

DeepFace on CCTV

Manual de usuario

- **NOMBRE Y APELLIDOS:** Ruth Torres Gallego
- **CORREO ELECTRÓNICO:** rutorresg@alumnos.unex.es

ÍNDICE

| | |
|---|----------|
| 1. Propósito..... | 3 |
| 2. Licencia..... | 3 |
| 3. Instalaciones previas..... | 4 |
| 4. Guía..... | 5 |
| 4.1. Organización del repositorio..... | 5 |
| 4.1.1. Archivo main.py..... | 5 |
| 4.1.2. Archivo Camera_CCTV.py..... | 5 |
| 4.1.3. Archivo AnalyzeAllImages.py..... | 5 |
| 4.1.4. Carpeta functionsSeparated..... | 5 |
| 4.1.5. Carpeta haarcascades..... | 5 |
| 4.1.6. Carpeta images..... | 5 |
| 4.1.7. Carpeta info..... | 5 |
| 4.1.8. Carpeta captures..... | 5 |
| 4.1.9. Carpeta intruders..... | 5 |
| 4.2. Ejecución del proyecto..... | 6 |

1. Propósito

El **objetivo** de este proyecto es crear una mini **cámara de vigilancia**, que sea capaz de detectar rostros, guardar una captura, y analizar dicho rostro. Los rasgos que detecta son: edad aproximada, sexo, raza, emoción, color de pelo y si lleva gafas o no.

Este proyecto está subido en un repositorio público de GitHub, el cual se puede ver a través del siguiente enlace:

<https://github.com/Ruth-Torres/DeepFace-on-CCTV>

2. Licencia

El proyecto usa una licencia MIT, tal y como se indica en el propio [repositorio del proyecto](#).

MIT License

Copyright (c) 2024 Ruth Torres Gallego

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

3. Instalaciones previas

Para poder ejecutar este programa, será necesario instalar algunas librerías:

- **Python** (La versión utilizada para este proyecto es la 3.12.6)
- **OS**: una librería que viene instalada con Python. Proporciona funcionalidad independiente del sistema operativo.
- **Datetime**: una librería que viene instalada con Python y sirve para manejar fechas en Python.
- **Subprocess**: una librería que viene instalada con Python. Es una herramienta que te permite ejecutar otros programas o comandos desde tu código Python.
- **OpenCV**: una librería de computación visual para el procesamiento de imágenes en Python. Se instala con el siguiente comando:

```
pip install opencv-python
```

- **Numpy**: una librería para crear vectores y matrices multidimensionales (arrays) en Python.
IMPORTANTE: para este proyecto se ha usado la versión **1.26.4**, ya que la librería MediaPipe necesita una versión de Numpy anterior a la 2.0.0 y la librería TensorFlow necesita una versión de Numpy entre la 2.1.0 (sin incluir) y la 1.26.0 (incluida). Para instalar Numpy-1.26.4, se ejecuta el siguiente comando:

```
pip install numpy==1.26.4
```

- **DeepFace**: una librería para la detección de caras y el análisis facial en Python.

```
pip install deepface
```

- **TensorFlow**: una librería de Machine Learning desarrollada por Google, necesaria para usar DeepFace.

```
pip install tf-keras
```

- **MediaPipe**: es un marco de trabajo de código abierto desarrollado por Google para construir pipelines de aprendizaje automático multimodales. Es especialmente útil para tareas de visión por computadora, como el reconocimiento facial, el seguimiento de manos, la segmentación de imágenes, entre otros.

```
pip install mediapipe
```

4. Guía

4.1. Organización del repositorio

4.1.1. Archivo **main.py**

El programa principal del proyecto. Ejecuta primero el archivo *Camera_CCTV.py* y una vez termine, ejecutará el archivo *AnalyzeAllImages.py*.

4.1.2. Archivo **Camera_CCTV.py**

El programa con toda la parte de capturar caras con la cámara. Abrirá la cámara y cada vez que detecte una cara, guardará una captura en la carpeta *captures* con su timestamp correspondiente.

4.1.3. Archivo **AnalyzeAllImages.py**

El programa con toda la parte de análisis facial. Analizará todas las imágenes en la carpeta *captures* y creará un cartel de **INTRUDER** en la carpeta *intruders* con sus datos.

4.1.4. Carpeta **functionsSeparated**

En esta carpeta se encuentran las distintas funciones utilizadas en el programa principal por separado.

4.1.5. Carpeta **haarcascades**

Son clasificadores ya entrenados e importados desde el repositorio de OpenCV. En esta carpeta se encuentran los clasificadores de **caras** y **ojos**.

4.1.6. Carpeta **images**

Una serie de imágenes de prueba para las distintas funciones.

4.1.7. Carpeta **info**

Archivos con información adicional sobre las librerías usadas.

4.1.8. Carpeta **captures**

Las imágenes con caras detectadas por la cámara.

4.1.9. Carpeta **intruders**

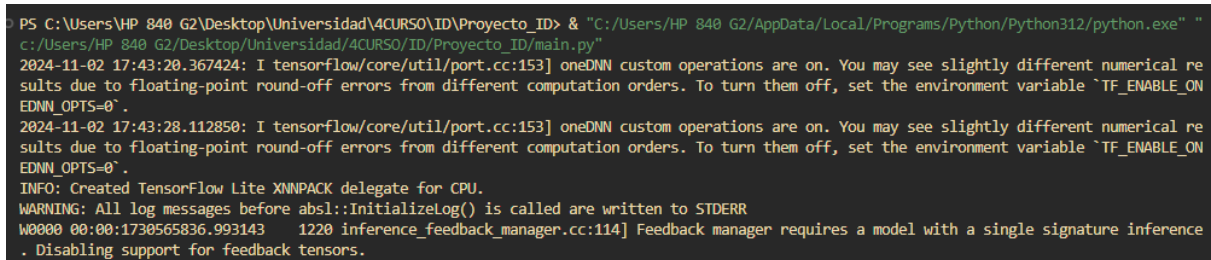
Las imágenes generadas con la descripción de la cara.

4.2. Ejecución del proyecto

Para ejecutar el proyecto, basta con ejecutar el archivo llamado `main.py`, ya sea a través del propio IDE o con el siguiente comando:

```
python main.py
```

Al ejecutarlo, es posible que aparezcan una serie de warnings, pero no impiden la ejecución y funcionamiento del proyecto (Figura 1).



```
PS C:\Users\HP_840_G2\Desktop\Universidad\4CURSO\ID\Proyecto_ID> & "C:/Users/HP_840_G2/AppData/Local/Programs/Python/Python312/python.exe" "
c:/Users/HP_840_G2/Desktop/Universidad/4CURSO/ID/Proyecto_ID/main.py"
2024-11-02 17:43:20.367424: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical re
sults due to floating-point round-off errors from different computation orders. To turn them off, set the environment variable `TF_ENABLE_ON
EDNN_OPTS=0`.
2024-11-02 17:43:28.112850: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical re
sults due to floating-point round-off errors from different computation orders. To turn them off, set the environment variable `TF_ENABLE_ON
EDNN_OPTS=0`.
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
WARNING: All log messages before absl::InitializeLog() is called are written to STDERR
W0000 00:00:1730565836.993143 1220 inference_feedback_manager.cc:114] Feedback manager requires a model with a single signature inference
. Disabling support for feedback tensors.
```

Figura 1. Captura de pantalla con warnings iniciales.

Una vez ejecutado, se encenderá la cámara y comenzará a capturar imágenes de las caras detectadas. Las imágenes capturadas se guardarán en la carpeta *captures*. Para cerrar la cámara, habrá que presionar la tecla 'q'.

Luego, empezará a analizar las imágenes capturadas en dicha carpeta. Cada vez que finalice el análisis de una imagen, mostrará los resultados por pantalla. Para pasar a la siguiente imagen, bastará con **presionar cualquiera tecla**.

Cuando ya no haya más imágenes que analizar, el programa terminará. Los resultados del análisis se podrán también ver en la carpeta *intruders*.