Histopathologic Cancer Detection

PROYECTO FINAL CURSO DEEP LEARNING

Iván Darío Gómez Marín.
C.C: 1.041.147.729

Docente: **Raúl Ramos Pollán**

UNIVERSIDAD
DE ANTIOQUIA
1 8 0 3

Facutad de ingeniería
Universidad de Antioquia

27 de septiembre de 2023

1 Avance del proyecto

1.1. Contexto de la aplicación

La aplicación propuesta proviene la una competencia publicada en la plataforma Kaggle, que puede encontrarse en el siguiente link: histopathologic-cancer-detection.
El reto de la competencia radica en **crear un algoritmo para identificar metástasis en pequeños parches de imágenes tomadas de exploraciones patológicas digitales de mayor tamaño**. Los datos para esta competición son una versión ligeramente modificada del conjunto de datos de referencia PatchCamelyon (PCam).PCam convierte la tarea clínica de detección de metástasis en una sencilla tarea de clasificación binaria de imágenes, similar a CIFAR-10 y MNIST.

1.2. Objetivo de ML

Dada una imagen etiquetada previamente indicando si se identifica tejido cancerígeno en ella, el objetivo es crear un modelo de clasificación binario que dada una imagen permita discriminarla, indicado si hay presencia de tejido cancerígeno o no; 1 indica la presencia de tejido cancerígeno en la imagen y 0 la ausencia de este.

1.3. Dataset

El conjunto completo de imágenes disponible para el entrenamiento y validación del modelo, se puede acceder a través del siguiente link: data kaggle competition.
El conjunto de datos consiste en imágenes microscópicas de tejido de ganglios linfáticos. Cada imagen tiene una resolución de 96x96 píxeles, y la tarea consistirá en identificar tejido canceroso metastásico en una región central de la imagen de 32x32 píxeles. Según la descripción de la competencia de Kaggle, la identificación de al menos 1 píxel de tejido tumoral etiquetaría efectivamente la imagen como positiva, es decir, con cáncer. El conjunto de datos original de entrenamiento consta de **220.025 imágenes**.
Nota: *En nuestro caso de estudio, por cuestiones pedagógicas y por temas de capacidad computacional, haremos uso de un subconjunto de* ***30.000 imágenes****. En el conjunto de datos original, aproximadamente el 60 % de las imágenes no contienen tejido cancerígeno; el subconjunto de*

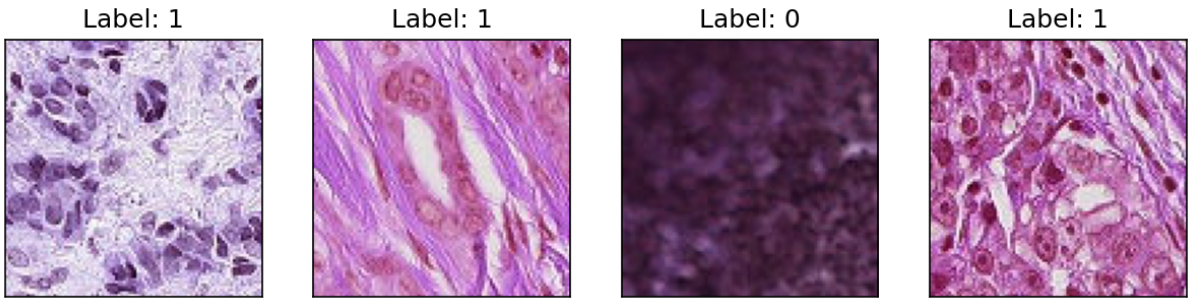


Figura 1.1: Imágenes ejemplo

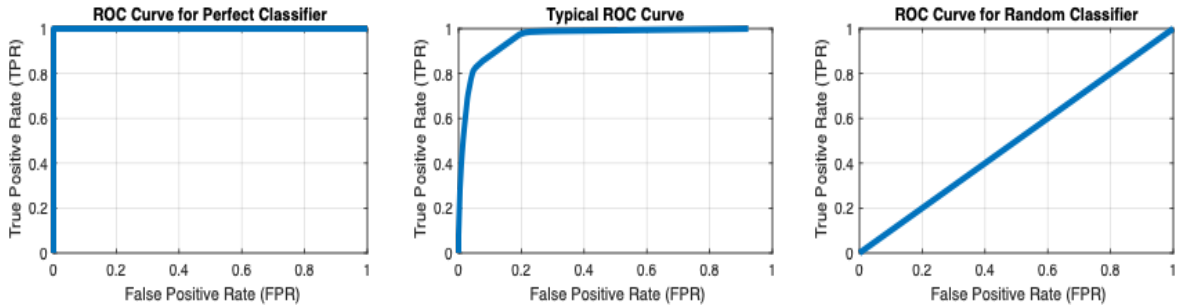


Figura 1.2: Curva ROC (Receiver Operating Characteristic Curve)

imágenes para este trabajo se eligió respetando esta misma proporción.

La figura 1.1 muestra tres imágenes elegidas aleatoriamente del dataset disponible para crear el modelo con su respectiva etiqueta (**1:** tejido cancerígeno; **0:** Tejido sin cáncer).

Cabe resaltar que el dataset está desbalanceado debido a que está conformado por 17.876 imágenes sin tejido cancerígeno y 12.124 con tejido cancerígeno.

1.4. Métricas de desempeño

En la competición, la evaluación del desempeño se hizo con base en la curva ROC; de acuerdo con la gráfica 1.2, en general cuanto más "arriba y a la izquierda" del diagrama se encuentre la curva ROC, mejor será el clasificador.

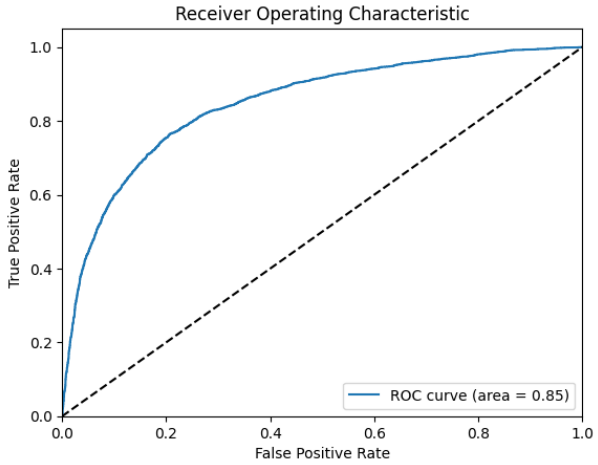
1.5. Referencias y resultados previos

Inicialmente se dividió el dataset en dos conjuntos: uno para entrenamiento y otro para validación en la proporción tradicional 70 : 30; posteriormente se planteó el modelo que se muestra en la figura 1.3 y se realizó su entrenamiento durante 10 epochs. La gráfica 1.4a muestra la curva ROC obtenida con este modelo; vemos que los resultados ya son buenos, sin embargo en el reporte de la clasificación (ver figura 1.4b) podemos apreciar que la precisión y el recall para la clase 1 (imágenes con tejido canceroso) son inferiores. El objetivo es aplicar métodos que ayuden a lidiar con el de clasificación desbalanceado. A continuación muestro los links a dos artículos que usaré como referencia para tal fin:

- Classification on imbalanced data
- Survey on deep learning with class imbalance

```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 sequential_2 (Sequential)   (None, 196, 196, 3)       0

 conv2d_2 (Conv2D)           (None, 194, 194, 64)      1792

 max_pooling2d_2 (MaxPoolin  (None, 97, 97, 64)        0
 g2D)

 conv2d_3 (Conv2D)           (None, 95, 95, 32)        18464

 max_pooling2d_3 (MaxPoolin  (None, 47, 47, 32)        0
 g2D)

 flatten_1 (Flatten)         (None, 70688)             0

 dense_3 (Dense)             (None, 256)               18096384

 dropout_1 (Dropout)         (None, 256)               0

 dense_4 (Dense)             (None, 128)               32896

 dense_5 (Dense)             (None, 1)                 129
...
Total params: 18149665 (69.24 MB)
Trainable params: 18149665 (69.24 MB)
Non-trainable params: 0 (0.00 Byte)
```

Figura 1.3: Modelo inicial propuesto



(a) Curva ROC

	precision	recall	f1-score	support
0	0.79	0.87	0.83	5363
1	0.77	0.65	0.71	3637
accuracy			0.78	9000
macro avg	0.78	0.76	0.77	9000
weighted avg	0.78	0.78	0.78	9000

(b) Reporte de clasificación

Figura 1.4: Resultados del modelo inicial