```python
# OpenCV program to detect face in real time
# import libraries of python OpenCV
# where its functionality resides
import cv2

# load the required trained XML classifiers
# Trained XML classifiers describes some features of some
# object we want to detect a cascade function is trained
# from a lot of positive(faces) and negative(non-faces)
# images.
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

# Trained XML file for detecting eyes
eye_cascade = cv2.CascadeClassifier('haarcascade_eye.xml')

# capture frames from a camera
cap = cv2.VideoCapture(0)

# Video Resolution
resW = 800      # Resolution width and
resH = (resW//16) * 9   # Height (aspect ratio must be 16:9)

cap.set(cv2.CAP_PROP_FRAME_WIDTH, resW)
cap.set(cv2.CAP_PROP_FRAME_HEIGHT, resH)

font = cv2.FONT_HERSHEY_SIMPLEX

# loop runs if capturing has been initialized.
while True:

    # reads frames from a camera
    ret, img = cap.read()

    # convert to gray scale of each frames
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    # Detects faces of different sizes in the input image
    faces = face_cascade.detectMultiScale(gray, 1.3, 5)
    print(faces, "\n")

    for (x, y, w, h) in faces:
        # To draw a rectangle on a face
        cv2.rectangle(img, (x, y), (x + w, y + h),(230, 220, 210), 2)

        # Draw line from face to center of screen
        cx_face = x + w//2
        cy_face = y + h//2
```

```
48          c_screen = (resW//2, resH//2)
49
50          cv2.line(img, (cx_face, cy_face), c_screen, (0, 255, 0), 2
    )
51
52
53          # Using cv2.putText() method
54          img = cv2.putText(img, 'human',(x, y-5), font,
55                            1, (30,220,210), 2, cv2.LINE_AA)
56
57          # Regions of interest
58          roi_gray = gray[y:y+h, x:x+w]
59          roi_color = img[y:y+h, x:x+w]
60
61          # Detects eyes of different sizes in the input image
62          eyes = eye_cascade.detectMultiScale(roi_gray)
63
64          # To draw a rectangle around eyes
65          for (ex,ey,ew,eh) in eyes:
66              cv2.rectangle(roi_color,(ex,ey),(ex + ew,ey +eh),(0,
    127,255),2)
67
68      # Display an image in a window
69      cv2.imshow('img', img)
70
71      cv2.imshow('facecam', gray)
72
73      # Wait for Esc key to stop
74      k = cv2.waitKey(30) & 0xff
75      if k == 27:
76          break
77
78  # Close the window
79  cap.release()
80
81  # De-allocate any associated memory usage
82  cv2.destroyAllWindows()
```