**Object and Face Detection with OpenCV and DeepFace**

This project implements object detection (focused on detecting persons) and face verification using OpenCV and DeepFace. It processes live video feed to detect objects and faces, save images, and verify detected faces against a reference directory.

**FeaturesObject and Face Detection with OpenCV and DeepFace**

* Object Detection: Uses OpenCV's DNN module and a pre-trained model to detect objects (with focus on persons).
* Face Detection: Applies OpenCV's Haar Cascade to detect faces within detected persons.
* Face Verification: Uses DeepFace to verify detected faces against a set of reference images.
* Video Capture: Processes frames from live video input for continuous person detection and face recognition.
* Multithreading: Verification tasks are handled asynchronously using Python's threading module.

**Project Structure**

* Object Detection: The model uses a pre-trained SSD MobileNet model to detect objects in a frame.
* Face Detection: Detected persons are further analyzed to detect faces using the haarcascade\_frontalface\_default.xml classifier.
* Face Verification: Detected faces are verified against reference images using DeepFace's face verification feature.
* Image Saving: Detected persons and faces are saved as images in specific directories.
* Logs: Metadata logs track processed images to avoid redundant verification of the same image.
* Dependencies

**Install the necessary dependencies using:**

1. bash
2. Copy code

pip install opencv-python deepface

1. Setup

**Ensure the following files are present in the Resources folder:**

* coco.names: Contains class names for object detection.
* ssd\_mobilenet\_v3\_large\_coco\_2020\_01\_14.pbtxt: Configuration for the object detection model.
* frozen\_inference\_graph.pb: Weights for the object detection model.
* haarcascade\_frontalface\_default.xml: Classifier for face detection.

**Create directories for:**

**Reference Images:** This folder should contain the images you want to use for face verification.

**Output:** Where detected persons and faces will be saved.

**How to Run**

Clone the repository and set up the directories.

Execute the script:

1. bash
2. Copy code

python script.py

The program captures video from your default camera, detects persons, identifies faces, and verifies them against reference images.

**Functions**

capture\_increase\_size(x, y, w, h, img)

Increases the crop size of detected objects to include more surrounding pixels.

capture\_analyze\_save(img, persons\_path, faces\_path)

Detects persons in a frame, crops their images, detects faces, and saves both the cropped persons and detected faces.

check\_face(faces\_path, persons\_path, reference\_dir, person\_metadata\_log\_file, face\_metadata\_log\_file)

Verifies detected faces against reference images and moves matched images to the reference directory.

video\_capture(output\_folder, reference\_folder, person\_metadata\_log\_file, face\_metadata\_log\_file)

Main loop that captures video frames, detects objects, saves results, and verifies faces.

getObjects(img, thres, nms, draw=True, objects=[])

Detects objects in an image using a pre-trained DNN model.

**Requirements**

OpenCV for video processing, object detection, and face detection.

DeepFace for face recognition and verification.

Python 3.x