

# Aplicación de redes neuronales convolucionales para el diagnóstico automatizado de patologías oftálmicas

Jorge Miguel Acosta Arcinieagas \*, Diego Fernando Malagón Saenz<sup>†</sup>, Gabriel Santiago Murillo Barragan<sup>‡</sup>

joacostaa@unal.edu.co \*, gdmalagons@unal.edu.co<sup>†</sup>, gsmurilloba@unal.edu.co<sup>‡</sup>

Universidad Nacional de Colombia

---

## Resumen

La detección temprana y precisa de patologías oculares es esencial para prevenir la pérdida de visión y mejorar la calidad de vida de los pacientes. Este proyecto propone una metodología para desarrollar un sistema automatizado de diagnóstico basado en redes neuronales convolucionales (CNN), diseñado para clasificar imágenes de fondo de ojo en cuatro categorías: Retina Normal, Retinopatía Diabética, Catarata y Glaucoma. Se utilizará un conjunto de 4217 imágenes provenientes de una base de datos en *Kaggle*, las cuales ya han sido sometidas a preprocesamientos básicos, como la ecualización del histograma y segmentación de la imagen, para mejorar el contraste y aislar las regiones de interés. Este estudio presenta el desarrollo de un sistema automatizado de diagnóstico para patologías oftálmicas utilizando redes neuronales convolucionales (CNN) y técnicas de transferencia de aprendizaje. Se empleó EfficientNetB0 preentrenada para la extracción de características en imágenes de fondo de ojo, las cuales fueron clasificadas en cuatro categorías: retina normal, retinopatía diabética, catarata y glaucoma. El modelo fue optimizado mediante fine-tuning y evaluado a través de métricas clave como exactitud, precisión, sensibilidad, F1-Score y AUC-ROC, alcanzando resultados prometedores con una exactitud superior al 75 % y AUC superiores a 0.90 en la mayoría de las clases. Los hallazgos demuestran el potencial de este enfoque para mejorar la detección temprana y la interpretación de enfermedades oculares, estableciendo una base sólida para futuras aplicaciones clínicas en el aprendizaje de máquina.

**Keywords**— Imágenes Olfálmicas, CNN, Clasificación, diagnóstico automatizado .

---

## 1. INTRODUCCIÓN

LA salud visual es un componente esencial de la calidad de vida, y la detección temprana de enfermedades oftálmicas resulta crucial para prevenir la pérdida irreversible de visión. En este contexto, el análisis de imágenes de fondo de ojo se ha consolidado como una herramienta diagnóstica fundamental para identificar patologías como la retinopatía diabética, el glaucoma y la catarata. Tradicionalmente, el diagnóstico se basa en la interpretación manual de estas imágenes por parte de especialistas, lo que puede derivar en variabilidad interobservador, errores humanos y un elevado costo en términos de tiempo y recursos.

La irrupción de la inteligencia artificial y, en particular, de las redes neuronales convolucionales (CNN) ha abierto nuevas perspectivas para automatizar y estandarizar el proceso diagnóstico. Diversos estudios han demostrado que las CNN pueden alcanzar resultados comparables a los de expertos clínicos en tareas de clasificación de imágenes médicas, lo que respalda su aplicación en el ámbito oftalmológico [1]. Sin embargo, la adopción

clínica de estos modelos requiere no solo de altos índices de precisión, sino también de una adecuada interpretabilidad que permita comprender y validar las decisiones del modelo.

El presente trabajo propone el desarrollo de un sistema automatizado de diagnóstico basado en CNN, enfocado en la clasificación de imágenes de fondo de ojo en cuatro categorías: Retina Normal, Retinopatía Diabética, Catarata y Glaucoma. Para ello, se utiliza EfficientNetB0 preentrenada en ImageNet como base para la extracción de características, complementada con capas de pooling y densas que realizan la clasificación final. Además, se incorporan técnicas de interpretabilidad como saliency maps, para identificar las regiones de las imágenes que influyen en las predicciones, facilitando así la validación clínica del sistema.

Este enfoque integral, que combina preprocesamiento avanzado, transferencia de aprendizaje, optimización de hiperparámetros y técnicas de interpretabilidad, tiene el potencial de mejorar significativamente la precisión diagnóstica y ofrecer una herramienta confiable para la práctica clínica.

En consecuencia, este estudio no solo aborda la necesidad de automatización en el diagnóstico de patologías oftálmicas, sino que también sienta las bases para futuras aplicaciones de inteligencia artificial en el campo de la salud.

### 1.1. Estado del Arte

Diversos estudios han abordado la aplicación de IA en imágenes oftálmicas. Entre ellos se destaca el trabajo de Gulshan et al. [1] donde se demuestra que las CNN pueden alcanzar sensibilidad y especificidad comparables a las de expertos humanos en la detección de retinopatía diabética. Por otro lado, arquitecturas con U-Net [2] se han consolidado en tareas de segmentación en el ámbito biomédico, permitiendo delimitar estructuras anatómicas con alta precisión.

Aunque ambos enfoques han sido ampliamente estudiados, existen limitaciones para cada uno: en clasificación, aunque se trata de un método robusto, no se proporciona información detallada sobre la localización de las lesiones; en segmentación se requiere el uso de datos anotados de forma precisa y puede ser más costosa en términos computacionales. Por ello, la integración de ambos enfoques en un sistema híbrido representa una innovación que puede aportar ventajas significativas en la práctica clínica, al combinar la capacidad diagnóstica global con la precisión en la delimitación de regiones de interés.[3]

## 2. METODOLOGÍA

### 1. Adquisición de Datos

Recopilaremos un conjunto de imágenes de fondo de ojo provenientes de una base de datos disponible en la plataforma kaggle, estos datos ya vienen etiquetados según el tipo de enfermedad que se presenta en cada muestra, en total se tienen 4217 fotos las cuales se distribuyen de la siguiente manera:

- Retina Normal (1074)
- Retinopatía Diabética (1098)
- Catarata (1038)
- Glaucoma (1007)

Para asegurar la consistencia del conjunto de datos, esta base de datos ya viene con los siguientes pasos de preprocesamiento:

- **Equalize Histogram:** Se aplica para mejorar el contraste. La ecualización del histograma redistribuye los valores

de intensidad de píxeles para que la imagen tenga un mejor rango dinámico de luminosidad, esto facilita la detección de características relevantes durante el entrenamiento del modelo.[4]

- **Image Segmentation:** Ayuda a aislar las áreas de interés en la imagen, como los vasos sanguíneos y otras estructuras importantes del fondo del ojo, eliminando el ruido y las partes irrelevantes de la imagen.[5]

Sobre el origen de estos datos no se tiene mayor información, sin embargo dado el origen del autor de la publicación se puede asumir que estas imágenes son de origen asiático.

### 2. Preparación de Datos

Para la preparación de los datos se hace uso de la herramienta Google Colab y el lenguaje de programación python, así mismo se hace uso de la librería de código abierto de TensorFlow y demás librerías gráficas.

En primer lugar se importan las librerías a utilizar, para luego declarar la ruta de la base de datos en su última versión, una vez realizado este proceso, se establece una carpeta para el acceso a estos datos y se hacen las respectivas verificaciones.

A pesar de que los datos ya vengan con un preprocesamiento es necesario ajustar los datos para facilitar el proceso de convolucionado:

- **Redimensionamiento:** Ajustar todas las imágenes a un tamaño uniforme para facilitar el procesamiento por la red convolucional.
- **Normalización:** Ajustar los valores de píxeles para tener una media de cero y una desviación estándar de uno, esto para evitar dominancia en el proceso de aprendizaje, asegurar la estabilidad numérica y mejorar la interpretabilidad del modelo.
- **Filtrado de la base de datos:** Es posible que sea necesario descartar las imágenes borrosas y con defectos lo suficientemente importantes como para afectar el entrenamiento del modelo.

### 3. Arquitectura de la Red Convolucional

Se utilizará una arquitectura de red convolucional (CNN) para entrenar el modelo de clasificación, debido a que esta es una de las mejores

herramientas para la clasificación de imágenes. Esta se puede dividir en tres grupos:

- **Capas de Convolución:** Las cuales utilizando kernels que recorren la imagen realizando convoluciones que finalmente extraen las características locales de las imágenes.
- **Capas de Pooling:** Para prevenir el sobreajuste estas capas reducen la dimensionalidad de las características dejando solo la información más importante.
- **Capas Densas:** Toman las decisiones finales de clasificación utilizando las características y/o reglas aprendidas.

#### 4. Entrenamiento del Modelo

Se compila el modelo utilizando el optimizador Adam y la función de pérdida categorical crossentropy. El modelo se entrenará utilizando los generadores de datos definidos anteriormente.

- Se divide el conjunto de datos en subconjuntos de entrenamiento (70 %) y prueba (30 %).
- Se propone usar validación cruzada para obtener una estimación más robusta del rendimiento del modelo.
- Se ajusta la tasa de aprendizaje y el número de épocas para asegurar una convergencia óptima.

#### 5. Evaluación del Modelo

El modelo se evaluará utilizando métricas de rendimiento como la precisión, la sensibilidad, la especificidad y el AUC-ROC. Estas métricas se describen a continuación.

- **Exactitud (Accuracy):** La exactitud mide la proporción de predicciones correctas sobre el total de muestras, y se define como:

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

donde:

- **TP (True Positives):** número de casos positivos correctamente clasificados.
- **TN (True Negatives):** número de casos negativos correctamente clasificados.

- **FP (False Positives):** número de casos negativos clasificados erróneamente de casos positivos clasificados erróneamente como negativos.
- **Precisión (Precision):** La precisión indica la proporción de verdaderos positivos sobre el total de predicciones positivas:

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

- **Sensibilidad (Recall o Tasa de Verdaderos Positivos):** La sensibilidad mide la capacidad del modelo para identificar correctamente los casos positivos:

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3)$$

- **Especificidad (Specificity):** La especificidad evalúa la capacidad del modelo para identificar correctamente los casos negativos:

$$\text{Specificity} = \frac{TN}{TN + FP} \quad (4)$$

- **F1-Score:** El F1-Score es la media armónica entre la precisión y la sensibilidad, y se define como:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (5)$$

$$= \frac{2TP}{2TP + FP + FN}$$

- **Área Bajo la Curva ROC (AUC-ROC):** La curva ROC relaciona la Tasa de Verdaderos Positivos (TPR) con la Tasa de Falsos Positivos (FPR) para distintos umbrales de decisión. El AUC-ROC se define como:

$$AUC = \int_0^1 TPR(FPR) d(FPR) \quad (6)$$

#### Validación final

Una vez que el modelo haya sido entrenado y evaluado, se buscará validar el rendimiento del modelo utilizando otro set de datos no utilizados anteriormente, de esta manera se emulará su desempeño en un entorno real.

### 3. DESARROLLO.

El código implementado para el desarrollo del proyecto se puede dividir en 4 pasos principales, La carga de los datos de entrada, la definición del modelo, el entrenamiento y la visualización de los resultados, esto teniendo en cuenta la metodología explicada anteriormente

#### a) Datos de entrada.

- La descarga del conjunto de datos se realiza utilizando kagglehub. Luego estos datos son organizados en carpetas con el nombre de su respectiva etiqueta.
- Se utiliza ImageDataGenerator para preprocesar las imágenes (redimensionamiento, normalización, aumento de datos) y dividir las en conjuntos de entrenamiento y validación.

#### b) Definición de la Arquitectura de la CNN.

La arquitectura del modelo se desarrollo con ayuda de keras, una libreria de python que simplifica el desarrollo de este tipo de redes. Esta arquitectura consta de 3 bloques de convolución,normalización por lotes y pooling para extraer características locales, seguidos una capa de aplanamiento para su posterior procesamiento en las capas densas las cual realizan la clasificación final. Esta estructura se puede ver resumida en la figura 1.

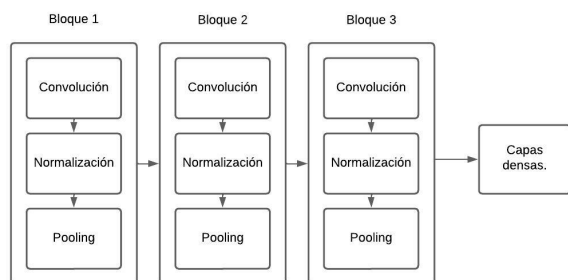


Figura 1: Arquitectura de la CNN.

Es importante aclarar que cada capa de convolución presente una función de activación relu, además se agrega un dropout al final de las capas densas el cual dado el momento desactiva aleatoriamente un porcentaje de neuronas en una capa específica, esto significa que esas neuronas no se utilizan para propagar la información hacia adelante ni para calcular los gradientes durante la retropropagación. Esto como técnica de regularización

para prevenir el sobreajuste en estas redes. Finalmente esta arquitectura no funcionó de manera que se esperaba, los resultados daban a entender que la red estaba clasificando de manera aleatoria. Es por esta razón que se opto por usar transfer learning usando arquitectura diferente con una red preentrenada, para esto se utilizó EfficientNetB0 como base del modelo, una arquitectura de redes neuronales convolucionales (CNN) eficiente y de alto rendimiento que se ha usado previamente en arquitecturas de reconocimiento y análisis de imágenes biomédicas. [6] Luego, se agregó una capa de Global Average Pooling que agrupa las redes de EfficientNet con la red que ya se tenia, una capa de Dropout y una capa densa con su respectiva función de activación softmax para la clasificación. Esta estructura final se puede ver observar más claramente en la figura 2.

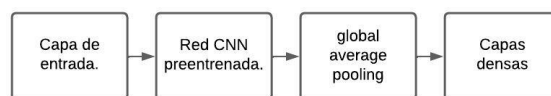


Figura 2: Arquitectura de la CNN.

Por otro lado en la figura 3 se puede observar con más detalle cada capa y sus parámetros.

Layer (type)	Output Shape	Param #
input_layer_2 (InputLayer)	(None, 224, 224, 3)	0
efficientnetb0 (Functional)	(None, 7, 7, 1280)	4,049,521
global_average_pooling2d (GlobalAveragePooling2D)	(None, 1280)	0
dropout (Dropout)	(None, 1280)	0
dense (Dense)	(None, 4)	5,124
Total params: 4,054,695 (15.47 MB)		
Trainable params: 5,124 (20.02 KB)		
Non-trainable params: 4,049,521 (15.45 MB)		

Figura 3: Arquitectura de la CNN.

#### c) Entrenamiento.

El modelo se compila con el optimizador Adam (Adaptive Moment Estimation) el cuales un algoritmo de optimización que ajusta los pesos de la red neuronal se eligió sobre otros debido a su eficiencia y capacidad para manejar gradientes dispersos.

El modelo se entrenó finalmente con un 70 % de los datos para entrenamiento, un 15 %

para validación y 15% para texteo, por lo que no fue necesario realizar una validación cruzada. Por otro lado, tras varias pruebas e intentos el mejor resultado obtenido fue con una tasa de aprendizaje de  $1e-5$  y con 20 épocas.

Luego, se realiza fine-tuning descongelando parte de la base de EfficientNetB0 y entrenando durante 10 épocas más con un learning rate reducido para ajustar los pesos de la base a las características específicas del dataset.

## 4. RESULTADOS Y DISCUSIÓN

En esta sección se presentan los resultados obtenidos luego del entrenamiento del modelo con EfficientNetB0 y se discuten los hallazgos en cada una de las etapas posteriores al entrenamiento inicial.

### 4.1. Evaluación del Modelo

Luego del entrenamiento inicial, en el cual se utilizó EfficientNetB0 preentrenada con sus pesos congelados y se entrenó únicamente la parte superior (capas de pooling, dropout y la capa densa final), se evaluó el desempeño del modelo en el conjunto de prueba. Las métricas obtenidas incluyeron:

- **Exactitud (Accuracy):** Se alcanzó una exactitud del 72% en el conjunto de prueba, para el modelo inicial antes del fine tuning lo cual indica la proporción de imágenes clasificadas correctamente.
- **Precisión, Sensibilidad y F1-Score:** El análisis del reporte de clasificación mostró que, en general, el modelo presenta una buena capacidad para detectar las diferentes patologías, aunque se identificaron algunas dificultades entre las clases de cataratas y la retina normal. Esto se puede evidenciar en la matriz de confusión inicial en la figura 4 y en los valores para precisión, sensibilidad y F1-Score para la figura 5

Estos resultados preliminares evidenciaron la robustez de la transferencia de aprendizaje, aunque se observó que la falta de ajuste fino en la base preentrenada podía limitar la adaptación del modelo a las características específicas del conjunto de datos oftálmico.

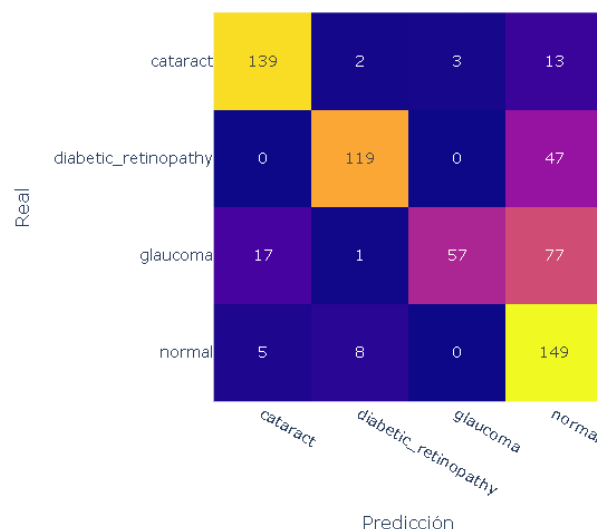


Figura 4: Matriz de confusión inicial

Accuracy inicial en test: 0.728414442700157  
Reporte de clasificación (inicial):

	precision	recall	f1-score	support
cataract	0.86	0.89	0.87	157
diabetic_retinopathy	0.92	0.72	0.80	166
glaucoma	0.95	0.38	0.54	152
normal	0.52	0.92	0.67	162
accuracy			0.73	637
macro avg	0.81	0.72	0.72	637
weighted avg	0.81	0.73	0.72	637

Figura 5: Precisión, sensibilidad y F1- Score inicial

### 4.2. Fine-Tuning y Ajuste del Modelo

Con el objetivo de mejorar la adaptación del modelo a la tarea específica, se procedió a realizar fine-tuning. En esta etapa se descongelaron las capas superiores de EfficientNetB0 (a partir de cierta capa, determinada empíricamente) y se recompiló el modelo con un learning rate reducido ( $1 \times 10^{-5}$ ). Además, se incorporaron *class weights* para mitigar el efecto del desbalance de clases.

Durante el fine-tuning se entrenó el modelo durante un número reducido de épocas. Los resultados indicaron una convergencia más estable y, en algunos casos, una mejora en la precisión de predicción para aquellas clases inicialmente menos representadas. El uso del learning rate reducido permitió ajustar los pesos sin sobrescribir el conocimiento previamente adquirido por la red.

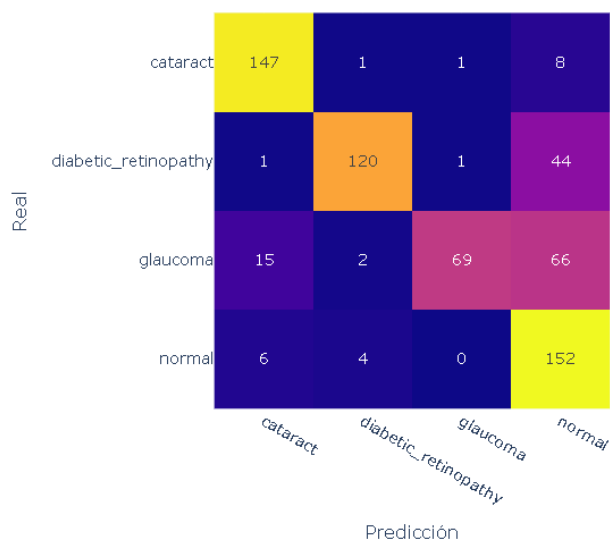


Figura 6: Matriz de confusión después del fine tuning

```
Accuracy final en test: 0.7660910518053375
Reporte final de clasificación:
```

	precision	recall	f1-score	support
cataract	0.87	0.94	0.90	157
diabetic_retinopathy	0.94	0.72	0.82	166
glaucoma	0.97	0.45	0.62	152
normal	0.56	0.94	0.70	162
accuracy			0.77	637
macro avg	0.84	0.76	0.76	637
weighted avg	0.84	0.77	0.76	637

Figura 7: Precisión, sensibilidad y F1- Score después del fine tuning

#### 4.3. Evaluación Final y Análisis de Métricas

Una vez finalizado el proceso de fine-tuning, se realizó una evaluación final del modelo en el conjunto de prueba. Las métricas clave fueron las siguientes:

- **Exactitud:** La exactitud mejoró hasta un 76 % evidenciando una leve mejora en el modelo.
- **Reporte de Clasificación:** Las métricas de precisión, sensibilidad y F1-Score presentaron mejoras, especialmente en aquellas clases con menor representación en el conjunto de entrenamiento como la clase normal. Estos resultados se muestran en la figura 7
- **AUC-ROC:** El Área Bajo la Curva ROC (AUC) se situó en valores superiores a 0.90 para la mayoría de las clases, lo que indica una alta capacidad de discrimina-

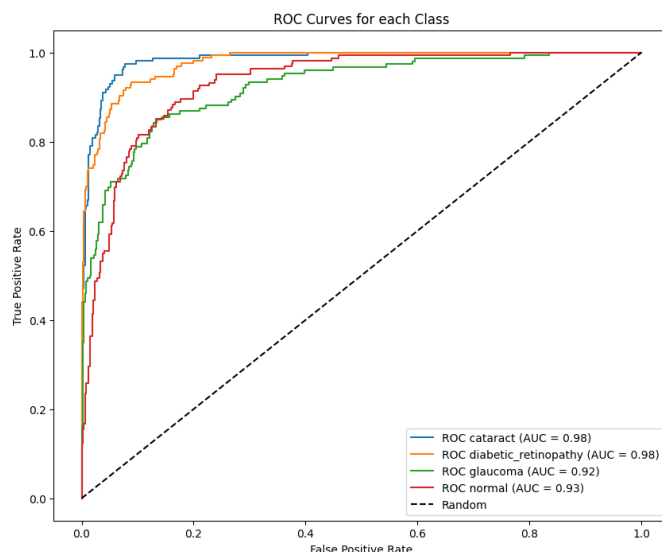


Figura 8: ROG-AUC para cada clase

ción del modelo. Esto se puede mirar en la figura 8

El análisis de estas métricas proporciona una perspectiva integral del desempeño del modelo, permitiendo identificar tanto su capacidad discriminativa global como su rendimiento por cada clase. Estas herramientas son fundamentales para validar la robustez del sistema en un entorno clínico, donde la precisión y la interpretabilidad son esenciales.

#### 4.4. Interpretabilidad y Análisis de Saliency Maps

La interpretabilidad de los modelos de aprendizaje profundo es esencial, especialmente en aplicaciones médicas, donde comprender y validar las decisiones del modelo es crucial para su adopción clínica. En este sentido, los *saliency maps* ofrecen una herramienta valiosa al resaltar las regiones de la imagen que tienen mayor influencia en la predicción del modelo.

Los *saliency maps* se generan calculando los gradientes de la salida correspondiente a una clase específica respecto a la imagen de entrada. La magnitud de estos gradientes indica la sensibilidad del modelo ante cambios en los píxeles, lo que permite identificar qué áreas contribuyen de manera significativa a la decisión final. Esta técnica, por tanto, proporciona una visualización intuitiva que ayuda a los expertos clínicos a validar si el modelo se centra en regiones relevantes del fondo de ojo, tales



como los vasos sanguíneos o zonas afectadas por patologías.

La Figura 9 ilustra un ejemplo comparativo: en la columna izquierda se presenta una imagen de prueba original, mientras que en la columna derecha se muestra su correspondiente *saliency map*. Como se observa, las áreas con mayor intensidad en el *saliency map* (representadas en colores cálidos) indican las regiones donde el modelo concentra su atención al realizar la clasificación.

Esta comparación es fundamental para:

- **Validar el modelo:** Asegurar que el sistema se centra en características relevantes y no en artefactos o regiones irrelevantes.
- **Aumentar la confianza clínica:** Permitir que los especialistas comprendan y verifiquen las bases de las predicciones, facilitando la integración del sistema en el flujo de trabajo clínico.
- **Identificar oportunidades de mejora:** Revelar posibles errores en el preprocesamiento o en la arquitectura del modelo, lo cual puede conducir a ajustes que optimicen la precisión diagnóstica.

En conclusión, la implementación de *saliency maps* no solo añade una capa de interpretabilidad al sistema, sino que también contribuye a validar la eficacia del modelo, permitiendo un análisis detallado de su comportamiento. Esto es especialmente relevante en el ámbito de la salud, donde la transparencia y la confianza en las predicciones son fundamentales para la toma de decisiones clínicas.

## 5. CONCLUSIONES.

- El modelo de clasificación basado en EfficientNetB0, complementado con un proceso de fine-tuning y técnicas interpretativas (por ejemplo, saliency maps), ha demostrado ser eficaz en la detección de diversas patologías oculares. En particular, se observó una alta tasa de verdaderos positivos en la detección de cataratas, lo que evidencia su capacidad para identificar características patológicas relevantes.
- No obstante, se identificaron desafíos importantes en la clasificación del glaucoma, puesto que el modelo presentó

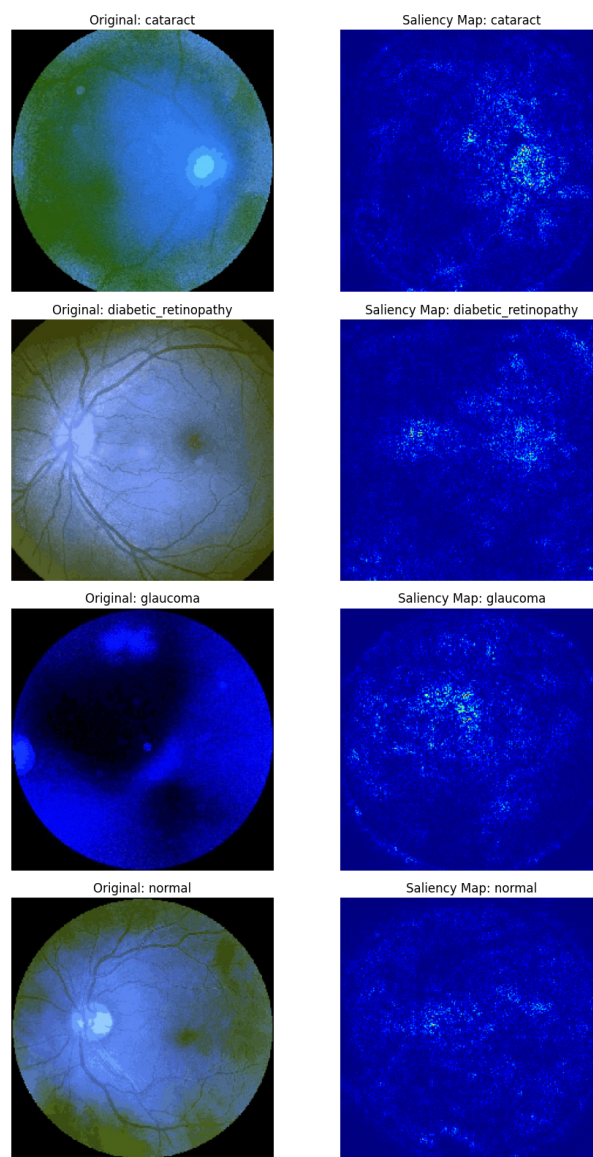


Figura 9: Comparación entre una imagen de fondo de ojo original (izquierda) y su *saliency map* correspondiente (derecha), que resalta las regiones de mayor importancia para la predicción del modelo.

mayores complicaciones en esta categoría. Esto sugiere la necesidad de ajustar el preprocesamiento o explorar modificaciones en la arquitectura para optimizar la sensibilidad y precisión en la identificación de esta patología.

- La aplicación del fine-tuning permitió mejorar la exactitud global del modelo, incrementándose de un 72 % a un 76 %, y se registraron mejoras en las métricas de precisión, sensibilidad y F1-Score. Estos resultados destacan la importancia de adaptar los pesos preentrenados a las ca-

racterísticas específicas del conjunto de datos oftálmico.

- En conjunto, la metodología implementada, que combina preprocesamiento avanzado, transferencia de aprendizaje y técnicas interpretativas, establece una base técnica robusta para el desarrollo de sistemas automatizados de diagnóstico en oftalmología. Este enfoque no solo mejora la precisión diagnóstica, sino que también añade un nivel de transparencia fundamental para la validación clínica.

## 6. PERSPECTIVAS FUTURAS

- **Implementación y Optimización:** Se recomienda migrar el código a entornos colaborativos (por ejemplo, GitHub) y aprovechar recursos de computación en la nube para superar las limitaciones de hardware actuales. Esto facilitará la experimentación con arquitecturas más complejas y entrenamientos a mayor escala.
- **Integración de Técnicas Interpretativas Adicionales:** Además de los saliency maps, se plantea explorar técnicas complementarias como Integrated Gradients u Occlusion Sensitivity, las cuales pueden ofrecer una visión más completa de las regiones críticas que influyen en las predicciones del modelo.
- **Ampliación y Diversificación del Conjunto de Datos:** Es fundamental incorporar imágenes provenientes de diversas fuentes, especialmente datos locales y de diferentes poblaciones, para evaluar la robustez y generalización del sistema, reduciendo posibles sesgos y mejorando la aplicabilidad clínica.
- **Mejoras en la Arquitectura del Modelo:** Se propone investigar la integración de CNN con Vision Transformers (ViT) o evaluar arquitecturas híbridas que permitan capturar tanto características locales como globales, optimizando así el desempeño en la clasificación de patologías complejas.
- **Validación en Entornos Clínicos Reales:** La realización de estudios prospectivos en colaboración con instituciones de salud es crucial para validar el sistema en condiciones reales de uso. La re-

troalimentación clínica permitirá ajustar parámetros, mejorar la interpretabilidad y garantizar que el modelo cumpla con los estándares requeridos en la práctica médica.

## REFERENCIAS

- [1] V. Gulshan, L. Peng, M. Coram, M. C. Stumpe, D. Wu, A. Narayanaswamy, S. Venugopalan, K. Widner, T. Madams, J. Cuadros, *et al.*, "Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs," *jama*, vol. 316, no. 22, pp. 2402–2410, 2016.
- [2] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical image computing and computer-assisted intervention—MICCAI 2015: 18th international conference, Munich, Germany, October 5–9, 2015, proceedings, part III 18*, pp. 234–241, Springer, 2015.
- [3] A. E. Ilesanmi, T. Ilesanmi, and G. A. Gbotoso, "A systematic review of retinal fundus image segmentation and classification methods using convolutional neural networks," *Healthcare Analytics*, vol. 4, p. 100261, 2023.
- [4] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Prentice Hall, 3rd ed., 2007.
- [5] W. N. Jasim and R. J. Mohammed, "A survey on segmentation techniques for image processing," *Iraqi Journal for Electrical & Electronic Engineering*, vol. 17, no. 2, 2021.
- [6] Z. Arif, R. Y. N. Fu'adah, S. Rizal, and D. Ilhamdi, "Classification of eye diseases in fundus images using convolutional neural network (cnn) method with efficientnet architecture," *JRTI (Jurnal Riset Tindakan Indonesia)*, vol. 8, no. 1, pp. 125–131, 2023.

Hecho en L<sup>A</sup>T<sub>E</sub>X