

# Assignment – 6

Name : V Surya Kumar  
Reg. No. : 19BCE10286  
Application ID : SPS\_APL\_20210013738

**Q.** Develop a python code to detect any object using Haar cascade classifier.

**Code:**

```
ObjectDetector.py - /Users/vsuryakumar/Downloads/ObjectDetector.py (3.9.5)
'''
Python Code Using OpenCV to detect face of from the primary camera.
'''

import cv2

face_classifier = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')
eye_classifier=cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_eye.xml')
smile_classifier=cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_smile.xml')

# Reading the First frame of the Video
cap=cv2.VideoCapture(0)

while True:
    _, img = cap.read()
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY) # Converting the Image into Grayscale

    faces = face_classifier.detectMultiScale(gray, 1.3, 5)
    # detectMultiScale function returns 4 values - x-coordinate, y-coordinate, width(w) and height(h) of
    # of the face. Based on these 4 values we will draw a rectangle around the face.

    for (x,y,w,h) in faces:
        cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
        roi_gray = gray[y:y+h, x:x+w]
        roi_color = img[y:y+h, x:x+w]
        eyes = eye_classifier.detectMultiScale(roi_gray)
        smiles = smile_classifier.detectMultiScale(roi_color)

        for (ex,ey,ew,eh) in eyes:
            cv2.rectangle(roi_color,(ex,ey),(ex+ew,ey+eh),(0,255,0),2)
        for (sx,sy,sw,sh) in smiles:
            cv2.rectangle(roi_color,(sx,sy),(sx+sw,sy+sh),(0,255,0),2)

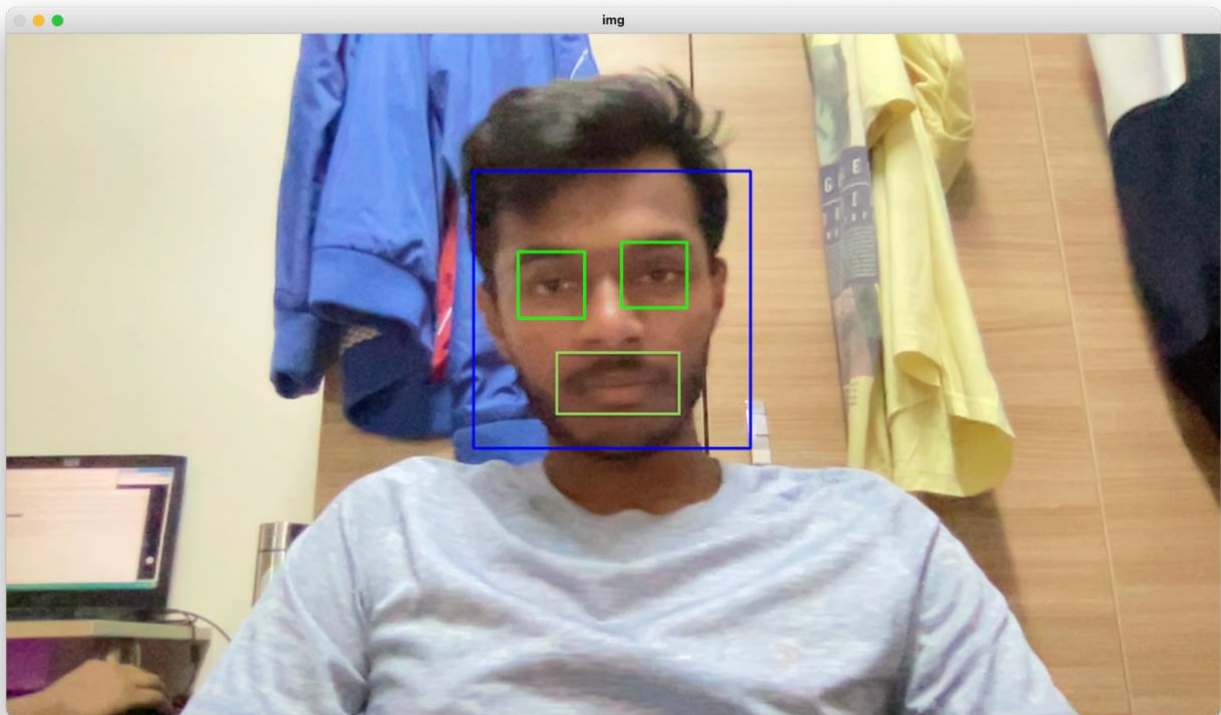
    cv2.imshow('img',img)

    k = cv2.waitKey(0)
    if k == 27:
        cv2.destroyAllWindows()
    elif k == ord('s'):
        cv2.imwrite('surya_photo.png',img)
        cv2.destroyAllWindows()

cap.release()
```

Ln: 21 Col: 8

# Output:



```
"""
Python Code Using OpenCV to detect face of from the primary camera.
"""

import cv2

face_classifier = cv2.CascadeClassifier(cv2.data.haarcascades +
'haarcascade_frontalface_default.xml')
eye_classifier=cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_eye.xml')
smile_classifier=cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_smile.xml')

# Reading the First frame of the Video
cap=cv2.VideoCapture(0)

while True:
    _, img = cap.read()
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY) # Converting the Image into
    Grayscale

    faces = face_classifier.detectMultiScale(gray, 1.3, 5)
    # detectMultiScale function returns 4 values — x-coordinate, y-coordinate, width(w)
    height(h) of
    # of the face. Based on these 4 values we will draw a rectangle around the face.

    for (x,y,w,h) in faces:
        cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
        roi_gray = gray[y:y+h, x:x+w]
        roi_color = img[y:y+h, x:x+w]
        eyes = eye_classifier.detectMultiScale(roi_gray)
        smiles = smile_classifier.detectMultiScale(roi_gray)
```

```
    for (ex,ey,ew,eh) in eyes:
        cv2.rectangle(roi_color,(ex,ey),(ex+ew,ey+eh),(0,255,0),2)
    for (sx,sy,sw,sh) in smiles:
        cv2.rectangle(roi_color,(sx,sy),(sx+sw,sy+sh),(0,255,0),2)

cv2.imshow('img',img)

k = cv2.waitKey(0)
if k == 27:
    cv2.destroyAllWindows()
elif k == ord('s'):
    cv2.imwrite('surya_photo.png',img)
    cv2.destroyAllWindows()

cap.release()
```