

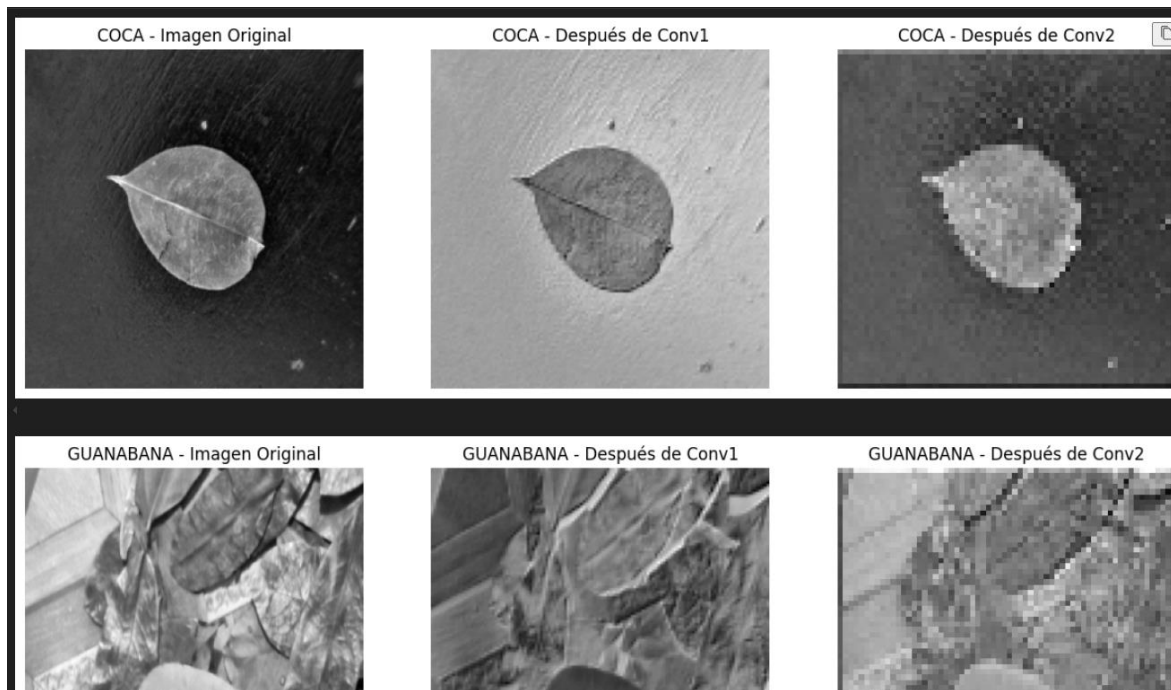
Informe Redes Neuronales Convolucionales – SIS421

Nombre: Arancibia Aguilar Daniel Andree

Carrera: Ingeniería en Ciencias de la Computación

Red Neuronal Convolutacional

En esta red aplicamos distintos filtros (kernel) en las convoluciones para que el modelo pueda entender mejor las imágenes y poder clasificarlas mejor que un MLP



```
100%|██████████| 125/125 [07:50<00:00, 3.76s/it]
Epoch [1/5] Loss: 1.1107 Acc: 0.7037 Val Loss: 0.3161 Val Acc: 0.8935
100%|██████████| 125/125 [08:13<00:00, 3.95s/it]
Epoch [2/5] Loss: 0.2971 Acc: 0.9018 Val Loss: 0.2290 Val Acc: 0.9290
100%|██████████| 125/125 [08:18<00:00, 3.99s/it]
Epoch [3/5] Loss: 0.1807 Acc: 0.9416 Val Loss: 0.1766 Val Acc: 0.9405
100%|██████████| 125/125 [06:32<00:00, 3.14s/it]
Epoch [4/5] Loss: 0.1062 Acc: 0.9621 Val Loss: 0.1561 Val Acc: 0.9535
100%|██████████| 125/125 [06:10<00:00, 2.97s/it]
Epoch [5/5] Loss: 0.0596 Acc: 0.9790 Val Loss: 0.1598 Val Acc: 0.9590
```

```
✓ 39.2s
Precisión del modelo en el conjunto de prueba: 95.90%
```

En el Laboratorio 2, se obtuvo una precisión del 89% utilizando un MLP después de 1000 epoch de entrenamiento. En cambio, al utilizar una red neuronal convolutacional, se logró una precisión del 95% en tan solo 5 epoch

Modelo Preentrenado – DeiT Small

En el Deep learning hay muchos modelos preentrenados que nos sirven para diferentes cosas, en mi caso elegí este modelo ya que fue hecho para la clasificación de imágenes con CNN

El modelo que elgi es DeiT Small (Data-efficient Image Transformer), que fue entrenado con ImageNet-1k, a una resolución de 224x224 píxeles

Tiene una arquitectura de 12 capas de transformadores el cual contiene 22 millones de parámetros haciendolo ligero en comparación con otros modelos

Luego de hacer la exploración de datos se hizo un entrenamiento con el modelo pero desde 0, es decir sin usar sus pesos y con las capas descongeladas

Entrenamiento desde 0

```
e:\LAB3SIS421\env\Lib\site-packages\timm\models\vision_trans
x = F.scaled_dot_product_attention(
Epoch [1/15], Loss: 1.7245, Accuracy: 0.2224
Epoch [2/15], Loss: 1.3471, Accuracy: 0.3977
Epoch [3/15], Loss: 1.0457, Accuracy: 0.5379
Epoch [4/15], Loss: 1.0292, Accuracy: 0.5509
Epoch [5/15], Loss: 0.8264, Accuracy: 0.6605
Epoch [6/15], Loss: 0.7447, Accuracy: 0.7026
Epoch [7/15], Loss: 0.6278, Accuracy: 0.7582
Epoch [8/15], Loss: 0.5377, Accuracy: 0.7939
Epoch [9/15], Loss: 0.4430, Accuracy: 0.8376
Epoch [10/15], Loss: 0.3951, Accuracy: 0.8518
Epoch [11/15], Loss: 0.3903, Accuracy: 0.8531
Epoch [12/15], Loss: 0.4148, Accuracy: 0.8424
Epoch [13/15], Loss: 0.4346, Accuracy: 0.8297
Epoch [14/15], Loss: 0.4061, Accuracy: 0.8440
Epoch [15/15], Loss: 0.4172, Accuracy: 0.8354
```

Se realizó un entrenamiento desde cero con DeiT Small, es decir, sin utilizar los pesos preentrenados y manteniendo las capas descongeladas, con esto se pudo ver una pérdida inicial considerable y una precisión máxima de 83% después de 15 épocas, con un tiempo de entrenamiento significativamente mayor.

Transfer Learning

```
Epoch [1/15], Loss: 0.4449, Accuracy: 0.8835
Epoch [2/15], Loss: 0.1518, Accuracy: 0.9706
Epoch [3/15], Loss: 0.0995, Accuracy: 0.9834
Epoch [4/15], Loss: 0.0755, Accuracy: 0.9880
Epoch [5/15], Loss: 0.0607, Accuracy: 0.9896
Epoch [6/15], Loss: 0.0503, Accuracy: 0.9924
Epoch [7/15], Loss: 0.0431, Accuracy: 0.9936
Epoch [8/15], Loss: 0.0375, Accuracy: 0.9948
Epoch [9/15], Loss: 0.0332, Accuracy: 0.9959
Epoch [10/15], Loss: 0.0297, Accuracy: 0.9962
Epoch [11/15], Loss: 0.0265, Accuracy: 0.9964
Epoch [12/15], Loss: 0.0240, Accuracy: 0.9972
Epoch [13/15], Loss: 0.0220, Accuracy: 0.9978
Epoch [14/15], Loss: 0.0199, Accuracy: 0.9982
Epoch [15/15], Loss: 0.0182, Accuracy: 0.9991
```

A diferencia del entrenamiento anterior, con transfer learning

El Transfer Learning utiliza pesos preentrenados para mejorar la eficiencia y precisión del modelo. Al congelar las capas excepto la capa de salida, se puede lograr una precisión del 88% desde el primer epoch, con una mejora notable en la velocidad de entrenamiento y una precisión final del 99% después de completar el entrenamiento

Fine Tuning

```
fit(model_c, train_loader, epochs=15)
2]
Epoch [1/15], Loss: 1.0751, Accuracy: 0.5436
Epoch [2/15], Loss: 0.3762, Accuracy: 0.8571
Epoch [3/15], Loss: 0.1724, Accuracy: 0.9389
Epoch [4/15], Loss: 0.0826, Accuracy: 0.9710
Epoch [5/15], Loss: 0.0842, Accuracy: 0.9732
Epoch [6/15], Loss: 0.0819, Accuracy: 0.9732
Epoch [7/15], Loss: 0.0634, Accuracy: 0.9779
Epoch [8/15], Loss: 0.0329, Accuracy: 0.9895
Epoch [9/15], Loss: 0.0264, Accuracy: 0.9914
Epoch [10/15], Loss: 0.0542, Accuracy: 0.9809
Epoch [11/15], Loss: 0.0501, Accuracy: 0.9841
Epoch [12/15], Loss: 0.0288, Accuracy: 0.9904
Epoch [13/15], Loss: 0.0087, Accuracy: 0.9970
Epoch [14/15], Loss: 0.0739, Accuracy: 0.9760
Epoch [15/15], Loss: 0.0344, Accuracy: 0.9891
```

El Fine Tuning implica el uso de pesos preentrenados sin congelar las capas, permitiendo que todas las capas sean ajustadas durante el entrenamiento. Aunque inicialmente la precisión es baja, el modelo mejora exponencialmente en 15 epoch, aunque el Fine Tuning puede lograr una alta precisión y una menor pérdida, el proceso es más lento en comparación con el Transfer Learning.

Conclusiones

- **Redes Neuronales Convolucionales (CNN):** Superan significativamente a los MLP en términos de precisión para la clasificación de imágenes, el uso de CNN permite obtener resultados de alta calidad en menos tiempo, al requerir menos epoch para alcanzar altas precisiones.
- **Modelos Pre entrenados:** Facilitan la tarea de ajuste del modelo al proporcionar una base sólida de pesos pre entrenados, reduciendo el esfuerzo necesario para encontrar los hiper parámetros óptimos
- **Transfer Learning:** Ofrece una combinación efectiva de alta precisión, menor pérdida y velocidad de entrenamiento al utilizar pesos pre entrenados con capas congeladas, resultando en una mayor eficiencia en comparación con el entrenamiento desde cero
- **Fine Tuning:** Proporciona mejoras adicionales en precisión y reducción de pérdida al ajustar todas las capas del modelo, sin embargo, este enfoque es más lento y puede ser menos eficiente en comparación con el Transfer Learning

En resumen, las CNN ofrecen una ventaja significativa en la clasificación de imágenes sobre los MLP, y la utilización de modelos preentrenados con Transfer Learning o Fine Tuning puede mejorar considerablemente la precisión y la eficiencia del entrenamiento.