Course Title: Computer Vision

Face Identification and Recognition

Submitted By:

Haris Ali Akhter (FA17-BCS-028)

Hamza Zaheer (FA17-BCS-026)

Submitted To: Ms. Shehla Saif

Dated: 5th August, 2020

Project Title: Face Identification and Recognition

Reasoning Behind The Project:

Facial recognition is a powerful technology, it brings immense advantage to the companies and end-users, helps them enhance their security and track down the trespassers. Humans have always had inmate ability to recognize and distinguish between faces. Computers being able to do the same now has opened tons of applications. A human interaction system for an automatic face recognition or facial expression detection has drawn the increasing attention from researchers in linguistics, computer science, neuroscience, psychology and related disciplines.
Face detection is used to improve the **Access and Security** of various hand held devices that we use daily in our lives. The **processing time** of recognizing the face takes a second or less and this is immensely fruitful to big companies and firms. Moreover, one of the biggest advantage of this technology is that it is **easily integrated**. A great chunk of **money is saved** on its integration. Before, security staff used to perform manual identification of a person that was time consuming and didn't raised accuracy. Facial recognition, on other hand, is completely **independent** in the identification process and not only takes seconds but is also incredibly accurate.

Methodology:

The project consists of two parts i.e. training the model on our dataset and then real time testing to identify and also recognize the faces. The first step is to create a dataset for the model to train on. This can be done manually and also through code. Images are stored in a directory named *Images* under a folder whose name is the label later used for identification. So for someone named *Ali* there images would be stored in the path /Images/Ali.

We are using the *Local Binary Pattern Histogram (LBPH)* algorithm in our project. LBPH is a simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number. Combined with histograms, we can represent the face images with a simple data vector. This algorithm will be used to train the model and test for recognizing the images.

Why Choose LBPH:

We have decided to use this algorithm because it the simplest of the face detection algorithms and is also easy to implement and understand. Firstly we wanted to use Eigenfaces but have changed our decision because this algorithm has given us better results compared to Eigenfaces.

Explanation:

We firstly do some data cleaning on our dataset i.e. providing labels for each image based on the label of the folder i.e. by taking the base name of the path. We are also using the *haar cascade classifier*.

We have also assigned an id for the labels i.e. 0, 1s and 2s etc. for the folder names which will be used to identify the images inside that folder. Then we are converting all the images into grayscale one by

one, resizing them and applying anti-aliasing filter on each image to soften the edges. Then we represent the image in a single data vector after which we use the *detectMultiScale* method of haar cascade to detect the face. We store the data into x and y list, x will be the image and y will be the label. We store the labels in a file and then train our data on the images. After training we store it into a file.

Next we detect the region of interest i.e. face from the images to further refine our data. This will help in increasing the accuracy of the face detection.

Now we move to the testing part of the project. First we read the file in which our trained data has been stored. We load our label dictionary as well now. We now use the opencv to start our video capturing, converting the frames into grayscale. The *detectMultiScale* method is used to detect any faces present in the video. Next we use our classifier to predict the face. The values we get are the id and the confidence values. Confidence is sort of like a percentage and if it greater than 45 then we can say that the algorithm is detection the face correctly.

If the confidence level is low, we can increase by generating a better dataset and we have done our best to generate a decent dataset given the low resources. Then the final step is to simply provide the label for the detected face which does have a slight chance of being wrong because of various factors such as lighting.

Limitations:

The limitations of the project are the following:

- Natural factors like lighting
- Not enough diversity of the dataset

Low lighting can provide wrong results in the face recognition part and sometimes it will either not detect the person or detect them but with the wrong label. This can be improved by taking pictures in various lighting and scenarios.

The same goes for diversity of dataset such as pictures from different angels, pictures at different stages and pictures at different times of the day etc.