Reconocimiento de Patrones

Máster Universitario en Visión Artificial

Práctica 2 - Deep Learning

Autores:

Juan Montes Cano

Hanae Igalla El Youssfi

Enero 2024

1. Contexto del proyecto

En este proyecto de aprendizaje profundo, se desarrolla un clasificador de imágenes para reconocer y categorizar imágenes de cinco clases excluyentes: Bosque, Setas, Hierba, Hojas y Ensalada. Utilizando técnicas avanzadas de Deep Learning, la memoria detallará el proceso de construcción del modelo, preprocesamiento de datos hasta la evaluación del rendimiento.

Para la realización del proyecto se han utilizado únicamente las imágenes aportadas junto con la práctica, que cuenta con 593 fotos de Bosques, 599 de Setas, 563 de Hierba, 544 de Hojas y 546 de Ensaladas. Las fotos por lo general tienen dimensiones de 150x150 y son RGB, sin embargo, algunas fotos están corruptas, tienen otra dimensión o están en blanco y negro.

Para este proyecto se ha creado un repositorio, el cual se encuentra en el siguiente enlace: Repositorio Prácticas RP.

Para la ejecución del proyecto se puede hacer uso del fichero `README.md`. Se usa Docker como forma de ejecución debido a las dificultades para usar GPU entre distintos sistemas operativos, lo usamos tanto en inferencia como entrenamiento.

2. Soluciones propuestas

A lo largo de este proyecto se han desarrollado varias soluciones evaluando distintas arquitecturas, de las que hemos escogido dos, una que se apoya en una red preentrenada y otra que no cuenta con ningún tipo de entrenamiento anterior.

Antes de comenzar el proceso de clasificación se realiza un filtrado de las imágenes en el que las imágenes en escala de grises o corruptas se filtran.

El único tratamiento que se hace a la imagen antes de inferir su clase es la normalización. A continuación, se muestran las particularidades de cada red.

2.1. Uso de red preentrenada

Tras un análisis de las posibles soluciones se pensó que sería útil comenzar con una red que ya obtuviese un mapa de características útil, de manera que ahorramos tiempo en el entrenamiento y de antemano contamos con aquellas características visuales importantes de las imágenes. En esta solución, se ha usado la red VGG16, aunque existen otras candidatas como ResNet o Inception.

La elección de apoyarnos en la red VGG16 se basa fundamentalmente en la estrategia de transferencia de aprendizaje (transfer learning). El concepto detrás de la transferencia de aprendizaje radica en aprovechar los conocimientos previamente adquiridos por un modelo entrenado en un conjunto de datos grande y generalizado, como en el caso de VGG16 entrenado en ImageNet. En lugar de empezar desde cero, utilizamos los pesos preentrenados de VGG16 para inicializar las capas convolucionales. Esto proporciona al modelo una ventaja inicial al poseer características visuales aprendidas en tareas previas.

Al realizar transferencia de aprendizaje, permitimos que el modelo herede representaciones visuales útiles y complejas que son efectivas para una variedad de imágenes. Posteriormente, adaptamos estas representaciones a nuestra tarea específica de clasificación de imágenes de Bosque, Setas, Hierba, Hojas y Ensalada.

Dentro de esta propuesta de solución se evaluaron dos posibilidades: la adición de una capa convolucional a la salida de VGG16 y la aplicación directa de una operación de aplanado (flatten).

En el primer enfoque, al introducir una capa convolucional después de la salida de VGG16, se buscaba permitir la extracción adicional de características y patrones visuales más complejos. Esta capa convolucional podría capturar correlaciones espaciales y semánticas más específicas antes de pasar a las capas densas de clasificación.

En el segundo enfoque, la aplicación directa de una operación de aplanado después de VGG16 se consideró para simplificar la representación de las características, transformándolas en un vector unidimensional antes de entrar en las capas densas de clasificación.

Hasta el momento, no se ha observado una diferencia significativa en los resultados al elegir una configuración sobre la otra, por lo que hemos optado por el segundo enfoque.

Se probó con otros *backbones*, como ResNet (obteniendo resultados peores) e Inception. Inception ofreció los mejores resultados. Sin embargo, en la capa de Flatten intentamos sustituirla por un GlobalAveragePooling y se obtuvieron buenos resultados y un menor tiempo para entrenar la red. Finalmente, optamos por tomar Inception como *backbone* utilizando la red entrenada con el dataset de ImageNet.

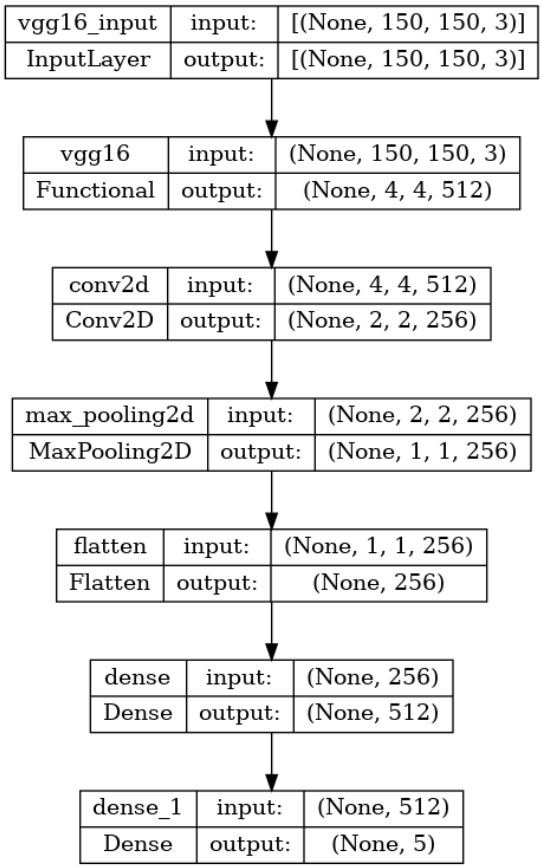

Figura 1: Arquitectura con VGG16 usando capa convolucional antes del Flatten

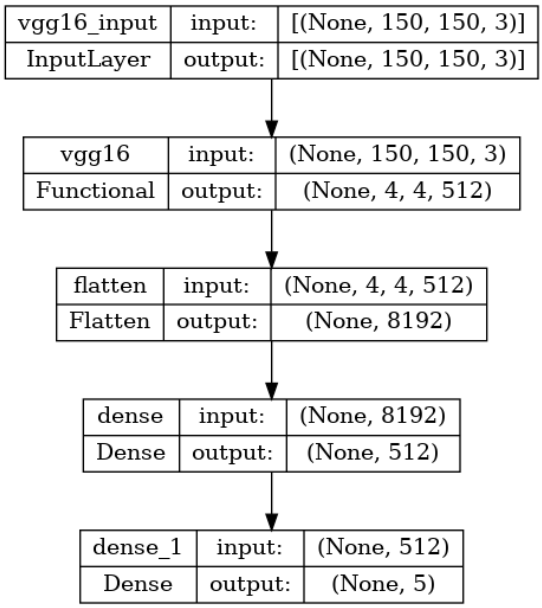

Figura 2: Arquitectura con VGG16 y ninguna capa convolucional extra

2.2. Uso de una red sin entrenamiento previo

La segunda propuesta de solución prescinde del uso de una red preentrenada, optando en su lugar por construir un modelo completamente basado en capas convolucionales desde cero. En este enfoque, las capas convolucionales se diseñan para aprender representaciones únicas y específicas del conjunto de datos de Bosque, Setas, Hierba, Hojas y Ensalada, sin depender de conocimientos previos de una tarea más general.

Este modelo personalizado basado en capas convolucionales permite una mayor flexibilidad en el diseño de la arquitectura, lo que facilita la adaptación a las características particulares del conjunto de datos objetivo.

La ausencia de transferencia de aprendizaje significa que el modelo se entrena desde cero, lo que puede resultar beneficioso cuando el conjunto de datos es lo suficientemente grande y específico como para permitir que el modelo aprenda patrones de manera autónoma.

Sin embargo, es importante tener en cuenta que este enfoque también puede requerir un conjunto de datos significativo para evitar el sobreajuste y lograr un rendimiento competitivo. Además, el tiempo de entrenamiento puede ser más extenso en comparación con la transferencia de aprendizaje, donde se inicia con pesos preentrenados.

La arquitectura que hemos seguido para la construcción de esta red es muy simple, constituida por varias capas convolucionales con una activación ReLU. En la siguiente figura

se muestra un diagrama con su arquitectura.

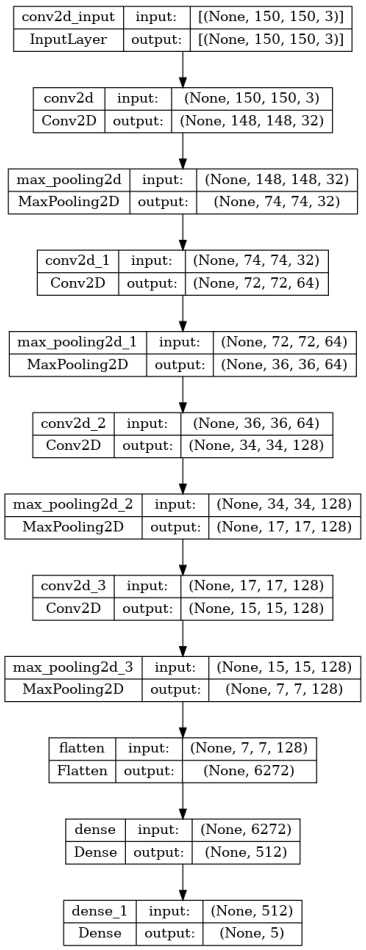

Figura 3: Arquitectura de la red usando capas convolucionales

Los resultados obtenidos con esta red son significativamente inferiores a los obtenidos usando transferencia de aprendizaje, sin embargo, es una red que únicamente se apoya en el entrenamiento con los datos del problema y que potencialmente se ajusta mejor al proyecto. El motivo por el que se incorpora esta solución es mostrar de una red completamente entrenada con los datos aportados por el enunciado.

Tabla 1: Resultados de Accuracy con diferentes arquitecturas de CNN.

Arquitectura	**Precisión (%)**
Sin Backbone	88.2
VGG	93.4
ResNet	76.7
InceptionV3	96.6

En cuanto a los resultados enviados a la competición, en el primer envio se uso la red sin *backbone* con 408 aciertos, y en el tercero se usó la red con Inception, logrando 454 aciertos.

3. Conclusión

El uso del transfer learning, como se evidencia en el caso de la red basada en Inception, ha demostrado ser una estrategia efectiva para mejorar el rendimiento de modelos de clasificación. Aprovechar el conocimiento preexistente de una red preentrenada proporciona beneficios como la rápida convergencia. Considerar la posibilidad de usar la red preentrenada con otro dataset para evaluar su comportamiento podría ofrecer una visión más completa de su desempeño, incluso se puede conjeturar una mejora de rendimiento utilizando distintos datasets.Los resultados son satisfactorios y desempeñan correctamente y con una alta tasa de acierto la clasificación.