Informe Práctica 4: Clasificación de lesiones cutáneas con YOLOv8

Mario Polaino, Antonio Moya, Leon Skalczynski, Fernando García

mariopolaino@uma.es, amoyam05@uma.es, leonskal@uma.es, nano_2005@uma.es
3º Ingeniería del Software. Universidad de Málaga

Abstract. En esta práctica damos un paso más dentro de la asignatura y trabajamos con modelos de aprendizaje profundo aplicados a imágenes. En concreto, utilizamos la versión de clasificación de YOLOv8 para distinguir entre distintos tipos de lesiones cutáneas a partir de fotografías dermatoscópicas del conjunto de datos ISIC 2019. Nuestro objetivo principal es construir un subconjunto balanceado del dataset, entrenar tres variantes del modelo (nano, small y medium), y comparar su rendimiento en términos de precisión global y capacidad para detectar correctamente el melanoma.

1 Introducción

En prácticas anteriores hemos trabajado con conjuntos de datos tabulares y técnicas clásicas como normalización, PCA y validación cruzada. En esta cuarta práctica nos centramos en el ámbito del aprendizaje profundo y el procesamiento de imágenes, concretamente en la clasificación de lesiones cutáneas. La detección temprana del melanoma es un problema importante en medicina, y el conjunto de datos ISIC 2019 recoge miles de imágenes etiquetadas en varias clases de lesiones (melanoma, nevus, carcinoma basocelular, etc.). Sobre este problema entrenamos tres modelos de la familia YOLOv8–cls (*nano*, *small* y *medium*) con el objetivo de estudiar cómo influye el tamaño del modelo en un escenario de pocos datos y qué compromiso se obtiene entre complejidad, tiempo de entrenamiento y capacidad de generalización.

En las siguientes secciones describimos cómo hemos preparado el subconjunto de datos, la configuración de los modelos, los resultados obtenidos y algunas conclusiones sobre su comportamiento.

2 Conjunto de datos y preparación

2.1 Descarga y formato original

Partimos del conjunto de datos ISIC 2019, que facilita un fichero `.csv` con las etiquetas y un archivo comprimido con las imágenes de entrenamiento. Las etiquetas vienen codificadas en formato *one-hot*, es decir, para cada imagen se incluyen varias columnas binarias indicando si pertenece o no a cada categoría. Cargamos el `csv` con `pandas` y utilizamos la función `idxmax` para recuperar, para cada fila, el nombre de la clase correspondiente. De este modo obtenemos una estructura sencilla donde cada imagen se asocia a una única etiqueta textual (por ejemplo, `MEL`, `NV`, `BCC`, etc.). En nuestro caso trabajamos con ocho clases de lesión: VASC, BKL, SCC, MEL, DF, NV, AK y BCC.

2.2 Construcción del subconjunto balanceado

El conjunto ISIC original está fuertemente desbalanceado, con muchas más imágenes de algunas clases (como nevus) que de otras. Para esta práctica se nos pedía trabajar con un subconjunto reducido pero balanceado, de forma que cada clase aportase la misma cantidad de ejemplos. Para ello, para cada una de las ocho clases seleccionamos aleatoriamente un total de 110 imágenes. De éstas, las primeras 100 se reservan para el conjunto de entrenamiento y las 10 restantes para el conjunto de test. De esta forma obtenemos:

– Conjunto de entrenamiento: $8 \times 100 = 800$ imágenes.
– Conjunto de test: $8 \times 10 = 80$ imágenes.

Finalmente, ejecutamos un pequeño script de comprobación que recorre las carpetas `train` y `test` y cuenta las imágenes de cada clase. La Tabla 1 resume el número de ejemplos resultante, confirmando que el subconjunto generado está perfectamente balanceado.

Table 1. Número de imágenes por clase en los subconjuntos generados.

Clase	# Train	# Test
VASC	100	10
BKL	100	10
SCC	100	10
MEL	100	10
DF	100	10
NV	100	10
AK	100	10
BCC	100	10

2.3 Organización de carpetas para YOLOv8

YOLOv8 en modo clasificación espera una estructura de directorios específica. A partir del muestreo anterior, copiamos las imágenes a una carpeta raíz `dataset`, que contiene dos subdirectorios principales: `train` y `test`. Dentro de cada uno de ellos creamos una subcarpeta por clase (AK, BCC, BKL, DF, MEL, NV, SCC y VASC) y movemos allí las imágenes correspondientes. De esta forma, por ejemplo, las imágenes de entrenamiento de melanoma se encuentran en `dataset/train/MEL` y las de test en `dataset/test/MEL`. La herramienta de YOLOv8 infiere automáticamente la etiqueta de cada imagen a partir del nombre de la carpeta en la que se encuentra.

3 Modelos y configuración experimental

3.1 Modelos YOLOv8 de clasificación

En lugar de utilizar YOLO en su modo de detección, en esta práctica empleamos la variante de clasificación de imágenes (`yolov8n-cls`, `yolov8s-cls` y `yolov8m-cls`). A nivel interno, estos modelos constan de una red convolucional que actúa como *backbone* para extraer características, seguida de una cabeza de clasificación que devuelve una probabilidad para cada una de las ocho clases. Las tres versiones difieren principalmente en profundidad y número de parámetros:

– **YOLOv8n-cls (nano)**: modelo más ligero y rápido, con menor capacidad de representación.
– **YOLOv8s-cls (small)**: tamaño intermedio, intenta equilibrar precisión y coste computacional.
– **YOLOv8m-cls (medium)**: modelo más grande y potente, con mayor capacidad de ajuste.

3.2 Hiperparámetros de entrenamiento

Entrenamos cada modelo sobre el mismo conjunto de entrenamiento para poder compararlos en igualdad de condiciones. De forma resumida, utilizamos la siguiente configuración:

– Número de épocas: 20. **¡ATENCIÓN!** (Según hemos visto, entre 20 y 50, suele ser suficiente para esta cantidad de imágenes. Nosotros hemos utilizado 20, aunque para ejecutar train.ipynb en el vídeo lo hemos puesto a 1. Igualmente eval.ipynb en el vídeo tomará el zip generado con 'epochs': 20, por eso el microcorte que habrá en el vídeo)
– Tamaño de lote (*batch size*): 16.
– Tamaño de las imágenes de entrada: 224×224 píxeles.
– Optimizador y resto de detalles: configuración por defecto de la implementación de YOLOv8.

Durante el entrenamiento, YOLOv8 genera automáticamente una carpeta `runs/classify/` con tres subdirectorios, uno por modelo, donde se almacenan los pesos finales (`best.pt`) y las curvas de entrenamiento y validación.

4 Resultados experimentales

4.1 Evolución de la precisión y de la pérdida

En la Figura 1 se muestra la evolución de la *top-1 accuracy* en validación a lo largo de las 20 épocas para los tres modelos. Las tres curvas siguen un comportamiento parecido al principio: un crecimiento rápido durante las primeras épocas hasta estabilizarse alrededor de la época 5–6. Se aprecia que el modelo *nano* (línea azul) se queda en torno a una precisión de validación de aproximadamente 0.62, mientras que el modelo *small* (verde) llega a valores similares, pero con más oscilaciones a partir de la mitad del entrenamiento. El modelo *medium* (rojo) es el que obtiene la mejor precisión en validación, alcanzando valores cercanos a 0.70 en las últimas épocas y manteniendo una tendencia ascendente más clara.

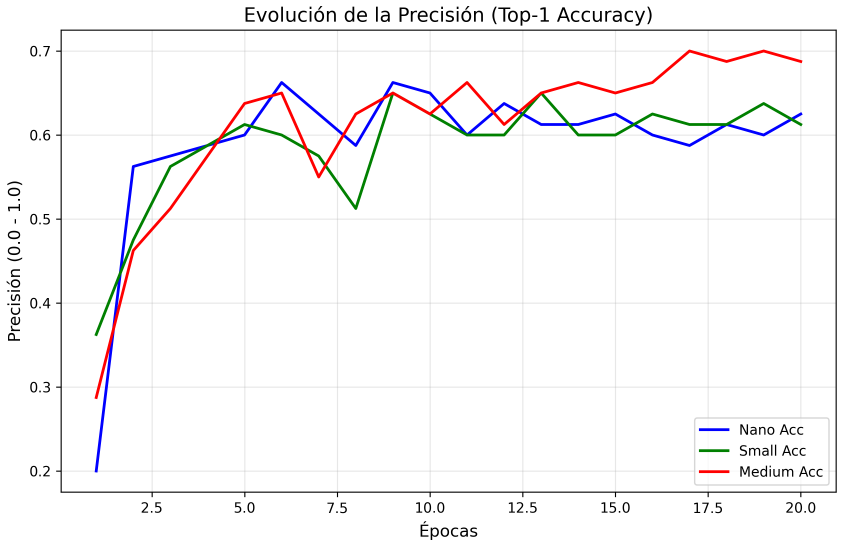

Fig. 1. Evolución de la precisión en validación para YOLOv8n, YOLOv8s y YOLOv8m.

En la Figura 2 comparamos la pérdida de validación de los tres modelos. Todas las curvas descienden bruscamente en las primeras épocas, lo que indica una fase inicial de aprendizaje rápido. A partir de ahí, el modelo *nano* y el *medium* se estabilizan alrededor de una pérdida de entre 1.0 y 1.1, mientras que el modelo *small* presenta una pérdida algo mayor y bastante más ruidosa, con subidas y bajadas entre épocas.

En general, no observamos un sobreajuste extremo en ninguno de los tres casos, aunque la inestabilidad del modelo *small* sugiere que le cuesta más encontrar un punto de equilibrio con tan pocos datos.

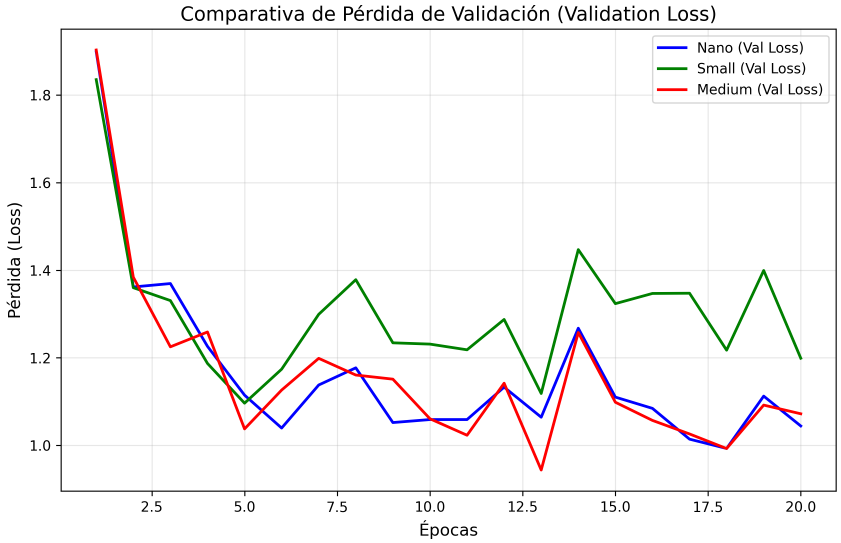

Fig. 2. Comparativa de la pérdida de validación para los tres modelos.

Además de las curvas, en la Tabla 2 resumimos, para cada modelo, el mejor valor de *top-1 accuracy* alcanzado durante el entrenamiento y la mínima pérdida de validación registrada (estos valores se han obtenido directamente del archivo `results.csv` generado por YOLOv8).

Table 2. Mejor precisión y mínima pérdida de validación alcanzadas por cada modelo durante el entrenamiento.

Modelo	Top-1 Accuracy (%)	Min Val Loss
YOLOv8n-cls	66.25	1.03
YOLOv8s-cls	65.00	1.07
YOLOv8m-cls	68.75	0.99

4.2 Matriz de confusión por modelo

Una vez finalizado el entrenamiento, evaluamos cada modelo sobre el conjunto de test. En las Figuras 3, 7 y 5 mostramos las matrices de confusión obtenidas para YOLOv8n, YOLOv8s y YOLOv8m respectivamente.

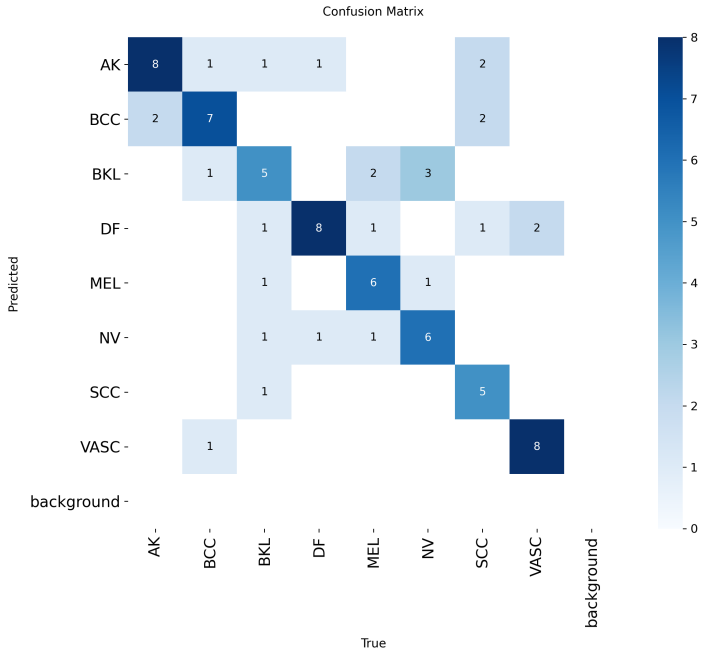

Fig. 3. Matriz de confusión en el conjunto de test para YOLOv8n-cls.

En el caso de YOLOv8n, la diagonal principal está razonablemente marcada, pero para varias clases el modelo comete entre dos y tres errores sobre las 10 imágenes disponibles. El melanoma (MEL) se clasifica correctamente en 5 de los 10 casos, mientras que los falsos negativos tienden a confundirse con nevus (NV) u otras lesiones benignas.

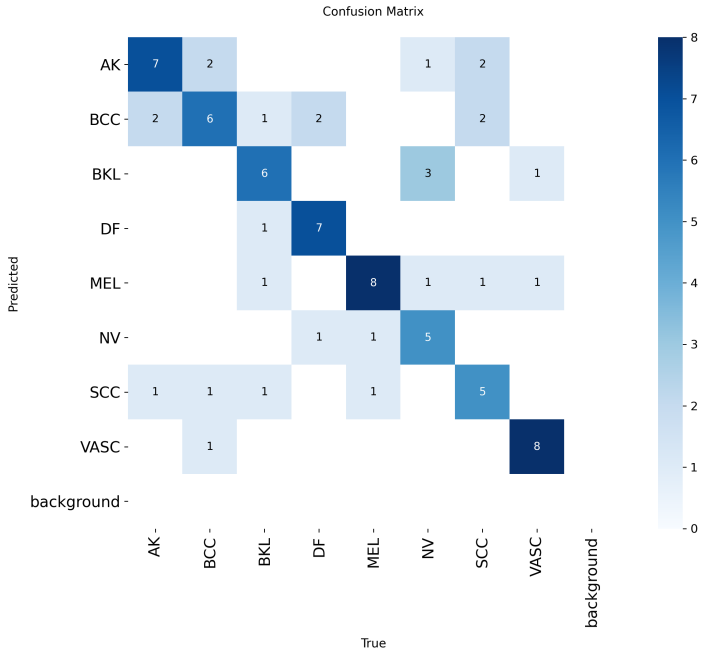

Fig. 4. Matriz de confusión en el conjunto de test para YOLOv8s-cls.

El modelo YOLOv8s mejora ligeramente la diagonal: por ejemplo, en melanoma acierta 6 de las 10 imágenes, y también aumenta el número de aciertos en clases como AK o BCC. Sin embargo, la matriz sigue mostrando bastantes confusiones entre lesiones visualmente similares, y en general el salto respecto al modelo *nano* no es muy grande.

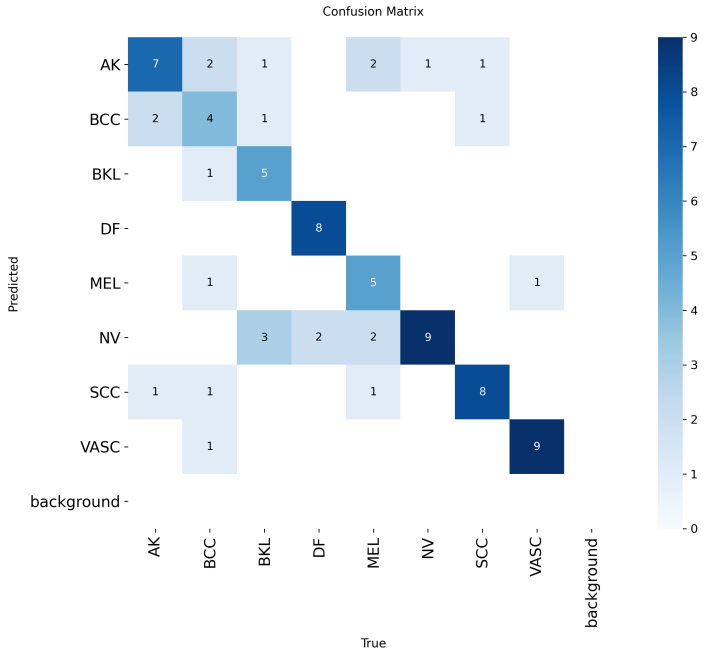

Fig. 5. Matriz de confusión en el conjunto de test para YOLOv8m-cls.

Finalmente, el modelo YOLOv8m es el que presenta la matriz de confusión más "limpia". En melanoma llega a detectar 8 de los 10 casos (recall de 0.8 en esta clase), y en otras categorías como DF, BKL o VASC también se alcanzan 7 u 8 aciertos sobre 10. Siguen apareciendo errores, especialmente en nevus, que se confunde con varias clases vecinas, pero globalmente la diagonal está más marcada que en los modelos anteriores.

Como resumen global del comportamiento en el conjunto de test, en la Tabla 3 comparamos la exactitud total (*test accuracy*) de cada modelo con la sensibilidad (*recall*) específica en la clase melanoma.

Table 3. Métricas finales en el conjunto de Test (Exactitud Global vs Sensibilidad en Melanoma).

Modelo	Test Accuracy (Global)	Recall Melanoma (Sensibilidad)
YOLOv8n (Nano)	66.25%	50% (5/10)
YOLOv8s (Small)	65.00%	60% (6/10)
YOLOv8m (Medium)	68.75%	80% (8/10)

5 Discusión

A partir de los resultados obtenidos podemos extraer varias ideas. En primer lugar, el uso de un subconjunto balanceado facilita la comparación entre clases y evita que el modelo optimice en exceso las categorías mayoritarias del dataset original. Sin embargo, el número total de imágenes sigue siendo reducido, lo que limita la capacidad de generalización y favorece que las métricas varíen bastante de una ejecución a otra.

Las curvas de entrenamiento muestran que los tres modelos aprenden rápido en las primeras épocas y tienden a estabilizarse después. El modelo *small* presenta una pérdida de validación más ruidosa y métricas inferiores a las esperadas. Planteamos la hipótesis de que esta arquitectura, al encontrarse en un punto

intermedio de complejidad (*tierra de nadie*), no se beneficia de la rápida convergencia de los modelos ligeros (como el *nano*) ni de la capacidad de memorización de los más profundos (como el *medium*). Es probable que el modelo *small* requiera un ajuste más fino de hiperparámetros, como una tasa de aprendizaje distinta o un mayor número de épocas, para estabilizar sus gradientes en este conjunto de datos específico. Por su parte, el modelo *nano* ofrece un rendimiento aceptable con muy pocos parámetros, mientras que el modelo *medium* aprovecha mejor su mayor capacidad y consigue las mejores precisiones tanto en validación como en el conjunto de test, tal y como se observa en la Tabla 3.

Analizando las matrices de confusión, las mayores confusiones se producen entre nevus (NV) y otras lesiones pigmentadas como melanoma (MEL) o queratosis benignas (BKL), algo razonable desde el punto de vista visual. En el mejor de los casos (YOLOv8m) se detectan 8 de los 10 melanomas, pero todavía persisten algunos falsos negativos que en un contexto clínico real serían especialmente preocupantes.

6 Conclusiones

En esta práctica hemos construido un sistema sencillo de clasificación de lesiones cutáneas utilizando la variante de clasificación de YOLOv8 sobre un subconjunto balanceado del dataset ISIC 2019. Hemos preparado el conjunto de datos, organizado la estructura de carpetas, entrenado tres modelos de distinta capacidad y evaluado su rendimiento en un conjunto de test independiente.

A partir de las curvas de entrenamiento, de las matrices de confusión y de las métricas finales en test, concluimos que los modelos más ligeros son capaces de alcanzar un rendimiento razonablemente alto con pocos datos, pero el modelo de tamaño medio (YOLOv8m) es el que mejor aprovecha la información disponible en este escenario. En concreto, es el que obtiene mayor exactitud global en el conjunto de test y la mayor sensibilidad en melanoma, lo que lo convierte en la opción más interesante desde el punto de vista práctico dentro de los tres evaluados.

Como trabajo futuro sería interesante ampliar el conjunto de entrenamiento, aplicar técnicas de aumento de datos más agresivas y explorar otras arquitecturas de clasificación o técnicas de *fine-tuning* más específicas para este dominio, con el objetivo de mejorar la detección de melanoma y reducir el número de falsos negativos.

7 Uso de inteligencia artificial

Para la realización de esta práctica hemos recurrido al uso de herramientas de inteligencia artificial como apoyo para conocer las librerías necesarias en Python para poder hacer frente al código.

UNIVERSIDAD
DE MÁLAGA