

Módulo 2. Implementación de un modelo de Deep Learning

Sobre el dataset

El dataset a utilizar fue obtenido de Kaggle (<https://www.kaggle.com/datasets/imspars/flowers-dataset>). Este se trata de un dataset de imágenes en el que se clasifican 5 tipos diferentes de flores. Para esta actividad se tomaron solo 4, siendo estas daisy, dandelion, sunflower y tulip.

El dataset se encuentra dividido en tres carpetas, la de train, la de test y la de validation.

- La carpeta de train contiene 1514 imágenes. De estas 357 son de daisies, 397 son dandelions, 374 sunflowers y 386 tulips.
- En cuanto a la carpeta de validation, esta cuenta con 108 imágenes de daisies, 101 imágenes de dandelions, 108 de sunflowers y 109 de tulips. Sumando 424 imágenes.
- Por último, en la carpeta de test se encuentran 551 imágenes, las cuales se dividen en 148 daisies, 134 dandelions, 138 sunflowers y 136 tulips.

Objetivo: generar un modelo que permita clasificar imágenes de flores en las 4 clases ya mencionadas anteriormente.

Código

Este fue iniciado con una exploración de los datos, además de que se agregaron algunas imágenes sumándose a las obtenidas de Kaggle para que el dataset pudiera estar mejor balanceado.

Primer modelo

Lo primero que se realiza es un preprocesado en el que se cambia el tamaño de la imagen.

El primer modelo realizado fue una red neuronal convolucional sencilla con la siguiente arquitectura:

- Capa convolucional con 8 filtros, tamaño de kernel de 3x3, sin padding y una función de activación 'relu'. Además de declarar el shape del input como (150, 150, 3) debido al tamaño de las imágenes y de que se trata de imágenes a color.
- Capa de MaxPooling
- Flatten layer
- Capa densa de 64 unidades y función de activación relu
- Capa densa de 64 unidades y función de activación relu
- Capa densa de 4 unidades (ya que tenemos 4 clases) y función de activación softmax

Este modelo se entrenó con 20 épocas, cada una conteniendo 10 steps.

Segundo modelo

Para iniciar se aplicó data augmentation para los datos de entrenamiento a través del escalamiento, rotación, width shift y height shift de las imágenes.

Después, se buscó hacer uso de transfer learning con VGG16. Esto es útil debido a que se está utilizando como base un modelo previamente entrenado. Teniendo la siguiente arquitectura:

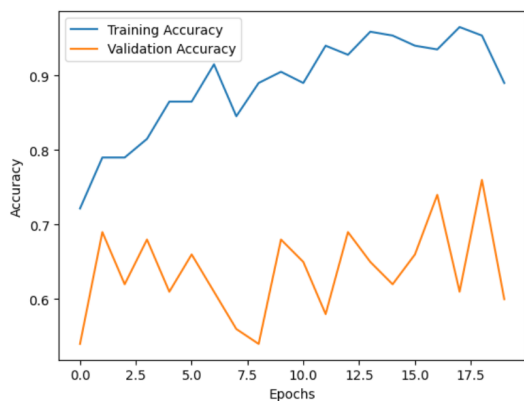
- Base VGG16 pre entrenada con ImageNet
- Flatten layer
- Capa densa de 64 unidades y función de activación relu
- Dropout layer
- Capa densa de 4 unidades (ya que tenemos 4 clases) y función de activación softmax}

Este modelo se entrenó con 20 épocas, cada una conteniendo 47 steps (por la división de $\text{train_generator.samples} / \text{train_generator.batch_size}$).

Evaluación comparativa

Para la evaluación de cada modelo se evaluó su precisión (accuracy) para train, validation y test, obteniendo los siguientes resultados.

Red Neuronal Convolucional

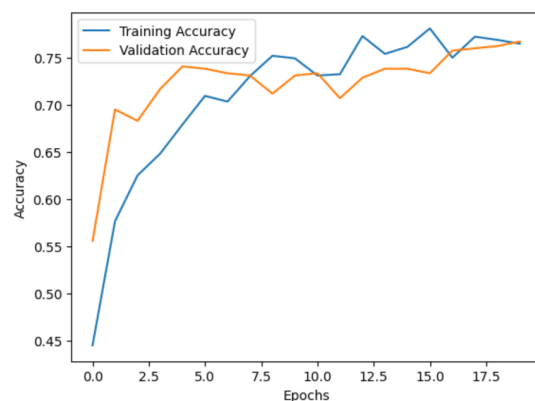


Al evaluar la precisión tanto en train como en validation se observa que, a pesar de que parece que está aprendiendo bastante bien por la precisión que se ve en el train. A la hora de aplicar la validation se nota que en realidad no es así.

Test accuracy = 64.5%

Ya que se hicieron las pruebas, donde se tomaron 10 steps, con el dataset de test, se puede observar que no se tiene una muy buena precisión.

Transfer learning

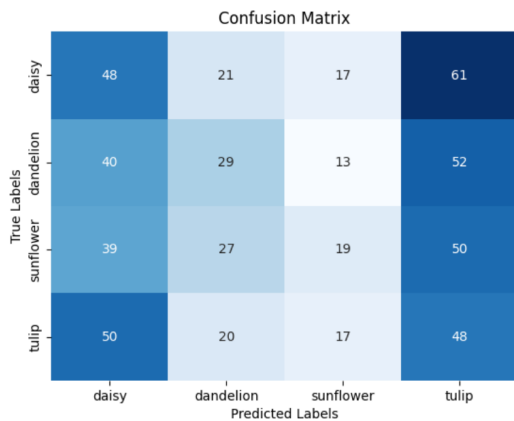


Para este segundo modelo se observa que tanto la precisión del train, como la de validation van incrementando conforme se aumentan las épocas.

Test accuracy = 82.2%

Al hacer de nuevo las pruebas con el dataset de test se aprecia una precisión mayor a la del modelo anterior.

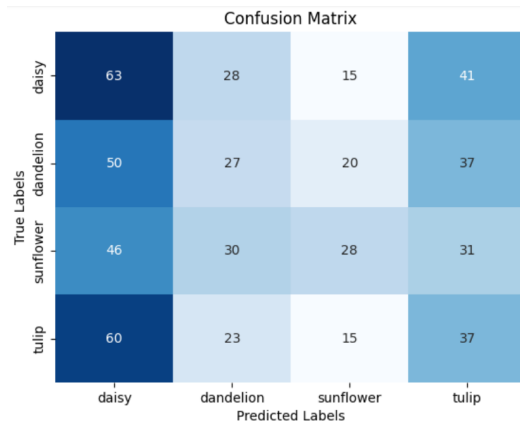
Matriz de confusión



Al observar la matriz de confusión se puede observar que el modelo tiende a predecir más tulipanes e incluso margaritas. Por lo que estas dos clases son las que mejor desempeño tienen cuando se trata de identificar correctamente dicha flor, sin embargo también identifica demasiados dientes de león y girasoles como tulipanes.

Clasificando 144 de las imágenes de prueba correctamente.

Matriz de confusión



A pesar de la mejora en la presión de este modelo en cuanto a las pruebas, se puede observar que sigue teniendo un sesgo hacía las margaritas.

Clasificando 155 de las imágenes de prueba correctamente.

En conclusión, al utilizar transfer learning se presenta una ligera mejoría, aunque aún se podría seguir modificando parámetros para mejorar su precisión.