

## Práctica 2. IA salud - DL e imágenes médicas.

FERNANDO JAVIER BORAO ZACCHEO - 100506847

ADRIÁN CERROS SÁNCHEZ - 100405342

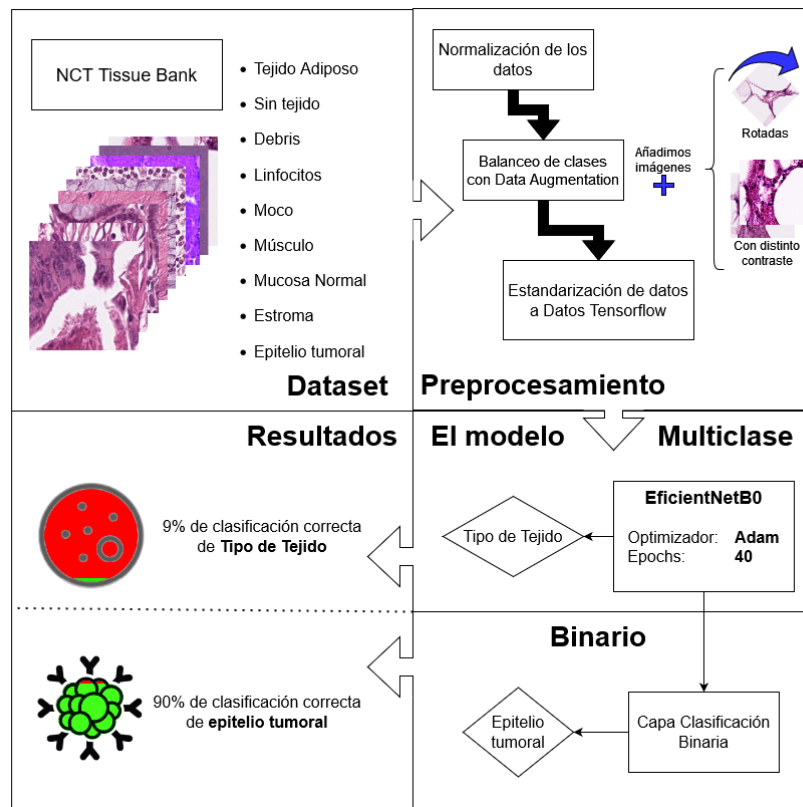
CLARA MIRANDA GARCÍA - 100518506

LUIS VICENTE ARÉVALO - 100505651

# Índice

Índice.....	2
Abstract Gráfico.....	3
Tabla de resultados.....	3
Introducción.....	4
Lenguaje de Programación, librerías y frameworks.....	4
Clasificador empleado y arquitectura de la red.....	5
Entrenamiento de la red.....	7
Recursos.....	7
Pruebas realizadas.....	8
Resultados.....	8
Conclusiones.....	10
Referencias.....	11
Anexo.....	11
Anexo 1: Gráfica de Accuracy para el modelo multiclase.....	11
Anexo 2: Gráfica de Accuracy para el modelo multiclase.....	12
Anexo 3: Curva ROC del modelo binario.....	12
Anexo 4: Matriz de confusión del modelo multiclase.....	13
Anexo 5: Matriz de confusión del modelo binario.....	14

## Abstract Gráfico



## Tabla de resultados

Tipo de tejido	Especificidad ( <i>Precision</i> )	Sensibilidad ( <i>Recall</i> )
Tejido Adiposo	0	0
Sin tejido	0	0
Debris	0	0
Linfocitos	0	0
Moco	0	0
Músculo	0	0
Mucosa normal	0	0
Estroma	0	0
Epitelio tumoral	0.09	1
<b>Clasificación Binaria</b>		
Tumor	0.91	1
No tumor	0	0

# Introducción

La histopatología es una pieza clave en el arsenal de herramientas disponibles para clínicos e investigadores, proporcionando imágenes detalladas de muestras de tejido humano que son esenciales en el diagnóstico y manejo de una variedad de enfermedades, incluido el cáncer.

El análisis de imágenes, tradicionalmente llevado a cabo por patólogos, implica una revisión manual y meticulosa, un proceso que puede ser tedioso, propenso a la subjetividad y errores debido a su dependencia en la interpretación individual.

Ante estos desafíos, la inteligencia artificial y específicamente el Deep Learning, ofrecen herramientas poderosas para mejorar la precisión y eficiencia de estos análisis.

Durante estos últimos años se ha observado un creciente interés y éxito en la aplicación de técnicas de aprendizaje automático para el análisis de imágenes médicas, marcando un camino hacia resultados más objetivos y replicables.

En este proyecto se va a implementar y evaluar modelos de Deep Learning para la clasificación de imágenes de tejido en una de las nueve clases posibles y también se creará un clasificador binario que pueda clasificar la imagen en tumor o cualquier otro tejido.

Es importante mencionar que este proyecto se inspira en las metodologías y enfoques presentados en el trabajo de fin de grado de Juan José Carballo Pacheco, 'Clasificación de imágenes médicas con técnicas de Deep Learning' realizado en la Universidad de Extremadura, el cual ha demostrado la eficacia de las técnicas de Deep Learning en la clasificación precisa de imágenes médicas.

## Lenguaje de Programación, librerías y frameworks

Para el desarrollo de estas soluciones, hemos empleado el lenguaje de programación Python, apoyado en una serie de librerías.

- **TensorFlow y Keras:** Se ha hecho uso de estas librerías para construir, entrenar y evaluar modelos de DL, facilitando la implementación de redes neuronales convolucionales en el código.
- **OpenCV (cv2):** Esta librería se utiliza para el preprocesamiento de imágenes, incluida la conversión de imágenes de BGR a RGB y el redimensionamiento.
- **NumPy:** Empleado para la manipulación eficiente de arrays y matrices, permite realizar operaciones matemáticas y lógicas sobre grandes volúmenes de datos.

- **Matplotlib:** Utilizada para la visualización de datos, así se pueden analizar los resultados del entrenamiento y la validación de modelos.
- **Scikit-learn:** Esta librería se emplea para dividir los datos en conjuntos de entrenamiento y validación, así como para implementar varias técnicas de preprocesamiento y evaluación de modelos.

## Clasificador empleado y arquitectura de la red

El clasificador principal empleado en este proyecto es una Red Neuronal Convolutiva (CNN), específicamente basada en la arquitectura EfficientNetB0.

EfficientNetB0 ofrece una serie de ventajas como una mayor precisión con menos parámetros y menor consumo de recursos computacionales en comparación con otras arquitecturas CNN y permite una gran escalabilidad, pues es posible trabajar con diferentes capacidades de cómputo y tamaños de datos sin comprometer significativamente la eficiencia.

La arquitectura EfficientNetB0 está compuesta por varias capas. A continuación se resumen las capas y su funcionamiento:

- **Entrada (InputLayer):** La primera capa, es la entrada de una imagen, para este caso se ha especificado una imagen con una resolución de 224x224 píxeles y 3 canales de color (RGB).
- **Preprocesamiento (Rescaling y Normalization):** Se aplica un reescalado y normalización a las imágenes de entrada para asegurar que estén en el rango adecuado para la red neuronal.
- **Capas Convolucionales Stem:** Una capa "Stem" en una red neuronal, se refiere al conjunto inicial de capas que procesan los datos de entrada antes de que pasen a través del cuerpo principal de la red. En el caso de nuestra arquitectura, el stem consta de los siguientes componentes:
  - Conv2D: convolución 3x3 con un paso (stride) de 2
  - Normalización por lotes (Batch Normalization)
  - Activación de la función de activación Swish
- La arquitectura de EfficientNetB0 se compone de múltiples bloques, cada uno de los cuales tiene una estructura similar. Cada bloque incluye una serie de capas que realizan ciertas funciones y se van repitiendo a lo largo de la duración del modelo
  - **Capas Convolucionales:** Las capas convolucionales son aquellas diseñadas para extraer patrones y características visuales de las imágenes. En las primeras capas convolucionales, el tamaño de los kernels en estas comienza siendo mayor, de esta manera se pueden

capturar características espaciales generales, así en las en capas subsiguientes esto se ajusta para enfocarse en detalles más finos y concretos.

- **Capas de Agrupación (Pooling):** Las capas de agrupación siguen a las convolucionales y hacen un sub-muestreo de la imagen. Sirven para reducir la dimensionalidad de los datos de entrada, conservando al mismo tiempo las características más importantes.
- **Capas de Dropout:** La regularización mediante dropout consiste en que mientras se entrena el modelo se omiten neuronas al azar. De esta manera, no hay ninguna neurona más importante que pueda sobre aprender en el entrenamiento.
- **Capas Densas (Fully Connected):** Hacia el final de la red, las capas densas se encargan de realizar la clasificación basándose en las características extraídas por las capas anteriores. Estas capas interpretan las representaciones de alto nivel aprendidas por la red y las mapean a las categorías objetivo, como diferentes tipos de tejidos o presencia de células tumorales.

Para la tarea de clasificación binaria, se ha personalizado la arquitectura de EfficientNetB0 retirando su capa superior de clasificación predeterminada. En su lugar, se ha integrado una capa adicional que aplica Global Average Pooling al mapa de características obtenido de la última capa convolucional del modelo base. Esta operación de pooling promedia los valores para cada característica a lo largo de todo el mapa, resultando en un vector unidimensional que conserva la información esencial de las características. Posteriormente, este vector se utiliza como entrada para una capa densa singular con una función de activación sigmoide, diseñada específicamente para clasificación binaria. Así, el modelo produce una salida que oscila entre 0 y 1, tejido con o sin tumor.

Respecto a la función de activación empleada, EfficientNetB0 hace uso de Swish, como se mencionó en la explicación de la capa Stem. En comparación a ReLU, Swish mitiga el problema conocido como “dying ReLU” o el problema de las neuronas moribundas. Este problema consiste en que las neuronas pueden volverse inactivas y sólo producir una salida igual a cero independientemente de la entrada. Esta inactividad resulta en gradientes cero durante la retropropagación, deteniendo el proceso de aprendizaje para esas neuronas. Por el contrario, ya que Swish no tiene gradientes cero excepto en menos infinito, se mantiene un flujo de gradientes incluso cuando el valor de entrada es negativo. Así, las neuronas permanecen activas y continúan actualizándose y aprendiendo durante el entrenamiento.

El tamaño de los kernels varía a lo largo de la red. Empieza con kernels más grandes en las primeras capas convolucionales y se ajustan según las necesidades específicas

de las capas subsiguientes para capturar adecuadamente las características espaciales de las imágenes.

Finalmente, se utiliza la entropía cruzada categórica (`categorical_crossentropy`) como función de pérdida. Se utiliza como un indicador para evaluar la capacidad del modelo de distinguir correctamente entre diferentes categorías, en nuestro caso, tipos de tejido o presencia de tumores. Esta métrica mide el grado de discrepancia entre las probabilidades predichas por el modelo y las etiquetas reales de los datos. Un valor de pérdida cercano a 0 indicaría que el modelo es altamente eficaz en la clasificación precisa de las imágenes, identificando correctamente las características relevantes que distinguen los tejidos entre ellos para el clasificador multiclase y las características más relevantes que distinguen a un tejido con un tumor respecto a uno sano para el clasificador binario.

## Entrenamiento de la red

El entrenamiento de la red se configura con los siguientes parámetros clave: se establece un número de 40 epochs para el entrenamiento, lo cual indica la cantidad de veces que el conjunto de datos completo será procesado por la red.

El tamaño de mini-batch, definido por la variable `BATCH_SIZE`, es de 64, lo que significa que en cada iteración del entrenamiento, se tomarán 64 imágenes para actualizar los pesos de la red.

En cuanto a los pesos, al definir el modelo se ha especificado que no se carguen pesos pre entrenados, lo que implica que la red utilizará la inicialización por defecto proporcionada por Keras para esta arquitectura.

Respecto al learning rate, no se especifica explícitamente en el código, lo que significa que se utiliza el valor predeterminado del optimizador Adam, que es de 0.001.

## Recursos

El modelo se ha entrenado y evaluado en un equipo con una CPU intel core i9 de decimotercera generación, GPU Nvidia Geforce RTX 4060 con 8GB de memoria y 32 GB de memoria RAM.

La lectura y la separación de los datos en los conjuntos de entrenamiento y validación, así como el balanceo de los datos que componen ambos, se llevó a cabo en 2 minutos y 24'7 segundos.

En lo relativo a la clasificación multiclase, el etiquetado de los datos se hizo en 1'5 segundos, el entrenamiento del modelo precisó de 2 horas, 7 minutos y 39'5 segundos y la validación se realizó en 10'6 segundos.

El modelo de clasificación binaria empleó 1'8 segundos en etiquetar los datos, 32 minutos y 23'4 segundos en entrenar el modelo y 10'6 segundos en evaluar el modelo.

## Pruebas realizadas

Para la realización de este proyecto, se han empleado únicamente las imágenes que conforman el conjunto de datos CRC-VAL-HE-7K, las cuales provienen del NCT tissue bank. Para cada tipo de tejido, se han dividido las imágenes en dos conjuntos independientes; un conjunto de entrenamiento y otro de validación.

El conjunto de validación, de cada grupo, está formado por el veinte por ciento de las imágenes que corresponden a dicho grupo.

Por otro lado, tanto el grupo de entrenamiento como el de validación, se han balanceado. Este balanceo se ha realizado mediante Data Augmentation; es decir, se han aumentado los elementos de cada etiqueta hasta conseguir que todas las etiquetas cuenten con el mismo número de elementos. Los elementos incluidos en cada grupo, consisten en elementos del mismo grupo rotados o con distinto contraste respecto al elemento original.

Con todo esto, el modelo se ha entrenado con el conjunto de entrenamiento aumentado, y se ha evaluado el modelo con el conjunto de validación aumentado.

## Resultados

En este apartado se explican los resultados obtenidos. Primero, se comentan los valores de "Accuracy" obtenidos para el modelo multiclase y el binario, véase Anexo 1 y 2 respectivamente. Estas gráficas muestran resultados a lo largo de 40 epochs, donde la línea azul representa el entrenamiento y la línea naranja la validación.

Respecto a los resultados para el modelo multiclase se observan variaciones significativas en los valores de entrenamiento mientras que los valores validación presentan una mayor constancia a lo largo de los epochs. Aun así, el accuracy del entrenamiento aumenta con el tiempo, lo que indica que el aprendizaje del modelo y su mejora para la clasificación correcta. Por otro lado, la validación no muestra ninguna tendencia de mejora, lo que puede indicar que el modelo no generaliza bien nuevos datos que no se vieron durante el entrenamiento.



Lo más destacable es la brecha entre la accuracy de entrenamiento y la accuracy de validación, con la accuracy de entrenamiento siendo más alta en casi todas las epochs, esto puede ser indicativo de overfitting.

En relación a la gráfica del modelo binario se aprecian ciertas características que llaman la atención. Primero, los valores tanto para entrenamiento como para validación son extremadamente altos. Estos valores indican que el modelo funciona excepcionalmente bien para clasificar las imágenes. Segundo, no hay desviaciones de los valores a lo largo de los epochs, si no todo lo contrario, tanto el valor para el entrenamiento como para la validación se mantienen completamente constantes. Este comportamiento indica que no hay ni overfitting ni underfitting y que el modelo alcance rápidamente su máximo rendimiento.

Siguiendo con el análisis de resultados, también se ha obtenido una curva ROC del modelo binario, véase Anexo 3. El eje Y de la curva ROC muestra la TPR(True Positive Rate), que es la proporción de positivos reales, imágenes de tejidos con tumor que se identificaron correctamente como tales por el modelo. El eje X muestra la FPR(False Positive Rate), que es la proporción de negativos reales, imágenes de tejidos sin tumor que se identificaron incorrectamente como positivos, es decir, con tumor. La leyenda de la gráfica muestra un área bajo la curva(AUC) de 0.65. Éste área mide la capacidad general del modelo para discriminar entre las clases positivas y negativas. Dicho valor demuestra que el modelo tiene aún margen de mejora, especialmente teniendo en cuenta que un valor de 0.5 sería equivalente a un rendimiento no mejor que el azar. En este caso, el modelo indica que tiene una capacidad del 65 por ciento de clasificar una instancia positiva(hay tumor) elegida aleatoriamente más alta que una negativa(no hay tumor).

Los últimos resultados obtenidos han sido una serie de matrices de confusión para el modelo multiclase y binario, véase Anexo 4 y Anexo 5 respectivamente.

Comenzando con matriz de confusión del modelo multiclase, la última columna correspondiente a la clase TUM(tumor), se puede ver una concentración alta de predicciones correctas, lo que indica que el modelo es capaz de identificar correctamente la mayoría de los tejidos con tumor.

Respecto a la matriz de confusión del modelo binario se muestran las predicciones para las categorías “No Tumor” y “Tumor”. Las celdas se detallan a continuación:

- Celda superior izquierda: True Negative, casos en los que el modelo predijo correctamente la ausencia de un tumor. Esta celda tiene una cantidad de instancias casi nula.

- Celda inferior izquierda: False Negative, casos en los que el modelo no detectó un tumor, pero en realidad había uno. Al igual que con el ejemplo anterior, también hay muy pocas instancias de este caso.
- Celda superior derecha: False Positive, casos en que el modelo indicó la presencia de un tumor cuando en realidad no había ninguno. Se observa que esta celda tiene más instancias respecto a las dos anteriores.
- Celda inferior derecha: True Positive, casos en que el modelo predijo correctamente la presencia de un tumor. Este caso es el que más instancias tiene respecto al resto con una diferencia muy significativa.

La ausencia de falsos positivos es favorable desde el punto de vista clínico, ya que indica que el modelo no está generando alarmas innecesarias sobre la presencia de tumores. Sin embargo, la aparente ausencia de verdaderos y falsos negativos podría ser un área de preocupación.

## Conclusiones

Los avances en inteligencia artificial (IA) y aprendizaje profundo (Deep Learning, DL) están revolucionando el campo de la histopatología, proporcionando nuevas vías para el análisis y la interpretación de imágenes médicas.

La IA, con su capacidad para discernir patrones complejos y detalles minúsculos que podrían no ser evidentes para el observador humano, promete no solo un soporte decisivo en el diagnóstico clínico, permitiendo evaluaciones más rápidas y precisas, sino también una interpretación más objetiva de los datos histopatológicos, potenciando así la exactitud y confiabilidad en la detección de enfermedades.

El modelo de clasificación binaria y multiclase desarrollado, demuestra la viabilidad de emplear estas técnicas en el contexto real de la medicina diagnóstica y la posibilidad de trasladar estas prácticas a labores clínicas. La utilización de EfficientNetB0 y las estrategias de entrenamiento y validación implementadas, nos acercan a un sistema que podría asistir a los patólogos en su labor diaria, reduciendo la carga de trabajo y proporcionando una segunda opinión automatizada.

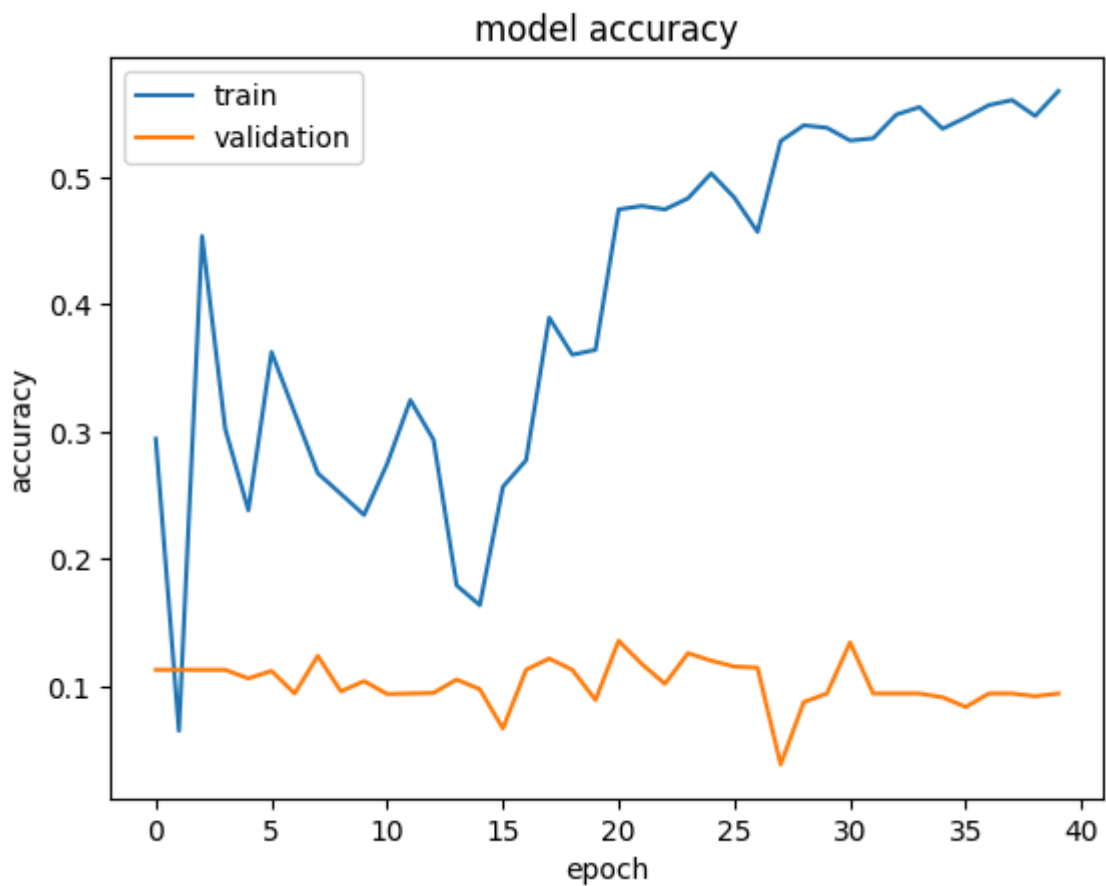
Sin embargo, para una implementación clínica efectiva, es crucial considerar varios pasos adicionales. Primero, sería necesario realizar validaciones extensas con conjuntos de datos independientes para asegurar la generalización de los modelos. Además, las regulaciones del ámbito sanitario requieren de procesos de certificación rigurosos para garantizar la seguridad y eficacia de los algoritmos utilizados. Finalmente, la aceptación clínica implica la integración de estos sistemas dentro del flujo de trabajo existente, ofreciendo interfaces intuitivas y procesos de decisión transparentes que inspiren confianza en los profesionales médicos.

## Referencias

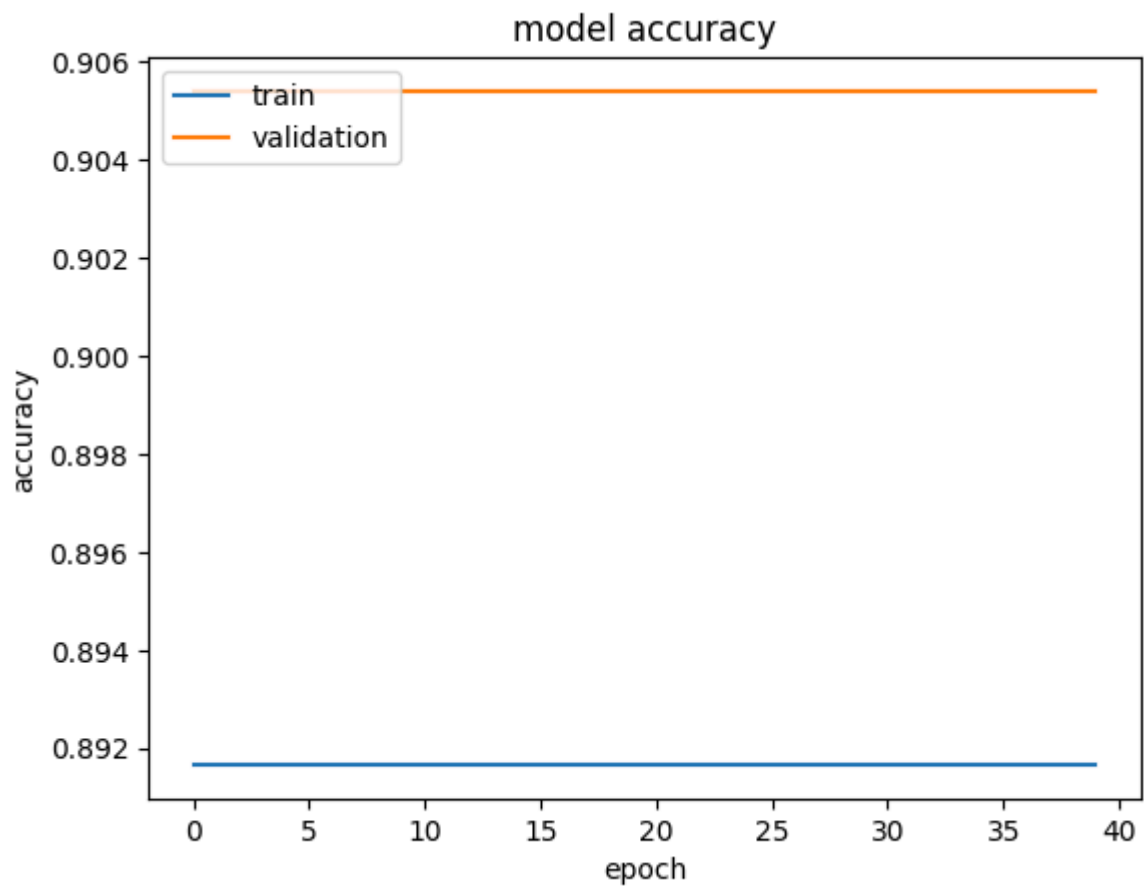
Carballo Pachecho, J. J. 2022. Clasificación de imágenes médicas con técnicas de Deep Learning [Trabajo de Fin de Grado, Universidad de Extremadura]. [https://dehesa.unex.es/bitstream/10662/16482/1/TFGUEx\\_2022\\_Carballo\\_Pacheco.pdf](https://dehesa.unex.es/bitstream/10662/16482/1/TFGUEx_2022_Carballo_Pacheco.pdf)

## Anexo

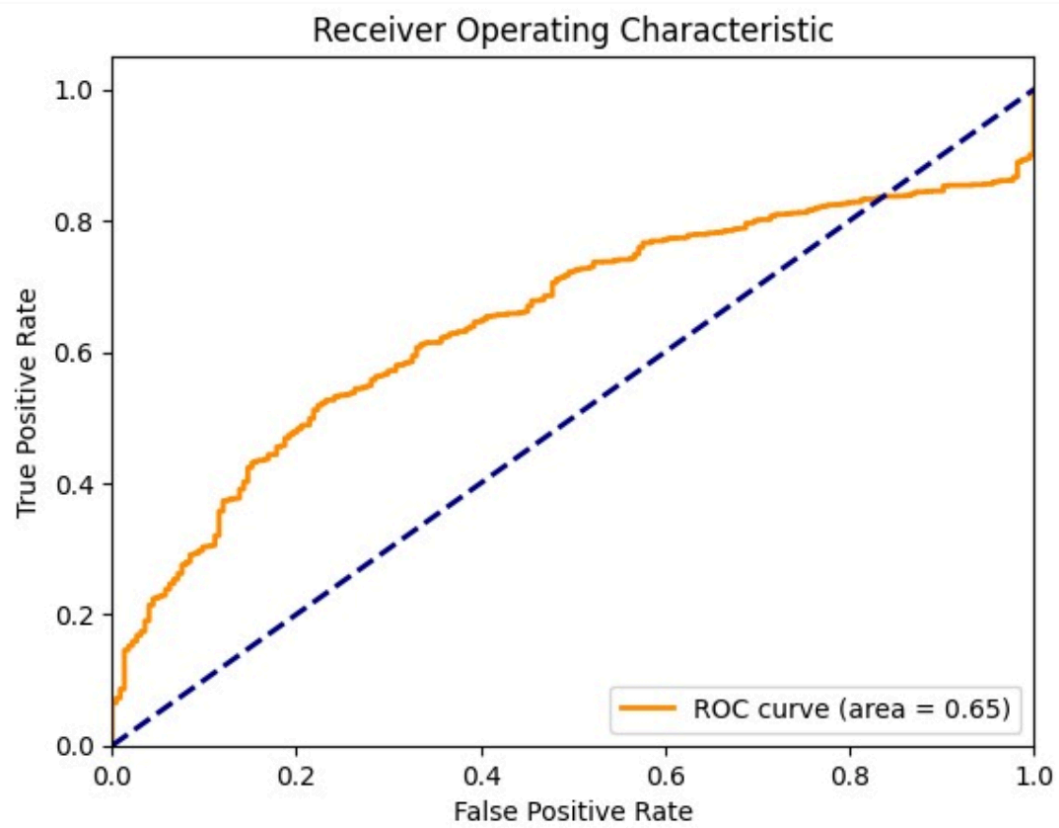
Anexo 1: Gráfica de Accuracy para el modelo multiclase



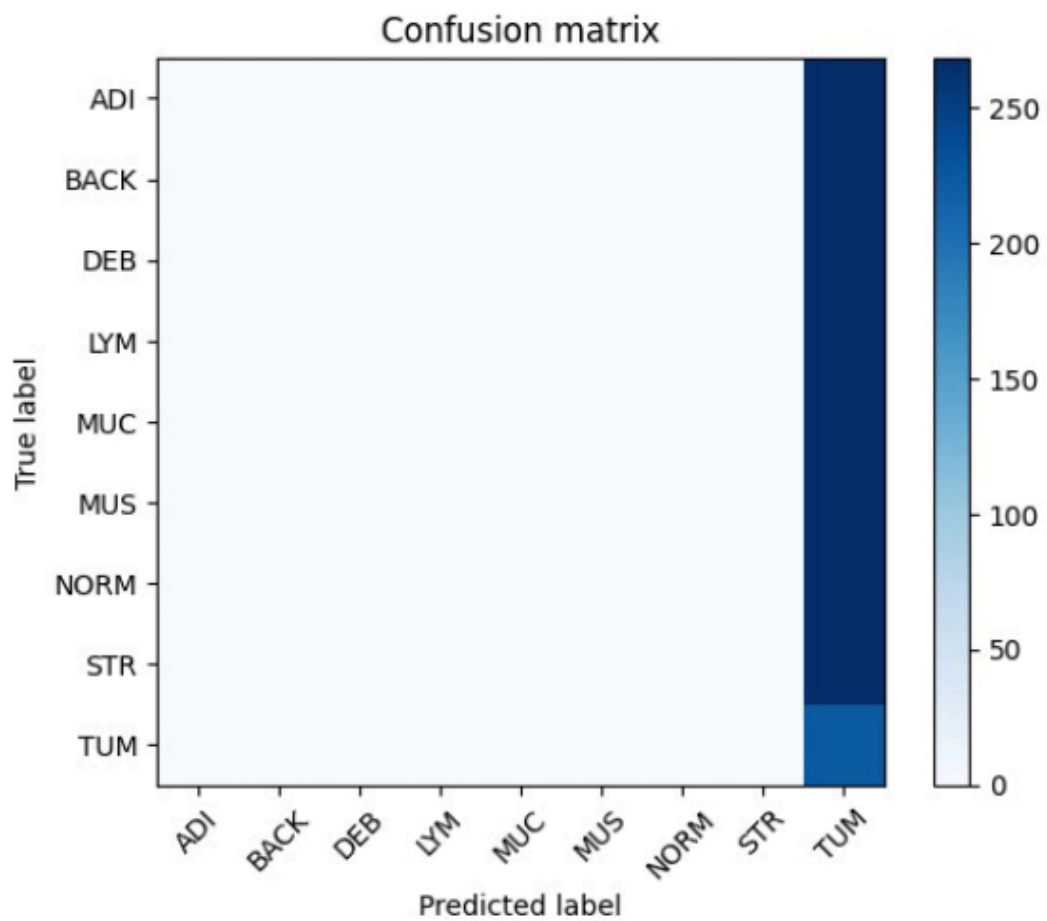
## Anexo 2: Gráfica de Accuracy para el modelo multiclase



### Anexo 3: Curva ROC del modelo binario



#### Anexo 4: Matriz de confusión del modelo multiclase



## Anexo 5: Matriz de confusión del modelo binario

