**Face Recognition Project**

**Main libraries used**:

Opencv

face\_recognition(Built using dlib’s state-of-the-art face recognition built with deep learning)

**The pipeline of the face recognition is as follows**:

first, look as a picture and find all the faces in it.

Second, focus on each face and be able to understand that even if a face is turned in a weird direction or in bad lighting, it is still the same person.

Third, be able to pick out unique features of the face that you can use to tell it apart form other people-like how big the eyes are , how long the face is, etc.

Finally, compare the unique features of that face to all the people you already know to determine the person’s name.

So, to encode faces in images we first need to detect them either by HOG or CNN. Between these two, HOG is faster than CNN but less accurate than CNN.

Encoding of faces is done using the pre-trained model(model is a ResNet network with 29 conv layes).

The network was trained form scratch on a dataset of about 3 million faces. On the Labeled Faces in the Wild (LFW) dataset the network compares to other state-of-the-art methods, reaching 99.38% accuracy.

**About LFW**:

Labeled faces in the wild , a database of face photographs designed for studying the problem of unconstrained face recognition. The data set contains more than 13,000 images of faces collected form the web. Each face has been labeled with the name of the person pictured. 1680 of the people pictured have two or more distinct photos in the dataset.

For making our own face recognition system for the employees of our company, we collected 5 to 6 images of everyone in differect poses.

After testing when we trained by 4 encodings or more i.e., 4 images of an employee and tested on other images, it gave 100% correct reuslts (tested on 132 employees).

**Similarity methods used**:

To find the similarity between the known encoding and test encoding we used two methods. In one method we used numpy library of pyton to convert the face encodings into a number i.e., it’s magnitude, then we calculated the distance between the magnitude of known encoding and unknown encoding, if it’s difference is less than the threshold, then it is considered as a match. Face id which got maximum matches is given as the final result.

In the second mehod we used KNN model trained it on the encoding of the known faces and Face encoding which is nearest to the test encoding is considered as a match as considered as result. It also gave 100% results when trained with more than 4 image encoding of an employee.

The results we have got on the test images,which are taken in same environment.

**Detailed results are as follows:**



