 # Face Recognition

Explanation: This is a comment indicating the purpose of the script.

 # Importing OpenCV

Explanation: Importing the OpenCV library which is used for image processing tasks.

import cv2

Explanation: Imports the OpenCV module into your script.

 # Face Detection Model

Explanation: Load the pre-trained Haar Cascade classifier for frontal face detection.

face_cap = cv2.CascadeClassifier("...haarcascade_frontalface_default.xml")

Explanation: Initializes the face detection model using the Haar cascade file provided by OpenCV.

 # Initializing Video Capture

Explanation: Start capturing video from the default camera (usually the webcam).

video_cap = cv2.VideoCapture(0)

Explanation: Creates a VideoCapture object to access the webcam.

 # Main Loop

Explanation: Start an infinite loop to read video frames and process them.

while True :

Explanation: Begin a continuous loop for real-time face detection.

ret , video_data = video_cap.read()

Explanation: Reads a frame from the webcam. `ret` is a boolean indicating success.

 # Converting Frame To Grayscale

Explanation: Convert the captured frame to grayscale as the Haar cascade model works on grayscale images.

col = cv2.cvtColor(video_data,cv2.COLOR_BGR2GRAY)

Explanation: Convert the frame from BGR (color) to grayscale.

 # Face Detection

Explanation: Detect faces in the grayscale frame.

faces = face_cap.detectMultiScale(...)

Explanation: Detects objects (faces) and returns coordinates of bounding boxes.

 # Drawing The Rectangles Around Faces

Explanation: Loop over detected face coordinates and draw rectangles.

for (x, y, w, h) in faces:

Explanation: Iterate over each detected face's position and size.

cv2.rectangle(video_data,(x, y),(x + w, y + h),(0, 255, 0),2)

Explanation: Draw a green rectangle around each detected face.

 # Displaying The Video Frame

Explanation: Show the processed video frame with rectangles drawn.

cv2.imshow("live_camera",video_data)

Explanation: Display the frame in a window titled 'live_camera'.

 # Breaking Loop

Explanation: Provide a condition to break out of the loop.

if cv2.waitKey(10) == ord("e"):

Explanation: Break the loop if the 'e' key is pressed.

break

Explanation: Exit the loop.

 # Releasing The Video Capture

Explanation: Release the camera resource after exiting the loop.

video_cap.release()

Explanation: Stop the video capture and free the camera.