Práctica 3

DeepFace - Detector de sentimientos

Profesor:
David Campoy Miñarro

¿Qué es Deepface?

Deepface es una tecnología de reconocimiento facial y análisis de atributos faciales (edad, género, emoción y raza) para Python.

Utiliza modelos ya entrenados de: VGG-Face, Google FaceNet, OpenFace, Facebook DeepFace, entre otros.

🔗 https://github.com/serengil/deepface

Paso 1. Instalación e importar librería

```
#!pip install deepface
from deepface import DeepFace
import matplotlib.pyplot as plt
import pandas as pd
```

- Clase DeepFace desde la librería deepface. Permite acceder a las funcionalidades y métodos proporcionados por DeepFace.
- matplotlib.pyplot bajo el alias plt. Esta librería para realizar y visualizar datos en Python.
- Pandas es una librería muy utilizada para analizar datos.

Paso 2. Modelos de reconocimiento facial

```
backends = ["opencv","ssd","dlib","mtcnn","retinaface","mediapipe"]
```

- **SSD (Single Shot MultiBox Detector):** Es un modelo de detección de objetos en tiempo real que puede ser usado para identificar y localizar múltiples objetos en imágenes.
- **Dlib:** Una librería que incluye algoritmos para detección facial.
- **MTCNN (Multi-Task Cascaded Convolutional Networks):** Es un algoritmo de detección facial que utiliza redes neuronales convolucionales.
- **RetinaFace:** Un detector de rostros que utiliza una red neuronal para localizar múltiples rostros en una imagen y proporcionar detalles precisos sobre las posiciones y atributos faciales.
- **MediaPipe:** Un marco de Google para aplicaciones de percepción de medios que incluye soluciones pre-entrenadas para la detección facial, seguimiento de manos, detección de pose, entre otros.

Paso 3. Identificamos los rostros

Vamos a comparar los diferentes modelos para identificar los rostros. Prueba con diferentes imágenes. Compara los resultados.

```
foto1 = "antonio-banderas.jpg"
foto2 = "michael-keaton.jpg"
foto3 = "bruce-willis.png"
model_name = "ArcFace"

fig, axs = plt.subplots(3,2, figsize=(15,10))
axs = axs.flatten()
for i, b in enumerate(backends):
  try:
    face = DeepFace.detectFace(foto1, target_size=(224,224), detector_backend=b)
    axs[i].imshow(face)
    axs[i].set_title(b)
    axs[i].axis("off")
  except:
    pass
plt.show()
```

Paso 3. Comparamos dos rostros

Para comparar dos rostros vamos a utilizar el método: *verify()*
Compara cada uno de los modelos de entrenamiento, ¿cuál es el mejor?

```
foto1 = "antonio-banderas.jpg"
foto2 = "michael-keaton.jpg"
foto3 = "bruce-willis.png"
model_name = "ArcFace"

models = ["VGG-Face","Facenet","Facenet512","OpenFace","DeepFace","DeepID","ArcFace","Dlib","SFace"]

result = DeepFace.verify(img1_path=foto1, img2_path=foto2, model_name=models[1],)

print(result)
```

Paso 4. Lo mostramos visualmente

```
fig, axs = plt.subplots(1, 2, figsize=(15, 5))
axs[0].imshow(plt.imread(foto1))
axs[1].imshow(plt.imread(foto2))
fig.suptitle(f'Parecido: {result["verified"]} - Distancia {result["distance"]:0}')
plt.show()
```

Paso 5. Probamos con distintos modelos

```
models = ["VGG-Face","Facenet","Facenet512","OpenFace","DeepFace","DeepID","ArcFace","Dlib","SFace"]

for model_name in models:
    result = DeepFace.verify(img1_path=foto1, img2_path=foto2, model_name=model_name)

    fig, axs = plt.subplots(1, 2, figsize=(15, 5))
    axs[0].imshow(plt.imread(foto1))
    axs[1].imshow(plt.imread(foto2))
    fig.suptitle(f'Modelo: {model_name} - Parecido: {result["verified"]} - Distancia: {result["distance"]:0.4f}')
    plt.show()
```

¿Qué modelo es el mejor comparando rostros?

Paso 6. Reconocer rostros

Imagina que tenemos un directorio con miles de fotografías de personas, y queremos identificar las fotos en las que aparece una persona.

```
result = DeepFace.find(img_path=foto1, db_path="fotos/" )
print(result)
```

```
[                      identity  source_x  source_y  source_w  source_h  \
0  fotos//antonio-banderas.jpg       204       426       927       927
1    fotos//michael-keaton.jpg       204       426       927       927
2      fotos//bruce-willis.png       204       426       927       927

   VGG-Face_cosine
0     6.661338e-16
1     3.535842e-01
2     3.595549e-01  ]
```

Hemos buscando a Antonio Banderas en el directorio con las imágenes.

Paso 7. Análisis de atributos faciales

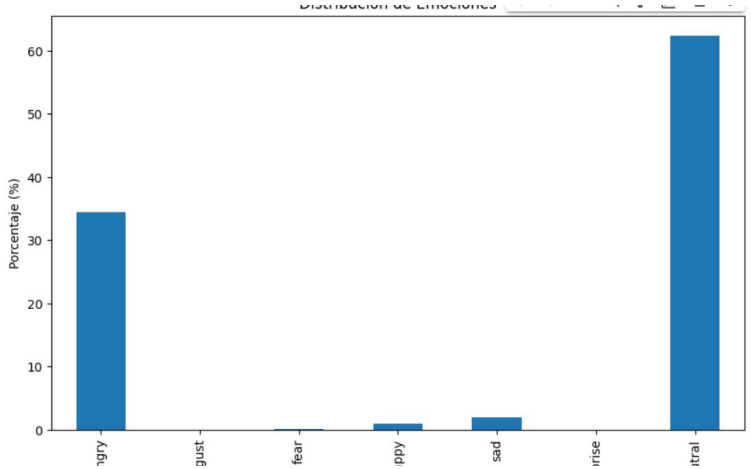
Con analyze podemos saber las características del rostro.
¿Prueba con diferentes rostros?

```
result = DeepFace.analyze(img_path=foto1)
print(result)
```

Paso 8. Emociones

Con analyze podemos saber las características del rostro.
¿Prueba con diferentes rostros?

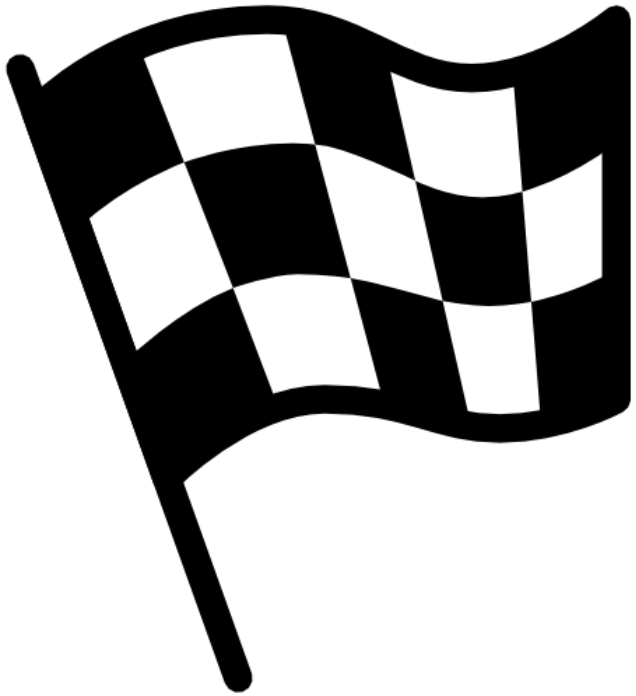
```
result = DeepFace.analyze(img_path=foto1)
print(result)
```




```
result = DeepFace.analyze(img_path=foto2)
emotions_data = result[0]["emotion"]

df_emotions = pd.DataFrame(emotions_data, index=[0])

df_emotions.T.plot(kind='bar', figsize=(10, 6), legend=None)
plt.title('Distribución de Emociones')
plt.xlabel('Emoción')
plt.ylabel('Porcentaje (%)')
plt.show()
```



“Lo que todos tenemos que hacer es asegurarnos de que estamos usando la IA de una manera que sea en beneficio de la humanidad, no en detrimento de la humanidad”

Tim Cook, actual director ejecutivo de Apple Inc