

Grupo 5			
Alumno/a:	Sara Rodríguez Galán	NIA:	100496119
Alumno/a:	Lucía Merinero López	NIA:	100496052
Alumno/a:	Jhonatan Barcos Gambaro	NIA:	100548615

1. Descripción

Describiremos las implementaciones y los cambios que han llevado a cabo cada miembro de este proyecto, donde discutiremos su finalidad y el resultado que hemos logrado tras su implementación.

Sara Rodríguez Galán

- **Exploración de diferentes arquitecturas de CustomNet.**

Se probaron diferentes arquitecturas para el CustomNet, modificando las capas convolucionales, la cantidad de filtros por capa y la profundidad de la sección lineal final, con el objetivo de encontrar una estructura que ofreciera un mejor desempeño para la clasificación. Pudimos ver que las redes aprendían bien en conjunto pero no lograba generalizar correctamente a datos no vistos.

- **Implementación de nuevas transformaciones.**

Se añadió y probó nuevas transformaciones, las cuales pensó que serían las más adecuadas para problemas dermatológicos. Algunas de estas transformaciones son: mejoras de contraste, recortar la mancha, alterar el color de la imagen... Empleando algunas logramos bastante mejora.

- **Implementación de Early Stopping para evitar overfitting.**

Se utilizó esta técnica durante el entrenamiento con el objetivo de detener el proceso de aprendizaje en el caso de que el modelo dejara de mejorar, para evitar entrenar innecesariamente cuando el modelo no presenta mejoras significativas. Nos resultó útil, cuando el modelo tendía a sobreajustarse después de cierto punto.

Lucía Merinero López

- **Evaluación de diferentes combinaciones de optimizador y scheduler.**

Se combinó el optimizador Adams y el scheduler ReduceLROnPlateau con el objetivo de mejorar el rendimiento del modelo. En el caso de CustomNet, permitió un entrenamiento estable, pero al aplicarlo a modelos preentrenados no observamos mejoras frente al optimizador y scheduler originales.

- **Implementación de Mixup para mejorar la generalización.**

Se empleó Mixup, que consiste en combinar aleatoriamente pares de imágenes y sus respectivas etiquetas durante el entrenamiento, usando un coeficiente extraído de una distribución Beta. La idea era forzarlo a aprender representaciones más generales frente a variaciones sutiles. Llegó a disminuir ligeramente el AUC.

- **Implementaciones de diferentes combinaciones de transformaciones.**

Se añadieron transformaciones de rotaciones, escala de grises y ruido gaussiano. Probó a hacer combinaciones con las ya añadidas anteriormente pero empeoró el resultado en el Codabench.

Jhonatan Barcos Gambaro

- **Implementación de muestreo ponderado para clases desbalanceadas.**

Se intentó compensar el desbalance de clases en el conjunto de entrenamiento, empleando WeightedRandomSampler para definir un muestreo balanceado con reemplazo. Pudo comprobar que los resultados en Codabench no mejoraron significativamente, lo que nos sugirió que el desbalance de clases no era el principal factor limitante en el rendimiento del modelo.

- **Implementación de composición de transformaciones y simulación en muestras.**

Se implementaron nuevas transformaciones y se combinaron de distintas formas con el objetivo de priorizar las características relevantes de las imágenes. Para ello, realizamos un estudio previo de la realidad, con lo que pudimos comprobar que las transformaciones más significativas serían las de orientación, desplazamiento, zoom, ruido, deformaciones y color.

- **Diseño de una nueva arquitectura CustomNet con normalización por lotes y mayor profundidad y prueba de modelos Fine-Tuning.**

Se propuso una versión diferente de CustomNet, aumentando las capas convolucionales e introdujo Batch Normalization después de cada capa para estabilizar el entrenamiento. No mejoró debido a que resultó demasiado profunda para la cantidad de datos disponibles. Así mismo se probaron distintos modelos Fine-Tuning como AlexNet, DenseNet, EfficientNet y ResNet, siendo este último el que mejores resultados obtuvo.

El objetivo perseguido en la realización de este proyecto era acercarse lo máximo posible a una predicción perfecta, es decir sin sobreajuste ni infra ajuste, de manera que la red al entrenarse se adapte a los cambios cuando se le pasan los datos reales. De esta forma, se busca alcanzar el diagnóstico correcto para así poder ayudar a identificar las distintas enfermedades utilizando predicciones a través de la inteligencia artificial.

2. Material Extra

A continuación, detallaremos las principales técnicas utilizadas para maximizar la calidad de nuestros datos así como el del entrenamiento de la red.

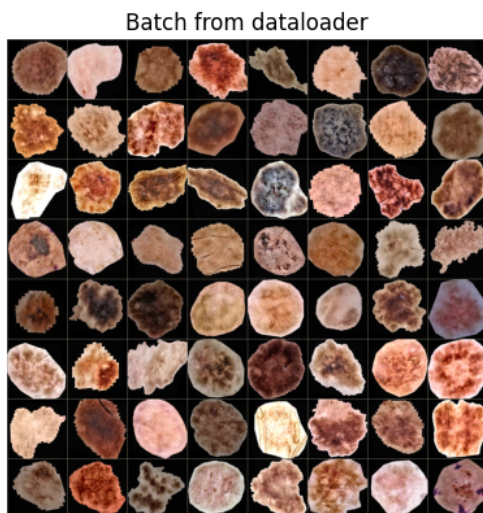
Preparación de los datos: Preprocesamiento y Balanceo.

Respecto a nuestro conjunto de datos, hemos comprobado que las clases no estaban balanceadas:

```
Distribución de clases en el dataset de entrenamiento:  
label  
0    1372  
1     374  
2     254  
Name: count, dtype: int64  
  
Distribución de clases en el dataset de validación:  
label  
0     78  
2     42  
1     30  
Name: count, dtype: int64
```

Para solucionar este problema, hemos realizado un ajuste de los pesos de la red para penalizar las clases en función de su frecuencia de aparición.

Por otra parte, para mejorar la variabilidad de los datos y una mejor detección de características relevantes, hemos aplicado técnicas de Data augmentation en el conjunto de entrenamiento. Para ello hemos utilizado una composición de transformaciones sobre las imágenes, tal que como resultado obtenemos estas nuevas imágenes:



Modelos de entrenamiento para la clasificación de imágenes.

Respecto a los modelos utilizados, tenemos la CustomNet, una arquitectura CNN dividida en dos partes principales: extracción de características y clasificación.

Dicha extracción de características la conseguimos a través de 5 capas convolucionales, cada una de ellas seguidas de una función de activación ReLU y un MaxPool para reducir el tamaño de la imagen de entrada. Tras esta parte, para una correcta clasificación, las salidas pasan por capas densas con funciones de activación ReLU y regularización con Dropout para evitar el overfitting. Finalmente, la capa de salida tiene 3 neuronas, correspondientes a las clases que queremos predecir: melanoma, queratosis y nevus.

Así mismo, hemos utilizado un modelo Fine-Tuning para la clasificación de estas imágenes. Tras una exhaustiva exploración con diferentes modelos preentrenados, el que mejor resultados ha obtenido ha sido el modelo ResNet18. Para poder utilizarlo en nuestro proyecto, se ha modificado la última capa de ese modelo ajustada a 3 clases específicas.

Este modelo ha sido entrenado con un optimizador SGD, ajustado con una reducción de Learning Rate cada 7 epochs y con la función de pérdida CrossEntropyLoss.

Declaración de uso de IA Generativa.

En la elaboración de dicho proyecto nos hemos apoyado en herramientas de IA Generativa, para ahorrar tiempos a la hora de redacción de código. Para mejorar la eficiencia de estas herramientas se han utilizado técnicas de Prompt Engineering, así como se ha guiado la creación de código de forma dinámica.