

1- Introduction :

In computer vision, one essential problem we are trying to figure out is to automatically detect objects in an image without human intervention. Face detection can be thought of as such a problem where we detect human faces in an image. There may be slight differences in the faces of humans but overall, it is safe to say that there are certain features that are associated with all the human faces. There are various face detection algorithms so using python and openCV, we will be able to make a face recognition program for recognizing faces in images as well as on a live webcam feed.

2-What is Face Recognition?

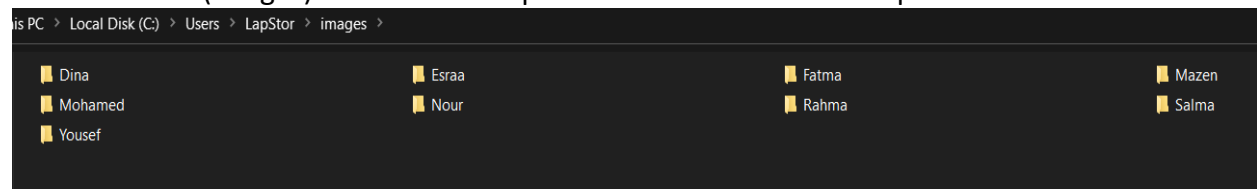
Face recognition is a method of recognizing an individual using their face. Face recognition systems can be used to identify individuals in photos, videos, or in real-time. The main motivation behind building a real-time face recognition system is to enable faster processing and seamless integration in various real-world applications such as public safety systems, authentication systems, retail-store systems, etc.

During the initial days, the speed of recognition of human faces in machines was slow and the accuracy seemed to be lower than manual recognition. However, with the development of deep learning and the increased application of Convolution Neural Network (CNN), the accuracy and speed of human-face recognition by machines are greatly improved.

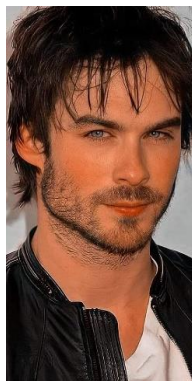
We try to build a real-time face recognition system using a suitable model which is capable of detecting and recognizing images from webcams and cameras accurately in real-time, we aim to develop the system such that it is capable of recognizing faces live, using a laptop webcam or cameras installed in a room but there is another way With OpenCV and Python, through a database, we compare the person's photo and we know how to identify it precisely, we not only identify the person by drawing a box on his face but we also know how to give a precise name.

3- Face Recognition on images:

To make face recognition work, we need to have a dataset of photos also composed of a several image per character and comparison photo. For example, in our example, we have a dataset folder (images) have folders of persons that contain several photos for them:



and in the comparison, we see that:



Will be compared with:



After calling the installed libraries **OpenCV** and **face_recognition** in our project.

With the usual OpenCV procedure, we extract the image, in this case, **Rahma.jpeg**, and convert it into RGB color format. Then we do the “face encoding” with the functions of the Face recognition library,
and doing same procedure for the second image, we only change the name of the variables and obviously the path of the second image, in this case: **images/r.jpeg**.

```
In [2]: img = cv2.imread("images/Rahma.jpeg")
        rgb_img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        img_encoding = face_recognition.face_encodings(rgb_img)[0]

In [3]: img2 = cv2.imread("images/r.jpeg")
        rgb_img2 = cv2.cvtColor(img2, cv2.COLOR_BGR2RGB)
        img_encoding2 = face_recognition.face_encodings(rgb_img2)[0]
```

With a single line, we make a simple face comparison and print the result. If the images are the same it will print True otherwise False.

```
In [4]: result = face_recognition.compare_faces([img_encoding], img_encoding2)
        print("Result: ", result)

Result: [True]
```

Here the result is True → The 2 images are for the same person

Next step is to encode all faces in dataset:

Now we have to encode all the images in our database, so that through the webcam video stream if it finds the match it shows the name otherwise it says “name not found”.

This is a function of the file I have prepared and it simply takes all the images contained in the **images/** folder and encodes them. In our case, there are 9 folders contain images.

```
In [6]: from simple_facerec import SimpleFacerec

In [7]: # Encode faces from a folder
        sfr = SimpleFacerec()

        sfr.load_encoding_images("images/*")

        24 encoding images found.
```

4- Face Recognition in real-time on a webcam:

First we will take webcam stream with a simple Opencv function, then we identify the face passing the frame of the webcam to this function **detect_known_faces(frame)**. It will give us the name of the person's folder and an array with the position at each moment of the movement like this:

```
[136 388 288 244]
[136 388 288 244]
[136 388 288 244]
[136 388 288 244]
[136 404 288 268]
[136 404 288 268]
[136 404 288 268]
[136 404 288 268]
[136 428 288 276]
```

Now we need to retrieve the region of interest (ROI).

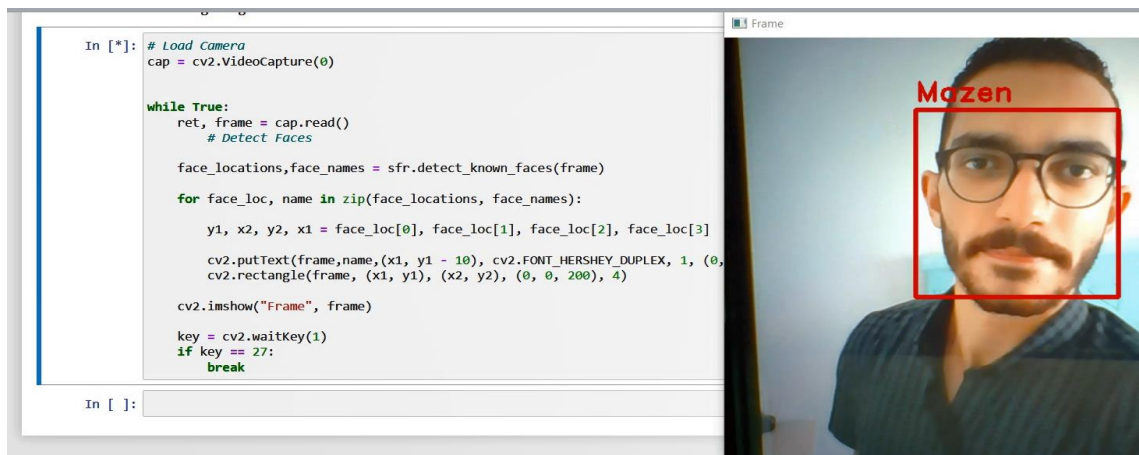
To do this we need to find the coordinates of:

-top left point (x1, y1)

-right bottom (x2, y2)

That we will use to show name and rectangle.

Here's how it shows Mazen and his name.



4- Some tests with facial recognition:

For the real-time test, I put the photos on my phone and showed them to the webcam. Even if the lighting conditions are not perfect it works quite well and obviously, in real conditions, it will work much better.

