CGC ASSIGNMENT – 7

Name: RAHUL VARMA

Roll No: S20200010212

Using Harr Cascade Eye Detection (Closed or Open):

First Code Detects the Face.

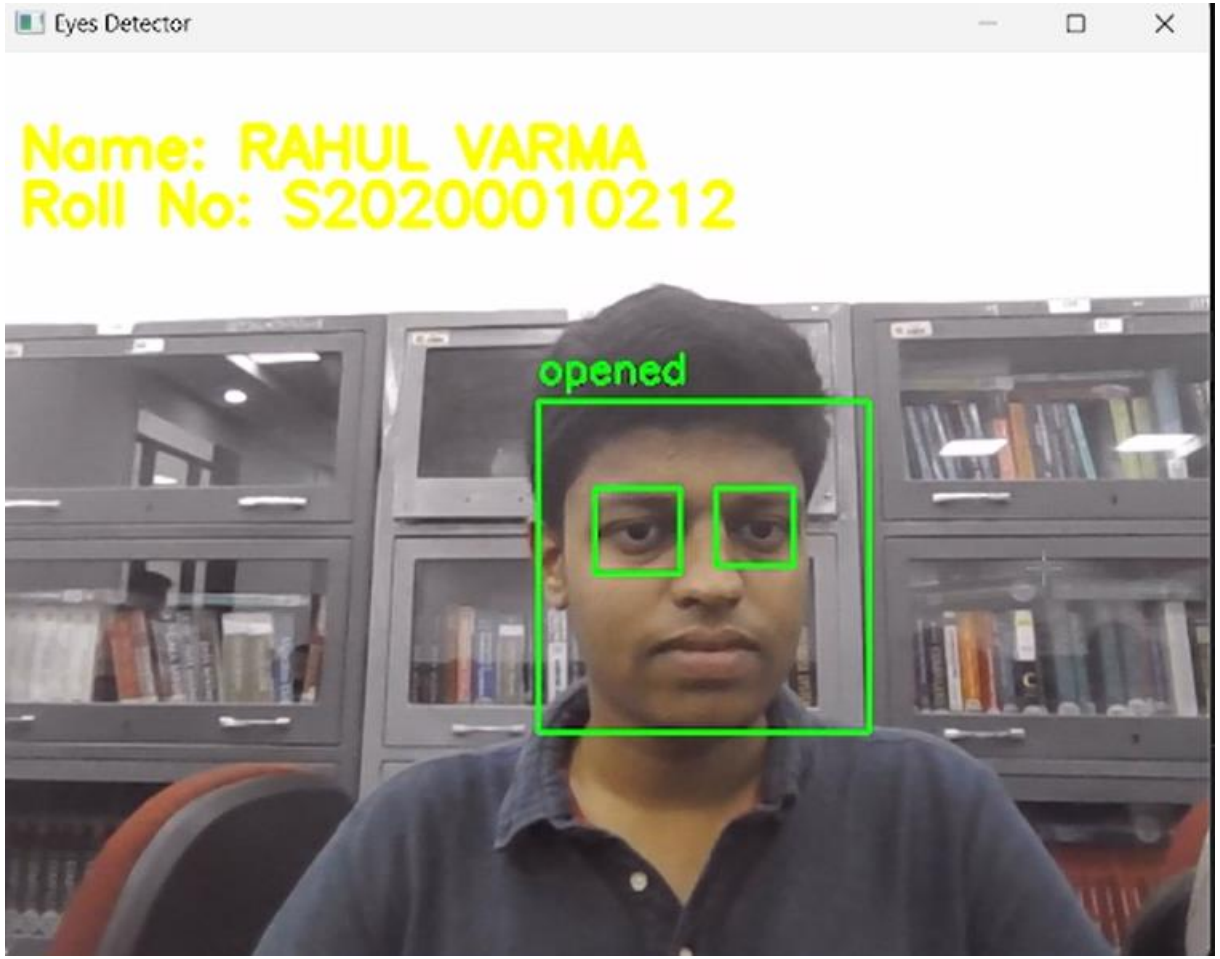
When Both Eyes are open then it shows in green color and shows that it is open.

And when Both eyes are closed then it shows in red color and also shows that it is closed.

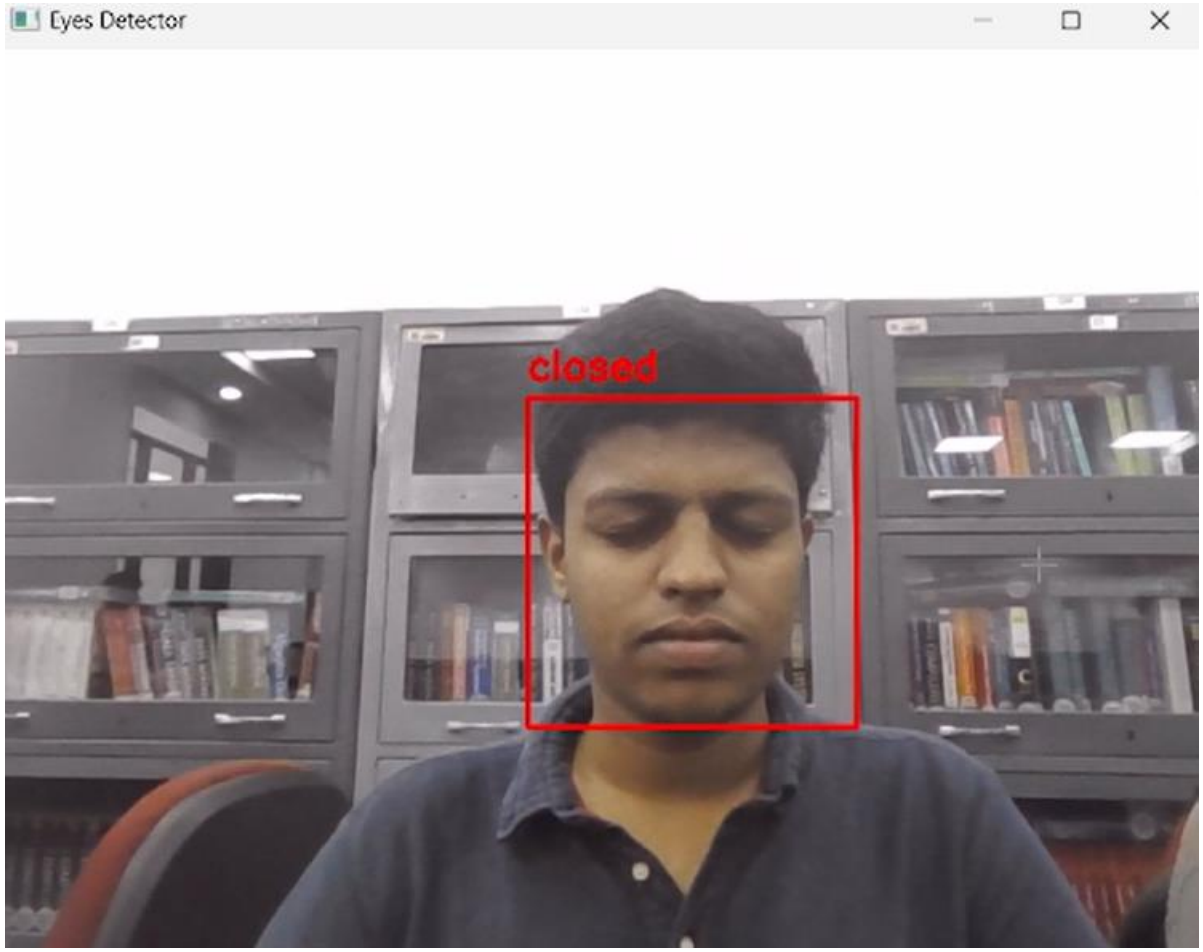
Using Harr Cascade.

Detail explanation is given in the end of the pdf.

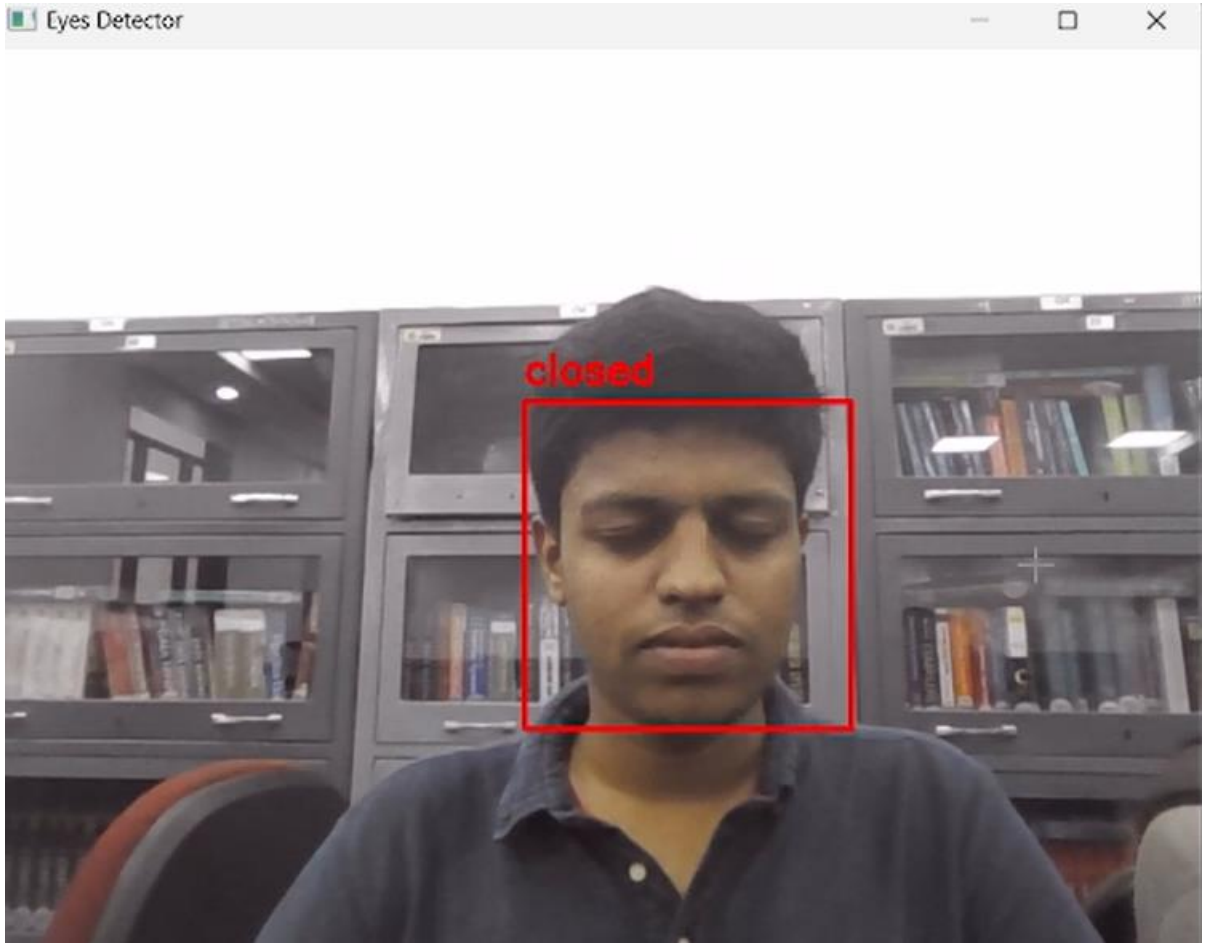
Eyes are opened:



Eyes are closed:



Closed Eyes:



Code:

```python
import cv2
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
eye_cascade = cv2.CascadeClassifier('haarcascade_eye.xml')

EYE_COUNT_THRESH = 1
cap = cv2.VideoCapture(0)

while True:
    ret, frame = cap.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray, 1.3, 5)

    # Loop through each face
    for (x,y,w,h) in faces:
        # Draw a rectangle around the face
        cv2.rectangle(frame,(x,y),(x+w,y+h),(255,0,0),2)
        gray = gray[y:y+h, x:x+w]
        frame_color = frame[y:y+h, x:x+w]
        eyes = eye_cascade.detectMultiScale(gray)

        count = 0
        for (ex,ey,ew,eh) in eyes:
            cv2.rectangle(frame_color,(ex,ey),(ex+ew,ey+eh),(0,255,0),2)
            count += 1

        if count < EYE_COUNT_THRESH:
            cv2.putText(frame, "closed", (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
            cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 0, 255), 2)
        else:
            cv2.putText(frame, "opened", (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 255, 0), 2)
            cv2.putText(frame, "Name: {}".format('RAHUL VARMA'), (10, 60), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 255), 3)
            cv2.putText(frame, "Roll No: {}".format('S20200010212'), (10, 90), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 255), 3)
            cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)

    cv2.imshow('Eyes Detector', frame)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```

```python
import cv2
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
eye_cascade = cv2.CascadeClassifier('haarcascade_eye.xml')

EYE_COUNT_THRESH = 1
cap = cv2.VideoCapture(0)

while True:
    ret, frame = cap.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray, 1.3, 5)

    # Loop through each face
    for (x,y,w,h) in faces:
        # Draw a rectangle around the face
        cv2.rectangle(frame,(x,y),(x+w,y+h),(255,0,0),2)
        gray = gray[y:y+h, x:x+w]
        frame_color = frame[y:y+h, x:x+w]
        eyes = eye_cascade.detectMultiScale(gray)

        count = 0
        for (ex,ey,ew,eh) in eyes:
            cv2.rectangle(frame_color,(ex,ey),(ex+ew,ey+eh),(0,255,0),2)
            count += 1

        if count < EYE_COUNT_THRESH:
            cv2.putText(frame, "closed", (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
            cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 0, 255), 2)
        else:
            cv2.putText(frame, "opened", (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 255, 0), 2)
```

```
            cv2.putText(frame, "Name: {}".format('RAHUL VARMA'), (10, 60), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255,
255), 3)
            cv2.putText(frame, "Roll No: {}".format('S20200010212'), (10, 90), cv2.FONT_HERSHEY_SIMPLEX, 1, (0,
255, 255), 3)
            cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)

    cv2.imshow('Eyes Detector', frame)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```

Detailed Explanation:

- The code is an implementation of an eye detection system using OpenCV (Open Source Computer Vision) library in Python. It detects human faces in the input video stream and then detects eyes in the face region. The system checks if the number of eyes detected is less than a threshold value, which implies that the person's eyes are closed. If the eyes are open, it displays the person's name and roll number on the video stream along with a green rectangle around the face, and if the eyes are closed, it displays "closed" along with a red rectangle around the face.

- The code starts by importing the necessary libraries, OpenCV, and Haar Cascade classifiers for face and eye detection. Haar Cascade classifiers are machine learning-based algorithms that are used to detect objects in an image or video.

- It then sets a threshold value for the number of eyes detected and captures the video feed from the default camera device (here, 0). The code then enters an infinite loop that reads each frame of the video stream and converts it to grayscale for face detection.

- It then applies the face detection algorithm to the grayscale frame using the detectMultiScale function. The function returns a list of rectangles, each representing a detected face. The code then loops through each face and applies eye detection to the face region by cropping the grayscale and colored frames.

- It then draws a rectangle around the detected face using the cv2.rectangle function and a blue color. If the eyes are detected, the code draws a green rectangle around each eye using the cv2.rectangle function, counts the number of detected eyes and checks if it is greater than the threshold value. If the eyes are open, it displays the person's name and roll number using the cv2.putText function with a yellow color, and if the eyes are closed, it displays "closed" using the cv2.putText function with a red color.

- Finally, the code displays the video stream using the cv2.imshow function and waits for a key press. If the key pressed is 'q', the loop breaks, and the code releases the video capture device and destroys all the OpenCV windows using the cap.release and cv2.destroyAllWindows functions, respectively.