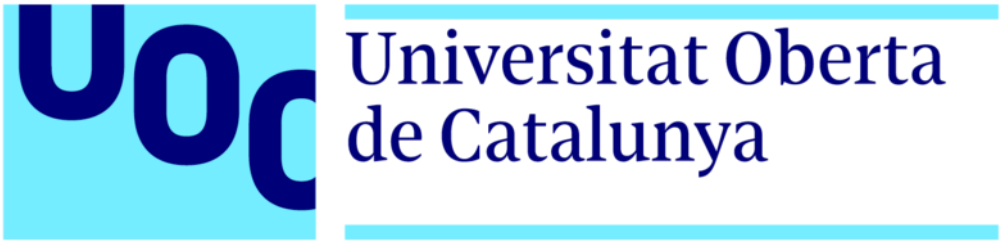


Máster universitario de Ciencia de Datos

Práctica Final – PRA

Aprendizaje profundo (*Deep Learning*) – Implementación de un clasificador neuronal para la detección de glaucoma.

Autor:

Mario Ubierna San Mamés

Índice de Contenido

Índice de ilustraciones

Índice de tablas

1. Introducción

El glaucoma es una patología que afecta al nervio óptico, y solventarlo es de vital importancia para tener una buena vista. Por norma general, esta enfermedad se produce debido a una presión en el ojo más alta de lo normal [1].

Esta patología es una de las principales causas de ceguera a nivel mundial, siendo la segunda causa por detrás de la diabetes. Suele aparecer a edades tardías, por lo que personas mayores de 70 años pueden empezar a desarrollar esta enfermedad.

Uno de los principales inconvenientes relativos al glaucoma es la dificultad en el diagnóstico del mismo, en otras palabras, esta enfermedad no presenta signos de advertencia. La pérdida de la visión se realiza de forma gradual y no hay una clara apreciación de la pérdida hasta que el glaucoma está en una fase avanzada.

Es aquí donde entra en juego el objetivo de esta práctica, implementar un clasificador de imágenes que nos permita detectar si un ojo presenta glaucoma o no, con el fin de mejorar la calidad de vida de la sociedad.

Tal y como ya se ha mencionado, el glaucoma está relacionado a una presión elevada en el ojo, la cual produce daños en el nervio óptico que son irreparables. Esta presión se debe principalmente a una acumulación de líquido dentro del ojo, este líquido si se produce en las cantidades adecuadas se suele drenar a partir de los tejidos del ojo. El problema sucede cuando dicha cantidad es excesiva, haciendo que el drenaje sea más laborioso o incluso nulo.

Otro aspecto a tener en cuenta en el diagnóstico es que esta enfermedad puede ser hereditaria, es decir, algunas personas con genes muy similares tienden a reproducir esta patología.

Finalmente cabe destacar los diferentes tipos de glaucoma que existen, en esta práctica no se va a entrar en detalle pero es importante tenerlo en cuenta para ver que el problema no es una tarea trivial. La clasificación glaucomas a día de hoy es la siguiente [1] [2]:

- **Glaucoma de ángulo abierto**: suele ser el tipo de glaucoma más común encontrado en la sociedad. La causa que origina este tipo de glaucoma se desconoce, pero a medida que pasa el tiempo aumenta la presión en el ojo dando lugar a una pérdida en la visión.

- **Glaucoma de ángulo cerrado**: ocurre cuando el líquido dentro del ojo está bloqueado y no puede salir. En este caso, la presión originada en el ojo es elevada e intensa, dando lugar a una situación de emergencia.

- **Glaucoma secundario**: en este caso sí que se conoce la causa, y los dos glaucomas mencionados anteriormente pueden ser también de este tipo. Las causas que originan este glaucoma son: medicamentos como los corticosteroides, enfermedades oculares como la uveítis, enfermedades como la diabetes y lesiones oculares.

- **Glaucoma congénito**: solamente ocurre en los recién nacidos, se produce cuando el ojo no se ha desarrollado de forma correcta en la primera etapa de vida del ser humano.

2. Redes neuronales convolucionales

Para la resolución de esta práctica se ha hecho uso de redes neuronales convolucionales. Este tipo de redes no es más que un tipo de red neuronal artificial dentro de lo conocido como *deep learning*. La principal característica de estas redes es que no necesitan que se les introduzca datos en sí para la clasificación, sino que son ellas mismas capaces de extraer características a partir de una imagen.

Al extraer características de forma eficiente permite que estas redes sean muy usadas para el reconocimiento de patrones en imágenes, es decir, reconocer caras, coches, animales…

El *input* de una red neuronal convolucional es la imagen, en nuestro caso la imagen del ojo, y el *output* va a ser la clasificación de la misma, es decir, si presenta glaucoma o no.

Uno de los mayores problemas que nos encontramos al hacer uso de redes neuronales convolucionales es el elevado consumo que se necesita para entrenar la misma. Con la idea de solventar esta problemática aparece el concepto de redes pre-entrenadas, la técnica que hace uso de redes pre-entrenadas se denomina *transfer learning*.

Estas redes se caracterizan porque una vez se ha realizado el entrenamiento para un problema se guarda dicha red, en otras palabras, se almacenan los pesos "óptimos" y la arquitectura en sí.

Con todo esto conseguimos solventar el problema del coste computacional, además de reducir el tiempo necesario para entrenar la red.

Existen numerosas redes pre-entrenadas para mejorar la precisión de los modelos de una forma eficiente y rápida. Para la solución de este proyecto se ha hecho uso de *EfficientNet* y más concretamente del modelo *EfficientNetB0*, perteneciente a *Google*.

Estas redes *EfficientNet* son similares en cuanto a la estructura de la red, es decir, presentan los mismos componentes generales. Sin embargo, varían en el número de capas dentro de cada componente general, a medida que aumenta el modelo usado de *EfficientNet* aumenta el número de parámetros a entrenar de la red.

Los componentes que contienen todos los modelos de *EfficientNet* son los siguientes:

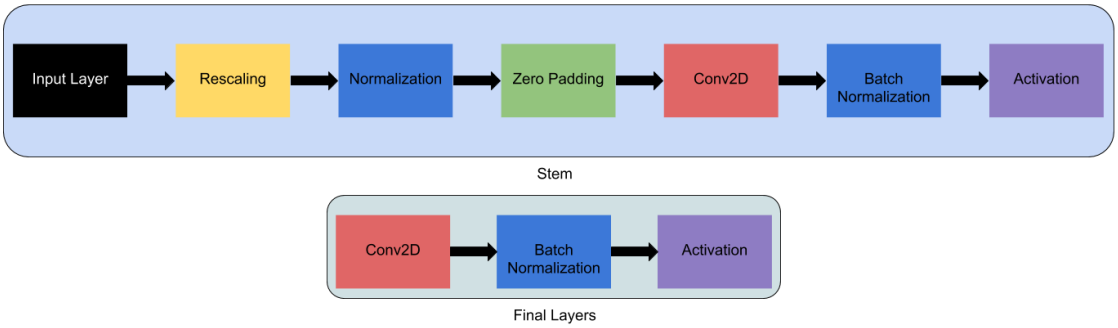


Ilustración 1 - Componentes generales de un modelo EfficientNet [3].

Cabe destacar que para realizar el *transfer learning* a veces solamente se ha entrenado las capas finales, mientras que cuando se ha aplicado la técnica de *fine tuning* se han ido descongelando más capas.

Otro punto a tener en cuenta del modelo *EfficentNetB0* es que el tamaño de imágenes es de 224x224 pixeles, cuanto mayor sea el modelo mayor es el tamaño de imágenes y de la red y más parámetros se necesitan para entrenar.

Red neuronal	**Parámetros entrenables**	**Parámetros no entrenables**	**Parámetros totales**
EfficientNetB0	3.969.373	86.599	4.055.972

Tabla 1 - Número de parámetros en EfficientNetB0.

3. Análisis exploratorio

En este apartado se va a realizar un análisis exploratorio de los datos que se nos han proporcionado. El objetivo es poder obtener información relevante sobre diferentes aspectos.

Lo primero de todo es comprobar el número de imágenes que hay para cada conjunto, es decir, para los conjuntos de entrenamiento, validación y test. En la siguiente ilustración podemos ver de forma visual la proporción y el número de imágenes que hay:

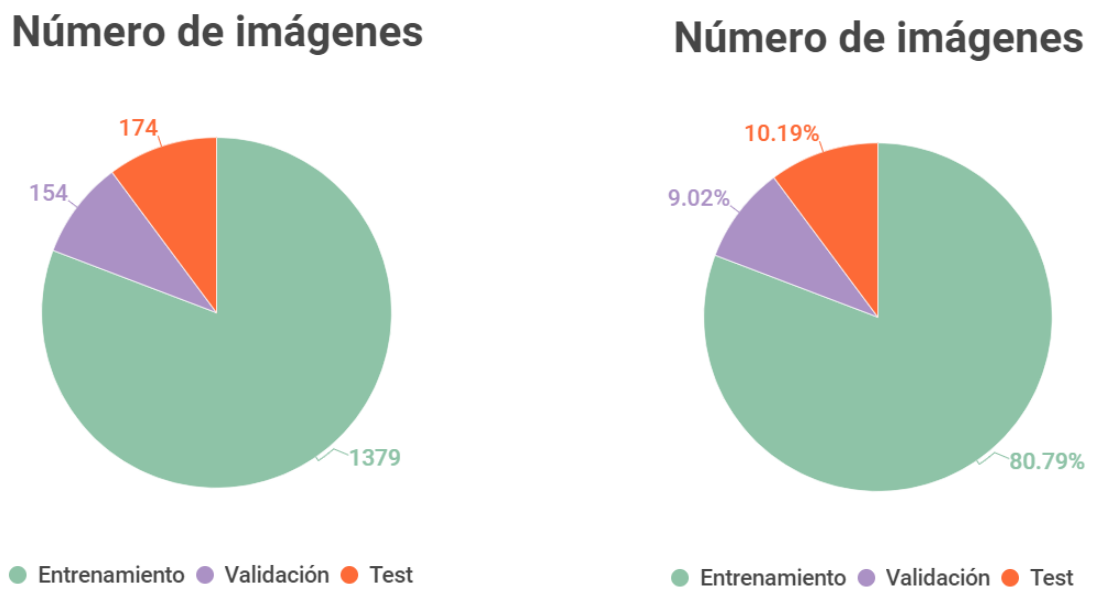


Ilustración 2 - Número de imágenes en el dataset.

Cabe destacar que de todas las imágenes el 80% representa al conjunto de entrenamiento, haciendo que los conjuntos de validación y test tengan un 10% aproximadamente. Otro punto clave del *dataset* es que todos los *folds* presentan el mismo número de imágenes, es decir, para todos los *folds* hay el mismo número de imágenes en cada conjunto.

El siguiente paso a analizar fue la detección de duplicados, tanto si había duplicados dentro de cada conjunto, es decir, si para el conjunto de entrenamiento/validación/test de cada *fold* había algún duplicado, como si los había independientemente del conjunto, es decir, dentro de cada *fold*.

El resultado del análisis de duplicados fue que no había ninguna imagen repetida en ninguno de los dos casos mencionados en el párrafo anterior.

Una vez que se sabía que no había duplicados, se analizó el número de casos con ojos con y sin glaucoma que había dentro de cada *fold*. En la siguiente ilustración podemos apreciar el número de casos y la proporción:

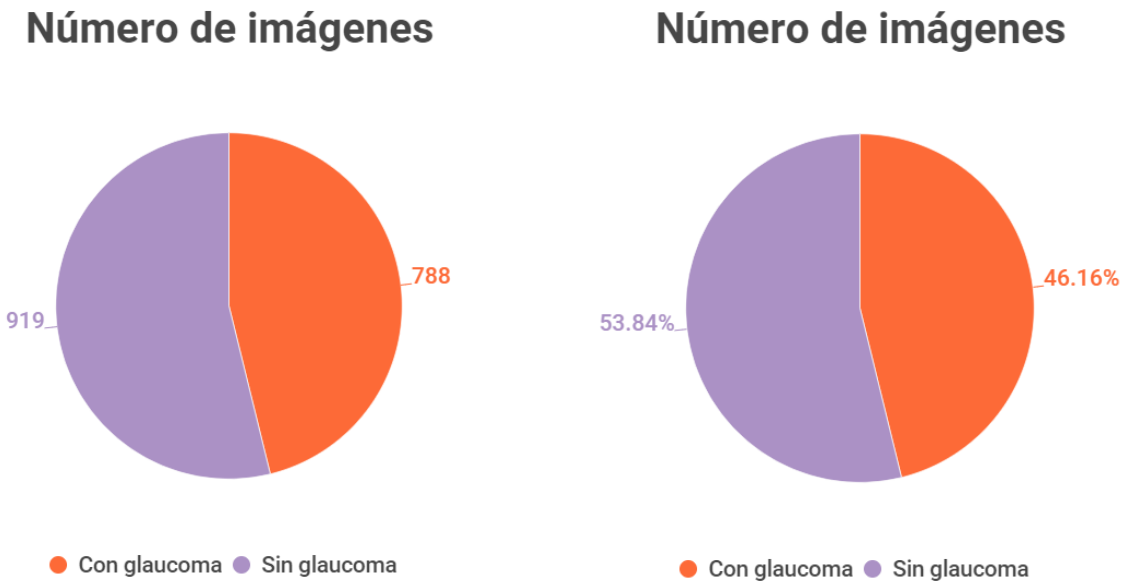


Ilustración 3 - Número de casos del dataset.

Como podemos apreciar en las anteriores gráficas los datos no están balanceados, en otras palabras, hay más casos de imágenes sin glaucoma que con glaucoma. Un aspecto importante en este punto es que todos los *folds* presentan los mismos casos con y sin glaucoma.

Finalmente, para finalizar el análisis de los datos se ha comprobado si en cada conjunto de cada *fold* hay el mismo número de casos de esta patología:

	Entrenamiento		Validación		Test	
	Con	Sin	Con	Sin	Con	Sin
Fold0	625	754	71	83	92	82
Fold1	639	740	66	88	83	91
Fold2	640	739	71	83	77	97
Fold3	636	743	69	85	83	91
Fold4	633	746	73	81	82	92
Fold5	621	758	83	71	84	90
Fold6	625	754	70	84	93	81
Fold7	642	737	72	82	74	100
Fold8	631	748	74	80	83	91
Fold9	646	733	72	82	70	104

Tabla 2 - Número de casos por conjunto.

A la vista de la anterior tabla podemos observar que no hay el mismo número de casos con y sin glaucoma para cada *fold*, es decir, depende de con qué *fold* entrenemos a la red para que tenga más casos de un tipo o del otro, lo cual va a sesgar el entrenamiento de nuestro modelo.

4. Entrenamiento y selección del modelo

En esta sección se van a detallar los hiperparámetros de los modelos entrenados, y cómo ha sido el entrenamiento junto con la selección del mejor modelo. Cabe destacar que en este informe no queda justificado por qué se han elegido unos hiperparámetros u otros, ni tampoco la discusión de la evaluación de cada modelo, esto ha quedado detallado en el *notebook* que se ha entregado junto a este informe.

4.1. Hiperparámetros

En la siguiente tabla se puede apreciar los diferentes hiperparámetros usados para cada modelo. Cabe destacar que el número de épocas depende del *EarlyStopping*, el cual se detallará en el siguiente punto. Los parámetros utilizados para cada modelo son:

Hiperparámetro	**Modelo 1**	**Modelo 2**	**Modelo 3**	**Modelo 4**	**Modelo 5**
Modelo base	Eff.NetB0	Eff.NetB0	Eff.NetB0	Eff.NetB0	Eff.NetB0
Aumento de datos	No	No	No	Sí	Sí
Optimizador	SGD	ADAM	SGD	ADAM	SGD
Learning rate	0.001	0.0001	0.1	0.0001	0.1
Batch	64	64	64	64	64
Épocas Max	100	100	100	100	100
Épocas	69	12	26	21	32

Capas descongeladas	Clasificador final	Últimas 20	Todas	Últimas 20	Todas

Tabla 3 - Hiperparámetros de cada modelo.

4.2. EarlyStopping

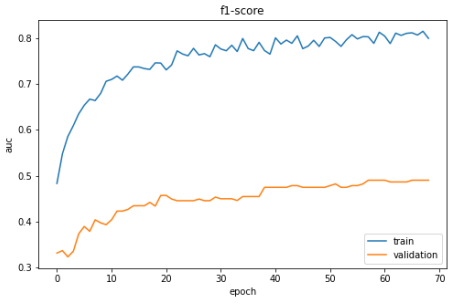
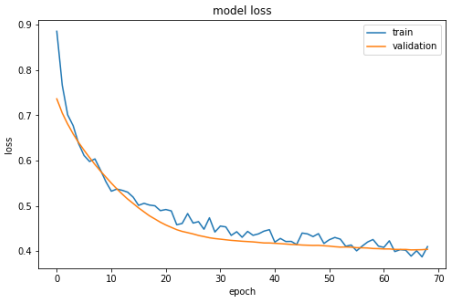
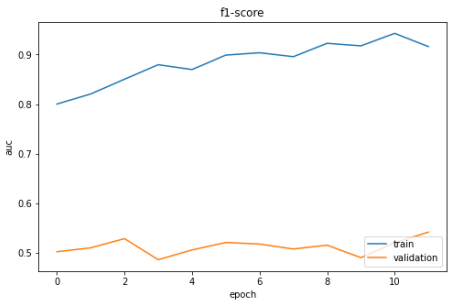
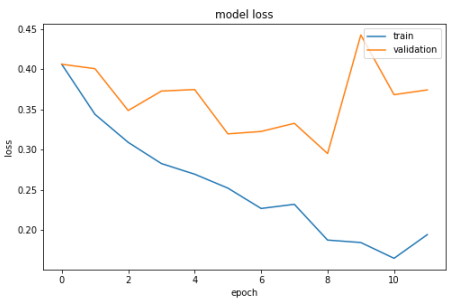
Los hiperparámetros utilizados de *EarlyStopping* para cada modelo mencionando en el punto anterior han sido:

Hiperparámetros	**Modelo 1**	**Modelo 2**	**Modelo 3**	**Modelo 4**	**Modelo 5**
Monitor	val_loss	val_loss	val_get_f1	val_get_f1	val_get_f1
Mode	min	min	max	max	max
Patience	3	3	5	3	5

Tabla 4 - Hiperparámetros de EarlyStopping.

4.3. Entrenamiento (f1-score y loss)

La evolución del entrenamiento de cada modelo es la siguiente:

	F1-score	**Loss**
Modelo 1		
Modelo 2		

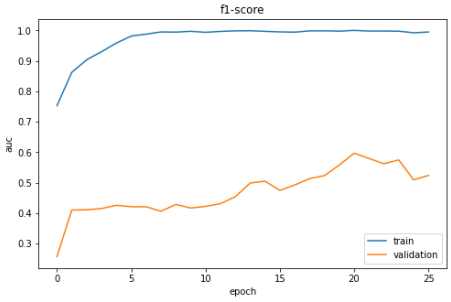
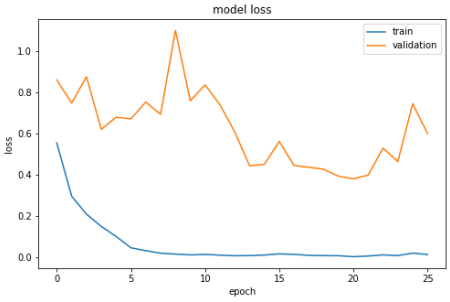
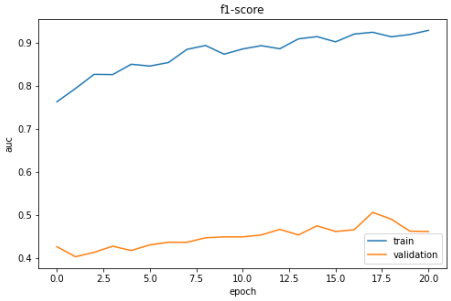
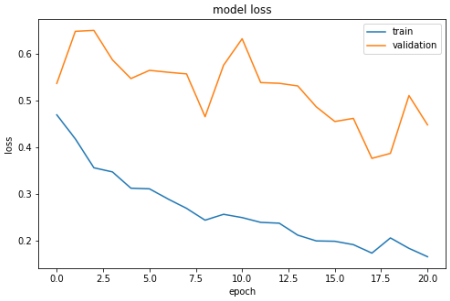
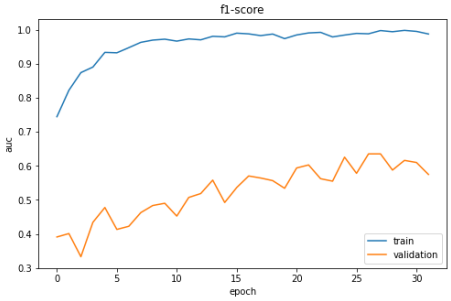
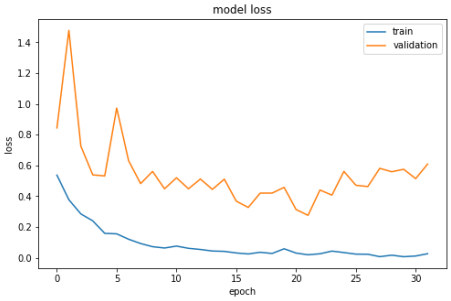
Modelo 3		
Modelo 4		
Modelo 5		

Tabla 5 - Evolución F1-score y loss.

4.4. Evaluación

En la siguiente tabla podemos apreciar diferentes métricas analizadas para cada modelo, cabe destacar que en el siguiente punto se hará y se explicará cuál es el mejor modelo obtenido:

	Exactitud	**Precisión**	**Recall**	**F1-score**
Modelo 1	0.82	0.90	0.75	0.82
Modelo 2	0.82	0.94	0.71	0.81
Modelo 3	0.86	0.95	0.78	0.86
Modelo 4	0.85	0.95	0.75	0.84

Modelo 5	0.90	0.97	0.84	0.90

Tabla 6 - Evaluación de los modelos.

4.5. Elección del mejor modelo

A la vista de la anterior tabla podemos concluir que el modelo que mejor se comporta con los datos es el modelo 5.

El modelo 5 se caracterizaba porque hacía uso de aumento de datos y se descongelaban todas las capas. Una nota aclaratoria es que a medida que se descongelan más capas y/o se hace uso de aumento de datos se consigue mejorar los resultados obtenidos.

Este modelo es capaz de obtener una elevada exactitud 0.90, es decir, es el que mejor identifica tanto los verdaderos negativos como los verdaderos positivos.

Respecto a la precisión obtenida, este modelo también es el que mejor clasifica, se consigue una precisión del 0.97, con esto se llega a la conclusión de que reduce muy bien los falsos positivos.

Una de las métricas más importantes es el *recall* o sensibilidad, ya que ésta nos determina cómo de bueno es un modelo para reducir los falsos negativos (aspecto importante en soluciones para el ámbito sanitario). Vemos que este modelo se comporta bien ya que obtiene un *recall* de 0.84.

Finalmente, la última métrica para seleccionar qué modelo es mejor es el *f1-score*, ésta mezcla tanto la precisión como la sensibilidad del modelo, por lo que a mayor valor mejor clasifica la red neuronal entrenada. En este caso, se obtiene un valor de 0.90, dicho valor es muy alto y podemos considerar este modelo como un muy buen modelo.

5. Validación cruzada

Una vez elegido el mejor modelo en la sección anterior, se procede a realizar la validación cruzada de cada uno de los *folds*. Cabe mencionar que por motivos de reducción en la longitud de este informe, la pregunta que se plantea en esta sección ha sido resulta en el *notebook* que se ha proporcionado junto al presente documento.

En la siguiente tabla podemos apreciar los *f1-score* obtenidos para cada fold:

	Fold0	**Fold1**	**Fold2**	**Fold3**	**Fold4**
F1-score	0.91	0.92	0.88	0.84	0.95
	Fold5	**Fold6**	**Fold7**	**Fold8**	**Fold9**
F1-score	0.86	0.91	0.89	0.85	0.86

Tabla 7 - Validación cruzada del modelo 5.

El valor *f1-score* medio obtenido ha sido de 0.88 y la desviación estándar es de 0.03.

Como podemos apreciar el *f1-score* medio es elevado para el conjunto de test aplicado sobre el modelo 5, la desviación estándar es baja. Con todo ello los resultados obtenidos son muy buenos independientemente del conjunto de datos utilizado.

Es por ello que llegamos a la conclusión de que el modelo 5 es el mejor, básicamente por hacer uso de una red pre-entrenada *EfficientNetB0*, realizar un aumento de datos y descongelar totalmente la red. En definitiva, todo esto influye considerablemente en la obtención de un alto rendimiento del modelo.

6. Análisis crítico

Respondiendo a la pregunta del apartado a), la estrategia de diseño que hubiera seguido sería diferente.

Un problema fundamental que tienen estas particiones es que no siempre hay el mismo número de imágenes con casos positivos que negativos, dicho en otras palabras no se sigue la misma distribución de valores. Al suceder esto se pueden estar creando modelos sesgados a favor de la clase mayoritaria que hay en cada conjunto de entrenamiento para cada *fold*.

Por lo tanto, esto lo podemos resolver de dos formas:

- Haciendo que la partición aleatoria de los datos tenga en cuenta la distribución de los casos, es decir, que sea similar en cada *fold* el número de casos positivos y el número de casos negativos.
- La segunda forma es realizando un aumento de datos sobre la clase minoritaria, así se consigue que para todos los conjuntos de entrenamiento haya el mismo número de casos positivos y negativos.

Teniendo en cuenta que es importante la distribución de valores para una buena generalización, en mi caso hubiera hecho una primera partición aleatoria, y a partir de ésta crear k particiones con distribuciones en el número de casos similar. De esta forma, conseguimos reducir el sesgo y mejorar la capacidad de generalización del modelo.

Respondiendo a la pregunta del apartado b), se detalla a continuación las resultados obtenidos y las conclusiones del proyecto.

En primer lugar se ha definido el modelo 1, el cual se caracterizaba porque hacía uso de *EfficientNetB0*. En este caso, solo se entrenaba el clasificador final, haciendo que el resto de capas estuvieran congeladas.

Los resultados obtenidos con ese modelo son buenos, se conseguía una elevada precisión para los casos en los que había una anomalía, pero la sensibilidad de los casos positivos era baja. El *f1-score* obtenido para la clase negativa era de 0.83 y para la positiva de 0.82. No es un mal modelo, pero lo idóneo es que el *f1-score* sea igual o superior a 0.9.

Por otro lado, estamos en un problema del ámbito sanitario, por lo que el objetivo es reducir los falsos negativos. En este modelo había 8 falsos positivos frente a los 23 falsos negativos. Por lo que no termina de resolver de forma adecuada la problemática que se plantea.

El siguiente modelo era el modelo 2, éste se basaba en el modelo 1 pero se descongelaban las última 20 capas.

Los resultados obtenidos para este modelo han sido similares al anterior en cuanto a la detección de no anomalías. Se consigue reducir los falsos positivos pero no lo falsos negativos. En este caso, el *f1-score* obtenido para la clase sin anomalía es de 0.83 y de 0.81 en los casos que sí que hay anomalía.

En resumen, a nivel numérico este modelo se comporta mejor, pero estamos en el ámbito sanitario, por lo que hay que priorizar los falsos negativos en vez de los falsos positivos, debido a esto este modelo no ha sido capaz de superar al anterior.

El siguiente modelo creado fue el modelo 3, igual que los dos modelos anterior pero descongelando totalmente el modelo.

Tras el entrenamiento se han obtenido muy buenos resultados, mejores que los anteriores. En este caso, se consigue una mayor precisión y sensibilidad para ambas clases, haciendo que el *f1-score* sea superior, para la clase 0 se obtiene un 0.87 y para la clase 1 un 0.86.

Al aumentar la precisión se consigue reducir los falsos positivos, pero al aumentar también la sensibilidad se consigue reducir los falsos negativos. Las conclusiones que obtenemos de este apartado es que al descongelar un mayor número de capas del modelo base, el modelo final es capaz de aprender mejor los datos y mejorar las salidas que proporciona el mismo.

El siguiente modelo es el modelo 4, similar al modelo 2 pero realizando un aumento de datos.

Comparando este modelo con el modelo 2 se ha conseguido mejorar los resultados a partir del aumento de datos. En este caso, hay mejoras tanto en la precisión como en la sensibilidad consiguiendo así aumentar el *f1-score*, para la clase 0 de 0.85 y para la clase 1 de 0.84. Además, se consigue reducir los falsos negativos.

En resumen, este modelo comparado con el modelo 2 se comporta mejor, pero aun así es algo peor que el modelo 3.

El último modelo creado ha sido el 5, similar al modelo 3 pero con un aumento de datos.

Los resultados obtenidos para este modelo han sido los mejore encontrados. Se mejora la precisión, la sensibilidad y por lo tanto el *f1-score*. En este caso para ambas clases el *f1-score* es de 0.9, haciendo que este modelo sí que sea muy bueno. Al igual que sucedía antes se consigue reducir los falsos negativos.

Las conclusiones obtenidas respecto a este modelo y al anterior es que al hacer un aumento de datos, y descongelar parcialmente o totalmente la red, se consigue mejorar de forma considerable el modelo.

En cuanto a la sección 3 del enunciado “Validación cruzada y discusión”, se ha hecho uso del modelo 5 para realizar la validación cruzada.

En este caso se ha obtenido un *f1-score* medio de 0.88 para los 10 *folds* con una desviación estándar de 0.03.

Como podemos apreciar, el *f1-score* medio es muy elevado para el conjunto de test, y la desviación estándar es baja, haciendo que los resultados obtenidos sean muy buenos independientemente del conjunto de datos utilizado.

7. Bibliografía

[1] «Glaucoma - Síntomas y causas - Mayo Clinic». https://www.mayoclinic.org/es-es/diseases-conditions/glaucoma/symptoms-causes/syc-20372839 (accedido 6 de junio de 2022).

[2] «Glaucoma: MedlinePlus enciclopedia médica». https://medlineplus.gov/spanish/ency/article/001620.htm (accedido 6 de junio de 2022).

[3] «Social Network for Programmers and Developers». https://morioh.com (accedido 13 de mayo de 2022).