



Bachelor's degree in Computer Science and Engineering

Visión Artificial

2023-2024

Práctica 1

“CNNs para la clasificación de imágenes”

Ignacio Arnaiz Tierraseca – 100428997

Luis Daniel Casais Mezquida – 100429021

Profesor:

Fernando Díaz de María

1. Introducción

En la presente práctica se plantea la aplicación de los conocimientos adquiridos acerca de redes convolucionales y procesamiento de imágenes para el tratamiento de imágenes dermatológicas de lunares, a fin de distinguir y clasificar si estos se tratan de melanoma (tumor maligno), nevus (benigno) o keratosis seborreica (benigno). Durante la realización de la práctica se han desarrollado transformaciones para aplicar sobre el conjunto de imágenes dadas a clasificar, con el fin de mejorar el entrenamiento de la red permitiendo así extraer más características diferentes a dicha red. Durante la práctica también se ha desarrollado la arquitectura de la red CNN partiendo de la base original, una vez implementada la red y las transformaciones sobre las imágenes, se han llevado a cabo múltiples pruebas para detectar posibles mejoras y evaluar los resultados obtenidos.

2. Desarrollo

2.1. Generación del *dataset* de entrenamiento

Para la práctica se cuenta con un conjunto de datos basados en imágenes de lunares a fin de que la red diseñada sea capaz de otorgar una puntuación para cada categoría posible de lunar.

A fin de poder mejorar el entrenamiento de la red, se han llevado a cabo una serie de transformaciones sobre el conjunto de imágenes originales para así proveer al conjunto de entrenamiento de diferentes variaciones que pueden tener lugar:

- *Horizontal Flip*: Esta transformación con una probabilidad del 50 % volteará la imagen seleccionada sobre el eje horizontal, haciendo que quede a modo de espejo de la original.
- *Rotation*: Esta transformación rotará la imagen original con un valor aleatorio dentro del rango provisto tanto en positivo como en negativo, por ejemplo si recibe como parámetro 20, rotará la imagen con un valor aleatorio entre -20 y 20 grados.
- *Color Jitter*: El objetivo de esta transformación es poder modificar los parámetros cromáticos de una imagen dada, para se modificarán los parámetros de brillo, contraste, saturación y matriz. En la implementación actual, tras experimentar con diferentes valores para cada parámetro los mejores valores se obtuvieron teniendo un 0.28 para el brillo, un 0.3 para contraste y saturación, y un 0.06 para matriz, que aunque pueda resultar bajo, permite mantener los colores en términos generales entre tonos marrones o amarillos/rojos oscuros, ya que a valores más elevados comienzan a aparecer colores que no serían naturales para este tipo de lesiones como tonos verdes o azules.

- *Gaussian Blur*: Con esta transformación se busca aplicar una difuminación sobre la imagen original, a fin de reducir el ruido de la imagen y suavizar los detalles. Esta función toma como entrada el tamaño del *kernel*, que debe ser un número impar mayor o igual a tres.

En la implementación actual, tras llevar a cabo múltiples pruebas, se determinó que dicho *kernel* fuese un número impar comprendido entre el 3 y 50.

Tras analizar cada transformación por separado, y obtener los valores correctos para los parámetros, se implementó una transformación compuesta que aplicase sobre cada imagen todas las transformaciones mencionadas previamente, además de las transformaciones ya predefinidas, *CropByMask*, *Rescale* y *RandomCrop*, ambas a 224px, ya que es el tamaño de entrada de la CNN.

Finalmente se decidió no incluir la transformación de rotación de imágenes, ya que en las pruebas llevadas a cabo a pesar de realizar el recorte antes de la rotación, en gran cantidad de imágenes quedan bordes de color negro en las esquinas, ya que al rotar la imagen, parte de su información quedaba fuera del campo visible dejando las mencionadas esquinas negras, que provocaban peores resultados en el modelo.

2.2. Diseño de la red neuronal convolucional

Para resolver este problema se diseñó una arquitectura de CNN básica, usando filtros convolucionales con ReLUs de 3×3 (dado que es un buen equilibrio entre tamaño de filtro y número de operaciones, para generar el mismo *receptive field*) usando *same padding*, y aumentando el número de canales progresivamente hasta 128.

Entre capas convolucionales, se aplica *batch normalization* para mejorar y estabilizar el entrenamiento, y *max pooling* con filtros de 2×2 y *stride 2* para reducir la resolución a la vez que aumentamos los canales.

A la salida de la última capa convolucional, se realiza un *flatten* para la entrada del clasificador (*fully connected network*). Se aplica un *dropout* bastante agresivo en esta capa, que debido a nuestras pruebas proporcionó un mejor resultado, y luego usamos simples capas lineales hasta reducir al output de tamaño 3.

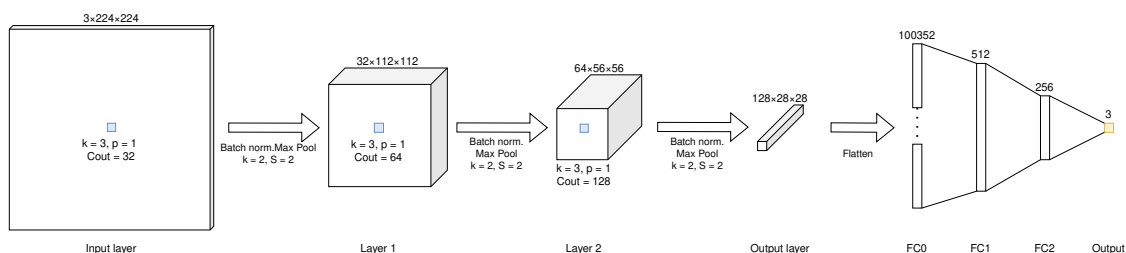


Figura 1: Arquitectura de la CNN propuesta

3. Conclusiones

El desarrollo de esta práctica nos ha parecido de gran utilidad para poder poner en uso los conocimientos teóricos obtenidos acerca del funcionamiento de las CNN y su arquitectura, así como el funcionamiento del tratamiento de imágenes y como la aplicación de diversas transformaciones tiene efecto en la precisión del modelo desarrollado.

Tras evaluar el modelo final, los resultados arrojados indicaban que el modelo se podría considerar un buen clasificador obteniendo un average de 0.734, con los siguientes parámetros detallados AUCs: mel 0.623333 sk 0.845459 avg 0.734396.

Durante el desarrollo, hemos encontrado ciertas dificultades principalmente a la hora de identificar que parámetros dentro de la arquitectura de la red son los que más afectan a su desempeño, así como el tiempo de cómputo, ya que especialmente en el caso de la evaluación final del dataset óptimo con maxSize=0, el tiempo de ejecución resultó bastante elevado.