-:DOCUMENTATION OF FACE DETECTION:-

**About Face Detection:**

Machine Learning techniques enables us to detect faces in an image, identify key facial features, and get the contours of detected faces.

With face detection, we can get the information we need to perform tasks like embellishing selfies and portraits, or generating avatars from a user's photo. Because ML Kit can perform face detection in real time, you can use it in applications like video chat or games that respond to the player's expressions.

Here in this we used OpenCV for detecting faces. Using Cascade Classifiers eyes, nose, mouth are detected.

The xml files of eyes, nose, face and mouth are downloaded and linked(the path Is given as reference) with the code.Boundaries of the detected features are given different colors like blue for face,red for eyes,green for nose,white for mouth.

Object Detection using Haar feature-based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones in their paper, “Rapid Object Detection using a Boosted Cascade of Simple Features” in 2001. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images.

Here we will work with face detection. Initially, the algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then we need to extract features from it.

In detect multiscale the parameters following are passed:-

* **image :** Matrix of the type CV\_8U containing an image where objects are detected.
* **scaleFactor :** Parameter specifying how much the image size is reduced at each image scale.
* **minNeighbors:** Parameter specifying how many neighbors each candidate rectangle should have to retain it.

OpenCV comes with a trainer as well as detector. If we train our own classifier for any object like car, planes etc. we can use OpenCV to create one using  **Cascade Classifier Training.**

Here we will deal with detection. OpenCV already contains many pre-trained classifiers for face, eyes, smile etc. Those XML files are stored in opencv/data/haarcascades/ folder.We created face and eye detector with OpenCV.

First we need to load the required XML classifiers. Here a window appears on the screen and the faces in it are detected and features (mouth, nose, eyes ) are also further detected. The faces in the image are found. If faces are found, it returns the positions of detected faces as rectangle(x,y,w,h). Once we get these locations, we can create a ROI for the face and apply eye detection on this ROI (since eyes are always on the face !!! ) and also mouth, nose.

**Code:**

import cv2

def draw\_boundary(img, classifier, scaleFactor, minNeighbors, color, text):

gray\_img = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

features = classifier.detectMultiScale(gray\_img, scaleFactor, minNeighbors)

coords = []

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

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

#cv2.putText(img, text, (x, y-4), cv2.FONT\_HERSHEY\_SIMPLEX, 0.8, color, 1, cv2.LINE\_AA)

coords = [x, y, w, h]

return coords

def detect(img, faceCascade, eyeCascade, noseCascade, mouthCascade):

color = {"blue":(255,0,0), "red":(0,0,255), "green":(0,255,0), "white":(255,255,255)}

coords = draw\_boundary(img, faceCascade, 1.2, 10, color['blue'], "Face")

if len(coords)==4:

# Updating region of interest by cropping image

roi\_img = img[coords[1]:coords[1]+coords[3], coords[0]:coords[0]+coords[2]]

coords = draw\_boundary(roi\_img, eyeCascade, 1.2, 14, color['red'], "Eye")

coords = draw\_boundary(roi\_img, noseCascade, 1.4, 10, color['green'], "Nose")

coords = draw\_boundary(roi\_img, mouthCascade, 1.7, 15, color['white'], "Mouth")

return img

faceCascade = cv2.CascadeClassifier(r'C:\Users\Rahul\Pictures\faceRecg\haarcascade\_frontalface\_default.xml')

eyesCascade = cv2.CascadeClassifier(r'C:\Users\Rahul\Pictures\faceRecg\haarcascade\_eye.xml')

noseCascade = cv2.CascadeClassifier(r'C:\Users\Rahul\Pictures\faceRecg\Nariz.xml')

mouthCascade = cv2.CascadeClassifier(r'C:\Users\Rahul\Pictures\faceRecg\Mouth.xml')

video\_capture = cv2.VideoCapture(0)

while True:

check, img = video\_capture.read()

img = detect(img, faceCascade, eyesCascade, noseCascade, mouthCascade)

cv2.imshow("face detection", img)

if cv2.waitKey(1) & 0xFF == ord('q'):

break

video\_capture.release()

cv2.destroyAllWindows()

**Conclusion:**

This project successfully explains the process of face detection with the help of openCV.