Universidad Francisco de Vitoria

Escuela Politécnica Superior

Deep Learning

Práctica Grupal 1: Computer Vision

Profesor:
D° Moisés Martínez Muñoz

Alumnos:
Alfredo Robledano Abasolo
Jorge Barcenilla González
Rubén Sierra Serrano
Pedro García Silgo

Martes 1 de abril de 2025

Resumen

El presente trabajo consiste en desarrollar modelos de *Deep Learning* para la clasificación de fauna marina a partir de imágenes. Para ello, se utilizarán redes neuronales convolucionales profundas (*Convolutional Neural Networks*, CNNs), empleando un *dataset* de acuarios que contiene fotografías de diversas especies, tales como peces, medusas, pingüinos, tiburones, frailecillos, mantarrayas y estrellas de mar.

El objetivo principal es diseñar un *pipeline* que permita identificar y clasificar estas especies con alta precisión. Para lograrlo, se implementarán técnicas avanzadas como el *transfer learning*, *fine tuning* y *data augmentation*, optimizando el rendimiento del modelo mediante el ajuste de hiperparámetros y el uso de arquitecturas modernas. [1]

Además, se evaluará el desempeño del modelo mediante métricas estándar como la precisión, la sensibilidad y la especificidad.

Palabras Clave

Deep Learning, Redes Neuronales Convolucionales, *Transfer Learning*, clasificación

Índice

1. Desarrollo

1.1. Descripción del *dataset*

El *dataset* con el que estamos trabajando posee tres carpetas: una para los datos de *train*, otra para los datos de *test* y, por último, una tercera para los datos de *validation*. En estas carpetas se encuentran las imágenes y un documento en formato `.csv` llamado `annotations`, que contiene la información de las imágenes. La siguiente tabla describe los atributos del documento

Atributo	**Tipo de dato**	**Descripción del atributo**
filename	string	Nombre de la imagen
width	int	Anchura de la imagen
height	int	Altura de la imagen
class	string	Clase de la imagen
xmin	int	Coordenadas de la clase detectada en la imagen
ymin		
xmax		
ymax		

Figura 1: Atributos del *dataset*

En cuanto a la distribución de las imágenes, tenemos 448 imágenes de *train*, 63 imágenes de *test* y 127 imágenes de *validation*. Por tanto, estamos ante un *dataset* bastante limitado, teniendo en cuenta que estamos trabajando con imágenes y que nuestro objetivo es clasificarlas. Esto abre la puerta al uso de técnicas de *data augmentation*, así como al empleo de arquitecturas de red preentrenadas para aplicar *transfer learning*.

Además, existe un notable desbalanceo en la cantidad de imágenes representativas de cada clase, lo cual refuerza la necesidad de implementar una estrategia adecuada de *data augmentation* para mejorar la diversidad del conjunto de datos y mitigar los efectos negativos del desequilibrio en el entrenamiento del modelo.

1.2. Organización del *dataset*

Para facilitar el trabajo con el *dataset*, hemos automatizado su descarga, extracción y organización en carpetas según la clase dominante de cada imagen. Para ello, subimos el archivo `.zip` a Google Drive y desarrollamos un código para descargarlo y descomprimirlo.

Luego, con el archivo `annotations`, identificamos la clase dominante de cada imagen según su área y frecuencia en la imagen, almacenándolas en carpetas con su respectivo nombre.

1.3. Preprocesamiento

1.4. *Data Augmentation*

1.5. *Pre-Train*

1.6. *Fine-Tuning*

Podemos observar

2. Conclusiones

2.1. Fase 1: Preparación del entorno

Pese a que en esta sección no se ha realizado ningún modelo, es importante destacar que la preparación del entorno ha sido un proceso fundamental para el desarrollo de la práctica.

También es resulta interesante mencionar el proceso de creación de imágenes sintéticas que al final no se usaron. Con el proceso que realizó, se generó una imagen una imagen sintética para cada imagen que tuviera 2 tipos de animales o más en la misma imagen y que la segunda clase no fuera de fish. Siguiendo esta idea solo se generaron unas 24 imágenes nuevas, y debido a la poca aportación de imágenes de clases poco representadas, se decidió no incluirlas en el *dataset* final.

Asimismo, resulta relevante mencionar el proceso de generación de imágenes sintéticas, el cual finalmente no fue utilizado. Mediante este procedimiento se creó una imagen sintética para cada imagen que contenía dos o más tipos de animales en la misma toma, siempre que la segunda clase más predominante no correspondiese a *fish*.

El proceso se llevó a cabo seleccionando las imágenes candidatas y eliminando los píxeles correspondientes a la primera clase predominante. Esto implicaba reemplazar las áreas ocupadas por los animales de dicha clase con el color del fondo original. A partir de esta modificación, se generó una nueva imagen sintética en la que la clase predominante pasó a ser la segunda más representativa de la imagen original.

Siguiendo esta línea, se produjeron aproximadamente 24 imágenes nuevas; sin embargo, dada la escasa producción de imágenes de clases menos frecuentes, se optó por no incluirlas en el *dataset* final.

Como aprendizaje fuera de estas imágenes sintéticas, se ha comprobado la utilidad de usar pipelines de proceso de imágenes, para automatizar el proceso de preparación de datos para el entrenamiento de modelos.

2.2. Fase 2: Implementación en modo Pre-Train

En esta seccion se ha implementado una red neuronal convolucional (CNN) y se han ido generando diferentes modelos a partir de esta red. Se ha ido modificando la arquitectura de la red, Teniendo así los siguientes resultados:

Modelo	Capas	LR Factor	LR Patience	Val Acc	Test Acc
1	4	0.20	5	0.7480	0.8730
2	3	0.10	3	0.7795	0.8254
3	5	0.30	8	0.7717	0.7460

Figura 2: Comparación de Modelos

Mejor Modelo: #1

- **Capas Convolucionales:** 4
- **Factor de Reducción LR:** 0.2
- **Paciencia para Reducción LR:** 5
- **Mejor Época:** 18
- **Precisión de Validación:** 0.7480
- **Precisión de Prueba:** 0.8730

Como se observa existe una diferencia entre la precisión de validación y la de prueba. Hay que tener en cuenta sin embargo que la precisión de test (prueba) es tan alta porque las imagenes del train son muy parecidas a las de test, ya que el dataset está hecho con imágenes un mismo video. Eso puede significar que el modelo no generalize fuera de este video o similares.

Como aprendizaje de esta fase, la limitación de calidad de las imágenes no tiene porqué afectar de forma muy negativa al modelo, y por lo tanto es una prueba que puede ser útil realizar en muchos otros casos debido a su bajo coste computacional.

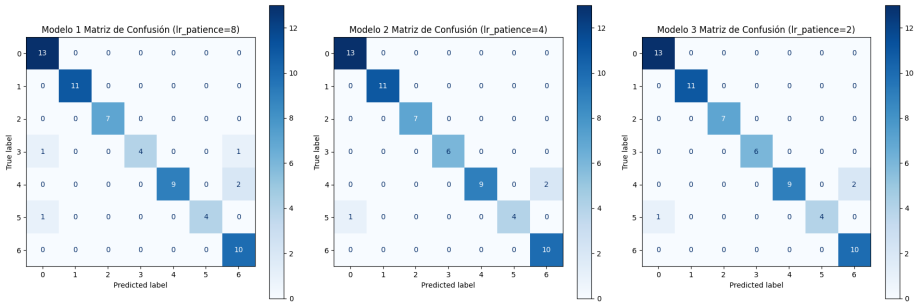

Figura 3: Matrices de confusión.

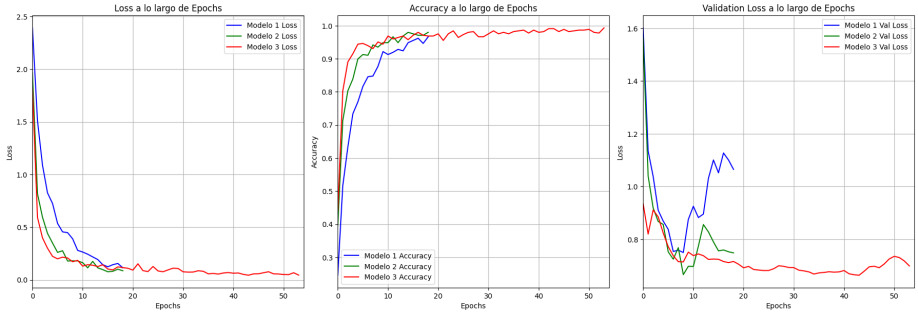

Figura 4: train_loss, accuracy y val_loss

2.3. Fase 3: Implementación en modo Fine-Tunning

[1]

Referencias

[1] Francois Chollet y François Chollet. *Deep learning with Python*. Simon y Schuster, 2021.