Practica 6

Especificar dependiendo de las características que llevan vistas hasta el momento de su conjunto de datos, cual es el tipo de modelo que van a utilizar, explicando cuales son los detalles que les hicieron pensar que ese era el mejor modelo.

El conjunto de datos es un grupo de fotografías tomadas en diferentes zonas de Estados Unidos, estas zonas son boscosas en su mayoría, y las imágenes estan etiquetadas con “Fire” y “Non fire”, según su clasificación.

La intención es generar un modelo de clasificación y posteriormente lograr una segmentación de las imágenes que nos permita localizar y focalizar la zona exacta de la imagen donde esto se encuentra.

Puntos clave:

- Modelo de Clasificación

  Se hace uso de Redes neuronales convolucionales binarias para la clasificación de ambos grupos. Para esto las herramientas seleccionadas son Pytorch y Albumentation.

  Inicialmente se definen funciones para descargar un conjunto de datos archivado y descomprimirlo, para poder posteriormente realizar la normalización del conjunto. Luego se dividen los archivos del conjunto de datos en conjuntos de entrenamiento y validación.

  Aquí es donde se hace uso de las herramientas más importantes para la clasificación, pues el conjunto de entrenamiento, se utiliza para definir un dataset de clase Pytorch, y además se hace uso de albumentaciones para definir funciones de transformación para los conjuntos de datos de entrenamiento y validación.

  Se definen los siguientes parámetros, y se crean todos los objetos y funciones necesarios para entrenamiento y validación.

```
params = {
    "model": "resnet50",
    "device": "cuda",
    "lr": 0.001,
    "batch_size": 64,
    "num_workers": 4,
    "epochs": 10,
```

  Se ejecuta el entrenamiento del modelo, con los parámetros definidos previamente:

```
model = getattr(models, params["model"])(pretrained=False,
num_classes=1,)

model = model.to(params["device"])
```

```
criterion = nn.BCEWithLogitsLoss().to(params["device"])
optimizer = torch.optim.Adam(model.parameters(), lr=params["lr"])
```

Logrando los siguiente resultados:

```
Epoch: 1. Train.      Loss: 1.266 | Accuracy: 0.616: 100%|██████████| 7/7 [00:23<00:00,  3.30s/it]
0it [00:00, ?it/s]
Epoch: 2. Train.      Loss: 0.450 | Accuracy: 0.796: 100%|██████████| 7/7 [00:12<00:00,  1.73s/it]
0it [00:00, ?it/s]
Epoch: 3. Train.      Loss: 0.327 | Accuracy: 0.866: 100%|██████████| 7/7 [00:10<00:00,  1.47s/it]
0it [00:00, ?it/s]
Epoch: 4. Train.      Loss: 0.340 | Accuracy: 0.831: 100%|██████████| 7/7 [00:11<00:00,  1.62s/it]
0it [00:00, ?it/s]
Epoch: 5. Train.      Loss: 0.271 | Accuracy: 0.865: 100%|██████████| 7/7 [00:12<00:00,  1.79s/it]
0it [00:00, ?it/s]
Epoch: 6. Train.      Loss: 0.263 | Accuracy: 0.887: 100%|██████████| 7/7 [00:12<00:00,  1.78s/it]
0it [00:00, ?it/s]
Epoch: 7. Train.      Loss: 0.352 | Accuracy: 0.838: 100%|██████████| 7/7 [00:11<00:00,  1.71s/it]
0it [00:00, ?it/s]
Epoch: 8. Train.      Loss: 0.299 | Accuracy: 0.847: 100%|██████████| 7/7 [00:10<00:00,  1.44s/it]
0it [00:00, ?it/s]
Epoch: 9. Train.      Loss: 0.232 | Accuracy: 0.896: 100%|██████████| 7/7 [00:12<00:00,  1.77s/it]
0it [00:00, ?it/s]
Epoch: 10. Train.      Loss: 0.236 | Accuracy: 0.908: 100%|██████████| 7/7 [00:12<00:00,  1.82s/it]
0it [00:00, ?it/s]
```

Se cumple el objetivo de una clasificación eficiente, por lo tanto se asume la selección de una herramienta adecuada para este paso.

- Segmentación

El dataframe creado con el id, el path y etiqueta de las imágenes fue de ayuda para extraer un código de segmentación en valores binarios similar a este formato: “0 0 0 1 0 34 0 35 0 39 0 40 0 41 0 4”. Este es usado para mostrar las imágenes con mascarás y posteriormente generar un dataframe con estos resultados obtenidos de esta transformación.

Se hizo uso de las siguientes librerías para la creación de la función que transformaría las imágenes y extraería las máscaras.

```
import cv2
import os
import matplotlib.pyplot as plt
import random
from skimage import color
```

Posteriormente, en el trabajo futuro se busca realizar una segmentación con U-Net que permita focalizar la zona exacta de fuego en la imagen.