

# Face Detection using OpenCV



## 1- Introduction:

In this task I used OpenCV to detect human faces in images in Python.

## 2- OpenCV:

It is an open source computer vision library, OpenCV provides a pre-trained models.

## 3- Detect Human Faces in image using OpenCV:

I used pretrained Haar cascade models to detect faces and eyes in an image, we need to download the trained classifier XML file (haarcascade\_frontalface\_alt.xml) and (haarcascade\_eye\_tree\_eyeglasses.xml), which is available in OpenCv's GitHub repository (<https://github.com/opencv/opencv/tree/master/data/haarcascades>) and (<https://github.com/opencv/opencv/tree/master/data/haarcascades>) and save it to your working location.

## • Libraries:

- 1- **cv2 Library:** OpenCV library.
- 2- **Numpy Library:** to search for the row and column values of the face NumPy ndarray. This is the array with the face rectangle coordinates.
- 3- **glob Library:** is used to retrieve files/pathnames matching a specified pattern.
- 4- **matplotlib Library:** to draw rectangle and circular around faces and eyes that detect in the image.

## • Steps:

- 1- Import all necessary libraries.
- 2- Load filenames for human images.
- 3- Extract pre-trained face and eye detectors.
- 4- Convert the color images to grayscale it is important step because detection works only on grayscale images.
- 5- Find faces and eyes in image using detectMultiScale function. 6- Get bounding box for each detected face and eye: Each face contains a set of coordinate for the rectangle regions where faces were found. We use these coordinates to draw the rectangles in our image.

```
In [78]: #importing necessary libraries.
import cv2
import numpy as np
from glob import glob
import matplotlib.pyplot as plt
%matplotlib inline

# load filenames for human images
human_files = np.array(glob("human_images/*"))
# extract pre-trained face and eye detectors
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_alt.xml')
eye_cascade = cv2.CascadeClassifier('haarcascade_eye_tree_eyeglasses.xml')
# load color (BGR) image
for image in human_files:
    img = cv2.imread(image)

    # convert BGR image to grayscale
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    # find faces in image
    faces = face_cascade.detectMultiScale(gray)
    eyes = eye_cascade.detectMultiScale(gray)
    # print number of faces and eyes detected in the image
```

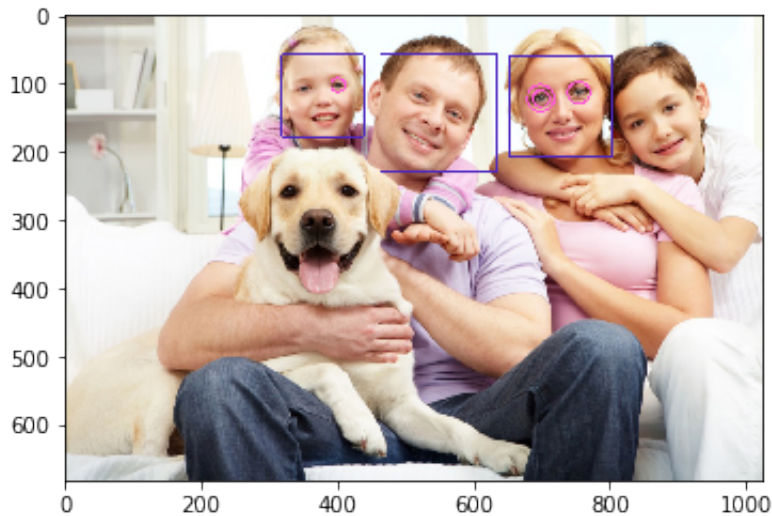
```

// print number of faces and eyes detected in the image
print('Number of faces detected:', len(faces))
print('Number of eyes detected:', len(eyes))
# get bounding box for each detected face and eye
for (x,y,w,h) in faces:
    # add bounding box to color image
    cv2.rectangle(img,(x,y),(x+w,y+h),(191,40,78),2)
for (x2,y2,w2,h2) in eyes:
    # add bounding box to color image
    radius = int(round((w2 + h2)*0.25))
    cv2.circle(img,(x2+w2//2,y2+h2//2),radius, (216,91,255), 2)
    # convert BGR image to RGB for plotting
cv_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    # display the image, along with bounding box
plt.imshow(cv_rgb)
plt.show()

```

Number of faces detected: 3

Number of eyes detected: 4



Number of faces detected: 1

## 4- References:

1- How to set up your computer to work with the OpenCV library:

[https://docs.opencv.org/master/df/d65/tutorial\\_table\\_of\\_content\\_introduction.html](https://docs.opencv.org/master/df/d65/tutorial_table_of_content_introduction.html)  
([https://docs.opencv.org/master/df/d65/tutorial\\_table\\_of\\_content\\_introduction.html](https://docs.opencv.org/master/df/d65/tutorial_table_of_content_introduction.html))

2- Face Detection in 2 Minutes using OpenCV & Python: <https://towardsdatascience.com/face-detection-in-2-minutes-using-opencv-python-90f89d7c0f81>

(<https://towardsdatascience.com/face-detection-in-2-minutes-using-opencv-python-90f89d7c0f81>)

3- Cascade Classifier: [https://docs.opencv.org/master/db/d28/tutorial\\_cascade\\_classifier.html](https://docs.opencv.org/master/db/d28/tutorial_cascade_classifier.html)  
([https://docs.opencv.org/master/db/d28/tutorial\\_cascade\\_classifier.html](https://docs.opencv.org/master/db/d28/tutorial_cascade_classifier.html))