Patricia Lázaro Tello
plazarotello@uoc.edu

Clasificador Neuronal para la detección de Glaucoma

Patricia Lázaro Tello

Introducción

El **glaucoma** es una enfermedad crónica e irreversible del nervio óptico que constituye la segunda causa de ceguera de los países occidentales, solo por detrás de la diabetes. Existen diferentes tipos de glaucomas, según su origen (primario y secundario) y según la amplitud angular (ángulo abierto o cerrado).

El ojo produce humor acuoso constantemente; este líquido es drenado por el ángulo de drenaje. Sin embargo, si el ángulo de drenaje no funciona correctamente (no drena tanto líquido como debería), el humor acuoso se acumula, aumentando la presión interior del ojo y dañando el nervio óptico en el proceso.

Entre las pruebas médicas necesarias para diagnosticar glaucoma se encuentran medir la presión ocular, revisar el ángulo de drenaje del ojo, examinar el nervio óptico, realizar pruebas de visión periférica, medir el espesor de la córnea y **realizar una imagen o medición por ordenador del nervio óptico**.

Esta última prueba, llamada **retinografía**, es el objeto de estudio de este trabajo. Actualmente, la mayoría de sistemas de detección automática de glaucoma que utilizan retinografías utilizan características creadas a medida; el objetivo de [2] y este trabajo es evaluar la viabilidad de un sistema que utilice características inferidas por el modelo subyacente.

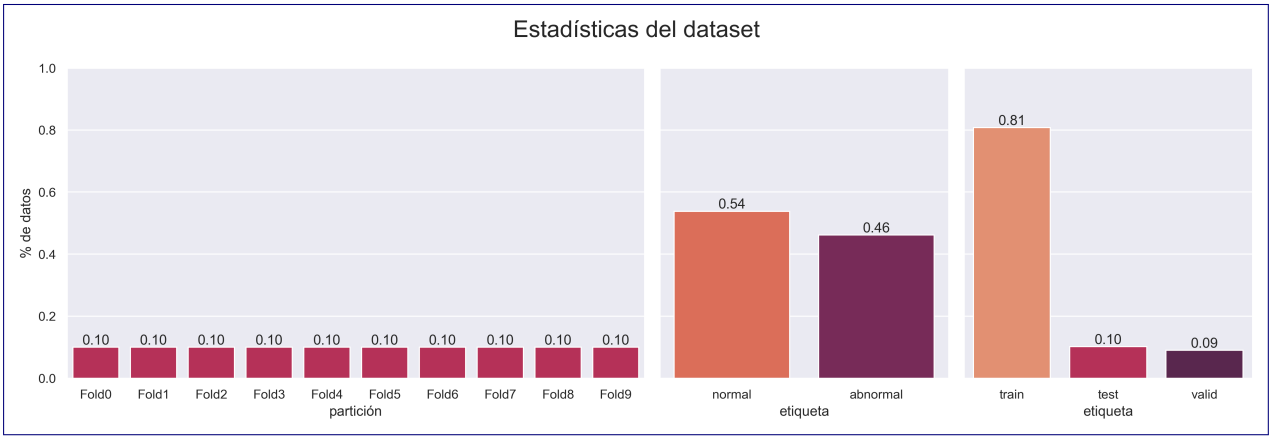
Por tanto, la finalidad de este trabajo es crear un **modelo de clasificador neuronal** basado en redes neuronales convolucionales para detectar glaucoma a partir de una retinografía o imagen de fondo de ojo. Para ello se utilizarán las técnicas de ***transfer learning*** **y** ***fine-tuning*** sobre varias redes neuronales basadas en modelos ya entrenados con ImageNet.

Este trabajo está basado mayoritariamente en *"CNNs for automatic glaucoma assessment using fundus images: an extensive validation"*[2] por Diaz-Pinto y col., en el que se concluye que una red neuronal convolucional entrenada sobre ImageNet y ajustada (*fine-tuned*) con retinografías puede ser una alternativa robusta para un sistema de detección automática de glaucoma frente a los extractores automáticos de características y clasificadores con características creadas a medida.

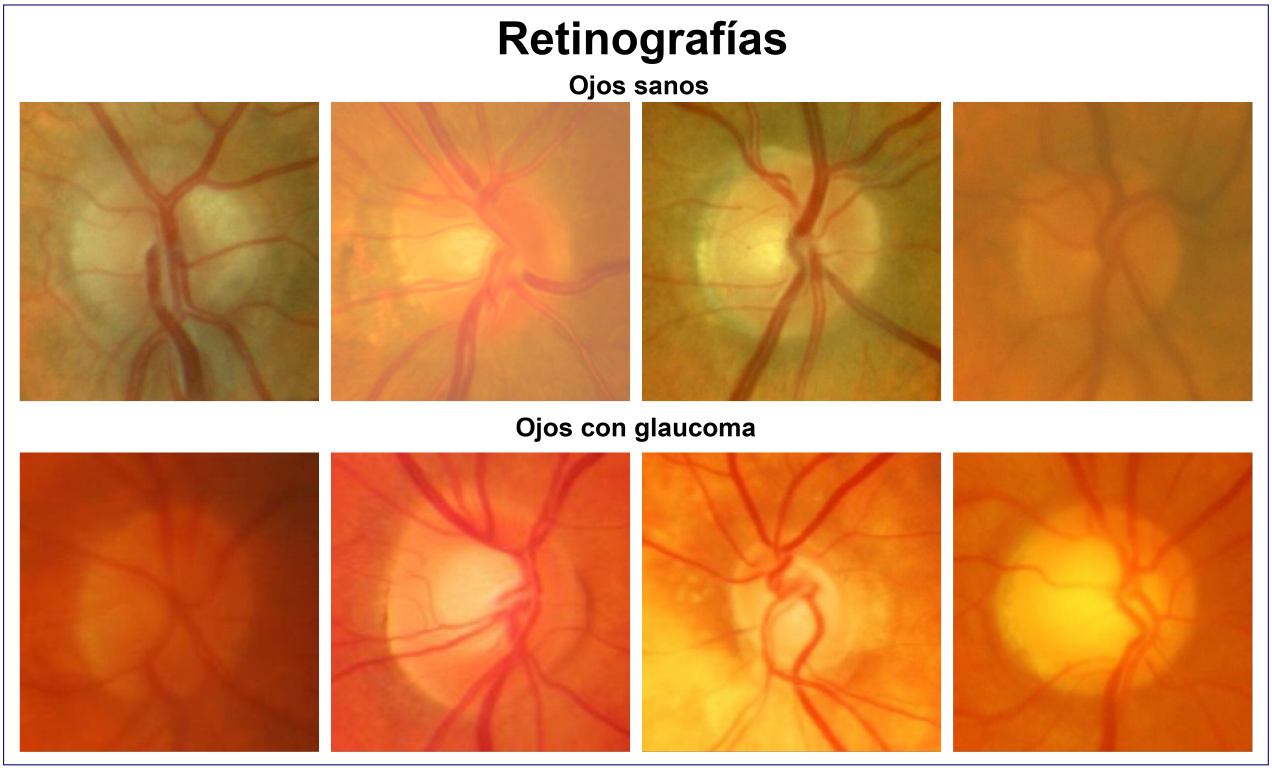
Patricia Lázaro Tello
plazarotello@uoc.edu

Análisis exploratorio de los datos

El conjunto de datos proporcionado consiste en 1.707 imágenes de 224x224 píxeles de retinografías de ojos sanos y con glaucoma. Estas imágenes se han replicado para obtener 10 particiones distintas, que se subdividen en conjuntos de entrenamiento, validación y test.



Se utiliza la etiqueta *normal* para los fondos de ojos sanos, mientras que los fondos de ojos con glaucoma se encuentran bajo la etiqueta *abnormal*. Se trata de un ***dataset* equilibrado**, con aproximadamente el mismo número de imágenes de ojos sanos que de ojos con glaucoma. Para cada partición se utiliza el 80 % de los datos para entrenamiento, dejando un 10 % de los datos para validación y otro 10 % para test.

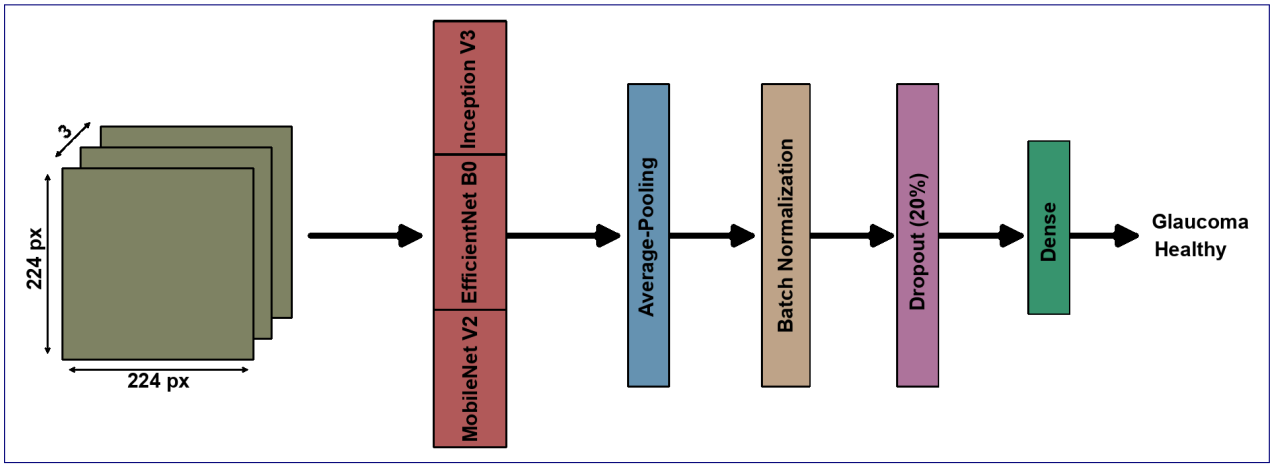


Patricia Lázaro Tello
plazarotello@uoc.edu

Arquitecturas de CNN utilizadas

Las arquitecturas de redes neuronales convolucionales utilizadas en este trabajo han sido ***EfficientNet B0***[6], ***Inception V3***[5] y ***MobileNet V2***[4]. Todas ellas han sido pre-entrenadas con el conjunto de datos ImageNet (la versión pre-entrenada se encuentra disponible en Keras).

Para adaptar las arquitecturas a la tarea de clasificación de retinografías en casos sanos y con glaucoma, se ha sustituido la última capa totalmente conectada de cada arquitectura por una capa de agrupamiento (*average-pooling*), una capa de normalización (*batch normalization*), una capa de *dropout* del 20 % y una capa totalmente conectada con 1 nodo por clase (2 nodos en total) con una función de activación *softmax*.



La arquitectura base del trabajo ha sido *EfficientNet B0*; las arquitecturas *Inception V3* y *MobileNet V2* se han elegido de los modelos disponibles en Keras[7] comparando el número de capas, tiempos de entrenamiento y número de parámetros con la arquitectura base. En este caso, se ha elegido *Inception V3* como modelo ligeramente más complejo y *MobileNet V2* como modelo más ligero y simple.

Preprocesamiento de los datos

Cada arquitectura de las mencionadas espera como entrada las imágenes como píxeles con un rango de valores determinado. Para adaptar la base de datos a las arquitecturas de redes neuronales convolucionales se han utilizado las funciones de preprocesamiento incluidas en Keras[7], que añaden capas extra a los modelos.

Concretamente, *EfficientNet B0* espera los píxeles en un rango de valores [0, 255] (utilizado por defecto) ya que el preprocesamiento en este caso viene incluido dentro del modelo; *Inception V3* y *MobileNet V2* esperan los píxeles en el rango de valores [-1, 1], por lo que se incluyen 2 capas de preprocesamiento en cada caso.

Arquitectura	Profundidad	Parámetros	Tiempo/paso [CPU | GPU]
EfficientNet B0	0 + 132 + 4	5.3M	46.0ms | 4.9ms
Inception V3	2 + 189 + 4	23.9M	42.2ms | 6.9ms
MobileNet V2	2 + 105 + 4	3.5M	25.9ms | 3.8ms

Experimentos realizados

> Los experimentos realizados en este trabajo se han ejecutado sobre una estación con CPU AMD Ryzen 3.600 y GPU Nvidia RTX 3.060 Ti *Founders Edition* (*overclocked* a -500MHz *Core Clock* y +1.000Mhz *Memory Clock*).

Para comparar el rendimiento de las 3 arquitecturas de redes neuronales convolucionales (*EfficientNet B0, Inception V3* y *MobileNet V2*) se han llevado a cabo múltiples experimentos sobre una única partición de la base de datos de retinografías con la intención de optimizar los tiempos de entrenamiento de los modelos.

En primer lugar, se ha buscado la **mejor combinación de hiperparámetros** (optimizador, *learning rate* y *batch size*) para cada arquitectura, entrenando únicamente las 4 últimas capas, es decir, las capas añadidas a la arquitectura base. El número de *epochs* de entrenamiento se ha fijado en 100, utilizando *early stopping*, que espera 20 *epochs* antes de determinar que el modelo no ha mejorado en la métrica de F_1Score.

Dadas las restricciones de tiempos de entrenamiento observadas en el experimento, la búsqueda de hiperparámetros se ha acotado, descartando los *learning rates* más grandes y pequeños, así como algunos optimizadores como Adagrad o Adadelta. Se han considerado las combinaciones de los siguientes elementos:

- Optimizador Adam
- Optimizador SGD
- *Batch size* 8
- *Batch size* 16
- *Batch size* 32
- *Learning rate* 0.0001
- *Learning rate* 0.001
- *Learning rate* 0.01

Este tipo de entrenamiento es el utilizado en *transfer learning*, una técnica de aprendizaje automático que consiste en transferir un modelo que ha sido entrenado para una tarea determinada (reconocer el *dataset* de ImageNet) y utilizarlo para resolver otra parecida (reconocer casos de glaucoma en retinografías) sin reentrenar el modelo original.

Después se ha realizado un experimento de **comparación entre el modelo con *transfer learning* y el mismo modelo entrenado en su totalidad**; es decir, realizando un *fine-tuning* total. Para entrenar el segundo modelo se ha utilizado la combinación óptima de hiperparámetros obtenida en el experimento anterior, disminuyendo el *learning rate* en 2 órdenes de magnitud.

Adicionalmente, para el modelo basado en *EfficientNet B0* se ha realizado un paso intermedio entre el modelo con *transfer learning* y el modelo con *fine-tuning* total, en el que se han entrenado las 4 capas añadidas y las 20 últimas capas de la arquitectura base, exceptuando las capas de normalización.

Asimismo, se han comparado todos los modelos entrenados con una única partición, tanto dentro de la misma arquitectura base como entre las diferentes arquitecturas. Para el entrenamiento y evaluación del rendimiento de todos los modelos se ha utilizado la métrica F_1Score, que da el mismo peso a la precisión y al *recall*.

La media del F_1Score se ha realizado de forma pesada (*weighted average*), por lo que el F_1Score de los casos de ojos sanos tiene un peso ligeramente superior al de los ojos con glaucoma.

Por último, se ha escogido el mejor modelo según el criterio de maximizar el F_1Score y **se ha evaluado su rendimiento utilizando la técnica de *cross validation*** o validación cruzada con $k = 10$.

Resultados de los experimentos

Como se menciona en la sección anterior, se han realizado un total de 3 experimentos más la búsqueda de hiperparámetros para cada arquitectura base. A continuación se desglosan los resultados obtenidos.

Búsqueda de hiperparámetros

Se han buscado los hiperparámetros sobre los modelos con *transfer learning* en los que solo se entrenan las nuevas capas añadidas. Para cada arquitectura base, se ha entrenado el modelo con diferentes combinaciones de optimizador, *batch size* y *learning rate*, fijando las épocas en 100 con *early stopping*.

Se ha decidido fijar el número de épocas en 100 ya que las 4 capas suponen un número de parámetros entrenables inferior a 10.000 y por tanto entrenar el modelo durante 100 épocas supondría un *overfitting* considerable. Gracias al *early stopping* añadido, el entrenamiento para antes de que se produzca dicho fenómeno.

El modelo basado en *EfficientNet B0* obtiene mejor rendimiento con lotes de 8 imágenes, aunque también supone el tiempo de entrenamiento más alto para cada combinación de optimizador y *learning rate*. Respecto al optimizador, Adam funciona mejor que SGD en todas las combinaciones probadas. Por último, se obtienen los mejores resultados con el *learning rate* más alto (0.01), lo que tiene sentido por el número de parámetros entrenables tan reducido.

El modelo basado en *Inception V3* también obtiene mejores resultados con el optimizador

Patricia Lázaro Tello
plazarotello@uoc.edu

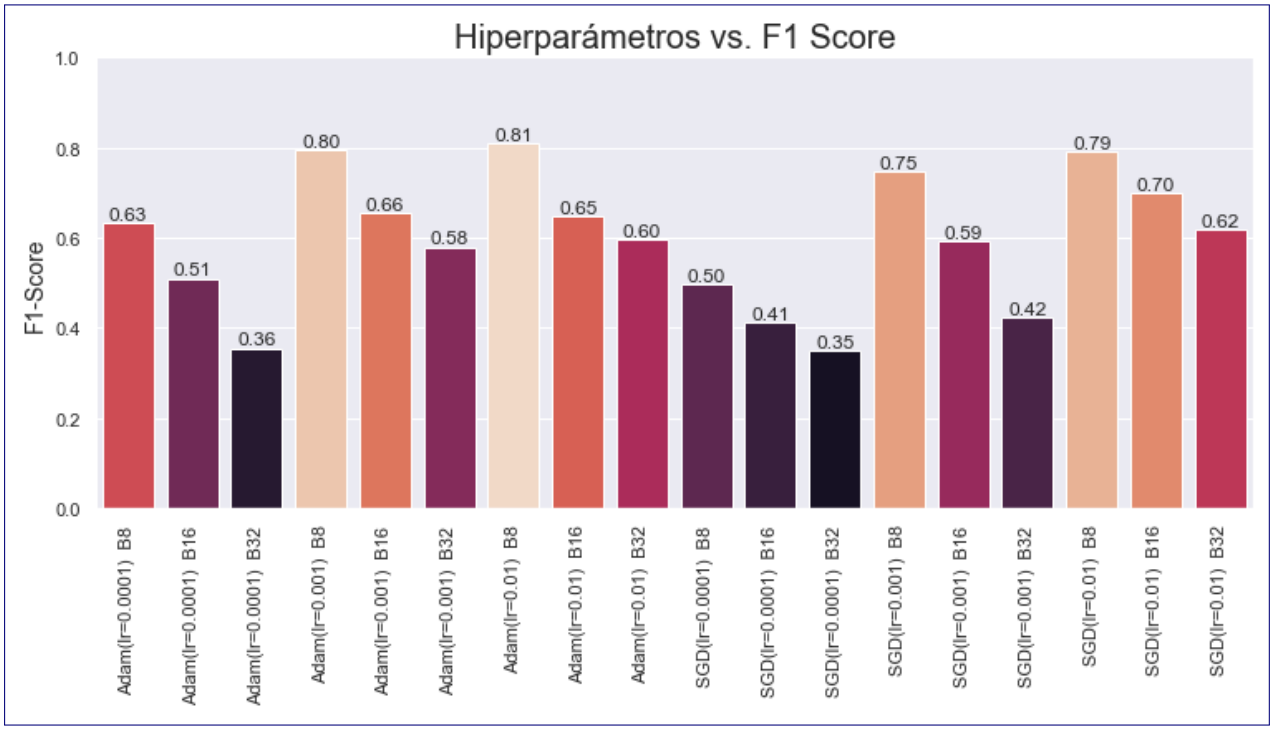


Figura 1: F_1Score frente a combinaciones de hiperparámetros para *EfficientNet B0*

Adam, salvo en el caso en que el *learning rate* es 0.01. En general, un *batch size* de 8 produce un mejor rendimiento, excepto en Adam con *learning rate = 0.001*, donde los lotes de 16 imágenes funcionan ligeramente mejor. Respecto al modelo de *EfficientNet B0*, esta arquitectura tiene casi el doble de parámetros entrenables, por lo que es esperable que un *learning rate* inferior proporcione un mejor rendimiento.

El modelo basado en *MobileNet V2* tiene el mismo número de parámetros entrenables que el modelo de *EfficientNet B0*; sin embargo, comparte combinación de hiperparámetros óptima con el modelo de *Inception V3*. Adam funciona mejor que SGD en todos los casos y, aunque en general los lotes de 8 imágenes dan mejores resultados, para el *learning rate* ganador (*lr = 0.001*) son los lotes de 16 imágenes los que proporcionan un rendimiento ligeramente superior.

Comparando los hiperparámetros elegidos para los 3 modelos con *transfer learning*, se observa que Adam es el optimizador que mejor funciona en todos los casos, y que *Inception V3* y *MobileNet v2* comparten mejor *batch size* y mejor *learning rate*, mientras que para *EfficientNet B0* es más adecuado utilizar un tamaño de lote inferior y un *learning rate* superior.

- ***EfficientNet B0*** : *Adam(lr = 0.01) batch_size = 8*
- ***Inception V3*** : *Adam(lr = 0.001) batch_size = 16*
- ***MobileNet V2*** : *Adam(lr = 0.001) batch_size = 16*

Patricia Lázaro Tello
plazarotello@uoc.edu

Comparación de modelos según las capas entrenadas

La comparación de modelos se ha realizado según sus valores de F_1Score total (*weighted average*); sin embargo, siendo el objetivo del trabajo minimizar el número de casos de glaucoma no detectados sería también necesario analizar el número de **falsos negativos** (casos de glaucoma clasificados como ojos sanos).

Modelo basado en *EfficientNet B0*

Como se menciona en secciones anteriores, se han entrenado 3 configuraciones del mismo modelo basado en *EfficientNet B0*, que corresponden con el modelo 1 (*transfer learning*), modelo 2 (*fine-tuning* parcial) y modelo 3 (*fine-tuning* total).

El modelo 1 consigue una **sensibilidad alta**, siendo capaz de detectar los casos de glaucoma aunque produciendo algunas falsas alarmas (ojos sanos clasificados como casos de glaucoma). La precisión y *recall* para ambas clases se encuentra balanceado, produciendo un F_1Score total de 0.81.

El modelo 2 resulta en métricas inferior a las del primer modelo, con una **especificadad muy alta** (casi todos los casos clasificados como glaucoma son efectivamente glaucoma) pero un *recall* para la clase glaucoma muy bajo. La precisión y el *recall* no están balanceados, produciendo un F_1Score total de 0.75.

El modelo 3 obtiene el mejor rendimiento, combinando los puntos fuertes de los modelos anteriores: posee una **especificidad y sensibilidad altas** para los casos de glaucoma, con una precisión y *recall* balanceados que producen un F_1Score total de 0.85.

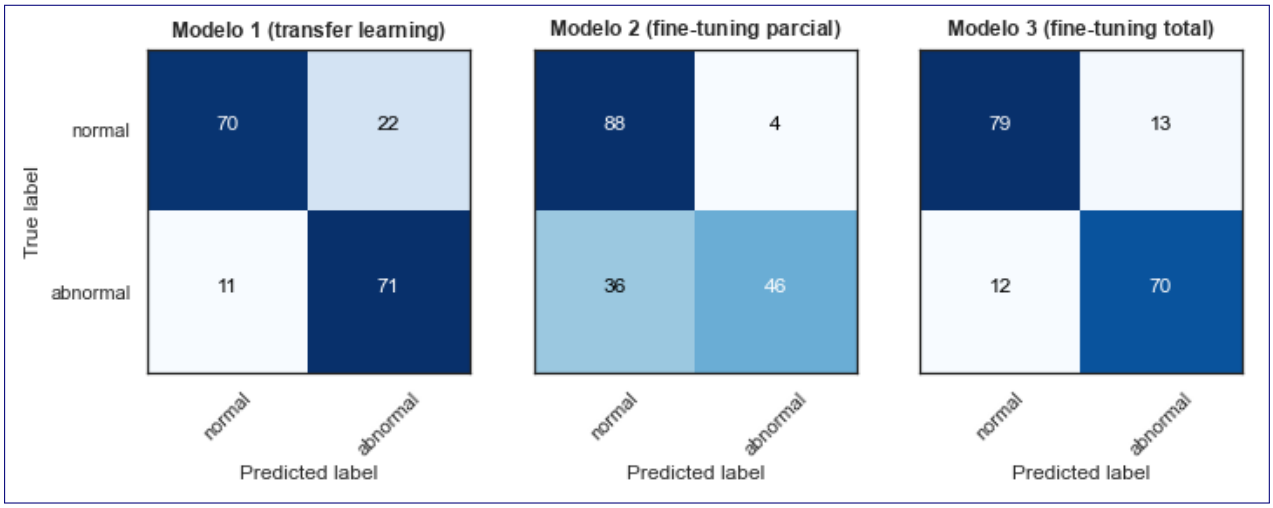


Figura 2: Matrices de confusión de las configuraciones de *EfficientNet B0*

Respecto al número de falsos negativos, el primer modelo sería el más recomendable de los 3, aunque analizando el resto de la matriz de confusión (figura 2) el modelo 3 es más aconsejable dado que reduce considerablemente el número de falsos positivos

manteniendo un número de falsos negativos muy razonable.

Modelo basado en *Inception V3*

Para el modelo basado en *Inception V3* se han entrenado 2 configuraciones, correspondientes al modelo 4-1 (*transfer learning*) y el modelo 4-2 (*fine-tuning* total). El primer modelo consigue una **sensibilidad alta** a costa de una **especificidad baja** (F_1Score = 0.76) frente al modelo 4-2 que obtiene una **especificidad muy alta** con una **sensibilidad baja** (F_1Score = 0.89). En el segundo modelo, la precisión y *recall* se encuentran más balanceados.

Respecto al número de falsos negativos, el modelo 4-1 es claramente superior al modelo 4-2 y es, de hecho, el modelo con mejor tasa de falsos negativos; sin embargo, también es el modelo con mayor tasa de falsos positivos, razón por la que obtiene un F_1Score más bajo que otros modelos. Por su parte, el modelo 4-2 es el que mejor especificidad posee para ojos sanos.

Modelo basado en *MobileNet V2*

Para el modelo basado en *MobileNet V2* se han entrenado 2 configuraciones, que corresponden con el modelo 5-1 (*transfer learning*) y el modelo 5-2 (*fine-tuning* total). El primer modelo consigue una **sensibilidad y especificidad moderadas** (F_1Score = 0.77) produciendo más falsas alarmas que otros modelos; el modelo 5-2 supone una mejora respecto al modelo anterior en la clasificación de ojos sanos, pero no en los de glaucoma, donde el *recall* es ligeramente inferior (F_1Score = 0.83).

Respecto al número de falsos negativos, el primer modelo sería el más recomendable de los 2 a la hora de clasificar los casos de glaucoma. El modelo 5-2 sería aconsejable si la tarea de clasificación fuera la contraria (clasificar ojos sanos) ya que obtiene una especificidad muy alta.

Selección del mejor modelo

Para realizar la validación cruzada (*cross validation*), se ha utilizado el mejor modelo de acuerdo a su F_1Score. Esto resulta en la elección del modelo basado en *Inception V3* entrenado en su totalidad (*fine-tuning*), ya que produce el F_1Score más alto (F_1Score = 0.87) frente a su competidor directo, el modelo basado en *EfficientNet B0* con *fine-tuning* total (F_1Score = 0.86). El tiempo de entrenamiento es una variable discriminadora, dado que el modelo 4-2 tarda aproximadamente 2 minutos menos que el modelo 3.

Si se utilizara únicamente el criterio de falsos negativos para la selección, el mejor modelo sería el denominado modelo 4-1, es decir, el modelo basado en *Inception V3* con *transfer learning*, ya que es el **modelo más sensible** de los entrenados.

Rendimiento del modelo *Inception V3 fine-tuned* con *cross-validation*

El rendimiento del modelo *Inception V3 fine-tuned* se ha medido tanto con la métrica F_1Score como con el tiempo de entrenamiento por partición.

Respecto a los tiempos de entrenamiento, se observa que de media el entrenamiento del modelo tarda algo más de 5 minutos, aunque existen casos especiales como la partición *k = 0*, que tarda menos de 4 minutos, o la partición *k = 3*, que tarda 11 minutos.

En relación con el F_1Score, se obtiene de media μ = *0.871*, con una desviación estándar de σ = *0.089*. En general, los F_1Score quedan comprendidos en torno a valores de [0.8, 0.9], con algunas excepciones como la partición *k = 1*, que alcanza el peor F_1Score (F_1Score = 0.65) y la partición *k = 9*, con un F_1Score casi perfecto (F_1Score = 0.97).

Análisis crítico

A continuación se procede a realizar un análisis crítico de algunas de las decisiones tomadas en el transcurso del trabajo, como son la métrica de evaluación escogida y el diseño de las particiones para *cross validation*.

Métrica de evaluación: Falsos Negativos (FN) y F_1Score

El glaucoma es una patología muy grave que puede causar ceguera si no es tratada de forma temprana. Es por ello que parece más adecuado centrarse en **reducir el número de falsos negativos** (o personas con glaucoma que el modelo detecta como personas sanas) en lugar de tratar de equilibrar precisión y *recall* con la métrica F_1Score, que es más apropiado para situaciones de clases desbalanceadas (no es el caso de este trabajo).

Utilizar el *recall* de la clase *abnormal* como métrica puede suponer que el modelo termine clasificando todas las imágenes de fondo de ojo como fondos de ojo con glaucoma, ya que así obtendría la máxima puntuación en la métrica. Para paliar este efecto, se podría utilizar una métrica similar a F_1Score que diera más peso al *recall*, como puede ser la **sensibilidad** o el **F_2Score**:

$$F_2 = (1 + 2^2) \times \frac{precision \times recall}{(2^2 \times precision) + recall}$$

Diseño de particiones para *cross validation*

El *dataset* utilizado contiene un número de imágenes muy reducido para la cantidad de capas y parámetros que se han de entrenar (que asciende al orden de millones, frente a las 1.707 imágenes del conjunto de datos). Para llevar a cabo el entrenamiento del mejor modelo con validación cruzada, se ha utilizado la técnica de *k-fold cross validation* con un valor de *k = 10*.

Sin embargo, entre las tareas a realizar en este trabajo se encuentra la búsqueda de hiperparámetros, por lo que una partición anidada (***nested k-fold***)[3] sería más adecuada.

Esta técnica utiliza *k-fold cross validation* y, para cada partición, se realiza otro *k-fold* con *k* = número de combinaciones de hiperparámetros para ajustar y evaluar las distintas combinaciones. En todas las particiones, el número de observaciones de ambas clases ha de estar balanceado, simulando las proporciones del *dataset* completo.

Por otro lado, tal y como se explica en *"CNNs for automatic glaucoma assessment using fundus images: an extensive validation"*[2], el glaucoma en retinografías es detectado principalmente por la afectación al disco óptico y sus alrededores, por lo que también sería recomendable aplicar un preproceso a los datos para centrar la identificación y clasificación del modelo en la afectación al disco óptico.

Por último, cabe destacar que, tratándose de un conjunto de datos reducido (1.707 imágenes disponibles), una de las maneras más sencillas de evitar el *overfitting* sería utilizar *data augmentation* para realizar una serie de transformaciones sobre los datos y conseguir mayor volumen de los mismos.

Todo este preprocesamiento (centrado en el disco óptico y *data augmentation* posterior) sería recomendable realizarlo *offline*, es decir, recortar y transformar las imágenes y guardarlas en disco para utilizarlas en el modelo principal sin llevar a cabo preprocesamiento. La razón detrás de esta preferencia se encuentra en la posibilidad de realizar retoques y ajustes manuales sin necesidad de volver a entrenar el modelo de preprocesado, etcétera.

Conclusiones

En este trabajo se han entrenado 7 (*transfer learning* y *fine-tuning* parcial y total) modelos basados en 3 arquitecturas de redes neuronales convolucionales (*EfficientNet B0, Inception V3* y *MobileNet V2*) pre-entrenadas con ImageNet para clasificar casos de glaucoma mediante retinografías.

Se observa que la sensibilidad de los modelos sin *fine-tuning* es superior a la de los modelos con *fine-tuning*, que debido a la métrica utilizada (F_1Score) renuncian a parte de su sensibilidad para obtener un aumento en la especificidad.

Para una única partición, el mejor modelo respecto a la sensibilidad es el basado en *Inception V3* sin *fine-tuning* (modelo 4-1), con un F_1Score de 0.76, una sensibilidad del 89.02 % y una especificidad del 65.22 %.

El mejor modelo de acuerdo a la métrica utilizada (F_1Score) es el modelo 4-2, basado en *Inception V3* con *fine-tuning*, y obtiene un F_1Score de 0.88, una sensibilidad del 79.27 % y una especificidad del 97.82 %.

La utilización de una métrica distinta como la sensibilidad o el F_2Score, y otras técnicas como *data augmentation*, puede suponer una mejora en la clasificación de glaucoma.

Referencias

[1] Anna Bosh Rué, Jordi Casas Roma y Toni Lozano Bagén. *Deep Learning: Principios y fundamentos*. Editorial UOC, 12 de jul. de 2019.

[2] Andres Diaz-Pinto y col. “CNNs for automatic glaucoma assessment using fundus images: an extensive validation”. En: *BioMedical Engineering OnLine* 18.1 (mar. de 2019). DOI: 10.1186/s12938-019-0649-y. URL: https://doi.org/10.1186/s12938-019-0649-y.

[3] Vladimir Lyashenko y Abhishek Jha. *Cross-validation in Machine Learning: How To Do It Right*. Mar. de 2022. URL: https://neptune.ai/blog/cross-validation-in-machine-learning-how-to-do-it-right.

[4] Mark Sandler y col. “MobileNetV2: Inverted Residuals and Linear Bottlenecks”. En: (2018). DOI: 10.48550/ARXIV.1801.04381. URL: https://arxiv.org/abs/1801.04381.

[5] Christian Szegedy y col. “Rethinking the Inception Architecture for Computer Vision”. En: (2015). DOI: 10.48550/ARXIV.1512.00567. URL: https://arxiv.org/abs/1512.00567.

[6] Mingxing Tan y Quoc V. Le. “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks”. En: (2019). DOI: 10.48550/ARXIV.1905.11946. URL: https://arxiv.org/abs/1905.11946.

[7] Keras Team. *Keras documentation: Keras Applications*. URL: https://keras.io/api/applications/#available-models.