# 6. FACE DETECTION METHOD IN OPENCV USING PYTHON

|  |  |
| --- | --- |
| **EX.N0 : 6** | **LOAD AND IMPLEMENT THE FACE DETECTION METHOD IN OPENCV USING PYTHON** |
| **DATE : 04/03/2025** |

**AIM:**

To load and implement real-time face detection using OpenCV and Haar Cascade Classifier.

# ALGORITHM:

Step 1: Import OpenCV library.

Step 2: Load the Haar cascade classifier for face detection. Step 3: Access webcam video using cv2.VideoCapture ().

Step 4: Read frames continuously and convert them to grayscale.

Step 5: Detect faces using detectMultiScale() method.

Step 6: Draw rectangles around detected faces and display the video

# PROGRAM:

import cv2

from google.colab.patches import cv2\_imshow # Import the patched cv2\_imshow

# Load the pre-trained face detection model

face\_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade\_frontalface\_default.xml')

# Read an image

image = cv2.imread('/content/photo.jpg') # Replace with your image path

gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY) # Convert to grayscale # Detect faces

faces = face\_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5, minSize=(30, 30))

# Draw rectangles around detected faces for (x, y, w, h) in faces:

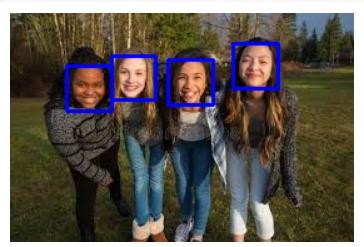
cv2.rectangle(image, (x, y), (x + w, y + h), (255, 0, 0), 2)

# Display the image with detected faces

cv2\_imshow(image) # Use cv2\_imshow instead of cv2.imshow cv2.waitKey(0)

cv2.destroyAllWindows()

# OUTPUT:

****

**RESULT:**

Thus the Program has been executed successfully and verified.