FACE DETECTION

Face Detection using Haar Cascade Classifier

Introduction:

Face detection is a crucial application of computer vision used in areas like surveillance, authentication systems, and image analysis. OpenCV provides a powerful and efficient method to detect faces using **Haar Cascade Classifiers** – a pre-trained model that can identify faces and other objects from images or video streams.

In this task, you will learn how to use Haar Cascades to detect faces from live webcam input and draw bounding boxes around detected faces.

What to Learn:

1. **Understanding Haar Cascade Classifiers:**
   - What is a Haar Cascade?
   - How does the Haar Cascade algorithm work? (Feature extraction, training process)
2. **Working with Pre-Trained Models in OpenCV:**
   - Introduction to OpenCV's haarcascade_frontalface_default.xml.
   - Using Haar Cascades to detect faces and other features (eyes, smile, etc.).
3. **Implementing Face Detection in OpenCV:**
   - Accessing the webcam feed.
   - Applying the face detection model.
   - Drawing rectangles around detected faces.

REFERENCE LINKS

https://docs.opencv.org/

https://docs.opencv.org/master/db/d28/tutorial_cascade_classifier.html

https://opencv-python-tutroals.readthedocs.io/en/latest/

1. Write a program to detect a face in a static image using a Haar cascade.
2. Create a script to detect and highlight both eyes in a static image.
3. Implement a program to detect multiple faces in an image and display the total count.
4. Write a program to detect faces from a webcam feed and draw rectangles around them.
5. Create a script to detect a smile in a static image using Haar cascades.
6. Implement a program to highlight the face region and save the cropped face to a folder.
7. Write a program to detect and highlight both the face and eyes from a video feed.
8. Create a script to count and display the number of faces detected in a live webcam feed.
9. Implement a program to detect faces in a grayscale image and highlight them.
10. Write a script to display "Face Detected" if a face is found in a static image.