

LAB: 09

CODE:

```
import cv2, os, numpy as np, matplotlib.pyplot as plt
def convertToRGB(image):
    return cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
def detect_faces(cascade, test_image, scaleFactor = 1.1):
    # create a copy of the image to prevent any changes to the original one.
    image_copy = test_image.copy()
    #convert the test image to gray scale as opencv face detector expects gray images
    gray_image = cv2.cvtColor(image_copy, cv2.COLOR_BGR2GRAY)
    # Applying the haar classifier to detect faces
    faces_rect = cascade.detectMultiScale(gray_image, scaleFactor=scaleFactor, minNeighbors=5)
    for (x, y, w, h) in faces_rect:
        cv2.rectangle(image_copy, (x, y), (x+w, y+h), (0, 255, 0), 15)
    return image_copy

#read image
test_image = cv2.imread('D:/AI_labs/lab-9/image.jpg')
#Loading the classifier for frontal face
cascade = detector_value = cv2.CascadeClassifier('D:/AI_labs/lab-9/haarcascade_frontalface_default.xml')
#calling the detect_faces function
image = detect_faces(cascade, test_image, scaleFactor = 1.1)
#plot resulted image in RGB
plt.imshow(convertToRGB(image))
plt.show()
```

OUTPUT:

