EX NO: 6	Multiple Object Detection using YOLO

Aim:

To load an image and implement face detection using OpenCV's Haar Cascade classifier in Python.

Algorithm:

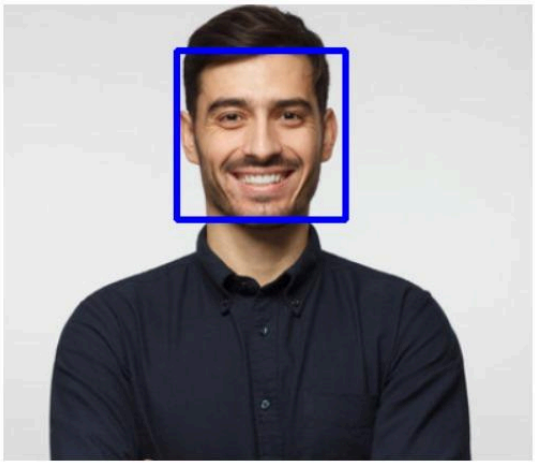
1. **Import Libraries**: Use OpenCV (`cv2`) and Matplotlib (`plt`) for image operations and visualization.
2. **Load Classifier**: Use OpenCV's pre-trained Haar Cascade XML file for face detection.
3. **Read Image**: Load the input image using `cv2.imread()`.
4. **Convert to Grayscale**: Convert the image to grayscale since Haar cascades work on grayscale images.
5. **Detect Faces**: Use `detectMultiScale()` to locate faces in the image.
6. **Draw Bounding Boxes**: For each detected face, draw a rectangle around it.
7. **Display Result**: Show the output using `matplotlib.pyplot`.

Code:

```
import cv2

import matplotlib.pyplot as plt

face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')
```

```
image_path = 'face_sample.jpg'  # Replace with your image path

img = cv2.imread(image_path)

faces = face_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5)

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

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

img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

plt.figure(figsize=(8, 6))

plt.imshow(img_rgb)

plt.axis('off')

plt.title('Detected Faces')

plt.show()
```

Output:



Result:

The program successfully detected and highlighted all human faces in the image using bounding boxes.