de predicción.

No máxima supresión: Se elimina la redundancia eliminando cajas delimitadoras con alta superposición que predicen el mismo objeto.

5.1.5. Métricas de rendimiento

A continuación, en la Tabla 2, se detallan las métricas de rendimiento consideradas en los experimentos de clasificación y detección.

Tabla 2. Métricas de rendimiento más relevantes para los experimentos realizados.

Medida	Fórmula	Descripción
Sensitividad (Se o recall)	$\frac{TP}{TP + FN}$	Relación entre los verdaderos positivos clasificados y los verdaderos positivos reales en la verdad básica. También conocido como tasa de verdaderos positivos (TPR). Donde TP son los verdaderos positivos y FN los falsos negativos.
Exactitud (Acc)	$\frac{TP + TN}{TP + TN + FP + FN}$	La exactitud mide la proporción de píxeles clasificados correctamente, con respecto al número total de píxeles en el campo de visión de la imagen (FOV).
Precisión (Pr)	$\frac{TP}{TP + FP}$	El porcentaje de casos positivos detectados.
Curva ROC	A modo de guía para interpretar las curvas ROC se han establecido los siguientes intervalos para los valores de AUC: [0.5]: Es como lanzar una moneda. [0.5, 0.6): Test malo. [0.6, 0.75): Test regular. [0.75, 0.9): Test bueno. [0.9, 0.97): Test muy bueno. [0.97, 1): Test excelente.	Un gráfico de curva ROC ilustra la sensibilidad y especificidad de cada uno de los posibles puntos de corte de un test diagnóstico cuya escala de medición es continua. En otras palabras, muestra la relación entre la tasa de verdaderos positivos (TPR) y la tasa de falsos positivos (FPR) del modelo.
Área bajo la curva (AUC)	$\frac{Se + Sp}{2}$	Área cubierta por la curva ROC cuando se logra la optimización.
Puntuación-F1	$2 * \frac{Se + Pr}{(Se + Pr)}$	Es el promedio ponderado de precisión y recuerdo. Tiene en cuenta tanto los falsos positivos como los falsos negativos.
Coefficiente Kappa	$\frac{p_o - p_e}{1 - p_e}$	El coeficiente Kappa refleja la concordancia interobservador, puede tomar valores entre -1 y +1. Mientras más cercano a +1, mayor es el grado de concordancia interobservador, por el contrario, mientras más cercano a -1, mayor es el grado de discordancia interobservador. Donde p_o es la proporción de acuerdo observado y p_e la proporción de acuerdo esperado por azar

Fuente: [48]

6. Materiales y Métodos

En esta sección se presentan todos los materiales y métodos empleados para el desarrollo de este proyecto. Primero, se describen las características de los conjuntos de datos (datasets) utilizados para el entrenamiento y las pruebas de rendimiento. Además, se detalla el preprocesamiento de imágenes previo al entrenamiento y la validación. El proyecto se subdivide en dos partes: la clasificación del grado de retinopatía y la detección de lesiones asociadas. Esto se realiza con el objetivo de proporcionar una perspectiva adicional sobre el panorama del grado de retinopatía diabética predicho, ya que los médicos se basan en las lesiones y otras características para diagnosticar las diferentes etapas de la enfermedad.

6.1. Datasets

A continuación, se detallan las características y la conformación de los conjuntos de datos empleados en el desarrollo de este proyecto:

6.1.1. DDR-dataset(principal)

La base de datos de imágenes de retinopatía diabética de China (DDR) incluye 13,673 imágenes de fondo de ojo recopiladas entre 2016 y 2018 de 147 hospitales en 23 provincias chinas. Estas imágenes fueron capturadas con diversas cámaras de fondo de ojo con un FOV de 45°, como Topcon D7000, Topcon TRC NW48, Nikon D5200 y Canon CR2. Las imágenes, desensibilizadas para su uso común, provienen de 9,598 pacientes de entre 1 y 100 años (edad promedio de 54.13 años), con una distribución de género del 48.23% masculino y 51.77% femenino. La resolución promedio de las imágenes es de 2514 x 1951 píxeles.

Las anotaciones del conjunto de datos se dividen en tres tipos: clasificación de RD a nivel de imagen, anotaciones a nivel de píxel y cuadros delimitadores de lesiones. Para este trabajo se utilizaron las anotaciones de clasificación de RD a nivel de imagen y los cuadros delimitadores. Todas las imágenes fueron clasificadas por oftalmólogos según la Clasificación Internacional de la Retinopatía Diabética, y se diferenciaron las imágenes no graduables debido a la baja calidad. Las imágenes se dividieron en seis clases: sin RD, RD no proliferativa leve, RD no proliferativa moderada, RD no proliferativa grave, RD proliferativa y no graduable. Los cuadros delimitadores se generaron automáticamente a partir de las anotaciones de nivel de píxel [19].

La división de la base de datos utilizada fue exactamente la misma que los autores originales, dicha distribución puede apreciarse a detalle en la Tabla 3:

Tabla 3. Número de muestras para cada clase en los conjuntos de train, test y validation.

	Train (Entrenamiento)	Test (Testeo)	Validation (Validación)	Total
Healthy (Sin RD)	3133	1253	1880	6266
Mild (RD leve)	315	126	189	630
Moderate (RD moderada)	2238	895	1344	4477
Severe (RD grave)	118	47	71	236
Proliferative (RD proliferativa)	456	182	275	913
Ungradable (No graduable)	575	230	346	1151
Total	6835	2733	4105	13673

Fuente: [19].

6.1.2. IDRiD

La base de datos de imágenes de retinopatía diabética de la India (IDRiD) es un conjunto de imágenes de fondo de ojo de personas de la India. Las imágenes de este conjunto de datos fueron adquiridas utilizando una cámara de fondo de ojo Kowa VX-10 con un FOV de 50°. Las imágenes poseen una resolución de 4288 × 4288 píxeles y están almacenadas en formato jpg. Dichas imágenes fueron tomadas en una serie de exámenes realizadas durante el período de 2009-2017, un grupo de expertos verificó que todas las imágenes tuvieran calidad adecuada, relevancia clínica, que no existieran duplicados y que estuviesen bien clasificadas según el grado de RD que poseían. El conjunto de datos se divide en 3 partes: la primera es segmentación y consiste en 81 imágenes de fondo de ojo subdivididas en un conjunto de entrenamiento y pruebas con 54 y 27 imágenes respectivamente. Estas imágenes poseen imágenes asociadas de la segmentación de las lesiones (hemorragias, microaneurismas, exudados duros y blandos) además de la segmentación de disco óptico. La segunda parte consiste en una carpeta de clasificación de la enfermedad, que posee 516 imágenes de fondo de ojo a color, con 413 imágenes de entrenamiento y 103 imágenes de prueba. Posee también archivos csv de entrenamiento y prueba con las etiquetas groundtruth para el grado de gravedad de la retinopatía diabética y edema macular diabética. La tercera parte consiste en las 516 imágenes anteriores con sus etiquetas para la ubicación de centro del disco óptico y la ubicación del centro de fovea [49].

6.1.3. EyePACS (KAGGLE dataset)

EyePacs es una aplicación web no propietaria basada en imágenes digitales para el almacenamiento y gestión de imágenes retinianas de pacientes.

EyePacs ha proporcionado al público un gran conjunto de datos de imágenes retinianas de programas de detección de la diabetes. El conjunto de datos está disponible en Kaggle. Consiste en 35,126 imágenes adquiridas con una variedad de cámaras de fondo de ojo [50].

Nota: Cabe destacar que para validación únicamente fueron utilizadas 15,768 imágenes de todo el dataset, lo que corresponde aproximadamente al 45% del dataset.

Un médico calificó la presencia de retinopatía diabética en cada imagen en un rango de 0 a 4 según la siguiente escala:

- 0 = Sin RD
- 1 = Leve
- 2 = Moderada
- 3 = Grave
- 4 = RD proliferativo

Las imágenes del conjunto de datos provienen de diferentes modelos y tipos de cámaras, lo que puede afectar la apariencia visual de la izquierda frente a la derecha. El conjunto de imágenes cuenta con resoluciones variadas, sin embargo, el valor promedio de resolución es de 3666 x 2490 píxeles, y todas las imágenes están en formato JPEG. Algunas imágenes se muestran cómo se vería la retina anatómicamente (mácula a la izquierda, nervio óptico a la derecha para el ojo derecho). Otros se muestran cómo se verían a través de la lente de condensación de un microscopio (es decir, invertidas, como se ve en un examen ocular típico en vivo). Como cualquier conjunto de datos del mundo real, encontrará ruido tanto en las imágenes como en las etiquetas. Las imágenes pueden contener artefactos, estar desenfocadas, subexpuestas o sobreexpuestas [51].

6.2. Preprocesamiento de imagen

A las imágenes de los dataset antes de pasarlos por una red neuronal convolucional es necesario aplicarles un procesamiento con el fin de resaltar y mejorar las características principales del o los objetos de interés a clasificar y/o detectar.

Con base en el estado del arte, se optó por utilizar el procesamiento de imagen utilizado por Alyoubi et al. [20] que a la vez utiliza la metodología propuesta por Zhou et al. [52]. Combinando dichas técnicas, pero aplicando ciertas mejoras con el fin de conseguir mejores resultados. Las mejoras implementadas incluyen la aplicación de un filtro de desenfoque gaussiano a la imagen y, finalmente, la aplicación de una máscara circular que permita trabajar únicamente la imagen de fondo de ojo y eliminando las partes de la imagen que no interesan. En la Figura 15 se puede apreciar la evolución de la imagen desde que entra al algoritmo de procesamiento de imagen hasta llegar al resultado final.

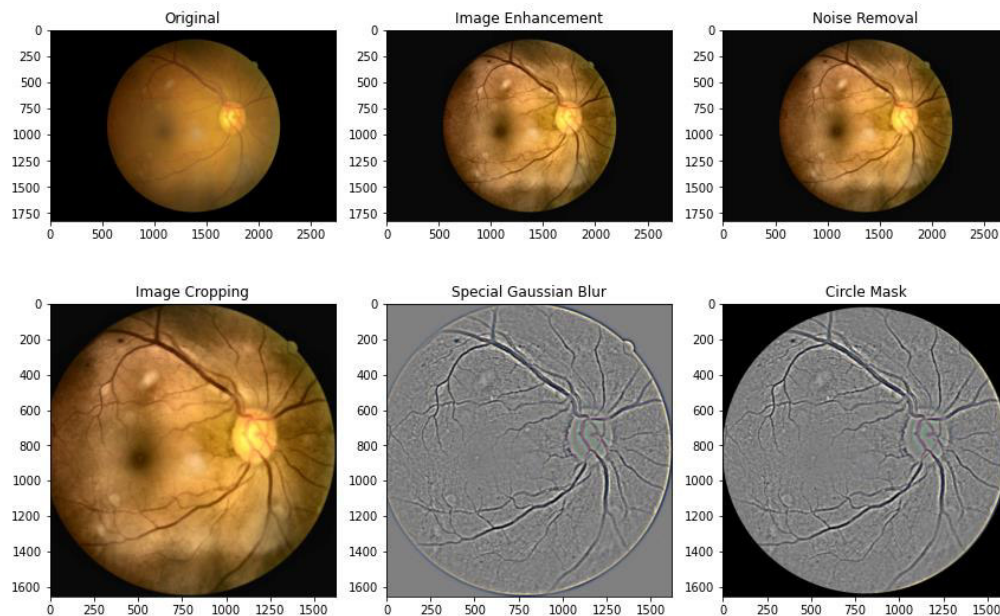


Figura 15. Etapas del procesamiento digital de imagen aplicado. Donde: “Image Enhancement” significa “Imagen Mejorada”, “Noise Removal” es “Eliminación de Ruido”, “Image Cropping” es “Imagen Recortada”, “Special Gaussian Blur” es “Desenfoque Gaussiano” y “Circle Mask” se refiere a “Máscara Circular”.

Nota: El código para realizar este procesamiento de imagen se puede encontrar en la sección: Código para el procesamiento de imagen, dicho código explica paso a paso la forma en que se implementó el algoritmo.

El procesamiento de imagen aplicado fue el siguiente:

- **Image Enhancement:** Para esta parte, se aplicaron dos métodos. El primero consiste en una mejora de luminosidad [52] y el segundo la aplicación de CLAHE por sus siglas en inglés (Contrast Limited Adaptive Histogram Equalization). El uso de CLAHE es bastante exitoso para la mejora del contraste de imágenes de fondo de ojo en imágenes médicas de bajo contraste [53], la configuración usada fue un kernel de tamaño de 8x8 y un Clip Limit de 5.0.
- **Noise Removal:** Debido a que CLAHE puede causar algo de ruido en las imágenes, es necesario removerlo [52] aplicando un filtro Gaussiano, siguiendo lo que se representa en la Ecuación 2).

$$G_0(x, y) = Ae^{-\frac{(x-\mu_x)^2}{2\sigma_x^2} - \frac{(y-\mu_y)^2}{2\sigma_y^2}}$$

2)

donde μ_x y μ_y es el promedio para cada una de las variables x y y , A es la amplitud y σ es la desviación estándar de cada una de las variables x y y .

- **Image Cropping:** Se recortan las imágenes con el fin de eliminar píxeles negros de fondo innecesarios alrededor de la retina.

Nota: A partir del proceso de Image Cropping, este y los siguientes pasos, se aplicaron únicamente para la etapa de clasificación. Para la detección de lesiones, a las imágenes únicamente se les aplicaron los dos pasos anteriores y se pasaron directamente al detector. Es importante destacar que ambos procesos (clasificación y detección) son independientes.

- **Color Normalisation:** Como se mencionó en la descripción de los dataset, las imágenes fueron capturadas de distintos pacientes de diferentes edades y etnias, teniendo diferentes niveles de iluminancia en el fondo de ojo. Estas condiciones tienen un efecto sobre los valores de intensidad de los píxeles de cada imagen y crean una variación en la misma. Para tratar esto las imágenes de fondo de ojo fueron normalizadas en intensidad luminosa en cada uno de sus canales RGB, este procedimiento se realizó utilizando la Ecuación 3).

$$z = \frac{(x - u)}{s}$$

3)

Donde x simboliza el canal RGB de las imágenes, u es la media de intensidad de todos los canales RGB de las imágenes y s es la desviación estándar del RGB de las imágenes de retina.

- **Special Gaussian Blur:** Se realiza un suavizado espacial en una imagen mediante un filtro de desenfoque gaussiano y una técnica de combinación de imágenes. Se aplica primero el desenfoque gaussiano para suavizar la imagen original. Luego, mediante una combinación ponderada de la imagen suavizada y la original, resalta los detalles y bordes mientras mantiene la suavidad general.
- **Circle mask:** Su finalidad es la eliminación eficiente del fondo de la imagen. Inicialmente, se determinan el centroide y el radio del círculo en función de las dimensiones de la imagen de entrada. A continuación, se genera una máscara binaria, donde un círculo sólido, representado por valores de píxeles blancos, se dibuja en un fondo negro. Este círculo se posiciona en el centroide de la imagen, garantizando una distribución uniforme. Posteriormente, se aplica la máscara circular a la imagen original mediante la operación bitwise AND. Esta operación permite preservar únicamente los píxeles de la imagen original que se superponen con los píxeles blancos de la máscara, resultando en la eliminación del fondo circundante.

EL resumen del procesamiento de imagen puede observarse en la Figura 16.

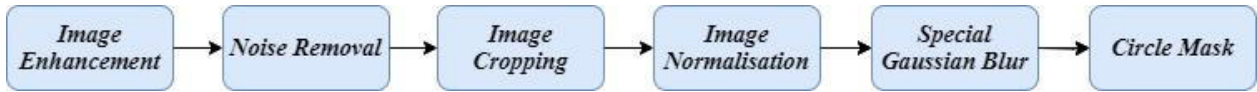


Figura 16. Diagrama de flujo del procesamiento de imagen realizado.

6.3. Clasificación de grado de la retinopatía diabética

Una vez realizado el preprocesamiento de imagen pertinente se procedió a realizar las siguientes actividades previo a iniciar el entrenamiento del modelo:

- **Ordenamiento de Clases:**

Es importante mencionar que antes de iniciar el entrenamiento es necesario ordenar los datos según el formato solicitado por YOLOv8, mostrado en su documentación [54]. Dicho formato es el siguiente:

```

DDR-dataset-classification/
|
|-- train/
| | |-- healthy/
| | | |-- 10008_healthy.png
| | | |-- 10009_healthy.png
| | | |-- ...
| | |-- ...
| |
|-- val/
| | |-- healthy/
| | | |-- 10_healthy.png
| | | |-- 11_healthy.png
| | | |-- ...
| | |-- ...
| |
|-- ...
  
```

- **Configuración de YOLOv8 para clasificación**

Para llevar a cabo el entrenamiento de este modelo, se utilizó una computadora equipada con 20 procesadores y una GPU NVIDIA GeForce RTX 4090 de 24GB, además de la siguiente configuración de parámetros para el entrenamiento del modelo que se pueden observar en la Tabla 4.

Tabla 4. Hiper parámetros de configuración de YOLOv8 para clasificación de grado de la retinopatía diabética.

<i>Configuration</i>	<i>Values</i>
Epochs	50
Batch	8
Imgsz	640x640
Optimizer	Auto
lr0	0.01
lrf	0.0001
Dropout	0.5
Model	yolov8n-cls.pt

Para la configuración restante del modelo, se mantuvo la que tenía por defecto, proporcionada por el equipo desarrollador del modelo, la cual se puede consultar en [55]. Hay que destacar que el submodelo de YOLO utilizado fue la versión nano, haciendo uso de Transfer Learning, con los pesos del dataset *imagenet* [56].

- **Entrenamiento del modelo**

A continuación, en la Tabla 5 se presenta el modelo de clasificación para la obtención de grados de la retinopatía diabética, en la tabla se muestra una sección de módulos, estos hacen referencia a partes específicas de la arquitectura de YOLOv8 mostrada en la Figura 14.

Tabla 5. Arquitectura de YOLOv8-cls nano para la clasificación de imágenes de grado de RD.

Layer	n	Params	Module
0	1	464	Conv
1	1	4672	Conv
2	1	7360	C2f
3	1	18560	Conv
4	2	49664	C2f
5	1	73984	Conv
6	2	197632	C2f
7	1	295424	Conv
8	1	460288	C2f
9	1	337926	Classify

Una vez que el entrenamiento se completaba, el siguiente paso consiste en la validación del modelo para así terminar con el despliegue de las métricas de rendimiento, las cuales pueden consultarse en la sección 5.1.5. El resumen del proceso de clasificación puede observarse en la Figura 17.

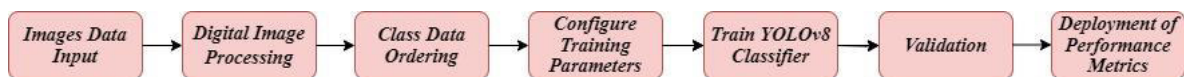


Figura 17. Diagrama de trabajo para el desarrollo del clasificador de grado de retinopatía diabética utilizando YOLOv8.

6.4. Detección de lesiones

Después de aplicar el mejoramiento de la imagen, el proceso a seguir para la detección de lesiones fue el siguiente:

- **Convertir archivos xml a formato YOLO**

La base de datos DDR cuenta con una muestra de imágenes tomadas para detección de lesiones, dicha muestra tiene asociado un archivo .xml que posee las coordenadas de todas cajas delimitadoras de las lesiones que posee dicha imagen (MA, HE, EX, SE), dicha muestra se subdivide en tres carpetas:

- Train: 383 imágenes
- Val: 225 imágenes
- Test: 149 imágenes

Cada una de estas imágenes puede poseer uno o varios tipos de lesiones. Por lo que el algoritmo es capaz de mostrar todas a la vez, de ser el caso.

Para que YOLOv8 sea capaz de leer esta información y poder entrenarlo es necesario que los archivos .xml sean convertidos al formato YOLO siguiendo las siguientes instrucciones que nos proporciona la documentación de YOLOv8: Las etiquetas para este formato deben exportarse al formato YOLO con un archivo .txt por imagen. Si no hay objetos en una imagen, no se requiere ningún archivo .txt. El archivo .txt debe tener el formato de una fila por objeto, en el formato: clase x_centro y_centro ancho alto. Las coordenadas de la caja deben estar en formato xywh normalizado (de 0 a 1). Si sus cajas están en píxeles, debe dividir x_centro y ancho por el ancho de la imagen, y y_centro y alto por la altura de la imagen. Los números de clase deben tener índice cero (comenzar con 0) [57]. Se puede observar de una forma más gráfica en la *Figura 18* lo explicado anteriormente.



Figura 18. Ejemplo de los valores que toma la etiqueta YOLO, para formarse. Fuente: [57]

Para cumplir con lo anterior se desarrolló un script en Python que facilitó la conversión de los archivos. Dicho script puede consultarse en la sección de Anexos.

- **Reordenamiento del dataset**

Una vez que se tienen los archivos de información de detección convertidos al formato YOLO, es necesario reorganizar el dataset de la siguiente forma:

```
DDR-dataset-detection/
|
|-- labels/
| | |-- val/
| | | |-- 10008_healthy.txt
| | |-- train/
| | | |-- 10009_healthy.txt
| |
```

```

| |
|-- images/
| |-- val/
| | |-- 10008_healthy.jpg
|-- train/
| | |-- 10009_healthy.jpg
| |

```

- **Configuración de archivo yaml**

Una vez que se ordena la base de datos con el formato solicitado, es necesario configurar un archivo yaml, dicho archivo contiene la ruta del dataset que se utilizará para entrenamiento y validación, además de las clases de objetos a detectar, dicha configuración queda de la siguiente forma:

```

path: /home/DDR-dataset-detection/
train: images/train
val: images/val

```

```

# Classes
names:
0: ex
1: ma
2: he
3: se

```

- **Configuración de YOLOv8 para la detección**

Al igual que en la parte de clasificación se utilizó una computadora con 20 procesadores y una tarjeta de video NVIDIA GeForce RTX 4090 de 24GB, los parámetros de configuración modificados se muestran en la Tabla 6.

Tabla 6. Hiper parámetros de configuración para el modelo de detección de lesiones.

<i>Configuration</i>	<i>Values</i>
Epochs	300
Batch	4
Imgsz	1920x1920
Optimizer	SGD
Conf	0.1
IoU	0.05
lr0	0.01
lrf	0.0001
Momentum	0.932
Dropout	0.5
Agnostic_nms	True
Deterministic	True
Model	yolov8n.pt

El resto de los parámetros con los que cuenta yolov8 se mantuvieron con los valores por defecto [55].

- **Entrenamiento del modelo**

En la Tabla 7, se pueden ver los parámetros utilizados para entrenar el modelo de detección y sus respectivos módulos utilizados, hay que destacar que modelo para detección de YOLOv8 nano. Para los pesos se hizo uso de la metodología de Transfer Learning, haciendo uso de los pesos del dataset COCO [58].

Tabla 7. Arquitectura del modelo de detección de YOLOv8 nano para detección de lesiones.

Layer	n	Params	Module
0	1	464	Conv
1	1	4672	Conv
2	1	7360	C2f
3	1	18560	Conv
4	2	49664	C2f
4	1	73984	Conv
6	2	197632	C2f
7	1	295424	Conv
8	1	460288	C2f
9	1	164608	SPPF
10	1	0	Upsample
11	1	0	Concat
12	1	148224	C2f
13	1	0	Upsample
14	1	0	Concat
15	1	37248	C2f
16	1	36992	Conv
17	1	0	Concat
18	1	123648	C2f
19	1	147712	Conv
20	1	0	Concat
21	1	493056	C2f
22	1	752092	Detect

Una vez que el entrenamiento se completaba el siguiente paso consistió en pruebas de validación para observar que tan bien clasifica los cuadros delimitadores en las lesiones y para así terminar con el despliegue de las métricas de rendimiento, las cuales pueden consultarse en la sección 5.1.5. El resumen del proceso para la detección se puede observar en la Figura 19.

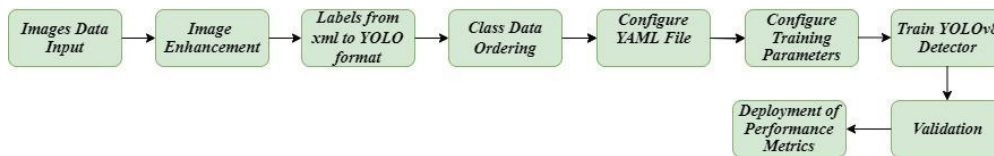


Figura 19. Diagrama de trabajo para el desarrollo del detector de lesiones de RD utilizando YOLOv8.

Nota: Todos los códigos utilizados durante el proceso de clasificación y detección se podrán encontrar en la sección de ANEXOS.

7. Resultados

7.1. Clasificador de grado de retinopatía diabética

Para probar la calidad de generalización del modelo, las pruebas de testeo se realizaron con tres Datasets, el primero DDR la utilizando las imágenes destinadas para test, el segundo IDRiD utilizando todas las imágenes del dataset y por último EyePACS de Kaggle utilizando únicamente una muestra de toda la base, debido a su gran tamaño y variabilidad.

Los resultados reportados para cada prueba de testeo retornan la curva ROC + AUC, un archivo .csv con las métricas de rendimiento más relevantes (Accuracy, Precision, F1 Score, Kappa) y por último la matriz de confusión general y normalizada.

Nota: Los valores de nan en IDRiD y EyePACS se deben a que no fueron clasificados en una clase por defecto. En otras palabras, la clase ungradable no existe en esos conjuntos de datos.

Previo a eso primero presentamos los resultados del entrenamiento del modelo:

Nota: En la Figura 20, el eje de las X representa el número de épocas, mientras que el eje de las Y representa la pérdida para los gráficos superiores y el accuracy top 1 o top 5, para los gráficos inferiores.

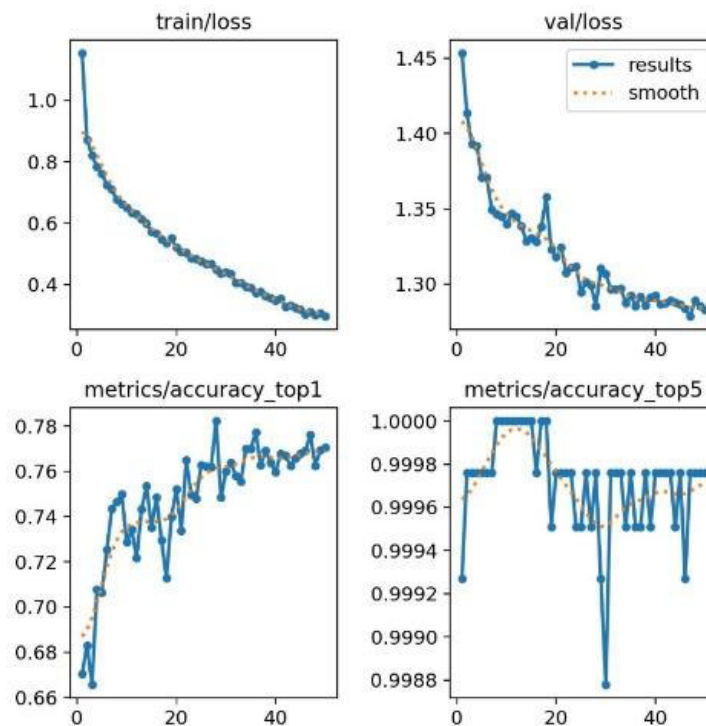


Figura 20. Gráficos de función de pérdida, accuracy top-1 y top-5 durante el entrenamiento del clasificador YOLOv8. El gráfico superior izquierdo muestra la disminución de la pérdida de entrenamiento ("train/loss") a lo largo de las épocas, mientras que el gráfico superior derecho muestra la pérdida de validación ("val/loss"). Los gráficos inferiores presentan la precisión del modelo, con el accuracy top-1 a la izquierda y el accuracy top-5 a la derecha, indicando la proporción de veces que la clase correcta estuvo entre las primeras 1 y 5 predicciones, respectivamente. Los resultados se presentan junto con una curva suavizada ("smooth") para facilitar la visualización de las tendencias. Para más detalles sobre la interpretación de estas métricas y su relevancia en la evaluación del modelo, se recomienda consultar fuentes especializadas. Para más información a fondo sobre cómo interpretar este tipo de métricas consultar: [59].

A continuación, se presentan las matrices de confusión general y normalizada para los datos de validación del entrenamiento. Donde las etiquetas se muestran en inglés, sin embargo, cada una de estas corresponda a: sin RD (healthy), RD no proliferativa leve (mild), RD no proliferativa moderada (moderate), RD no proliferativa grave (severe), RD proliferativa (proliferative) y no graduable (ungradable).

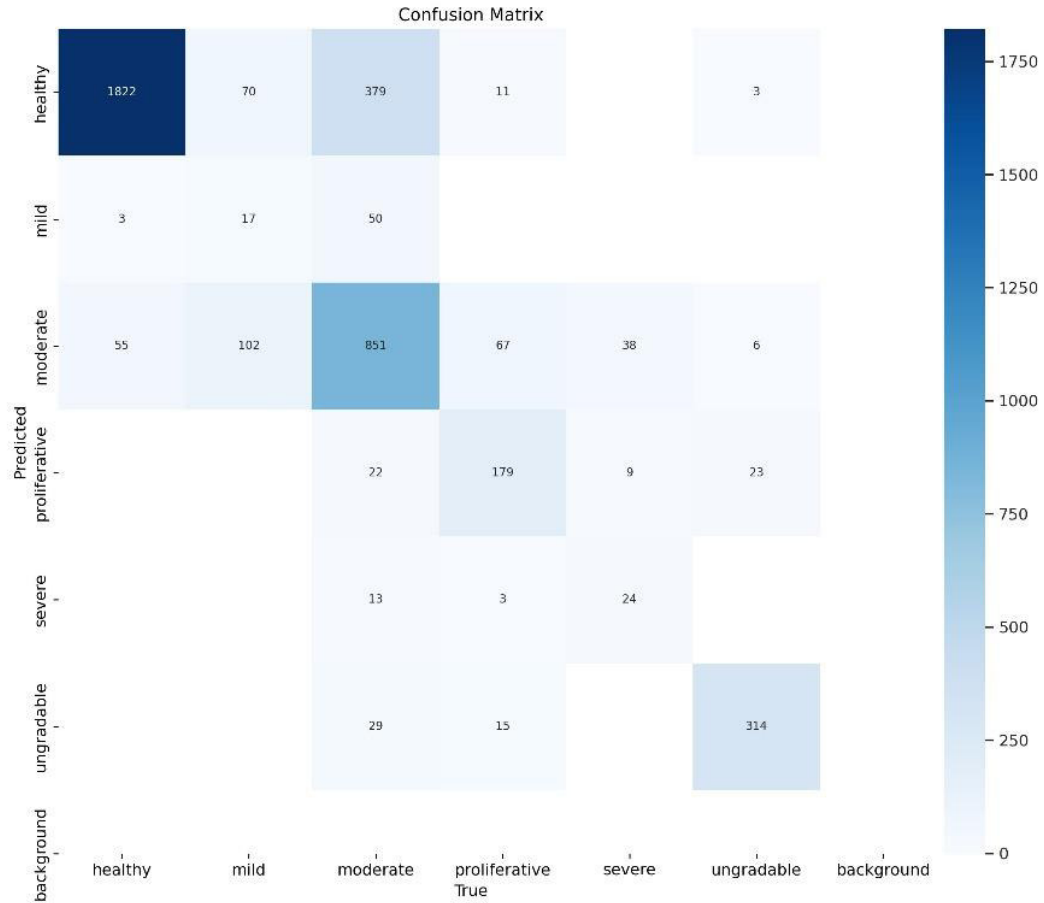


Figura 21. Matriz de confusión para los datos de validación durante el entrenamiento del modelo. La matriz presenta en el eje X las etiquetas verdaderas y en el eje Y las etiquetas predichas por el modelo. Cada celda muestra el número de instancias clasificadas en una clase específica. La diagonal central es la que muestra el número correcto de clasificaciones por cada clase.

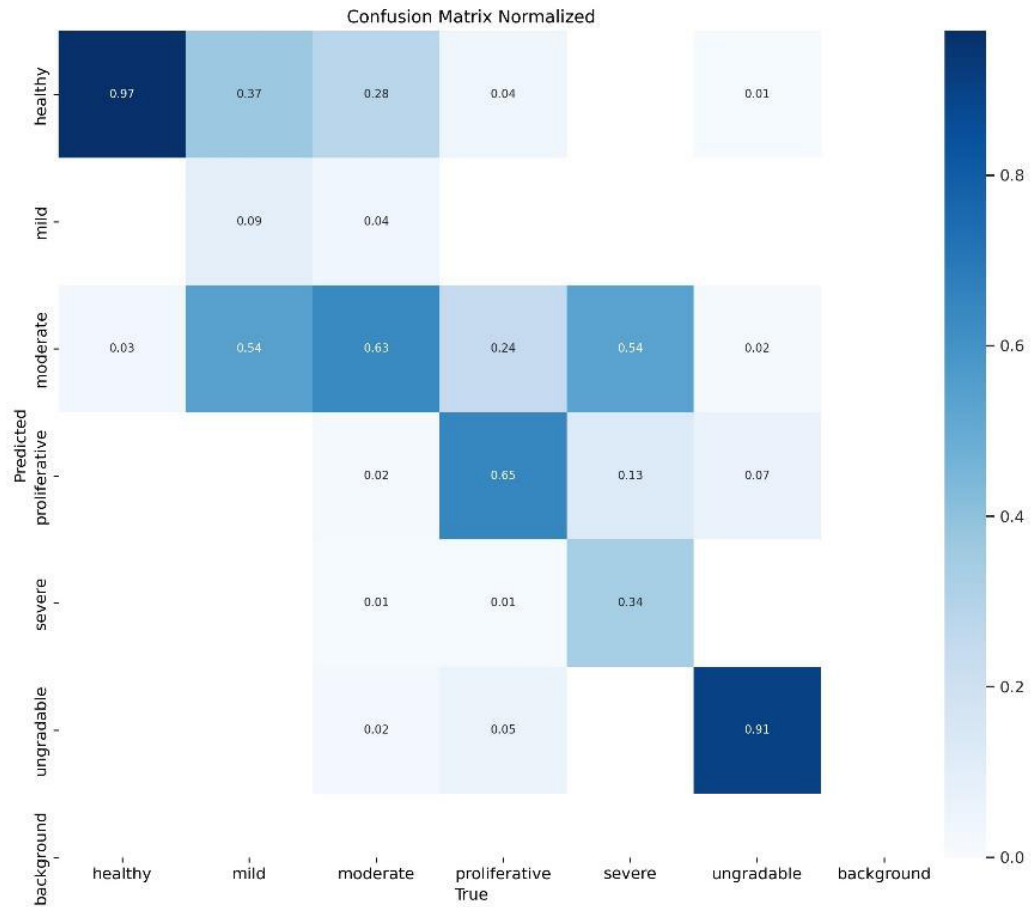
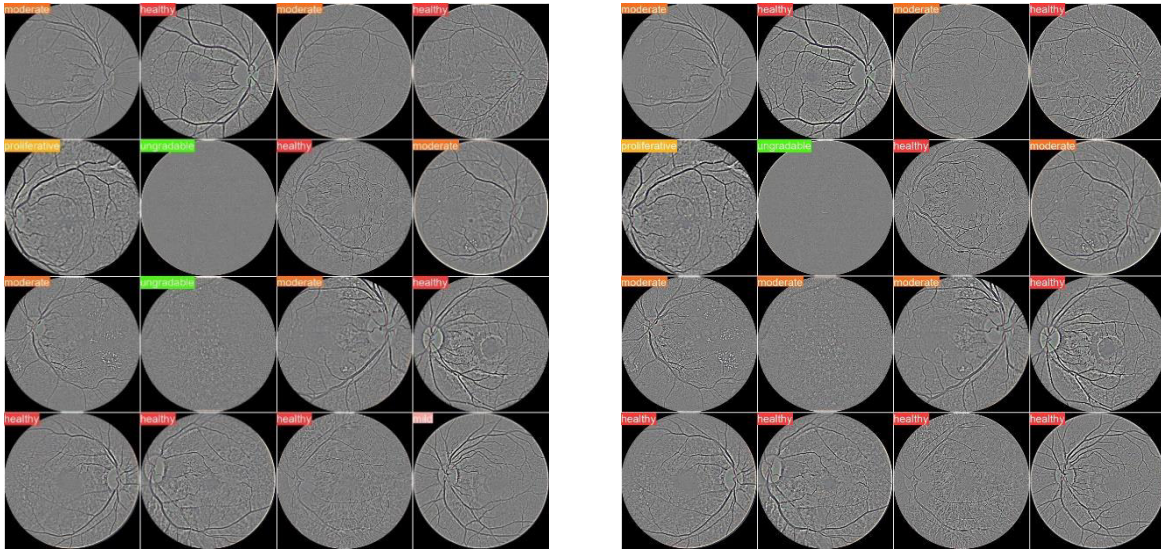


Figura 22. Matriz de confusión normalizada para los datos de validación durante el entrenamiento del modelo. La matriz presenta en el eje X las etiquetas verdaderas y en el eje Y las etiquetas predichas por el modelo. Cada celda muestra el número de instancias clasificadas en una clase específica. La diagonal central es la que muestra el número correcto de clasificaciones por cada clase.

En la Figura 23, se puede observar una muestra de las imágenes reales contra las predichas por el clasificador, en los datos de validación.



(a)

(b)

Figura 23. Muestra de las salidas de predicción del modelo de clasificación, la imagen (a) son las imágenes con sus etiquetas reales, y (b) las etiquetas predichas por el modelo. Se puede ver que el modelo tiene una predicción bastante precisa para esta muestra de imágenes, de las 16 imágenes únicamente 2 de ellas fueron erróneamente clasificadas.

A continuación, se presentan los datos más relevantes de las pruebas realizadas post entrenamiento utilizando los distintos datasets:

Tabla 8. Precisión del modelo entrenado con YOLOv8 para datos generalizados de tres distintos datasets.

YOLOv8						
Dataset	No RD	Mild	Moderate	Severe	Proliferative	Ungradable
DDR	0.98	0.12	0.75	0.62	0.77	0.77
IDRiD	0.89	0.08	0.77	0.62	0.52	N/A
EyePACS	0.77	0.10	0.66	0.23	0.22	N/A

Nota: En los datasets que poseen valor N/A en la clase “Ungradable” es debido a que los datos originales no cuentan con esta clase.

Tabla 9. Métricas generales de YOLOv8 para los datos de prueba de los distintos dataset.

General Metrics					
DATASETS	Accuracy	Precision	Recall	F1-Score	Kappa
DDR	0.8276	0.8231	0.8276	0.8192	0.7371
IDRiD	0.7170	0.7061	0.7170	0.7062	0.6098
Eye PACS	0.5740	0.5510	0.5740	0.5443	0.3506

- A continuación, se muestran las matrices de confusión y las curvas ROC para el dataset DDR:

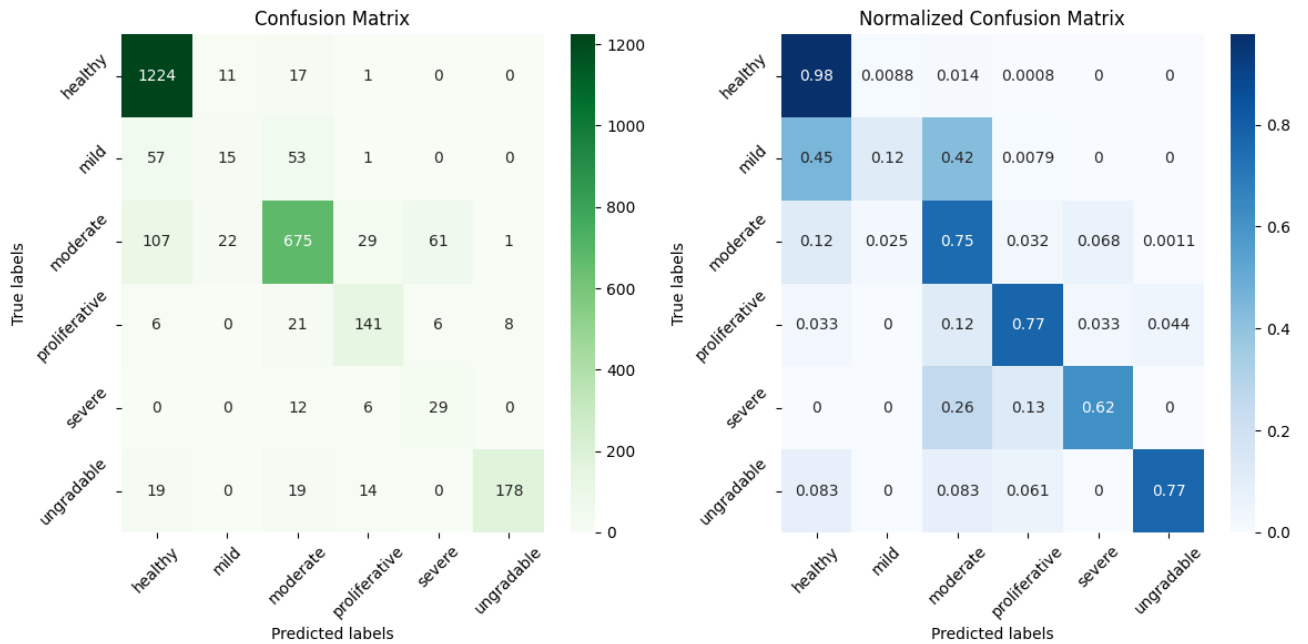


Figura 24. Matrices de confusión general (izquierda) y normalizada (derecha) para los datos de prueba del conjunto de datos DDR. La matriz de confusión general muestra el número absoluto de predicciones realizadas para cada clase, mientras que la matriz normalizada presenta las proporciones de cada predicción en relación con el total de casos verdaderos por clase. Las clases incluyen "healthy", "mild", "moderate", "proliferative", "severe" y "ungradable". En la matriz general, se observa que la mayoría de las instancias de la clase "healthy" fueron correctamente clasificadas (1224), con muy pocos errores (98% de precisión). En la matriz normalizada, se destaca la alta precisión para las clases "healthy" (0.98) y "ungradable" (0.77), así como un buen desempeño para las clases "moderate" (0.75) y "proliferative" (0.77). Esto indica que el modelo tiene un rendimiento especialmente bueno en la identificación de casos "healthy" y "ungradable", con menos precisión en clases como "mild" (0.12) y "severe" (0.26), donde se observa una mayor dispersión en las predicciones.

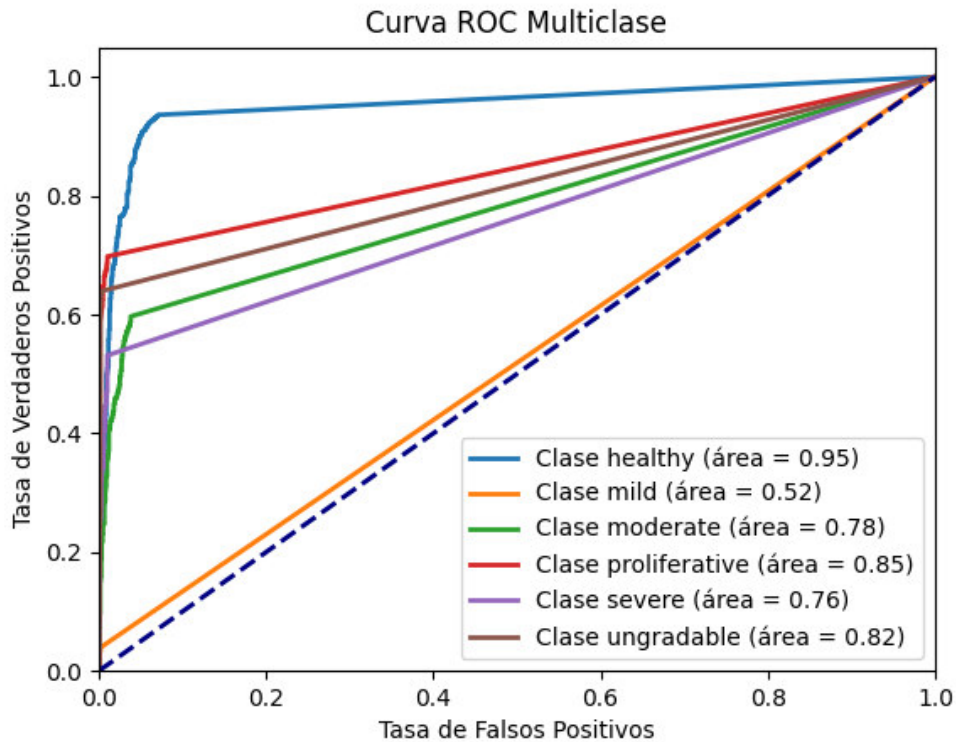


Figura 25. Curvas ROC multiclásificadas para el conjunto de datos DDR, mostrando el rendimiento del modelo en la clasificación de diferentes categorías: "healthy", "mild", "moderate", "proliferative", "severe" y "ungradable". Las curvas se generan considerando un umbral de confianza del 75% en las predicciones. El área bajo la curva (AUC) refleja la capacidad del modelo para distinguir entre las clases, con valores más altos indicando mejor rendimiento. Las clases con un desempeño relativamente bueno incluyen "healthy" con un AUC de 0.95, "proliferative" con un AUC de 0.85, "ungradable" con un AUC de 0.82 y "moderate" con un AUC de 0.78. Estos resultados indican que el modelo es particularmente eficaz en la identificación de casos "healthy" y "proliferative", mientras que las clases "mild" y "severe" presentan mayores desafíos con AUCs de 0.52 y 0.76, respectivamente.

- A continuación, se muestran las matrices de confusión y las curvas ROC para el dataset IDRiD

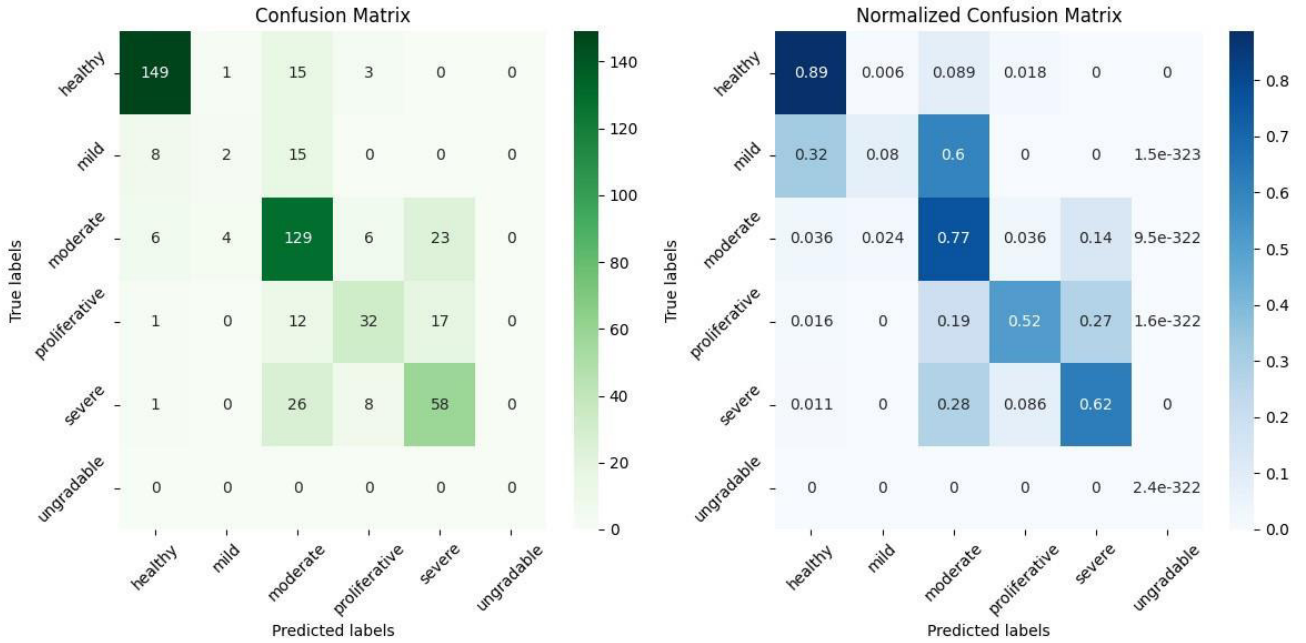


Figura 26. Matrices de confusión general (izquierda) y normalizada (derecha) para los datos de prueba del conjunto de datos IDRiD. La matriz de confusión general muestra el número absoluto de predicciones para cada clase, mientras que la matriz normalizada presenta las proporciones correspondientes. Las clases incluyen "healthy", "mild", "moderate", "proliferative", "severe" y "ungradable". En la matriz general, se observa que la mayoría de las instancias de la clase "healthy" se clasifican correctamente (149), mientras que algunas se clasifican erróneamente como "moderate" (15). La matriz normalizada muestra que el 89% de las instancias "healthy" fueron clasificadas correctamente, destacando la mayor precisión en esta categoría. En contraste, otras clases como "mild" y "proliferative" muestran mayor dispersión en las predicciones, indicando desafíos en la clasificación precisa de estas categorías.

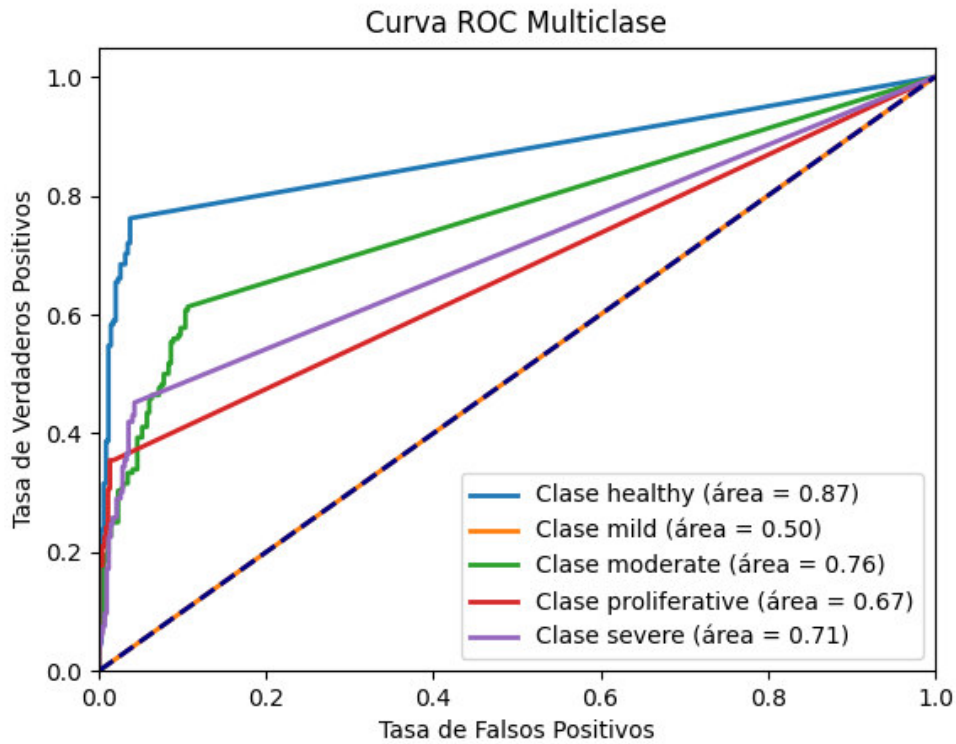


Figura 27. Curvas ROC multiclasicadas para el conjunto de datos IDRiD, mostrando el rendimiento del modelo en la clasificación de diferentes categorías: "healthy", "mild", "moderate", "proliferative", y "severe". Las curvas ROC se construyeron considerando un umbral de confianza en la predicción del 75%. El área bajo la curva (AUC) para cada clase indica la capacidad del modelo para distinguir entre las categorías. Los valores de AUC son 0.87 para "healthy", 0.50 para "mild", 0.76 para "moderate", 0.67 para "proliferative" y 0.71 para "severe", mostrando un buen rendimiento general, especialmente en la clasificación de la clase "healthy".

- A continuación, se muestran las matrices de confusión y las curvas ROC para el dataset EyePACS:

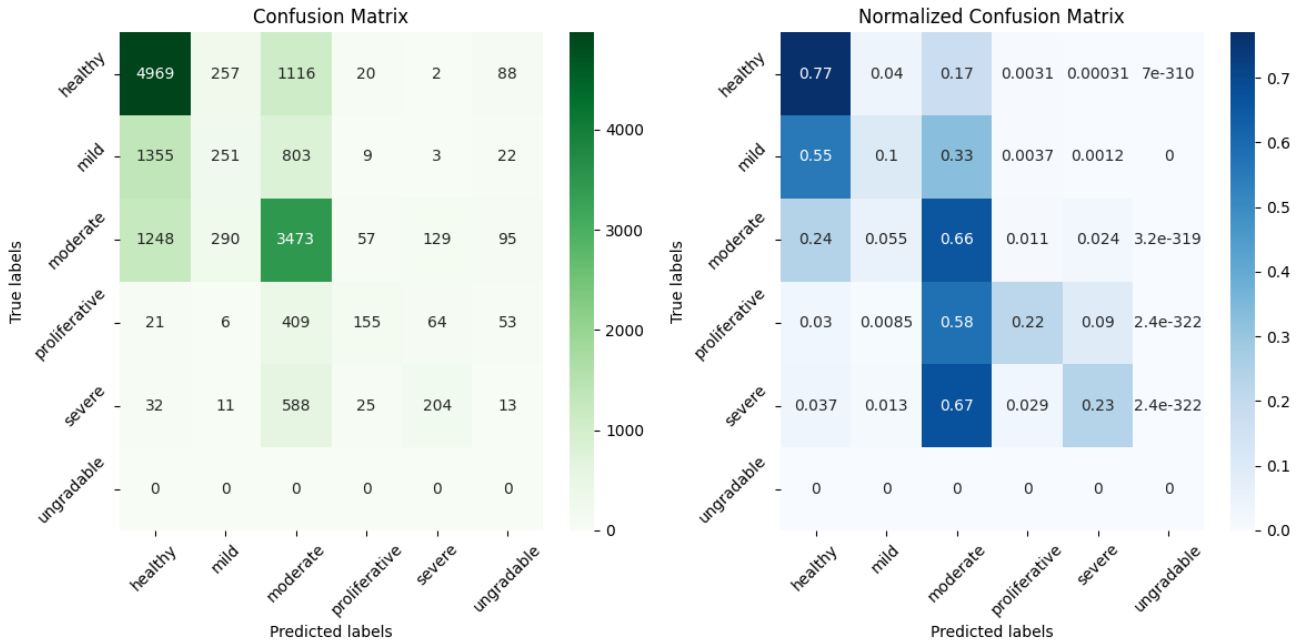


Figura 28. Matrices de confusión general (izquierda) y normalizada (derecha) para los datos de prueba del conjunto de datos EyePACS. La matriz de confusión general muestra el número absoluto de predicciones realizadas por el modelo para cada clase, mientras que la matriz normalizada muestra las proporciones de cada predicción respecto al total de verdaderos positivos por clase. Las clases incluyen "healthy", "mild", "moderate", "proliferative", "severe" y "ungradable". Los valores diagonales indican el número de casos correctamente clasificados, y los valores fuera de la diagonal representan errores de clasificación. La normalización facilita la comparación entre clases, revelando que la mayor precisión se obtuvo para la clase "healthy" con un 77%, mientras que las clases "mild" y "moderate" muestran más confusiones con otras categorías.

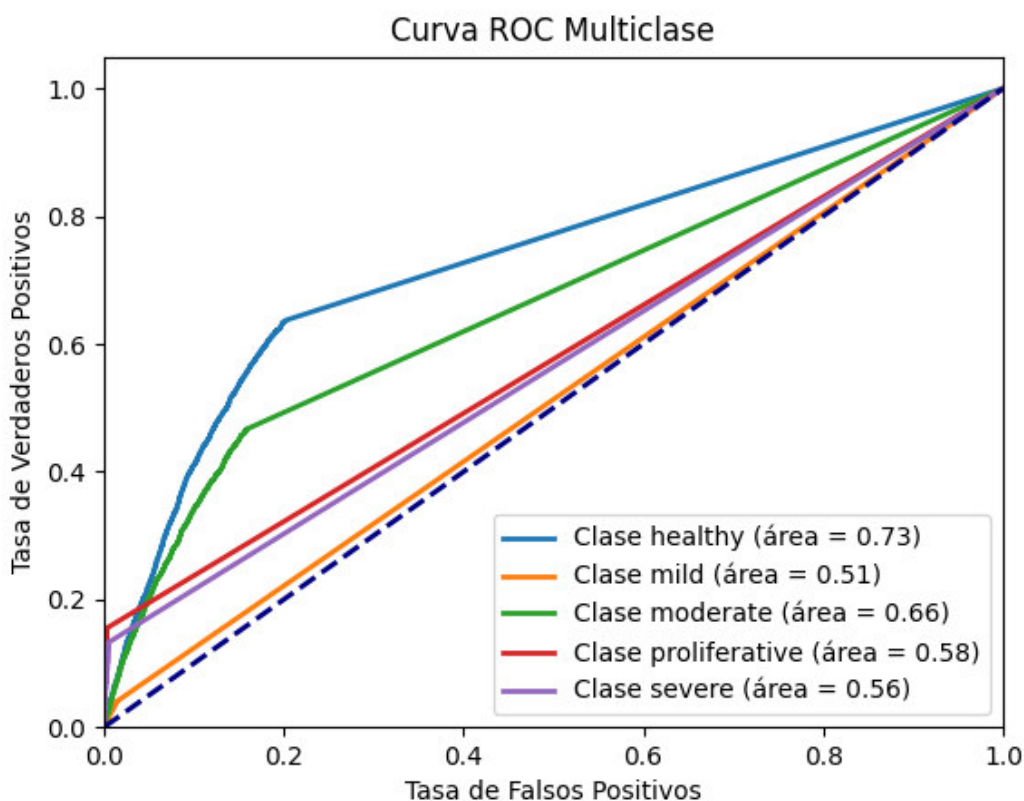


Figura 29. Curvas ROC multiclásificadas para el conjunto de datos Eye PACS, mostrando el rendimiento del modelo en la clasificación de distintas categorías: "healthy", "mild", "moderate", "proliferative", y "severe". Cada curva representa la relación entre la tasa de verdaderos positivos (sensibilidad) y la tasa de falsos positivos para una clase específica. El área bajo la curva (AUC) proporciona una medida de la capacidad del modelo para distinguir entre clases, con valores más altos indicando un mejor rendimiento. Los AUCs obtenidos son: 0.73 para "healthy", 0.51 para "mild", 0.66 para "moderate", 0.58 para "proliferative", y 0.56 para "severe". Las curvas se calculan con un umbral de confianza del 75%.

7.2. Detector de lesiones

YOLOv8 fue entrenado con el dataset DDR el cual a su vez contaba con una subsección enfocada a la localización de lesiones, cabe mencionar que este conjunto de datos cuenta con el tipo de lesión y el dibujo del bounding box de ésta. El conjunto de datos para entrenamiento del detector de lesiones cuenta con 608 imágenes para entrenamiento y 149 para validación. Después de realizar múltiples pruebas aplicando distintas configuraciones, se obtuvo un alto mAP (mean Average Precision) comparado con los resultados del estado del arte reportados por Tao Li et al. [19] y por Alyoubi et al. [20].

Se puede ver en la Tabla 10 la comparativa con respecto el estado del arte que ha utilizado la misma base de datos.

Tabla 10. Comparación entre el modelo de YOLOv8 y el estado del arte sobre el dataset DDR.

<i>Model</i>	<i>mAP</i>
Tao Li et al. [19]	0.092
Alyoubi et al. [20]	0.215
YOLOv8 + SGD optimizer + dropout	0.36

Ahora se presentan los gráficos de rendimiento durante el entrenamiento y proceso de validación del detector de lesiones utilizando los datos de la base DDR:

- Matriz de confusión de las lesiones de fondo de ojo por RD**

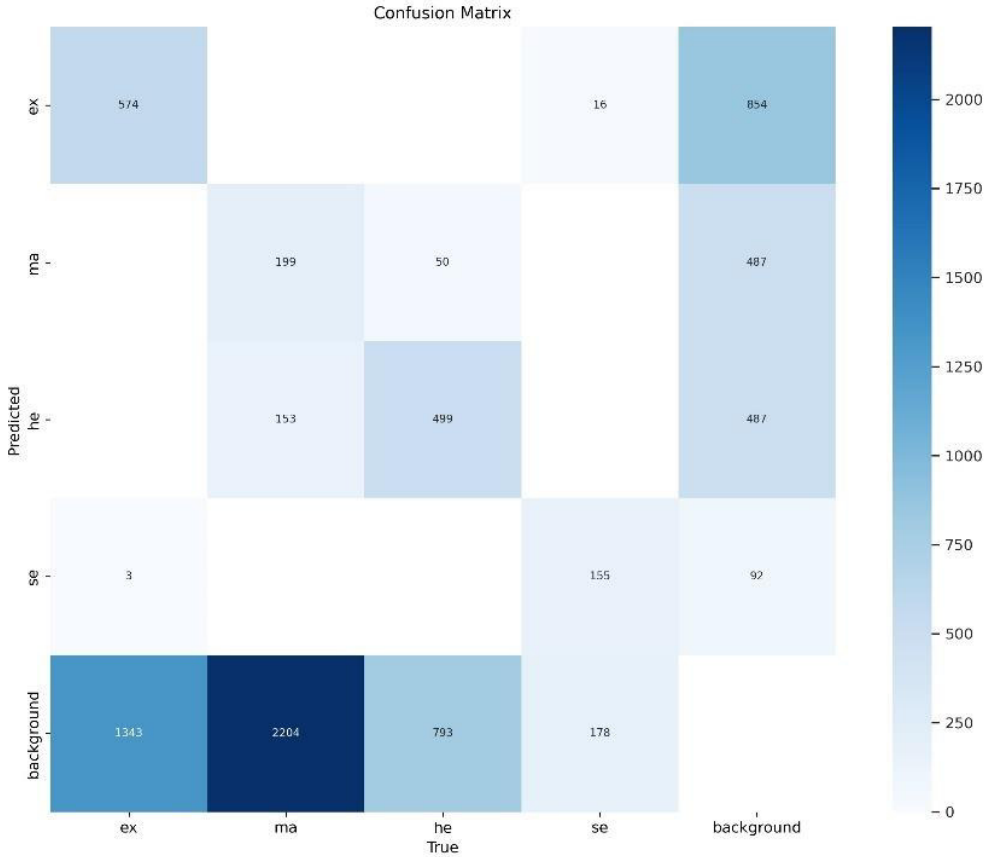


Figura 30. Matriz de confusión no normalizada para el modelo YOLOv8, mostrando el número absoluto de predicciones para cada clase de lesiones: exudados (ex), microaneurismas (ma), hemorragias (he), otras lesiones (se) y fondo (background). Cada celda contiene el número de instancias predichas para una clase particular; con la diagonal principal representando las predicciones correctas. Por ejemplo, se predijeron correctamente 574 casos de exudados, mientras que 1343 casos de exudados fueron erróneamente clasificados como fondo. Esta matriz permite una evaluación directa de la cantidad de errores de clasificación y aciertos por clase, facilitando el análisis del rendimiento absoluto del modelo.

- **Matriz de confusión normalizada de las lesiones de fondo de ojo por RD**

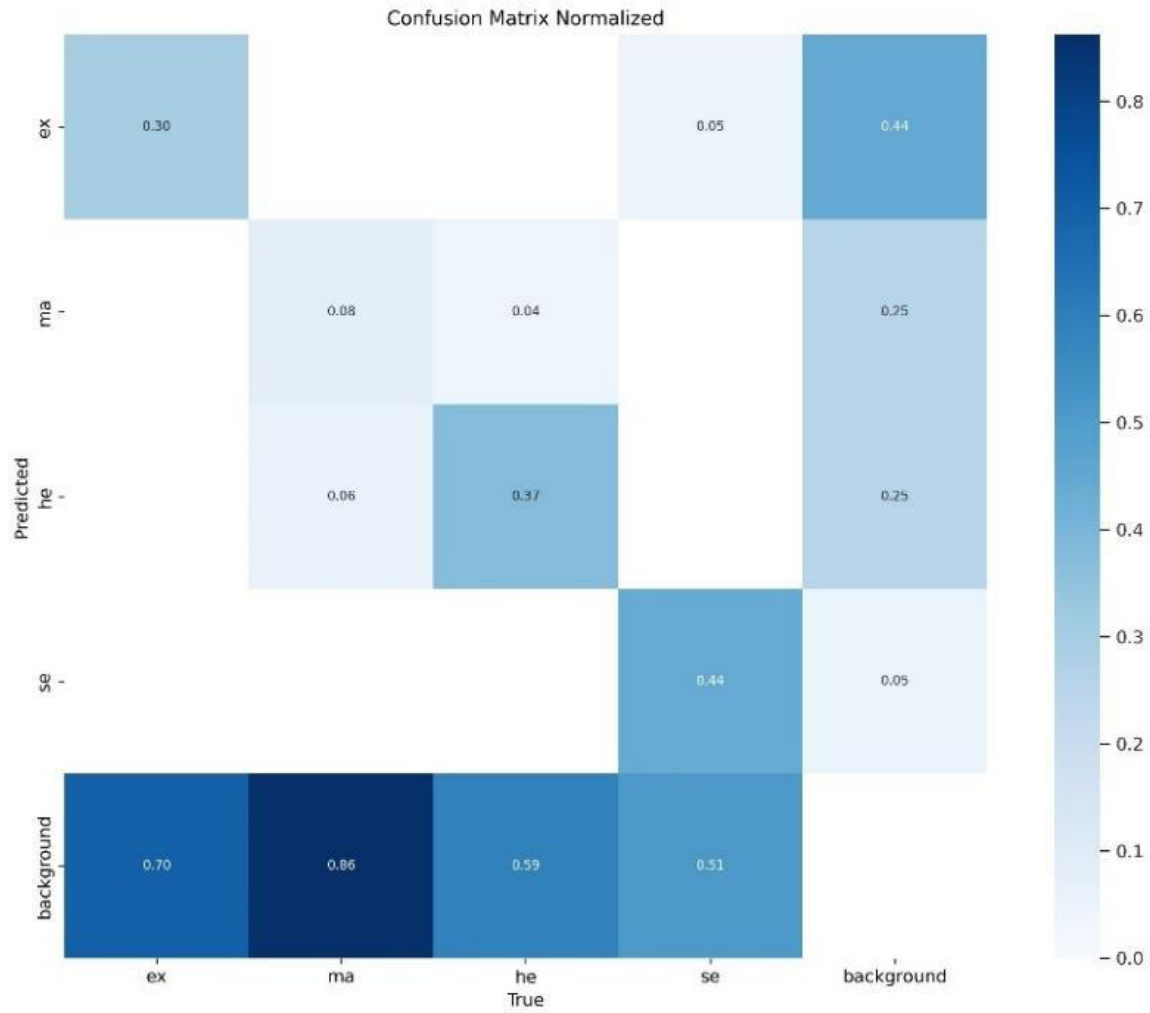


Figura 31. Matriz de confusión normalizada para el modelo YOLOv8, representando las predicciones de distintas clases de lesiones: exudados (ex), microaneurismas (ma), hemorragias (he), otras lesiones (se) y fondo (background). Los valores dentro de las celdas indican la proporción de instancias clasificadas en cada categoría. La diagonal principal muestra las predicciones correctas, mientras que las demás celdas representan los errores de clasificación. Por ejemplo, el modelo clasifica correctamente los exudados el 30% del tiempo, pero confunde los exudados con el fondo en el 70% de los casos. Esta matriz ayuda a visualizar y cuantificar la precisión del modelo y las posibles confusiones entre clases.

- **Curva de confianza F1**

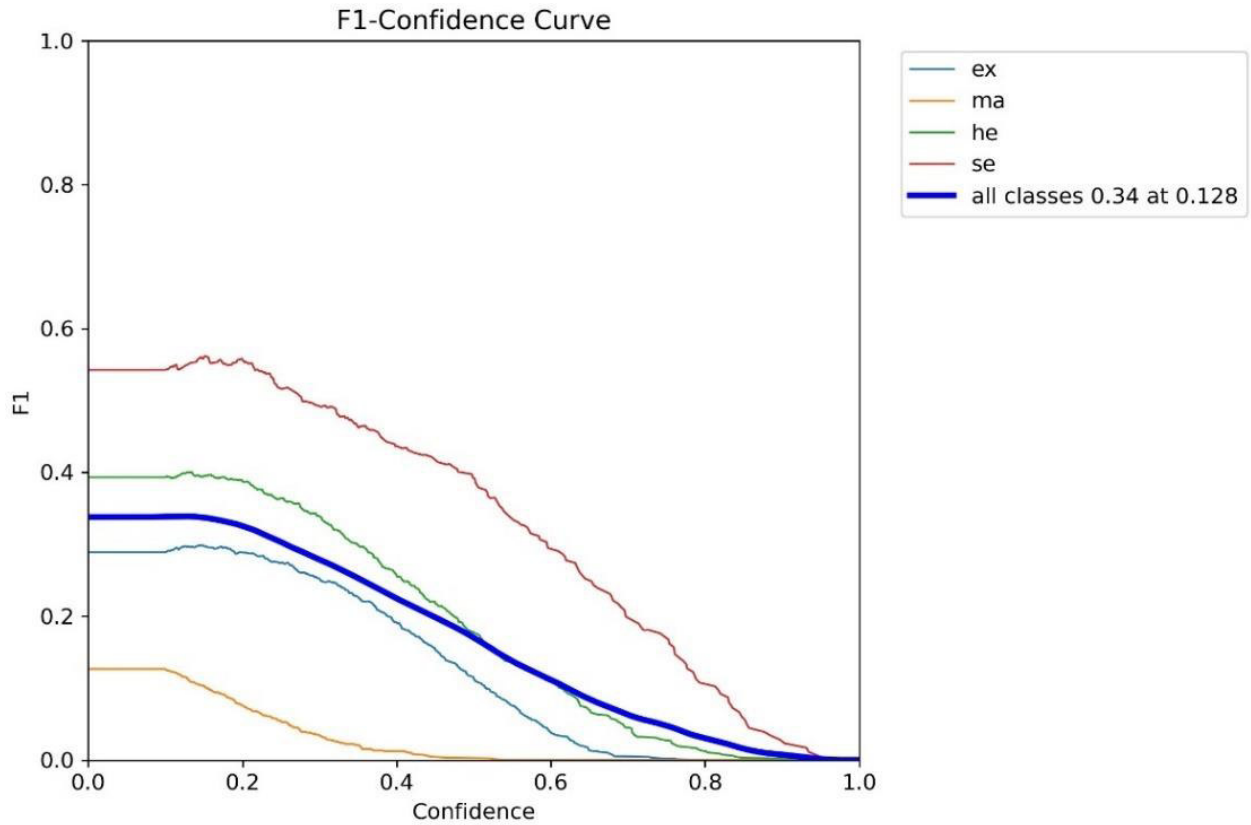


Figura 32. Curva de F1-Confianza para el modelo YOLOv8, ilustrando cómo varía el puntaje F1 con el nivel de confianza de las predicciones para diferentes clases de lesiones: exudados (ex), microaneurismas (ma), hemorragias (he) y otras lesiones (se). Cada línea de color corresponde a una clase específica, mientras que la línea azul gruesa representa el rendimiento promedio para todas las clases combinadas. El puntaje F1, que equilibra precisión y recall, alcanza su valor máximo de 0.34 a un umbral de confianza de 0.128 para todas las clases. Este gráfico ayuda a identificar el nivel de confianza óptimo para obtener un equilibrio adecuado entre precisión y recall, destacando la variación del rendimiento del modelo según el tipo de lesión detectada.

- **Precisión-Confianza**

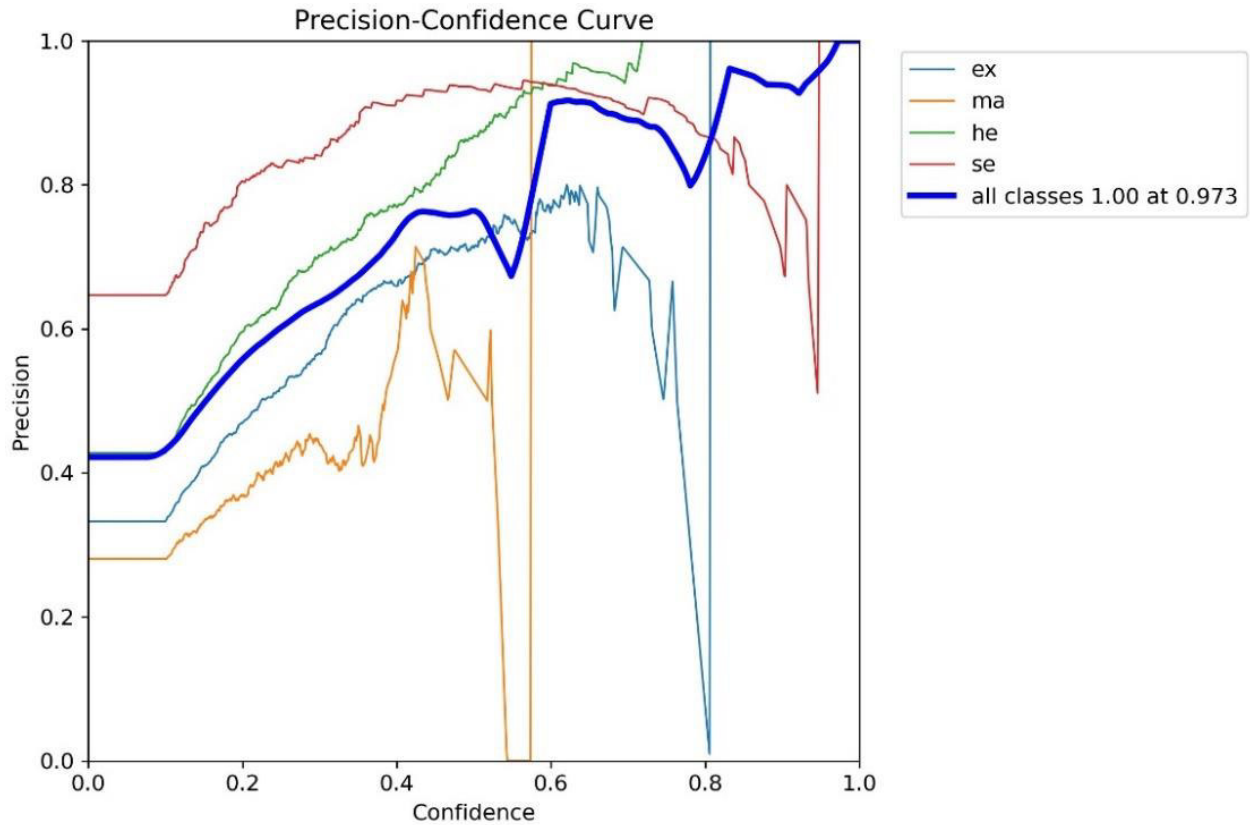


Figura 33. Curva de precisión-confianza para la detección de distintas lesiones con el modelo YOLOv8. La gráfica ilustra cómo la precisión varía según el nivel de confianza de las predicciones para cada clase de lesión: exudados (ex), microaneurismas (ma), hemorragias (he) y otras lesiones (se). Las líneas de diferentes colores representan cada clase específica, mientras que la línea azul gruesa muestra el comportamiento promedio para todas las clases combinadas. La precisión promedio alcanza un valor de 1.00 con una confianza de 0.973 para todas las clases, destacando el punto donde el modelo logra su máxima precisión. Las curvas individuales permiten observar cómo la precisión para cada clase aumenta con un mayor umbral de confianza, reflejando la relación entre la certeza del modelo y la calidad de las predicciones.

- **Curva de precisión-recall**

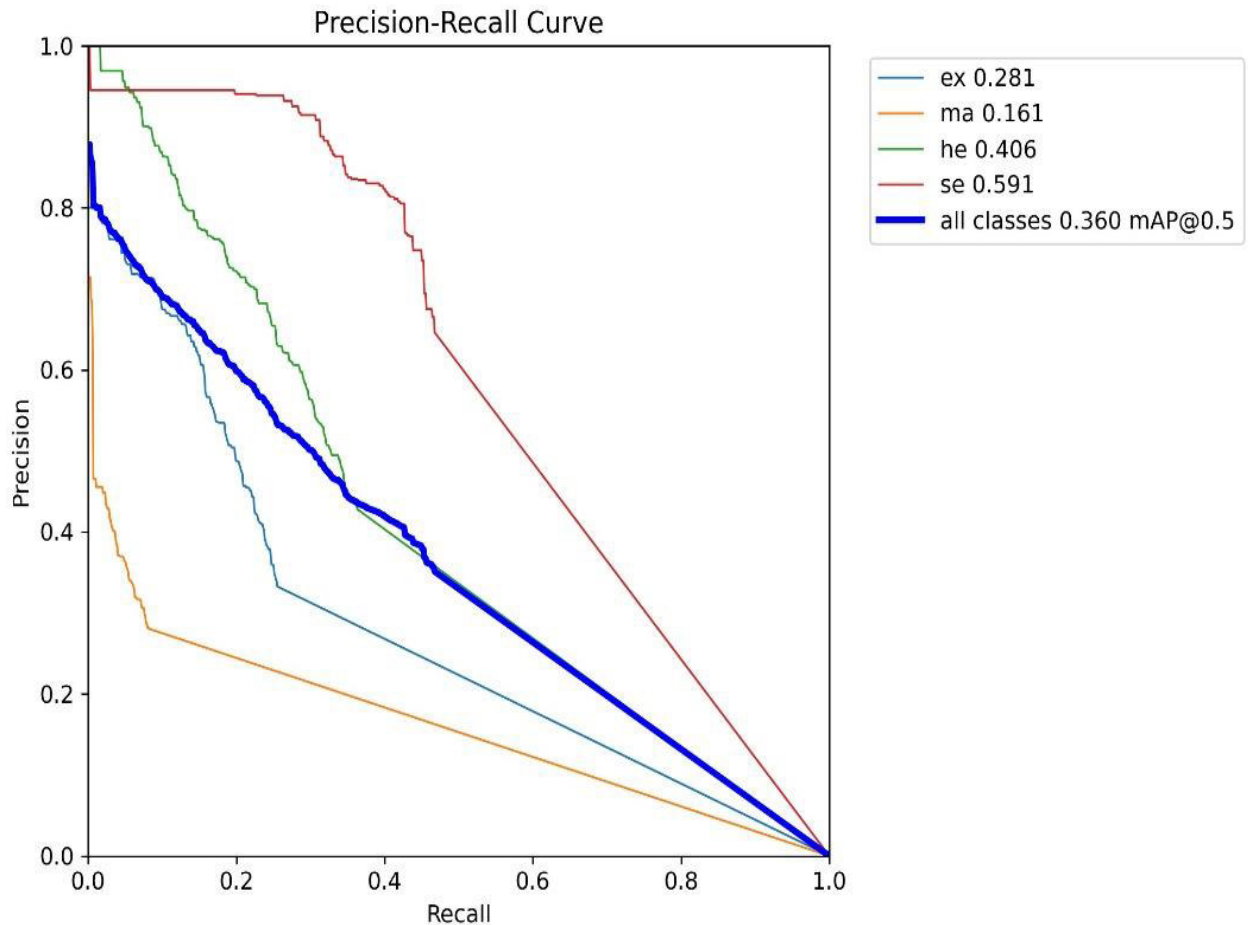


Figura 34. Curva de precisión-recall para la detección de diversas lesiones utilizando el modelo YOLOv8. La gráfica muestra la relación entre la precisión y el recall para cada clase de lesión específica: exudados (ex), microaneurismas (ma), hemorragias (he), y otras lesiones (se), representadas por líneas de diferentes colores. La línea azul gruesa indica el comportamiento combinado para todas las clases, con un mAP@0.5 de 0.360. Los valores numéricos junto a cada clase en la leyenda representan el área bajo la curva (AUC) para esa clase específica, mostrando la efectividad del modelo para detectar cada tipo de lesión. Un mayor área bajo la curva indica un mejor rendimiento del modelo, equilibrando precisión y recall. La clase "se" (en rojo) presenta el mejor rendimiento con una AUC de 0.591, mientras que la clase "ma" (en naranja) tiene la AUC más baja con 0.161.

- **Curva recall de confianza**

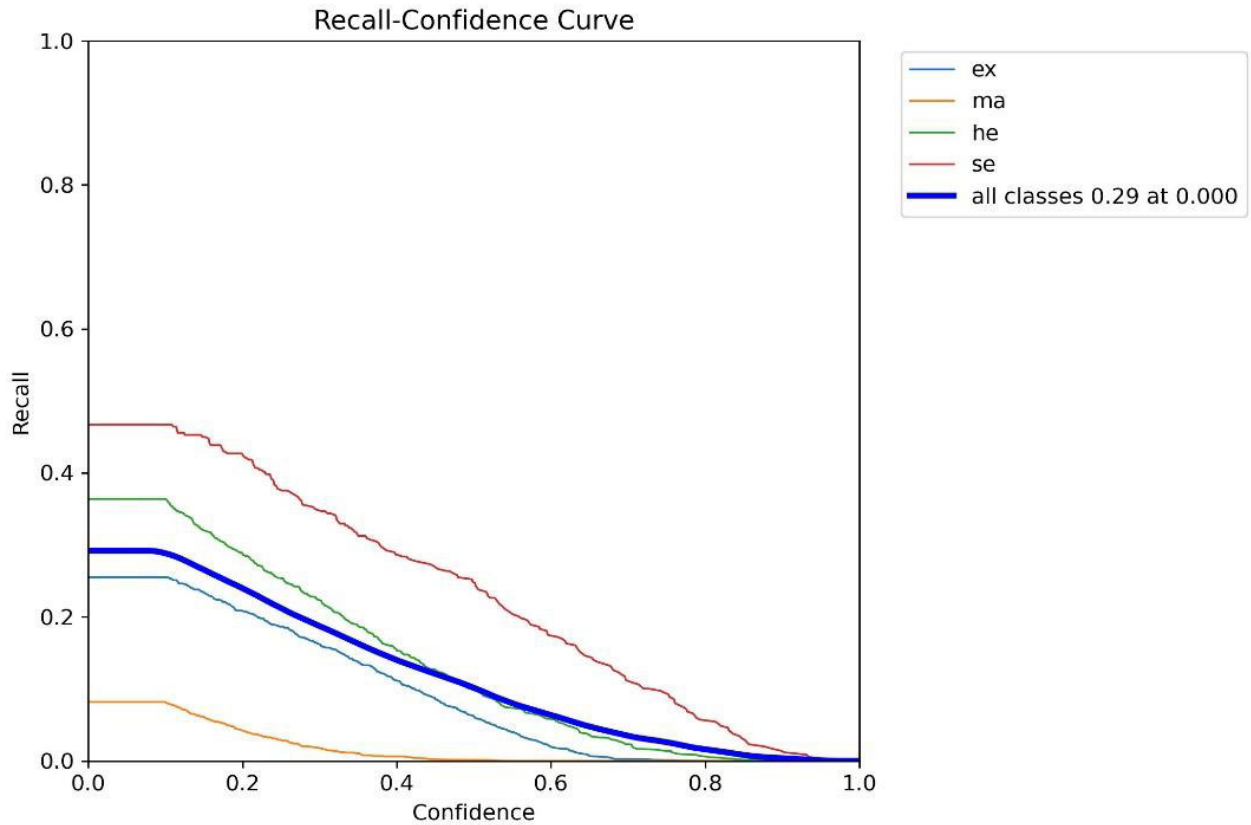


Figura 35. Curva de confianza-recall para la detección de distintas lesiones en un modelo YOLOv8. La gráfica muestra cómo varía el recall de cada clase (ex, ma, he, se) y de todas las clases combinadas en función del umbral de confianza. Cada línea de color representa una clase específica: exudados (ex), microaneurismas (ma), hemorragias (he) y otras lesiones (se). La línea azul gruesa indica el comportamiento combinado para todas las clases, con un recall promedio de 0.29 cuando la confianza es igual a 0.000. La curva refleja cómo el recall disminuye a medida que aumenta el umbral de confianza, lo que implica una mayor precisión a expensas de una menor detección de verdaderos positivos.

- **Gráficos de resultados generales**

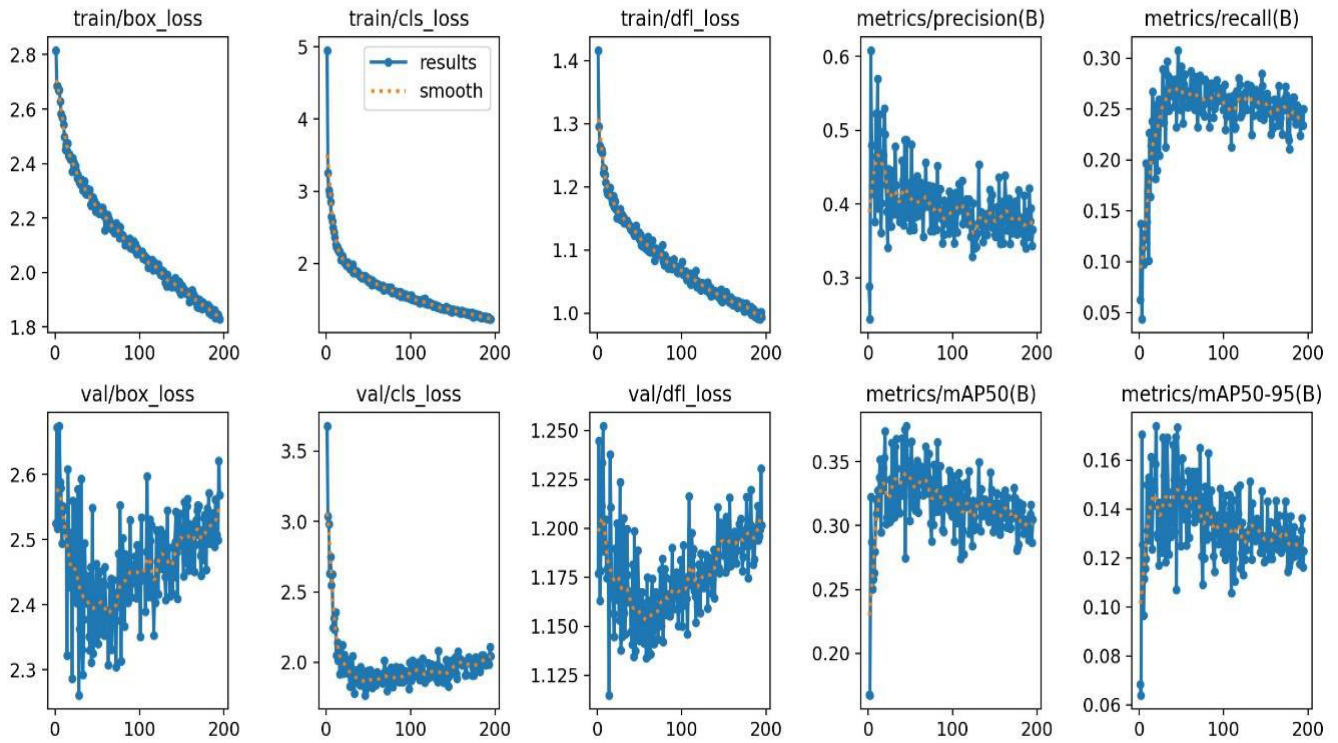
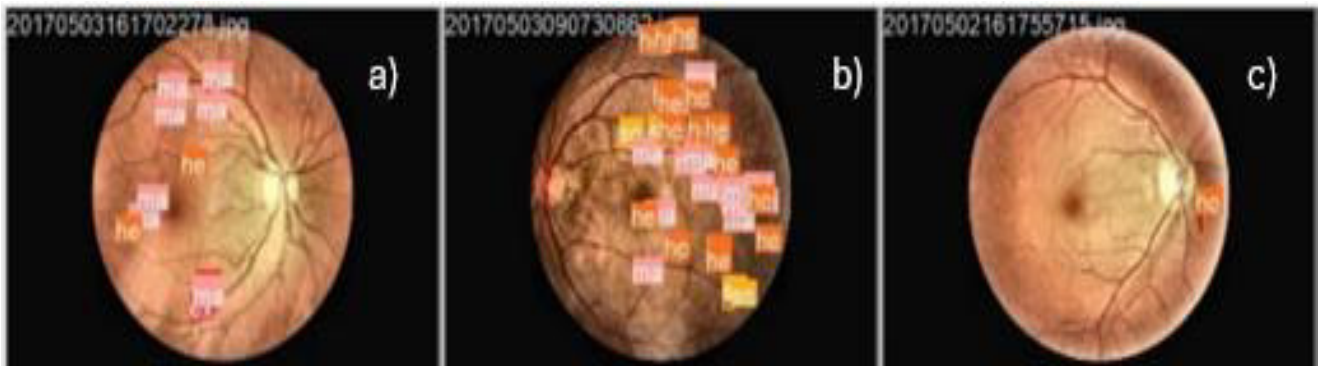


Figura 36. Gráficos variados de los distintos resultados obtenidos durante el proceso de entrenamiento y validación de YOLOv8 para detección de lesiones. En la fila superior se observan las métricas relacionadas con el entrenamiento, incluyendo la pérdida de la caja delimitadora (train/box_loss), la pérdida de clasificación (train/cls_loss), la pérdida de distribución de etiquetas (train/df_l_loss), la precisión (metrics/precision), y el recall (metrics/recall). En la fila inferior se muestran las métricas correspondientes para la validación, así como el mAP a diferentes umbrales de IoU (metrics/mAP50 y metrics/mAP50-95).

- Comparativa visual de etiquetas reales y predichas

ETIQUETAS REALES:



PREDICCIÓN DEL MODELO:



Figura 37. Comparativa de algunas imágenes muestra entre las etiquetas reales en las imágenes vs las etiquetas generadas por el modelo de predicción de YOLOv8. Se puede observar que en el caso a) su predicción detecta solamente una pocas lesiones en comparación a las etiquetas reales, para el caso b) se nota un incremento en la predicción, pero aun así queda a deber un poco, por último, para el caso c) observamos que la predicción fue correcta; sin embargo, en general se puede observar que la precisión de la detección de lesiones tiene un umbral de confianza bajo.

Para obtener más información de las métricas de rendimiento se puede consultar en la página oficial de documentación del equipo de Ultralytics [60].

8. Discusión

Los resultados obtenidos con YOLOv8 para la clasificación de la retinopatía diabética y la detección de lesiones son alentadores y abren nuevas posibilidades para el diagnóstico y el manejo de esta enfermedad. Sin embargo, a pesar de esto, es necesario destacar individualmente las tareas de clasificación y detección para tener una mejor perspectiva de lo que se obtuvo:

- **Clasificador**

En general el modelo de clasificación demostró un rendimiento que es bastante satisfactorio, pero a pesar de esto es crucial remarcar y analizar críticamente sus fortalezas y debilidades en este contexto específico:

Fortalezas:

- El modelo logró una precisión general prometedora, con un 82.7% en el conjunto de datos DDR, 71.7% en IDRiD y 57.4% en EyePACS. Estos resultados sugieren la capacidad del modelo para aprender patrones discriminativos relevantes a la clasificación de la RD.
- En base a las AUC se puede decir que: El modelo muestra una excelente capacidad para identificar imágenes sin retinopatía, con AUCs de 0.73, 0.87 y 0.95 en las diferentes gráficas.
- Las mejoras en la arquitectura del modelo reflejan un rendimiento creciente en la discriminación de varias clases, particularmente en las categorías "moderate" y "proliferative".
- Las mejoras en la arquitectura de YOLOv8, incluyendo el módulo C2f, el Decoupled Head, y la función de pérdida optimizada, contribuyen a la eficiencia del entrenamiento y a la capacidad de generalización del modelo.
- El uso de Transfer Learning, con pesos preentrenados en el dataset ImageNet, permite al modelo aprovechar el conocimiento previo sobre características visuales, acelerando el entrenamiento y mejorando el rendimiento.

Yolov8 posee potencial como una herramienta de apoyo al diagnóstico de la RD, ya que su capacidad para clasificar la gravedad de la enfermedad podría contribuir en la agilización del flujo de trabajo en los entornos clínicos, de tal forma que utilicen el modelo como primera línea de análisis, identificando casos que requieren atención médica inmediata. Además de tener potencial de mejorar la eficiencia del cribado, debido a su capacidad de procesar imágenes de forma rápida y precisa, podría facilitar la detección temprana de RD en poblaciones a gran escala, lo que nutrirá más el modelo.

➤ ***Debilidades:***

- ❖ Dificultad en la clasificación de etapas tempranas. El modelo muestra complicaciones para diferenciar correctamente las imágenes de RD leve y moderada, esto posiblemente debido a la sutileza de las lesiones en estas etapas. La baja precisión en estas clases (12% para mild en DDR, 8% en IDRiD y 10% en EyePACS) limita la capacidad del modelo para detectar la RD en las fases iniciales, lo cual compromete una de las justificaciones de la realización de esta investigación.
- ❖ Con base a las AUC se puede decir que: La AUC cercana a 0.5 para la clase "mild" en todas las gráficas indica una discriminación similar al azar, lo que es una debilidad significativa

para la detección temprana. Por lo cual sería óptimo buscar nutrir el modelo con datos de este estilo para poder mejorar este punto.

- ❖ Variabilidad en el rendimiento entre conjunto de datos. Hay una variabilidad significativa del clasificador entre los distintos dataset utilizados. Esta variabilidad podría ser indicativa de la sensibilidad del modelo a factores como la calidad de imagen, el equipo de adquisición, las características demográficas de la población y el protocolo de etiquetado utilizado en cada conjunto de datos.

- **Detector**

El detector de lesiones basado en el modelo YOLOv8 presentó un rendimiento destacable, mostrando una alta precisión en comparación con el estado del arte existente utilizando el mismo conjunto de datos, alcanzando el mAP más alto. Aun así, es importante revisar los pros y contras observados:

- **Fortalezas:**

- ✓ El modelo obtuvo una precisión general del 0.36, lo cual de forma normal puede verse como un resultado “bajo”, sin embargo, considerando que los datos reportados en el estado del arte haciendo uso del mismo conjunto de datos (DDR), reportan un mAP de 0.092 y 0.215, se puede concluir que se ha obtenido el valor más alto hasta ahora.
- ✓ Se observa una alta precisión en la detección de lesiones para niveles de confianza moderados a altos, lo cual se puede observar en el gráfico de *Precision-Confidence* (Figura 33).
- ✓ Se obtuvo un buen rendimiento para exudados duros (ex) y suaves (se), este detector es particularmente eficaz a la hora de clasificar estas dos clases, mostrando un buen equilibrio entre precisión y recall.
- ✓ Buen recall para detección de hemorragias (he), se mantiene un alto recall incluso a niveles de confianza bajos como se puede observar en la gráfica de *Recall-Confidence* (Figura 35).
- ✓ El modelo posee un punto de operación óptimo con un bajo umbral de confianza, obteniendo 29% de recuperación de todas las lesiones con una confianza de 0, lo que significa que esto es adecuado en aplicaciones donde la sensibilidad sea una prioridad.

- **Debilidades:**

- ❖ Se presentan dificultades para detectar microaneurismas (ma) de forma precisa y exhaustiva, lo que se evidencia en los bajos valores de precisión y recall. Esto puede deberse a la sutileza que poseen este tipo de lesiones en específico.
- ❖ Conservadurismo al asignar alta confianza a sus predicciones, sacrificando recall para obtener una mayor precisión. Esto se observa en la disminución del recall a medida que aumenta la confianza en la gráfica *Recall-Confidence* (Figura 35) y en el descenso gradual del F1 en la gráfica *F1-Confidence* (Figura 32).
- ❖ Algunas clases muestran caídas repentinas en la precisión para ciertos niveles de confianza, y curvas irregulares en los gráficos *Precision-Confidence* (Figura 33) y *Precision-Recall*

(Figura 34). Esto podría indicar la presencia de ejemplos atípicos, un posible sobreajuste a los datos de entrenamiento o dificultades para distinguir ciertas clases de lesiones.

9. Conclusiones

En el desarrollo de esta tesis se establecieron una serie de objetivos que fueron abordados mediante la implementación del modelo YOLOv8. A continuación, se presentan las conclusiones alcanzadas, resaltando la manera en que se lograron cumplir estos objetivos y las observaciones pertinentes:

Implementación del modelo YOLOv8 para la clasificación de imágenes de fondo de ojo en diferentes grados de la RD: Se logró implementar exitosamente el modelo YOLOv8, el cual demostró ser eficaz en la clasificación de imágenes de fondo de ojo en diversos grados de retinopatía diabética (RD), utilizando principalmente el conjunto de datos DDR como fuente principal de entrenamiento. El modelo fue capaz de identificar y clasificar con alta precisión las clases "healthy", "moderate" y "proliferative", con AUCs de 0.95, 0.78 y 0.85, respectivamente. Estas clases presentaron un desempeño sobresaliente, destacando la capacidad del modelo para diferenciar correctamente estos estados. Sin embargo, las clases "mild" y "severe" mostraron desafíos, con AUCs de 0.52 y 0.76, lo que refleja la dificultad para distinguir entre estas etapas más sutiles de la enfermedad.

Detección y localización de lesiones asociadas a la RD: El modelo mostró habilidades prometedoras en la detección y localización de lesiones como exudados duros, exudados suaves, microaneurismas y hemorragias. Se destacó un rendimiento sólido en la detección de exudados, con una alta precisión y un buen equilibrio entre precisión y recall. No obstante, se observó que la detección de microaneurismas fue menos precisa, posiblemente debido a la sutileza de estas lesiones en comparación con otras.

Evaluación del rendimiento del modelo en términos de las métricas de rendimiento más comunes para clasificadores y detectores: El rendimiento del modelo fue evaluado mediante diversas métricas, incluyendo precisión, recall, F1-score y AUC. El conjunto de datos DDR fue el principal utilizado, con resultados que destacan la capacidad del modelo para clasificar correctamente en la mayoría de las clases y detectar lesiones con eficacia. Las clases "healthy" y "proliferative" obtuvieron resultados particularmente altos, mientras que las clases "mild" y "severe" presentaron desafíos debido a la complejidad de las imágenes y las características visuales menos marcadas. En cuanto al detector, se evaluó la precisión en la localización de las lesiones identificadas, confirmando la capacidad del modelo para señalar con exactitud la presencia de exudados y hemorragias. Esta evaluación permitió identificar fortalezas y áreas de mejora, especialmente en la clasificación de casos más complejos y la detección precisa de lesiones más pequeñas.

Análisis de las limitaciones del modelo y potenciales propuestas de mejora: A lo largo del desarrollo del modelo, se identificaron ciertas limitaciones, como la dificultad para clasificar correctamente las etapas tempranas de la RD, especialmente la clase "mild". Además, la variabilidad en los resultados entre diferentes conjuntos de datos, como IDRiD y EyePACS, resalta la sensibilidad del modelo a la calidad de las imágenes y otras características del dataset. Para mejorar estas áreas, se sugiere ampliar la diversidad y cantidad de datos de entrenamiento, ajustar la arquitectura del modelo y considerar técnicas de preprocesamiento avanzadas.

En conclusión, la implementación de YOLOv8 ha demostrado ser un paso significativo hacia la mejora del diagnóstico y la gestión de la retinopatía diabética, ofreciendo un rendimiento robusto en múltiples clases y lesiones. Sin embargo, la investigación futura debe centrarse en perfeccionar el modelo, especialmente en la clasificación de las etapas iniciales de la RD y en la detección precisa de lesiones menos evidentes. Con estas mejoras, YOLOv8 tiene el potencial de convertirse en una herramienta esencial para el diagnóstico temprano y preciso de la RD en entornos clínicos.

Perspectivas futuras:

- Desarrollo de modelos de predicción de la progresión de la RD, más allá de la clasificación estática, se deben desarrollar modelos que puedan predecir la progresión de la RD en un

paciente individual, basándose en las características de las lesiones y otros factores de riesgo. Esto permitiría intervenciones tempranas y personalizadas para prevenir la pérdida de la visión.

- Integrar los datos clínicos y genómicos, explorar esta opción de añadir como parámetros a un modelo la historia clínica del paciente, el control glucémico y otros factores, podrían traer consigo el desarrollo de un modelo más robusto, preciso y personalizado.
- Explorar la integración de diferentes modalidades de imagen, como imágenes de fondo de ojo en color, angiografía con fluoresceína y tomografía de coherencia óptica, para obtener una visión más completa de la patología de la RD y mejorar la precisión de la detección.
- Mejoramiento en la detección de lesiones sutiles, aplicando técnicas de aumento de datos e investigando arquitecturas de red especializadas en la detección de objetos pequeños, como redes de atención o las redes piramidales.

10. Declaratoria de Autenticidad

Declaro que la presente tesis es el resultado de mi trabajo personal, y que todas las fuentes consultadas y citadas han sido debidamente acreditadas. Asimismo, certifico que he seguido todas las normas académicas de integridad y originalidad en la elaboración del presente documento.

En algunos párrafos de la tesis, se ha utilizado una herramienta de Inteligencia Artificial exclusivamente para la verificación ortográfica y de sintaxis, garantizando que el contenido y la estructura de estos fueran producto de mi autoría. Cualquier otra asistencia proporcionada por la herramienta ha sido revisada y modificada para cumplir con los estándares académicos y citada adecuadamente cuando ha sido necesario.

Esta declaración se incluye con el propósito de informar al público sobre el uso responsable y ético de las herramientas de apoyo disponibles en la actualidad.

11. Anexos

11.1. Licencia de uso de YOLOv8

Este proyecto cuenta con la licencia de uso tipo: GNU Affero General Public License v3.0, la cual puede consultarse a detalle en: [61].

11.2. Código para el procesamiento de imagen

```
import os
import cv2
import numpy as np
from tqdm import tqdm

def enhance_luminance_and_contrast(image):
    # 1. Mejora de la luminancia
    # Convertir la imagen al espacio de color HSV
    hsv_image = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
    h, s, v = cv2.split(hsv_image)

    # Calcular la luminancia como el máximo de los canales R, G y B
    luminance = np.maximum.reduce([image[:, :, 0], image[:, :, 1],
    image[:, :, 2]])

    # Calcular la ganancia de luminancia
    luminance_gain = v / np.maximum(luminance, 1)

    # Aplicar la corrección gamma
    gamma = 1/2.2 # Gamma según las especificaciones
    luminance_enhanced = np.power(luminance_gain * luminance, gamma)

    # Normalizar la luminancia mejorada al rango [0, 255]
    luminance_enhanced = np.uint8(255 * luminance_enhanced /
    np.max(luminance_enhanced))

    # 2. Mejora del contraste
    # Convertir la imagen al espacio de color LAB
    lab_image = cv2.cvtColor(image, cv2.COLOR_BGR2LAB)
    l, a, b = cv2.split(lab_image)

    # Aplicar CLAHE al canal L (luminosidad)
    clahe = cv2.createCLAHE(clipLimit=5, tileGridSize=(8, 8))
    l_enhanced = clahe.apply(l)

    # Fusionar los canales LAB nuevamente
    enhanced_lab_image = cv2.merge([l_enhanced, a, b])
```

```

    # Convertir la imagen de nuevo a espacio de color RGB
    enhanced_image = cv2.cvtColor(enhanced_lab_image,
cv2.COLOR_LAB2BGR)

    return enhanced_image

#Limpiar Ruido de la imagen
def apply_gaussian_filter(image):
    # Aplicar el filtro gaussiano
    blurred_image = cv2.GaussianBlur(image, (0, 0), sigmaX=2.5,
sigmaY=2.5)
    return blurred_image

def Gaussian_blur(image):
    Gauss = cv2.addWeighted(image, 4, cv2.GaussianBlur(image, (0, 0),
8), -4, 128)
    return Gauss

# Función para recortar la imagen centrada en la región de interés
(fondo de ojo)
def crop_eye_region(image):
    # Convertir la imagen a escala de grises
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

    # Aplicar umbral adaptativo para resaltar los bordes
    _, thresh = cv2.threshold(gray, 0, 255, cv2.THRESH_BINARY +
cv2.THRESH_OTSU)

    # Encontrar contornos en la imagen umbralizada
    contours, _ = cv2.findContours(thresh, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)

    # Encontrar el contorno más grande (asumiendo que es la región de
interés)
    contour = max(contours, key=cv2.contourArea)

    # Calcular el rectángulo delimitador mínimo que rodea el contorno
    x, y, w, h = cv2.boundingRect(contour)

#     # Recortar la región centrada en el contorno
    cropped_image = image[max(0, y - 10):min(y + h + 10,
image.shape[0]), max(0, x - 10):min(x + w + 10, image.shape[1])]

    return cropped_image

```

```

# Función para calcular el valor mínimo y máximo de los píxeles en
todas las imágenes de la carpeta de entrada
def calculate_min_max_pixel_values(input_folder):
    min_pixel_value = float('inf')
    max_pixel_value = float('-inf')

    image_files = os.listdir(input_folder)
    for filename in image_files:
        image_path = os.path.join(input_folder, filename)
        image = cv2.imread(image_path)

        if image is not None:
            min_pixel_value = min(min_pixel_value, np.min(image))
            max_pixel_value = max(max_pixel_value, np.max(image))

    return min_pixel_value, max_pixel_value

# Función para normalizar la imagen
def normalize_image(image, min_pixel_value, max_pixel_value):
    normalized_image = (image - min_pixel_value) / (max_pixel_value -
min_pixel_value)
    normalized_image = np.uint8(normalized_image * 255)
    return normalized_image

#Dibujar una máscara circular para quitar el fondo
def draw_circle(img):
    width = img.shape[1]
    height = img.shape[0]
    x = int(width/2)
    y = int(height/2)
    r = np.amin((x, y))
    circle_img = np.zeros((height, width), np.uint8)
    cv2.circle(circle_img, (x, y), int(r), 1, thickness=-1)

    # Asegurarse de que la máscara sea del mismo tipo de datos que la
imagen
    circle_img = circle_img.astype(np.uint8)

    # Asegurarse de que la máscara tenga el mismo tamaño que la imagen
    if circle_img.shape[:2] != img.shape[:2]:
        circle_img = cv2.resize(circle_img, (img.shape[1],
img.shape[0]))

    img = cv2.bitwise_and(img, img, mask=circle_img)

```

```

    return img

# Ruta de la carpeta de entrada y salida
input_folder = 'D:\TESIS_MAESTRIA\KAGGLE_DATASET\KAGGLE_IDRID SIN
MODIFICAR\\test_KAGGLE\severe'
output_folder = 'D:\TESIS_MAESTRIA\KAGGLE_DATASET\KAGGLE_IDRID SIN
MODIFICAR\\test_KAGGLE_PDI\severe'

# Crear el directorio de salida si no existe
if not os.path.exists(output_folder):
    os.makedirs(output_folder)

# Calcular los valores mínimos y máximos de píxeles en todas las
imágenes de la carpeta de entrada
min_pixel_value, max_pixel_value =
calculate_min_max_pixel_values(input_folder)

# Procesar cada imagen
image_files = os.listdir(input_folder)
for filename in tqdm(image_files, desc="Procesamiento de imagen",
unit="imagen"):
    input_image_path = os.path.join(input_folder, filename)
    image = cv2.imread(input_image_path)

    if image is not None:
        # Mejorar luminancia y contraste
        enhanced_image = enhance_luminance_and_contrast(image)

        # Aplicar filtro gaussiano
        image_cleaned = apply_gaussian_filter(enhanced_image)
        image_cleaned = crop_eye_region(image_cleaned)

        # Normalizar la imagen
        normalized_image = normalize_image(image_cleaned,
min_pixel_value, max_pixel_value)
        normalized_image = Gaussian_blur(normalized_image)
        normalized_image = draw_circle(normalized_image)

        # Guardar la imagen procesada en la carpeta de salida
        output_image_path = os.path.join(output_folder, filename)
        cv2.imwrite(output_image_path, normalized_image)
    else:
        print(f"No se pudo cargar la imagen: {input_image_path}")

```

11.3. Código para transformar etiquetas xml a yolo

```
from xml.dom import minidom
import os

lut = {}
lut["ex"] = 0
lut["ma"] = 1
lut["he"] = 2
lut["se"] = 3

def convert_coordinates(size, box):
    dw = 1.0 / size[0]
    dh = 1.0 / size[1]
    x = (box[0] + box[1]) / 2.0
    y = (box[2] + box[3]) / 2.0
    w = box[1] - box[0]
    h = box[3] - box[2]
    x = x * dw
    w = w * dw
    y = y * dh
    h = h * dh
    return (x, y, w, h)

def convert_xml2yolo(lut, xml_folder, yolo_folder):
    for xml_file in os.listdir(xml_folder):
        if xml_file.endswith(".xml"):
            xml_path = os.path.join(xml_folder, xml_file)
            xmldoc = minidom.parse(xml_path)
            fname_out = os.path.join(yolo_folder,
os.path.splitext(xml_file)[0] + '.txt')

            with open(fname_out, "w") as f:
                itemlist = xmldoc.getElementsByTagName('object')
                size = xmldoc.getElementsByTagName('size')[0]
                width =
int(size.getElementsByTagName('width')[0].firstChild.data)
                height =
int(size.getElementsByTagName('height')[0].firstChild.data)

                for item in itemlist:
                    classid =
item.getElementsByTagName('name')[0].firstChild.data
                    if classid in lut:
                        label_str = str(lut[classid])
```

```

        else:
            label_str = "-1"
            print("warning: label '%s' not in look-up
table" % classid)

            xmin =
int(item.getElementsByTagName('bndbox')[0].getElementsByTagName('xmin'
)[0].firstChild.data)
            ymin =
int(item.getElementsByTagName('bndbox')[0].getElementsByTagName('ymin'
)[0].firstChild.data)
            xmax =
int(item.getElementsByTagName('bndbox')[0].getElementsByTagName('xmax'
)[0].firstChild.data)
            ymax =
int(item.getElementsByTagName('bndbox')[0].getElementsByTagName('ymax'
)[0].firstChild.data)
            b = (float(xmin), float(xmax), float(ymin),
float(ymax))
            bb = convert_coordinates((width, height), b)
            f.write(label_str + " " + " ".join(["%.6f" % a)
for a in bb]) + '\n')

        print("Wrote %s" % fname_out)

def main():
    xml_folder = "/home/andres/DATASETS/DDR-
dataset/data_detection_normalized_COPIA/labels/val/" # Ruta de la
carpeta que contiene los archivos XML
    yolo_folder = "/home/andres/DATASETS/DDR-
dataset/data_detection_normalized_COPIA/labels/val/" # Ruta de la
carpeta donde se guardarán los archivos YOLO
    convert_xml2yolo(lut, xml_folder, yolo_folder)

if __name__ == '__main__':
    main()

```

11.4. Código para entrenar el clasificador de YOLOv8

```
from IPython import display
display.clear_output()

import ultralytics
ultralytics.checks()

#%%
from ultralytics import YOLO

from IPython.display import display, Image

#%%
#Pasar ruta del DATASET
#DATA_DIR = "/home/andres/DATASETS/DDR-dataset/sano_enfermo"

from ultralytics import YOLO

# Cargar el modelo
model =
YOLO('https://github.com/ultralytics/assets/releases/download/v8.2.0/yolov8n-cls.pt') # load yolov8n-cls.pta pretrained model (recommended for training)

# Entrenar el modelo
results = model.train(data="/home/andres/DATASETS/DDR-dataset/DR_GRADING_NORMALIZED/DATA_TRAIN_NORMALIZED", epochs=50, patience=100, batch=8, imgsz=640, device=0, seed=42, optimizer='auto', lr0=0.0001, lrf=0.01, dropout=0.5)
```

11.5. Código para entrenar el detector de YOLOv8

```
from IPython import display
display.clear_output()

import ultralytics
ultralytics.checks()

#%%
from ultralytics import YOLO

from IPython.display import display, Image
#%%
```



```

#Cargar el modelo
model =
YOLO("https://github.com/ultralytics/assets/releases/download/v8.1.0/y
olov8n.pt")

#Entrenar el modelo
results = model.train(data='config_detection.yaml', epochs=300,
patience=150, imgsz=1920, batch=4,optimizer='SGD', line_width=3,
iou=0.05, conf=0.1,dropout=0.5 ,agnostic_nms=True, augment=True,
lr0=0.01, lrf=0.0001, momentum=0.932, weight_decay=0.0005)

```

11.6. Código para obtener los resultados del clasificador

```

import os
import numpy as np
from ultralytics import YOLO
from sklearn.metrics import confusion_matrix, accuracy_score,
precision_score, recall_score, f1_score, roc_curve, auc,
cohen_kappa_score
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import time
from sklearn.preprocessing import label_binarize
from sklearn.multiclass import OneVsRestClassifier
from sklearn.linear_model import LogisticRegression

# Cargar el modelo YOLO
model = YOLO('D:/TESIS_MAESTRIA/KAGGLE_DATASET/weights/last.pt')

# Directorio base de las imágenes
base_directory = 'D:/TESIS_MAESTRIA/KAGGLE_DATASET/KAGGLE_IDRID SIN
MODIFICAR/test_KAGGLE'

# Directorio de salida
timestamp = time.strftime("%Y%m%d_%H%M%S")
output_directory =
f'D:/TESIS_MAESTRIA/KAGGLE_DATASET/KAGGLE_RESULTADOS_MUESTREO/last_{ti
mestamp}_{os.path.basename(base_directory)}'

# Crear el directorio de salida si no existe
os.makedirs(output_directory, exist_ok=True)

class_directories = sorted(os.listdir(base_directory))

```

```

# Listas para almacenar las etiquetas, probabilidades y predicciones
por clase
predicted_labels = []
true_labels = []
class_names = []
predicted_probabilities = []
predicciones_por_clase = []

# Iterar sobre las imágenes y realizar predicciones
for class_directory in class_directories:
    class_path = os.path.join(base_directory, class_directory)
    if os.path.isdir(class_path):
        image_files = [
            f for f in os.listdir(class_path) if
f.lower().endswith((".jpg", ".jpeg", ".png"))
        ]

        for image_file in image_files:
            image_path = os.path.join(class_path, image_file)
            results = model(image_path) # Predicción de la imagen
            names_dict = results[0].names
            probs = results[0].probs.data.tolist()

            predicted_label = names_dict[np.argmax(probs)]
            predicted_labels.append(predicted_label)

            # Asignar true_label aquí
            true_label = class_directory

            # Obtener la etiqueta verdadera (nombre de la carpeta)
            true_labels.append(class_directory)

            # Almacenar el nombre de la clase
            class_names.append(class_directory)

            # Almacenar las probabilidades predichas
            predicted_probabilities.append(probs)

            # Obtener el score de la predicción máxima
            max_score = max(probs)

            # Agregar información a la lista de predicciones por clase
            predicciones_por_clase.append({'Imagen': image_file,
                                           'Clase Real': true_label,

```

```

                                                                 'Clase Predicha':
predicted_label,
                                                                 'Score Máximo':
max_score}) # Agregar el score máximo al diccionario

# Convertir las etiquetas a formato numérico
class_mapping = {class_name: i for i, class_name in
enumerate(class_directories)}
true_labels_numeric = [class_mapping[label] for label in true_labels]
predicted_labels_numeric = [class_mapping[label] for label in
predicted_labels]

# Binarizar las etiquetas para cada clase
y_true_bin = label_binarize(true_labels_numeric,
classes=list(range(len(class_directories))))
y_pred_bin = label_binarize(predicted_labels_numeric,
classes=list(range(len(class_directories))))

# Calcular la matriz de confusión
conf_matrix = confusion_matrix(true_labels, predicted_labels,
labels=class_directories)

# Calcular la suma de cada fila en la matriz de confusión
row_sums = conf_matrix.sum(axis=1)

# Evitar la división por cero: asignar NaN a los elementos donde la
suma de la fila es cero
conf_matrix_normalized = np.divide(conf_matrix.astype('float'),
row_sums[:, np.newaxis], where=row_sums != 0)
conf_matrix_normalized[np.isnan(conf_matrix_normalized)] = 0.0 #
Reemplazar NaN con 0

# Calcular las métricas de evaluación
accuracy = accuracy_score(true_labels, predicted_labels)
precision = precision_score(true_labels, predicted_labels,
average='weighted')
recall = recall_score(true_labels, predicted_labels,
average='weighted')
f1 = f1_score(true_labels, predicted_labels, average='weighted')
kappa = cohen_kappa_score(true_labels, predicted_labels)

print("Accuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)
print("F1 Score:", f1)

```

```

print("Kappa : ", kappa)

# Guardar las métricas en un archivo CSV
metrics_df = pd.DataFrame({
    'Metric': ['Accuracy', 'Precision', 'Recall', 'F1 Score',
              'Kappa'],
    'Value': [accuracy, precision, recall, f1, kappa]
})
metrics_file_path = os.path.join(output_directory, 'metrics.csv')
metrics_df.to_csv(metrics_file_path, index=False)

# Graficar la matriz de confusión como una imagen
plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Greens',
            xticklabels=class_directories, yticklabels=class_directories)
plt.xlabel('Predicted labels')
plt.ylabel('True labels')
plt.title('Confusion Matrix')
plt.xticks(rotation=45)
plt.yticks(rotation=45)

plt.subplot(1, 2, 2)
sns.heatmap(conf_matrix_normalized, annot=True, cmap='Blues',
            xticklabels=class_directories, yticklabels=class_directories)
plt.xlabel('Predicted labels')
plt.ylabel('True labels')
plt.title('Normalized Confusion Matrix')
plt.xticks(rotation=45)
plt.yticks(rotation=45)

plt.tight_layout()
confusion_matrix_file = os.path.join(output_directory,
                                     'confusion_matrix.png')
plt.savefig(confusion_matrix_file)

# Crear un DataFrame con las predicciones por clase
df_predicciones = pd.DataFrame(predicciones_por_clase)

# Guardar el DataFrame en un archivo CSV
archivo_predicciones = os.path.join(output_directory,
                                     'predicciones_por_clase.csv')
df_predicciones.to_csv(archivo_predicciones, index=False)

# Calcula la curva ROC considerando un umbral de confianza del 0.75

```

```

df_predicciones = pd.read_csv(archivo_predicciones)

# Convertimos las clases a formato adecuado
classes = df_predicciones['Clase Real'].unique()
y_true = label_binarize(df_predicciones['Clase Real'],
classes=classes)

# Inicializa los arrays para almacenar las predicciones y los puntajes
y_score = np.zeros((len(df_predicciones), len(classes)))

# Rellenamos y_score con los puntajes, aplicando el umbral de
confianza del 0.75
for idx, class_label in enumerate(classes):
    y_score[:, idx] = df_predicciones.apply(lambda row: row['Score
Maximo'] if row['Clase Predicha'] == class_label and row['Score
Máximo'] >= 0.75 else 0, axis=1)

# Calcular ROC y AUC para cada clase
fpr = dict()
tpr = dict()
roc_auc = dict()

for i in range(len(classes)):
    fpr[i], tpr[i], _ = roc_curve(y_true[:, i], y_score[:, i])
    roc_auc[i] = auc(fpr[i], tpr[i])

# Graficar la curva ROC para cada clase
plt.figure()
for i in range(len(classes)):
    plt.plot(fpr[i], tpr[i], lw=2, label=f'Clase {classes[i]} (área =
{roc_auc[i]:.2f})')

plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('Tasa de Falsos Positivos')
plt.ylabel('Tasa de Verdaderos Positivos')
plt.title('Curva ROC Multiclase')
plt.legend(loc="lower right")
plt.show()

```

12. Referencias

- [1] K. Boyd, «AMERICAN ACADEMY OF OPHTHALMOLOGY,» 27 November 2023. [En línea]. Available: <https://www.aao.org/salud-ocular/enfermedades/retinopatia-diabetica#symptoms>. [Último acceso: 12 February 2024].
- [2] IAPB, «Retinopatía diabética - Agencia Internacional para la Prevención de la Ceguera,» La Agencia Internacional para la Prevención de la Ceguera, 03 08 2023. [En línea]. Available: <https://www.iapb.org/es/learn/knowledge-hub/eye-conditions/diabetic-retinopathy/>. [Último acceso: 07 05 2024].
- [3] R. Lalitha y P. Ajith, «Detection of Diabetic Retinopathy in Retinal Fundus Image Using YOLO-RF Model,» *2021 Sixth International Conference on Image Information Processing (ICIIP)*, vol. 6, pp. 105-109, 2021.
- [4] C. Santos, M. Aguiar, D. Welfer y B. Belloni, «A New Approach for Detecting Fundus Lesions Using Image Processing and Deep Neural Network Architecture Based on YOLO Model.,» *Sensors*, vol. 22, n° 17, p. 6441, 2022.
- [5] M. Chetoui y M. A. Akhloufi, «Explainable end-to-end deep learning for diabetic retinopathy detection across multiple datasets,» *Journal of Medical Imaging*, vol. 7, n° 4, p. 044503, 2020.
- [6] M. Merino Hernandez, M. Jauregui Garcia, J. Agirrezabala Iturralde, A. Urruzola Lizarribar, A. Guibelalde Gonzalez, M. Prat Madrazo, I. Vergara Mitxelorena y M. Soto Ruíz de Gordo, «Desarrollo de un software de cribado automático de retinopatía diabética,» *International Journal of Integrated Care*, vol. 19, n° 4, p. 74, 2019.
- [7] P. L. Akella y R. Kumar, «An advanced deep learning method to detect and classify diabetic retinopathy based on color fundus images,» *Graefe's Archive for Clinical and Experimental Ophthalmology*, vol. 262, pp. 231-247, 2023.
- [8] A. R. Wahab Sait, «A Lightweight Diabetic Retinopathy Detection Model Using a Deep-Learning Technique,» *Diagnostics*, vol. 13, n° 19, p. 3120, 2023.
- [9] S. G. Kobat, N. Baygin, E. Yusufoglu, M. Baygin, P. D. Barua, S. Dogan, O. Yaman, U. Celiker, H. Yildirim, R.-S. Tan, T. Tuncer, N. Islam y U. R. Acharya, «Automated Diabetic Retinopathy Detection Using Horizontal and Vertical Patch Division-Based Pre-Trained DenseNET with Digital Fundus Images.,» *Diagnostics*, vol. 12, n° 8, p. 1975, 2022.
- [10] M. Praveen y K. Yugal, «Smart detection and diagnosis of diabetic retinopathy using bat based feature selection algorithm and deep forest technique,» *Computers & Industrial Engineering*, vol. 182, p. 109364, 2023.
- [11] J. R. Dinesh Kumar y K. Priyadharsini, «Analysis of CNN Model based Classification of Diabetic Retinopathy Diagnosis,» *2021 2nd International Conference on Secure Cyber Computing and Communications (ICSCCC)*, pp. 180-185, 2021.
- [12] P. Prayas, S. Kundu y D. A. Kumar, «Detection of red lesions in retinal fundus images using YOLO V3,» *Journal of Ophthalmic Research Group*, vol. 7, n° 2, pp. 49-53, 2020.
- [13] D. Suvajit, C. M. Bonthala, M. B. Syed, D. C. Ronnie y N. C. S. N. Iyengar, «Classification of Diabetic Retinopathy Images by Using Deep,» *International Journal of Grid and Distributed Computing*, vol. 11, n° 1, pp. 89-106, 2018.
- [14] A. Gangwar Kumar y V. Ravi, «Home Evolution in Computational Intelligence Conference paper,» *Advances in Intelligent Systems and Computing*, vol. 1176, pp. 679-689, 2021.
- [15] E. G. Regalado Pozo, *Análisis de clasificadores supervisados para detectar la estructura vascular en imágenes retinianas patológicas*, Quito, 2019.
- [16] W. Aucchuasi, E. Flores, F. Sernaque, J. Cueva, M. Diaz y E. Oré, «Recognition of hard exudates using Deep Learning,» *Procedia Computer Science*, vol. 167, pp. 2343-2353, 2020.
- [17] F. Moreira, B. Schaun, J. Schneiders, M. Reis, M. Serpa y P. Navaux, «Impacto da Resolução na Detecção de Retinopatia Diabética com uso de Deep Learning,» *2020: ANAIS DO XX SIMPÓSIO BRASILEIRO DE COMPUTAÇÃO APLICADA À SAÚDE*, pp. 494-499, 2020.
- [18] S. Long, X. Huang, Z. Chen, S. Pardhan y D. Zheng, «Automatic Detection of Hard Exudates in Color Retinal Images Using Dynamic Threshold and SVM Classification: Algorithm Development and Evaluation,» *Hindawi-BioMed Research International*, vol. 2019, p. 13, 2019.

- [19] T. Li, Y. Gao, K. Wang, S. Guo, H. Liu y H. Kang, «Diagnostic assessment of deep learning algorithms for diabetic retinopathy screening,» *ELSEVIER-Information Sciences*, vol. 501, pp. 511-522, 2019.
- [20] W. L. Alyoubi, M. F. Abulkhair y W. M. Shalash, «Diabetic Retinopathy Fundus Image Classification and Lesions Localization System Using Deep Learning,» *Sensors*, vol. 11, n° 3704, p. 21, 2021.
- [21] R. Navaneethan y H. Devarajan, «Enhancing Diabetic Retinopathy Detection through Preprocessing and Feature Extraction with MGA-CSG Algorithm,» *Expert Systems with Applications*, p. 123418, 2024.
- [22] NIH, «Diabetic Retinopathy | National Eye Institute,» Nih.gov, 22 02 2024. [En línea]. Available: <https://www.nei.nih.gov/learn-about-eye-health/eye-conditions-and-diseases/diabetic-retinopathy>. [Último acceso: 2024 05 12].
- [23] CDC, «La diabetes y la pérdida de la visión,» Centers for Disease Control and Prevention, 19 12 2022. [En línea]. Available: <https://www.cdc.gov/diabetes/spanish/living/diabetes-vision-loss.html#:~:text=protegerle%20la%20vista,-.Retinopat%C3%ADa%20diab%C3%A9tica,parte%20de%20atr%C3%A1s%20del%20ojo..> [Último acceso: 12 05 2024].
- [24] Mayo Clinic, «¿Existe el riesgo de pérdida de la vista relacionada con la diabetes?-Retinopatía diabética - Diagnóstico y tratamiento - Mayo Clinic,» 21 02 2023. [En línea]. Available: <https://www.mayoclinic.org/es/diseases-conditions/diabetic-retinopathy/diagnosis-treatment/drc-20371617>. [Último acceso: 12 05 2024].
- [25] P. Olmos, A. Araya-Del-Pino, C. González, P. Laso, V. Irribarra y L. Rubio, «Pathophysiology of diabetic,» *Rev Méd Chile*, n° 137, pp. 1375-84, 01 10 2009.
- [26] M. Mookiah, U. Acharya, C. Chua, C. Lim, E. Ng y A. Laude, «Computer-aided diagnosis of diabetic retinopathy: A review,» *Computers in Biology and Medicine*, vol. 43, n° 12, pp. 2136-2155, 1 December 2013.
- [27] N. Salamat, M. M. Saad Missen y A. Rashid, «Diabetic retinopathy techniques in retinal images: A review,» *Artificial Intelligence in Medicine*, vol. 97, pp. 168-188, 13 June 2019.
- [28] K. Mittal y V. Rajam, «Computerized retinal image analysis - a survey,» *Multimed Tools Appl*, vol. 79, pp. 22389-22421, 23 May 2020.
- [29] M. Badar, M. Haris y A. Fatima, «Application of deep learning for retinal image analysis: A review,» *Computer Science Review*, vol. 35, p. 100203, February 2020.
- [30] Instituto Mexicano del Seguro Social, «Detección de RETINOPATÍA DIABÉTICA en Primer Nivel de Atención. Guía de Referencia Rápida Catálogo Maestro de Guías de Práctica Clínica: IMSS-735-14,» 2014. [En línea]. Available: <https://www.imss.gob.mx/sites/all/statics/guiasclinicas/735GRR.pdf>. [Último acceso: 12 05 2024].
- [31] D. Ferreira Piñeiro, I. Olmos Pineda y A. Olvera López, «Análisis de técnicas para la detección y clasificación de la retinopatía diabética mediante imágenes digitales,» *Lenguaje, conocimiento y tecnología educativa: avances recientes*, pp. 21-30, 2021.
- [32] Instituto Mexicano del Seguro Social, «Diagnóstico,» *Revista Médica del Instituto Mexicano del Seguro Social*, vol. 49, n° 5, pp. 551-562, 2011.
- [33] The International Agency for the Prevention of Blindness, «Retinopatía Diabética En Mexico - The International Agency for the Prevention of Blindness,» September 2021. [En línea]. Available: <https://www.iapb.org/news/retinopatia-diabetica-en-mexico/>. [Último acceso: 12 February 2024].
- [34] M. Sarah, «Ethical Considerations in AI-Driven Healthcare,» *News-Medical*, 6 11 2023. [En línea]. Available: <https://www.news-medical.net/health/Ethical-Considerations-in-AI-Driven-Healthcare.aspx>. [Último acceso: 17 05 2024].
- [35] M. M.-I. y H. L. Y.S. Abu-Mostafa, «Learning from Data: A Short Course,» 2012. [En línea]. Available: [Amlbook.com](http://amlbook.com).
- [36] R. S. y A. Barto, *Reinforcement Learning: An Introduction*, MIT Press, 2018.
- [37] A. Gerón, *Hands-On Machine Learning with Scikit-Learn, Keras, and Tensorflow.*, O'Reilly .
- [38] A. Gerón, «Chapter 10: Introduction to Artificial Neural Networks with keras,» de *Hands-On Machine Learning with Scikit-Learn, Keras, and Tensorflow*, O'Reilly.

- [39] S. Rozada Raneros, «Universidad de Valladolid, E.T.S.I. TELECOMUNICACIÓN,» 01 2021. [En línea]. Available: <https://uvadoc.uva.es/bitstream/handle/10324/45359/TFM-G1316.pdf>. [Último acceso: 19 05 2024].
- [40] C. L. C. D. d. León, «Desarrollo de una Red Neuronal Convolutiva para la identificación,» 2023.
- [41] E. d. H. Dorado, «DISEÑO, IMPLEMENTACIÓN Y EVALUACIÓN DE UNA RED NEURONAL CONVOLUCIONAL PARA LA DETECCIÓN DE PUNTOS PRINCIPALES EN NARANJAS,» 2020.
- [42] G. Boesch, «A Guide to YOLOv8 in 2024 - viso.ai,» viso.ai, 18 12 2023. [En línea]. Available: [https://viso.ai/deep-learning/yolov8-guide/#:~:text=The%20architecture%20consists%20of%20a,Feature%20Pyramid%20Network%20\(FPN\)..](https://viso.ai/deep-learning/yolov8-guide/#:~:text=The%20architecture%20consists%20of%20a,Feature%20Pyramid%20Network%20(FPN)..) [Último acceso: 19 03 2024].
- [43] A. Bochkovskiy, C.-Y. Wang y H.-Y. Liao, «YOLOv4: Optimal Speed and Accuracy of Object Detection,» arXiv.org, 23 04 2020. [En línea]. Available: <https://arxiv.org/pdf/2004.10934.pdf>. [Último acceso: 19 03 2024].
- [44] G. Maindola, «A Brief History of YOLO Object Detection Models From YOLOv1 to YOLOv5 - MLK - Machine Learning Knowledge,» MLK - Machine Learning Knowledge, 26 08 2021. [En línea]. Available: <https://machinelearningknowledge.ai/a-brief-history-of-yolo-object-detection-models/>. [Último acceso: 21 03 2024].
- [45] G. Jocher, A. Chaurasia y J. Qiu, «Ultralytics YOLO,» 10 01 2023. [En línea]. Available: <https://github.com/ultralytics/ultralytics>.
- [46] S. Jacob, «What is YOLOv8? The Ultimate Guide. [2024,]» Roboflow Blog, 11 01 2023. [En línea]. Available: <https://blog.roboflow.com/whats-new-in-yolov8/>. [Último acceso: 21 03 2024].
- [47] RangeKing, «Brief summary of YOLOv8 model structure #189,» Github, 10 01 2023. [En línea]. Available: <https://github.com/ultralytics/ultralytics/issues/189>. [Último acceso: 22 03 2024].
- [48] J. Cuello Navarro, C. Barraza Peña y J. Escorcía-Gutiérrez, «Una revisión de los métodos de Deep Learning aplicados a la detección automatizada de la retinopatía diabética,» *Revista Sextante*, n° 23, pp. 12-27, 2020.
- [49] P. S. K. R. K. M. D. G. S. V. M. F. Porwal P, «Indian Diabetic Retinopathy Image Dataset (IDRID): A Database for Diabetic Retinopathy Screening Research,» *Data*, vol. 3, n° 3, p. 25, 2018.
- [50] B. G. Cuadros J, «EyePACS: an adaptable telemedicine system for diabetic retinopathy screening,» *J Diabetes Sci Technol*, vol. 3, n° 3, pp. 509-16, 2009.
- [51] E. Dugas, J. J. y W. Cukierski, «Diabetic Retinopathy Detection,» Kaggle, 2015. [En línea]. Available: <https://kaggle.com/competitions/diabetic-retinopathy-detection>. [Último acceso: 23 02 2024].
- [52] M. Zhou, K. Jin, S. Wang y D. Qian, «Color Retinal Image Enhancement Based on Luminosity and Contrast Adjustment,» *IEEE Transactions on Biomedical Engineering*, vol. 65, n° 3, March 2018.
- [53] G. Huang, Z. Liu, L. van der Maaten y K. Weinberger, «Densely Connected Convolutional Networks,» de *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA., 2017.
- [54] G. Jocher y W. Xin, «Image Classification Datasets Overview,» ultralytics, 12 11 2023. [En línea]. Available: <https://docs.ultralytics.com/datasets/classify/#supported-datasets>. [Último acceso: 27 02 2024].
- [55] Ultralytics, «ultralytics-Configuration,» ultralytics, 12 11 2023. [En línea]. Available: <https://docs.ultralytics.com/es/usage/cfg/>. [Último acceso: 27 02 2024].
- [56] IMAGENET, «Image-net.org,» 2021. [En línea]. Available: <https://www.image-net.org/about.php>. [Último acceso: 22 04 2024].
- [57] J. Glenn, «Ultralytics YOLO format,» ultralytics, 12 11 2023. [En línea]. Available: <https://docs.ultralytics.com/datasets/detect/#ultralytics-yolo-format>. [Último acceso: 04 03 2024].
- [58] COCO, «COCO - Common Objects in Context,» 2020. [En línea]. Available: <https://cocodataset.org/#overview>. [Último acceso: 22 04 2024].
- [59] Ultralytics, «Results,» 2023. [En línea]. Available: <https://docs.ultralytics.com/reference/engine/results/>. [Último acceso: 25 04 2024].

- [60] Ultralytics, «YOLO Performance Metrics,» 2023. [En línea]. Available: <https://docs.ultralytics.com/guides/yolo-performance-metrics/#speed-metrics>. [Último acceso: 26 04 2024].
- [61] ultralytics, «ultralytics/LICENSE at main · ultralytics/ultralytics,» Github, 14 04 2023. [En línea]. Available: <https://github.com/ultralytics/ultralytics/blob/main/LICENSE>. [Último acceso: 01 08 2024].
- [62] International Council of Ophthalmology, «ICO Guidelines for Diabetic Eye Care,» 2017.
- [63] I. Kandel y M. Castelli, «Transfer Learning with Convolutional Neural Networks for Diabetic Retinopathy Image Classification. A Review,» *Medical Artificial Intelligence*, vol. 10, n° 6, 2020.



Universidad
de Guanajuato

CAMPUS LEÓN
DIVISIÓN DE CIENCIAS E INGENIERÍAS
DEPARTAMENTO DE FÍSICA

29 de Agosto del 2024

DR. DAVID Y. G. DELEPINE
DIRECTOR DE LA DIVISIÓN DE CIENCIAS E INGENIERÍAS
PRESENTE

Por este medio le comunico que he revisado la tesis titulada **DESARROLLO DE UN SISTEMA DE DETECCIÓN Y CLASIFICACIÓN DE RETINOPATÍA DIABÉTICA UTILIZANDO YOLOv8**, escrita por el estudiante JOSÉ ANDRÉS ECHEVESTE VÁZQUEZ, para efecto de presentarla y obtener el grado de MAESTRO EN CIENCIAS APLICADAS. El texto de la tesis se encuentra completo.

En mi opinión, la tesis cumple con los elementos necesarios para ser defendida ante el comité sinodal designado en fecha próxima que sea acordada de manera conjunta.

Agradeciendo su amable atención, aprovecho la ocasión para enviarle un cordial saludo.

Alma X. González Morales
Departamento de Física
División de Ciencias e Ingenierías
Universidad de Guanajuato



León, Guanajuato. 30 de agosto de 2024

ASUNTO: Revisión de Tesis

DR. DAVID YVES GISLAIN DELEPINE
DIRECTOR
DIVISIÓN DE CIENCIAS E INGENIERIAS

Por medio de la presente, le informo que he concluido la revisión de la tesis del *Ing. José Andrés Echeveste Vázquez*, cuyo título es "*Desarrollo de un sistema de detección y clasificación de retinopatía diabética utilizando YOLOv8*". Así también constato que he entrevistado y cuestionado a Andrés sobre los temas y tópicos desarrollados en su tesis; con lo que considero que Andrés tiene un pleno dominio de estos. Por lo cual consiento que su tema de tesis cumple con la calidad y solidez para poder tener su disertación.

Sin más por el momento le envío saludos cordiales.

Atentamente,

Una firma manuscrita en tinta azul que parece decir "C. Wiechers".

Dr. Carlos Herman Wiechers Medina
Profesor Investigador
Departamento de Física

e-mail: carherwm@fisica.ugto.mx

e-mail: ch.wiechers@ugto.mx



Dr. David Yves Ghislain Delepine
Director División de Ciencias e Ingenierías
Campus León

PRESENTE:

Por medio de este documento hago constar que una vez revisado el trabajo de tesis titulado: "Clasificación de Grado de Retinopatía Diabética y Detección de Lesiones Utilizando Yolov8", que presenta el Ing. José Andrés Echeveste Vázquez para obtener el grado de Maestro en Ciencias Aplicadas, lo encuentro satisfactorio y considero que cumple con los criterios de originalidad y calidad que se requiere en una tesis de posgrado, por lo que avalo se realice la defensa pública de esta tesis, para obtener el título mencionado.

Morelia, Michoacán a 28 de agosto de 2024

Atentamente

Dr. Pablo Genaro Martínez Torres
Profesor Investigador
Instituto de Física y Matemáticas

Instituto de Física y Matemáticas, Edificio C-3, Ciudad Universitaria, CP. 58040 Morelia
Michoacán, México. Tel: (443) 322 3500 Ext. 4140

#HumanistaPorSiempre



León, Guanajuato, 27 de agosto de 2024

Dr. David Yves Ghislain Delepine
Director División de Ciencias e Ingenierías
Campus León

PRESENTE:

Por medio de este documento hago constar que una vez revisado el trabajo de tesis titulado: "Clasificación de Grado de Retinopatía Diabética y Detección de Lesiones Utilizando Yolov8", que presenta el Ing. José Andrés Echeveste Vázquez para obtener el grado de Maestro en Ciencias Aplicadas, lo encuentro satisfactorio y considero que cumple con los criterios de originalidad y calidad que exige nuestra institución, por lo que avalo se realice la defensa pública de esta tesis, para obtener el título mencionado.

Atentamente
"La verdad os hará libres"

Una firma manuscrita en tinta azul que parece ser "C. Villaseñor Mora".

Dr. Carlos Villaseñor Mora
Profesor – Investigador Titular A