



CONECTANDO CADA GRANO CON SU HISTORIA Y SU FUTURO

Clasificación de Granos de Café con Deep Learning

Este proyecto tiene como objetivo la clasificación de imágenes del "conjunto de datos de granos de café ajustados en tamaño". Se trata de un proyecto de visión computacional que categoriza los granos de café según sus colores, utilizando 4 categorías: Oscuro, Verde, Claro y Medio. Los datos ya se encuentran organizados en carpetas destinadas a entrenamiento y prueba. En el proceso, se emplean técnicas de visualización con frecuencia, ya que es crucial comprender bien los datos antes de avanzar con el modelado.

Parfraseado de los Datos:

El archivo CSV que se utiliza contiene cuatro columnas de datos:

- La columna *filepaths* especifica la ruta relativa hacia el archivo de imagen.
- La columna *labels* incluye la etiqueta de texto correspondiente a cada archivo de imagen.
- La columna *data set* indica el conjunto de datos al que pertenece la imagen (entrenamiento o prueba).
- La columna *class index* asigna un número entero a cada clase de imagen, siendo:
 - "Oscuro" con el índice 0.
 - "Verde" con el índice 1.
 - "Claro" con el índice 2.
 - "Medio" con el índice 3.



CONECTANDO CADA GRANO CON SU HISTORIA Y SU FUTURO

[1] El código descarga el dataset de granos de café

```
import os

#walk through the directory and list the numbers of files
for dirpath, dirnames, filenames in os.walk("/kaggle/input/coffee-bean-dataset-resized-224-x-224"):
    print(f"There are {len(dirnames)} directories and {len(filenames)} images in {dirpath}")

print('Data source import complete.')
```

[2] El código descarga el dataset de granos de café

```
num_of_coffee_bean_classes = len(os.listdir("/kaggle/input/coffee-bean-dataset-resized-224-x-224/train"))
num_of_coffee_bean_classes
```

[4]

```
import pathlib
import numpy as np

data_dir = pathlib.Path("/kaggle/input/coffee-bean-dataset-resized-224-x-224/train")
class_names = np.array(sorted([item.name for item in data_dir.glob("*")])) # creating a list of class names from subdirectory
print(class_names)
```

['Dark' 'Green' 'Light' 'Medium']

[5] Creemos una función para visualizar imágenes aleatoriamente

```
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import random

def view_random_image(target_dir, target_class):
    # setting up the image directory
    target_folder = target_dir + target_class

    #get a random image path
    random_image = random.sample(os.listdir(target_folder), 1)

    #read image and plotting it
    img = mpimg.imread(target_folder + "/" + random_image[0] )
    plt.imshow(img)
```



CONECTANDO CADA GRANO CON SU HISTORIA Y SU FUTURO

```
plt.title(target_class)
plt.axis("off")

print(f"Image shape: {img.shape}")

return img
```

