Authentication based on Liveness test

The application I made consist of main three modules:

1. Face Recognition
2. Capture the face through webcam(liveness test),
3. Face Comparison

Face recognition:

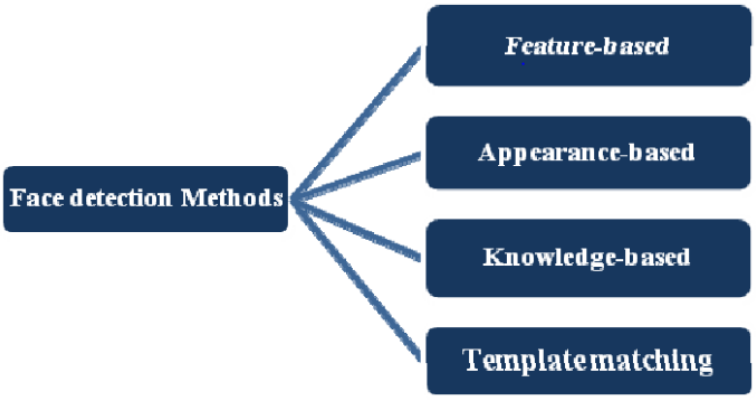
In last few years, face recognition is of the most used and one of the most admired work done in the field of image processing. Face recognition has many sub parts but the face detection is considered as one of the initial task. The technique of face detection in photos is difficult due to variability present across human faces like position, expression, pose, and orientation, skin tone, worn spectacles or hairs across face like beard, differences in camera gain, amount of light, resolution and picture quality.

Object detection is the computer technology that is link with to the computer vision technology and image processing. It interacts with identifying category of objects that fall in picture like human faces, architectural construction, flora and fauna, vehicles, etc. The primary goal of face detection algorithms is to identify if there is any human face present in an image or not.

In past few years, people have proposed considerable amount of potential in the field of Face Detection and Recognition so it becomes advanced and results are more accurate.

Face Detection Methods:-

Group of Yan, Kriegman, and Ahuja gave a classification for different face detection methods. Methods are classified into four categories, and there are chances that face detection algorithms could belong to two or more categories. These categories are as follows-



1.Knowledge based:-

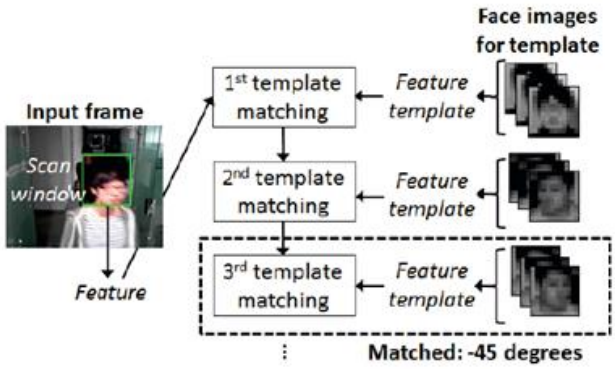
The knowledge-based method is based on human knowledge to detect the faces and accordingly the set of rules are formed. Example- As we all know, A face consist of a nose, eyes, and mouth within certain distances and positions with reference to each other. The main challenge with these methods is defining an appropriate set of rules. We cant make rules more general as it gives more number of False positive. SO this method standalone wont work perfect, It needed to merger with other method.

2.Feature Based:-

This method extracts structural features of the face and locates is. Before using it to differentiate facial and non-facial regions, It is trained as a classifier. This is to overcome the limits of our instinctive knowledge of different human faces. The whole approach is divided into several steps. They report a success rate of 94% in even photo with multiple faces.

3.Template Matching:-

This method already has pre-defined face templates. It uses those templates to locate or detect the faces by the correlation between the templates and input images. This approach is simple to implement, but it is not appropriate and accurate for face detection. However, we can use deformable templates to overcome these problems.



4.Appearance-Based:-

This method is used to find out face models. The method uses a set of delegate training face images to complete the task. This method is better in case of performance compare to other. It is depended on machine learning techniques and techniques from statistical analysis. It find the appropriate characteristics of face images. This method also used in for face recognition for feature extraction.

It has few sub-methods:

4.1. Eigenface-Based:-

4.2. Distribution-Based:-

4.3. Neural-Networks:-

4.4. Support Vector Machine:-

4.5. Sparse Network of Winnows:-

4.6. Naive Bayes Classifiers:-

4.7. Hidden Markov Model:-

4.8. Information Theoretical Approach:-

4.9. Inductive Learning:-

How does face recognition works?

Technologies vary, but here are the basic steps that are followed:

Step 1. A picture of is captured of your face either from photo or from video stream. There might me your picture alone or multiple other faces in picture. Your picture might have a straight face or side profile up to certain degree.

Step 2. The software used for face recognition reads the geometry of your face. There are some Key factors which include the distance between your eyes and few others. All the faces have facial landmarks and the software identifies them. The system identifies 68 of them. Your face is distinguished based on these landmarks. As a result you get your unique face signature.

Step 3. Thee is pre-loaded database of know Faces. The facial signature found in the above step is now compared with those images in database.

Step 4. Determination is mada and there are chances that your face signature may match with the image present in the existing database in a facial recognition system database.

Capture photo from the Live webcam on Smile to verify the liveness:

Process Overview

1. Using the dlib library, we would get to get the mouth coordinates using facial landmark detector.

2. Use mouth aspect ratio (MAR)

3. Live stream using Webcam

4. Click picture on smile

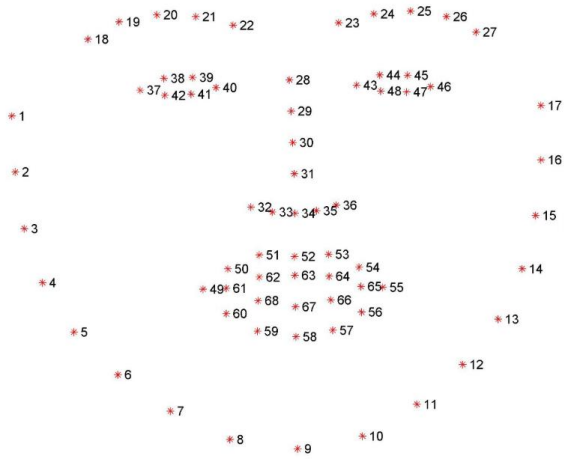
5. Saving the picture

6. Close webcam feed

Facial landmark detector

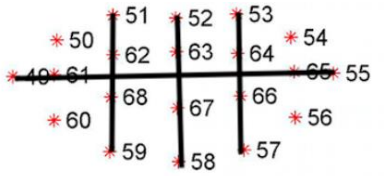
The facial landmark detector is an API pre-loaded in dlib library. It gives 68 x- y-coordinates of the face marking landmarks.

This would look something like:



As we are going to capture the image on SMILE, we would focus mainly on coordinates over the mouth [49,…,68].

I have created here the Mouth Aspect Ratio (MAR) so that we can calculate a range from which we give the webcam order to click the picture.



$$\frac{\|p_{51} - p_{59}\| + \|p_{52} - p_{58}\| + \|p_{53} - p_{57}\|}{3\,\|p_{48} - p_{55}\|}$$

MAR equation

We would write MAR=L/D.

As per this, I considered smile to be between the MAR range of <.3 or >.38. D will not be same for all, as people have different mouth shapes, so we cannot just consider D as value.

Video capture

Access the webcam:

cv2.namedWindow command will open webcam in new window.

Face Detection

As a First step, we click a single picture and convert that picture to grayscale for simple computation.

Auto-capture

When the MAR value I between 0.3 and 0.38 for continuous 15 frame, we click the 16th frame.

Quit video streaming

We press ‘q’ when we need to quit the video stream window.