PROJECT REPORT

Entitled

"FACE RECOGNITION USING DEEP LEARNING"

Submitted to the Department of
Electronics & Communication Engineering
In Fulfillment of the Completion of
Summer Research Internship (2019-20)

: Presented & Submitted By :

Mr. Dhruv Rajendra Patel (Roll No. U16EC053)
B. TECH

: Guided By :

Dr. K.P.UPLA

Assistant Professor, ECED

(Year : 2019-20)

DEPARTMENT OF ELECTRONICS ENGINEERING
Sardar Vallabhbhai National Institute of Technology
Surat-395007, Gujarat, INDIA.

Sardar Vallabhbhai National Institute of Technology

Surat-395 007, Gujarat, INDIA.

ELECTRONICS & COMMUNICATION ENGINEERING DEPARTMENT

CERTIFICATE

This is to certify that the **PROJECT REPORT** entitled **"FACE RECOGNITION USING DEEP LEARNING"** is presented & submitted by Candidate **Mr. Dhruv Rajendra Patel** bearing **Roll No. U16EC053** in the fulfilment of the completion of **Summer Research Internship (2019-20)** in **Electronics & Communication Engineering Department.**

He has successfully and satisfactorily completed his Research Project in all respect. We, certify that the work is comprehensive and complete.

Dr. K.P.Upla
Project Guide
Assistant Professor

Dr. A.D.Darji
Associate Professor &
Head, ECED.

S.V. National Institute of Technology * SURAT * ECED

DEPARTMENT SEAL

ACKNOWLEDGEMENT

I would like to express my utmost gratitude to **Dr. K.P.Upla**, **Assistant Professor**, **Electronics & Communication Engineering**, **Sardar Vallabhbhai National Institute of Technology** for providing his valuable assistance and guidance throughout the duration of this project. In addition, I feel obliged to use the lab facilities and the resources available at the **Image Processing & Computer Vision Laboratory, Department of Electronics & Communication Engineering, SVNIT**. I would also like to expand my deepest gratitude to all those who have directly and indirectly guided me.

Dhruv Patel

(ROLL NO.: U16EC053)

ABSTRACT

Biometrics refers to unique human characteristics. Each unique characteristic may be used to label and describe individuals and for automatic recognition of a person based on such characteristics. One of the most natural and the most popular biometric trait is a face. Such type of systems can be helpful for security purposes in military, payment transactions, access and security purposes, criminal identification, health care, advertising etc. So concluding, its applications in every of the mentioned field is increasing. So, we intend to develop a system which detects and recognizes the faces in real-time. Face detection is performed on live real-time images. Then a classification method is integrated with the system which provides the identity from the dataset generated with 25 people. The developed system has acceptable performance on the generated dataset within intended limits.

INDEX

1 INTRODUCTION

1.1 PROBLEM STATEMENT

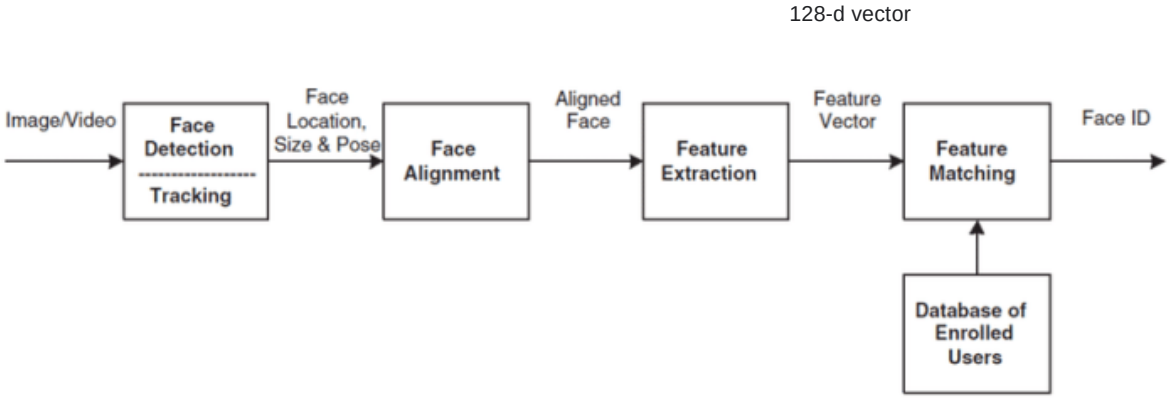
The Face Recognition System deals with detection and locating of the faces in real-time live feed of video frames and thereby using a CNN model to recognize or generate the labels for the corresponding faces detected. In order to effectively execute said process, the model must have information (facial features) of the identities or it must be trained with the images of the identities that it want to recognize. Various modules like Face detection, Alignment, Face Recognition model and its training with the custom dataset to execute the whole process is discussed in the report.

1.2 AVAILABLE APPROACHES

The available approaches for building a face recognition system can be classified into holistic, feature-based, model-based and hybrid approach. Holistic approach mainly deals with the whole facial similarities between classes based on AI and statistical approaches. Feature based approach deals with individual features like eyes, mouth etc. and extracting features by edge detection. Model based approach deals with 3D models generated by infrared or stereo imaging for depth sensing. Hybrid approach combines the holistic and feature based approach to extract information by passing the data of the entire face into the model developed.

Each approach is discussed in detail in the *Related Work* Section. Hybrid approach was chosen for the system that was required to develop due to its advantages and high performance over the other approaches discussed.

1.3 SYSTEM OVERVIEW



In order to achieve the complete functioning, the entire system can be divided into 3 subsections:

1. Face detection in a video frame
2. Cropping and aligning the detected faces.
3. Face Recognition using CNN Model.

Face detection involves detection and location of faces in an image. Face Cropping and Alignment involves the cropping of faces based on coordinates located by the detection model and thereby aligning it. Now comes the face recognition part, which identifies the faces based on the similarities of the embedding vectors of the labels in the dataset.

2 RELATED WORK

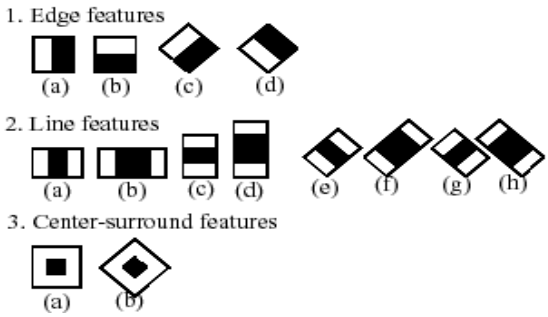
2.1 FACE DETECTION

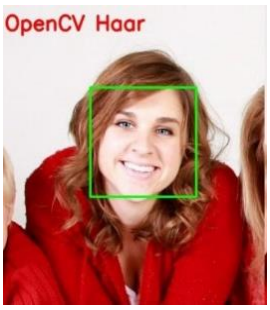
For building a face recognition system, face detection in an image or a video frame is an integral part as the system should be able to detect the faces in the image so as to recognize its identity through the CNN model designed. As it is the initial step in the system, the face should be detected in every conditions like different orientations, low illumination, glasses, facial hair and so on. Some of the face detection models available are:

1. Haar Cascade Face Detector OpenCV
2. HoG Face Detector dlib
3. CNN Face Detector dlib
4. MTCNN based Face Detector

2.1.1 Haar Cascade Face Detector OpenCV

It is a ML based approach trained from many positive images (with faces) and negative images(without faces).It extracts Haar features from each image by the below feature windows:

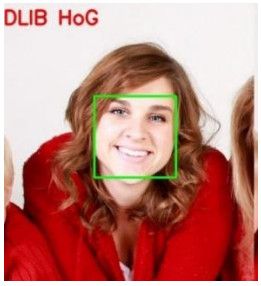




Each of the above mentioned window is applied on an image and the feature is calculated by subtracting the sum of pixels in white part from the black part. Still there are many irrelavent features in the image. So instead of applying all the features on a window, the features are grouped into different stages of classifiers one-by-one and if the windows fails we don’t apply the rest features on it else we apply the rest features in cascade fashion. It does not work under occlusion and detects only frontal images or faces. Apart from this, it also provides false predictions.

2.1.2 HoG Face Detector dlib

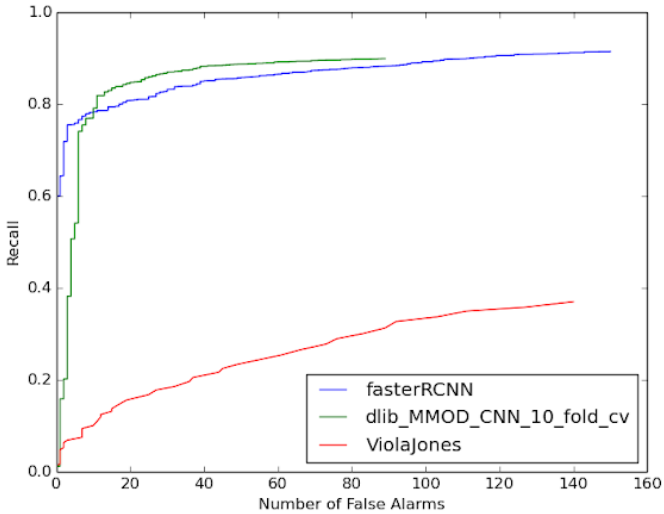
It is the fastest method on CPU and works on slightly non-frontal and frontal faces. In this type of detector, a feature descriptor i.e. Histogram of Oriented Gradients is generated from the image and then is fed into an image classification algorithm like Support Vector Machines(SVMs) to produce good results. In HoG descriptor, the histograms of directions of gradients is used as features. Gradients of an image is useful as it provides more information regarding the edges and the corners.



It does not work for side face and extreme non frontal faces. It also fails under substantial occlusion.

2.1.3 CNN based Face Detector dlib

It uses a Maximum-Margin Object Detector (MMOD) with CNN based features. It inherits very low training data requirements.



From the graph above, X axis is the no. of false alarms produced over the entire FDDB dataset of 2845 images and Y axis is the fraction of faces found by the detector. Most interestingly, the blue curve is a state-of-the-art result from the paper Face Detection with the Faster R-CNN. The red line(dlib CNN) in the graph shows better results than the blue line even after getting trained on only 4600 faces rather than the blue line which is trained on 159,424 faces.

It is more robust to occlusion and is able to detect faces at angles also. But it requires high processing power.

2.1.4 MTCNN based Face Detector

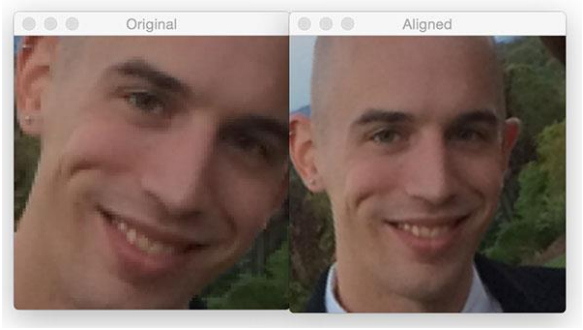
In Multi-Task Cascaded Convolutional Neural Network, Face detection and Alignment are done jointly in a multi-task training fashion.This ideally improves the performance of the system. In the "FaceNet: A Unified Embedding for Face Recognition and Clustering" , it uses MTCNN model to detect facial landmarks and align it as the paper mentions the failure of dlib detectors under partial occlusion, silhouettes, etc. But the real-time results of MTCNN showed that it misses to detect faces under low illumination conditions and thereby reducing the performance.

MTCNN Face Detector

Face Detector Used: *CNN based Face Detector dlib* is used here. The reasons for choosing this detector are as: more robust to occlusion, large detection angles. One of the cons of choosing it can be said to be requirement of high processing power.

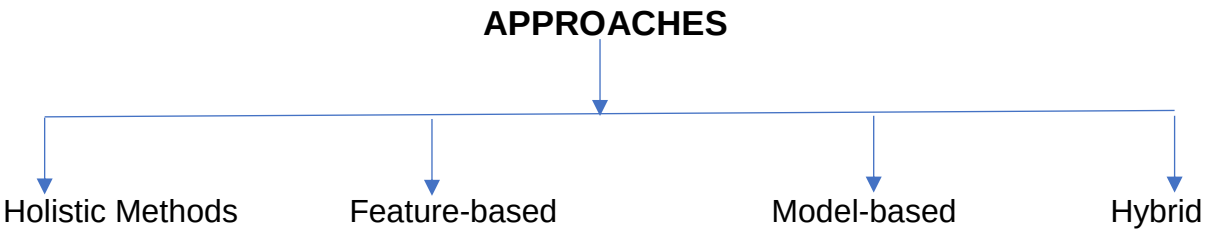
2.2 FACE ALIGNMENT

To attempt a canonical alignment on the basis of translation, rotation and scale of the face. It is a kind of data normalization which is similar to setting a feature vector via zero centering or scaling prior training it. It is very common to align the faces before training the face recognition model. It increases the performance of the model as the facial features are at the same location in the cropped image which results into more generation of more accurate embedding.



Facial landmarks are detected with shape predictor and then the facial image is thereby aligned by taking the location of facial landmarks into consideration.

2.3 FACE RECOGNITION



2.3.1 Holistic Approach:

Complete face is taken as a single feature for detection and recognition. It compares facial similarities ignoring the individual features like eyes, mouth, face etc. It includes AI based approach and Statistical approaches.

2.3.2 Feature-based Approach:

It works by means of hexagonal features detection. It works on the basis of edge detection for face recognition using hexagonal features. This approach mainly focused on nose portion of the acquired images followed by gray scale conversion and transformation of intensity. Fiscal features like eyes, nose and mouth were taken as point and Gabor filter applied for feature extraction. HSV colour space is used for extracting skin features using hue and saturation values.

2.3.3 Model-based Approach:

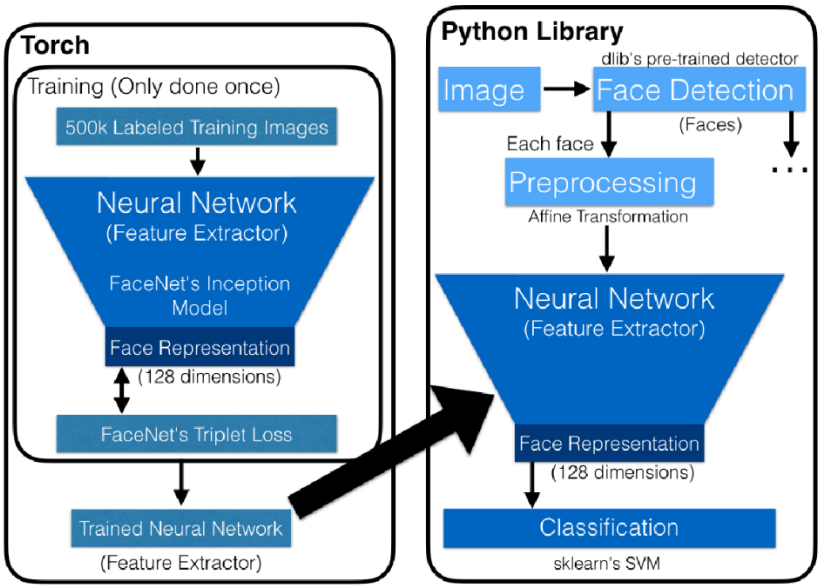
This approach mainly relies on the 3D image of the subject obtained by infrared laser projection beam or stereo imaging which uses 2 or more cameras for sensing the depth information of the image. It is computationally intensive and requires installation of multiple cameras.

2.3.4 Hybrid Approach :

It is mainly a combination of holistic and feature based approach. It firstly extracts features like nose location, its tip etc. from the image and applies it to a model which is trained using a training algorithm and is compared to a training set.

Approach Used:

Hybrid Approach is used here considering its performance and its advantages over the other approaches discussed above. The system initially takes an image, detects faces and crops it, aligns the face to fit in the model developed. The CNN model then predicts a given 128d embedding. The similarity of the face in the image is compared in the custom dataset generated by using the Euclidean distance and if it comes out under a threshold set, the identity is obtained with image in the dataset that matches with more similarity.



L2 Norm(Euclidean distance):

L2 Norm or the Euclidean distance stated as above is given by the formula below.

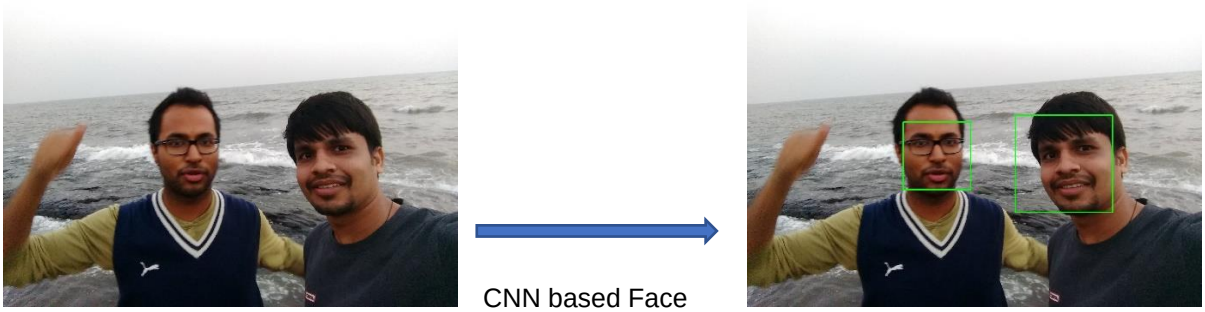
$$d_{L2}(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Here, xi can be considered as the input image and yi maybe any image in the dataset.

3 METHODOLOGY

3.1 FACE DETECTION

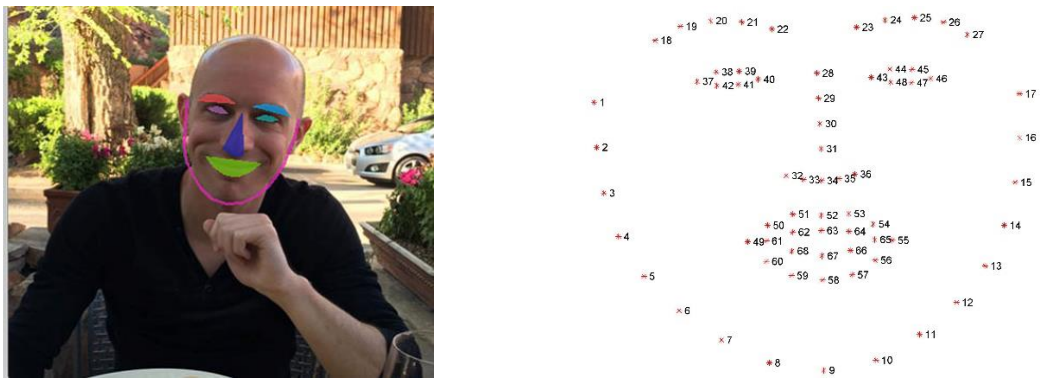
From all the models discussed above in the Related Work section, *CNN based face detector from dlib* is used because of its robustness in detecting face at odd angles, and independency of the occlusion.



As shown in the image above, input image is passed to the CNN Face Detector Dlib. The detector produces the bounding box around the face and fetches the coordinates of the bounding box.

3.2 FACE ALIGNMENT

The Facial image is then cropped according to the coordinates of the bounding box. This cropped image is then aligned using the 68 facial landmarks wherein position of the facial features like eyes, nose etc. tried to kept at the same location in the image.



68 Facial Landmarks represented

For aligning purpose, the facial landmarks model used here is the *shape_predictor_68_face_landmarks.dat.* This model uses 68 facial points like the mouth, eyebrows, eyes, jaw, nose etc.

Face Cropped and aligned from the coordinates of the bounding box.

3.3 FACE RECOGNITION

In this part, the face is feeded into the Neural Network and a face embedding(i.e. a vector that represents features extracted from the face) is generated. For identification of the face, there should be a face of a particular class in the dataset whose face embedding is close to the generated embedding.

This system uses FaceNet. FaceNet is a system that directly learns a mapping from face images to a compact Euclidean space where distances directly correspond to a measure of face similarity. FaceNet maps a face into a 128D Euclidien space. The L2 distance(or Euclidien norm) between two faces embeddings corresponds to its similarity. This is exactly like measuring the distance between two points in a line to know if they are close to each other.

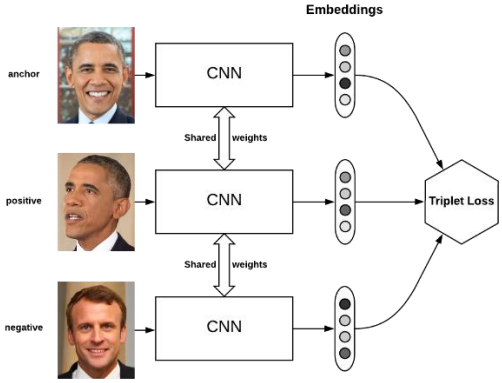
$$d_{L2}(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

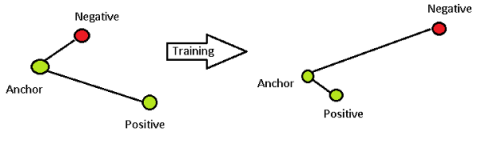
3.3.1 Why FaceNet ??

It achieves state-of-the-art face recognition performance using only 128-bytes per face. On the widely used Labeled Faces in the Wild (LFW) dataset, it achieves a new record accuracy of 99.63%. On YouTube Faces DB it achieves 95.12%. The system cuts the error rate in comparison to the best published result by 30% on both datasets. FaceNet, directly learns a mapping from face images to a compact Euclidean space where distances directly correspond to a measure of face similarity. Once this space has been produced, tasks such as face recognition, verification and clustering can be easily implemented using standard techniques with FaceNet embeddings as feature vectors.

3.3.2 Training a FaceNet Model

To train a Facenet model, a triplet of images is generated from the dataset-anchor,positive and negative. An anchor is the test image, positive is a different image of the same class as anchor and negative represents image from a different class in the dataset. The idea is to minimize the Euclidien distance between anchor and positive images and increase the distance between the anchor and the negative image during training.





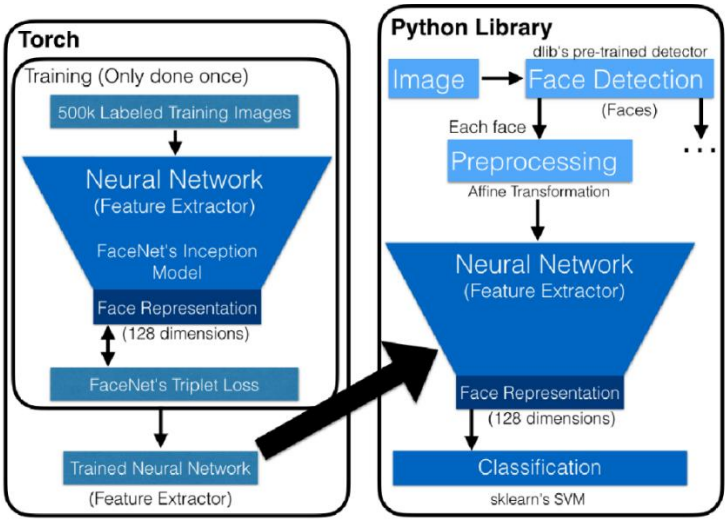
After training using triplet loss

$$Loss = \sum_{i=1}^{N} \left[\| f_i^a - f_i^p \|_2^2 - \| f_i^a - f_i^n \|_2^2 + \alpha \right]_+$$

a->anchor, p->positive, n->negative. Alpha here works as a threshold value in the loss equation. Alpha -> Similarity.

3.3.3 CNN ARCHITECTURE

The architecture used is a variant of nn4 inception architecture identified as nn4.small2 model specified in the OpenFace project. OpenFace is an open source library that drives the performance of the private state-of-the-art systems. It focuses on the real-time performance of the system with very less data available for training.



Comparison between different CNN Models

Looking at the below image, nn4.small2 has the least no. of parameters as it can be thought of from the name itself.

Model Definitions

The number of parameters are with 128-dimensional embeddings and do not include the batch normalization running means and variances.

Model	Number of Parameters
nn4.small2	3733968
nn4.small1	5579520
nn4	6959088
nn2	7472144

3.3.4 Accuracy and Performance Comparisons

As shown in the image below, the runtime for nn4.small2.v1 is the least on CPU as well as GPU compared to its variants. Also looking at the Accuracy obtained on LFW datasets, all the versions of nn4 architecture have comparable accuracy but still nn4.small2.v1 leads the chart looking at the highest accuracy obtained.

Performance

The performance is measured by averaging 500 forward passes with util/profile-network.lua and the following results use OpenBLAS on an 8 core 3.70 GHz CPU and a Tesla K40 GPU.

Model	Runtime (CPU)	Runtime (GPU)
nn4.v1	75.67 ms ± 19.97 ms	21.96 ms ± 6.71 ms
nn4.v2	82.74 ms ± 19.96 ms	20.82 ms ± 6.03 ms
nn4.small1.v1	69.58 ms ± 16.17 ms	15.90 ms ± 5.18 ms
nn4.small2.v1	58.9 ms ± 15.36 ms	13.72 ms ± 4.64 ms

Accuracy on the LFW Benchmark

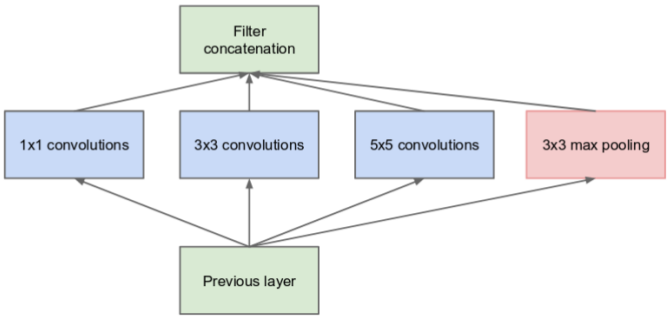
Even though the public datasets we trained on have orders of magnitude less data than private industry datasets, the accuracy is remarkably high on the standard LFW benchmark. We had to fallback to using the deep funneled versions for 58 of 13233 images because dlib failed to detect a face or landmarks.

Model	Accuracy	AUC
nn4.small2.v1 (Default)	0.9292 ± 0.0134	0.973
nn4.small1.v1	0.9210 ± 0.0160	0.973
nn4.v2	0.9157 ± 0.0152	0.966
nn4.v1	0.7612 ± 0.0189	0.853
FaceNet Paper (Reference)	0.9963 ± 0.009	not provided

3.3.5 NN4 INCEPTION ARCHITECTURE

Total parameters of the model: 37,43,280.

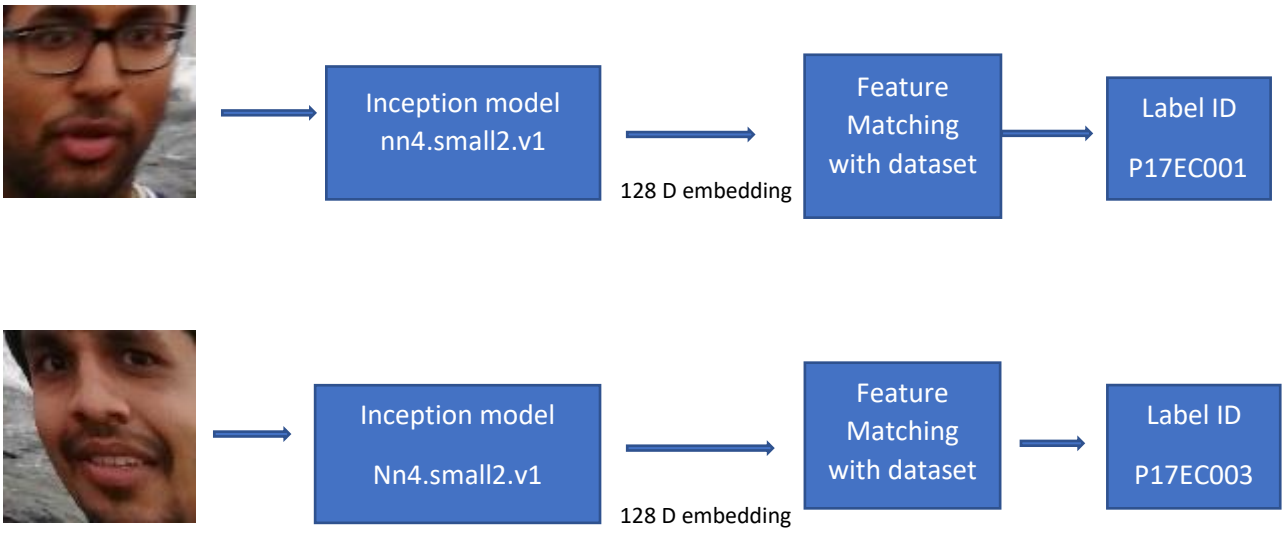
Looking at the above architecture, though it is mentioned above that it is the smallest architecture among then nn variants but it is quite dense. Here, is the basic representation of the main feature of the inception network.



(a) Inception module, naïve version

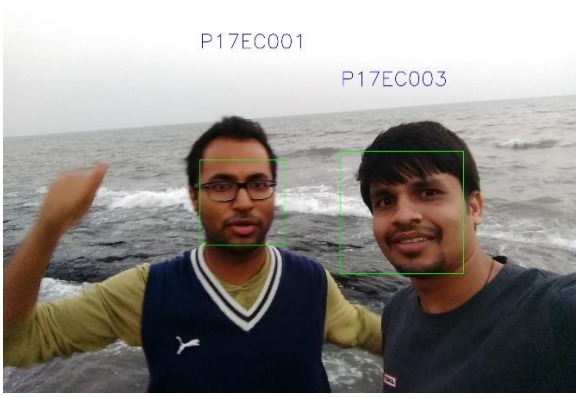
The network here concatenates the outputs obtained from various filters like 1*1, 3*3, 5*5 and concatenates them using pooling to provide the output to the next layer.

3.3.6 Final Step:



Feature Matching with the dataset involves finding the maximum similarity of an image corresponding to a label(among 17 labels) in the dataset. This involves finding the Euclidean distance or L2 norm of all the images in the dataset. And label corresponding to maximum similarity represents the identity of the person.

Final Output :




```
Total params: 3,743,280
Trainable params: 3,733,968
Non-trainable params: 9,312
_________________________________________________________________
P17EC001
2019-08-26 04:05:52.555754: I tensorflow/stream_executor/
P17EC002
P17EC003
P17EC004
P17EC005
P17EC006
P17EC007
P17EC008
P17EC009
P17EC010
P17EC011
P17EC012
P17EC013
P17EC014
P17EC015
P17EC017
(1637,)
it's P17EC001, the distance is 0.15042363
P17EC001
it's P17EC003, the distance is 0.26595327
P17EC003
```

Some of the predicted labels by the model are:

The model at present predicts the face according to the L2 Norm of its embedding with faces of 17 labels present in the dataset.

4 RESULTS

Total no. of parameters: 37,43,280.

Trainable parameters: 94,336

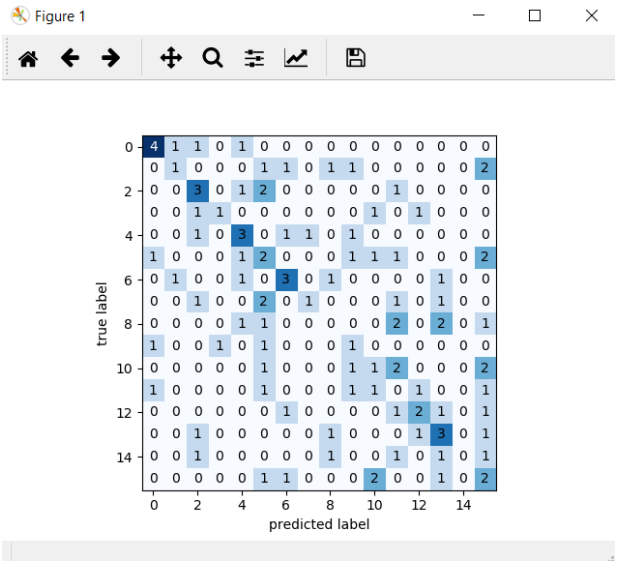
Non-Trainable parameters: 36,48,944.

As the no. of parameters of the model are very large, it is not possible to train the entire model with custom generated dataset as these type of models are trained with millions of images with public datasets from YouTube, Google etc. So, transfer learning was used to load the model with pretrained weights before starting the training. Initial layers of the model were kept fix as it did the task of feature extraction and only the final layers of the network were trained to avoid overfitting.

Confusion Matrix is used to plot or show the predicted and the true label. The diagonal entries having more value would lead to greater accuracy.

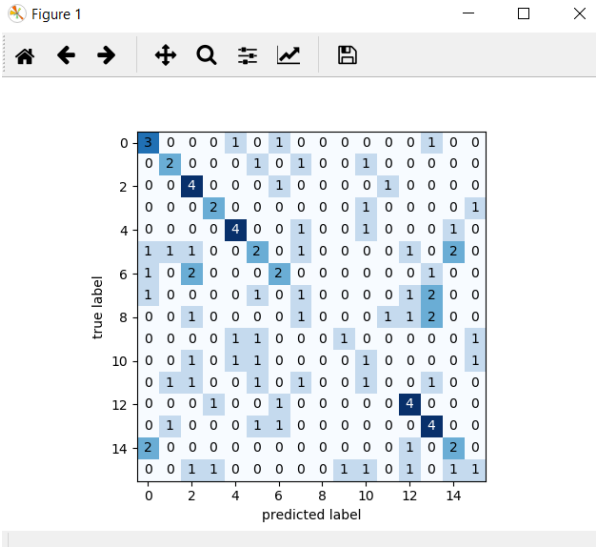
4.1 Comparison of Accuracy and Confusion Matrix over no. of iterations:

1. **100 iterations**



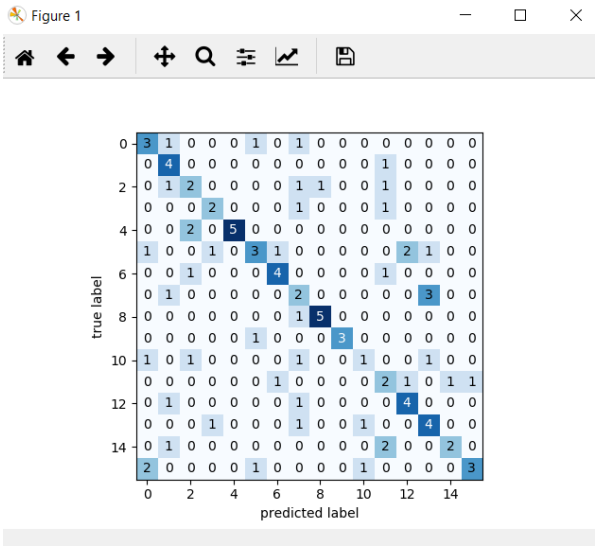
Accuracy: 27 faces correctly recognised out of 96 facial images (28.96%)

2. 300 iterations



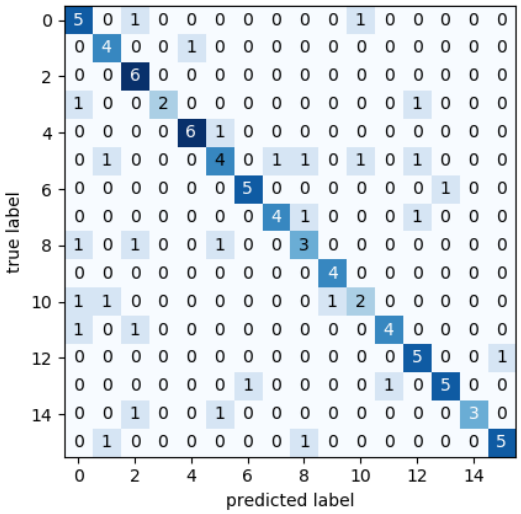
Accuracy: 33 images out of 96 correctly recognised (34.375%)

3. 500 iterations



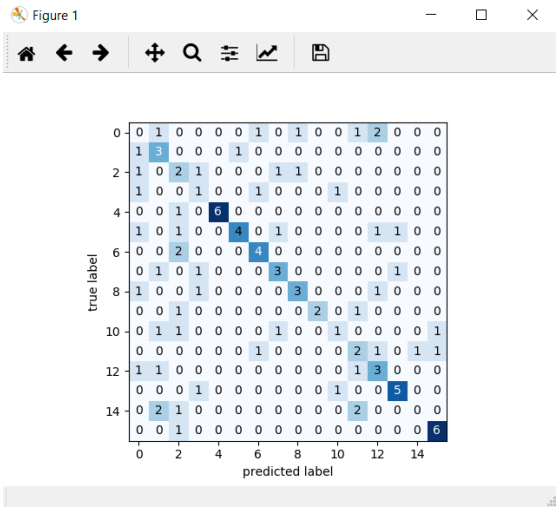
Accuracy: 49 images correctly recognised out of 96 (51.04%)

4. **1000 iterations:**



Accuracy: 67 faces correctly recognised out of 96 (69.79%)

5. **1200 iterations:**

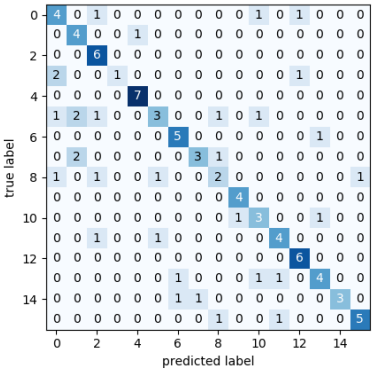


Accuracy: 45 faces correctly recognised.(46.875%)

CONCLUSION: Model starts to overfit after 1000 iterations .

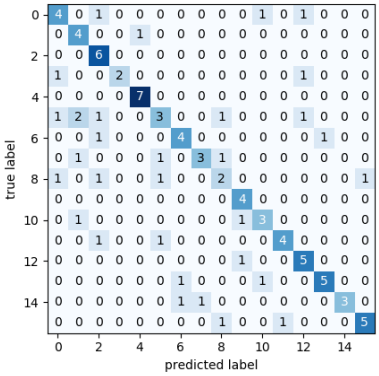
4.2 Comparison of Accuracy and Confusion Matrix over alpha:

1. **Alpha = 0.2**



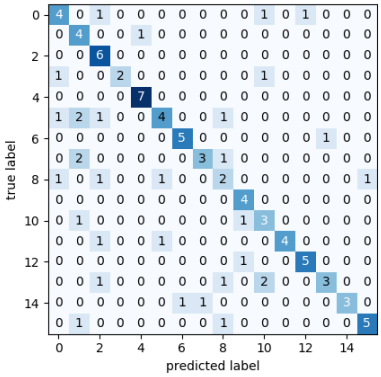
Accuracy: 66.67%

2. **Alpha = 0.3**



Accuracy: 66.67%

3. **Alpha = 0.4**



Accuracy: 64.583%

Model Accuracy does not vary significantly with change in value of alpha.

4.3 HYPERPARAMETERS

Optimizer: Adam

Alpha: 0.2

Learning rate: 0.001 (default)

beta_1: 0.9, **beta_2**:0.999 (default)

epsilon: 1e-8 (default)

Decay: 0.0

Batch Size: 128

Steps per Epoch: 50

Accuracy Achieved: 69.79%

Github: https://github.com/Dhruv2012/Deep-Learning

Dataset: https://drive.google.com/open?id=11AkGtS_zUU9ORqcWYhhtiJkZUOAPreA4

REFERENCES

1) Y. Bengio, J. Louradour, R. Collobert, and J. Weston. *Curriculum learning*. In Proc. of ICML, New York, NY, USA, 2009.
2) I. J. Goodfellow, D. Warde-farley, M. Mirza, A. Courville, and Y. Bengio. *Maxout networks*. In In ICML, 2013.
3) G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. *Labeled faces in the wild*: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.
4) F.Schroff, D. Kalenichenko, J.Philbin. *FaceNet: A Unified Embedding for Face Recognition and Clustering,* Google Inc.
5) Davidsandberg. Github Repository FaceNet available at https://github.com/davidsandberg/facenet.
6) TessFerrandez. *FaceNet Paper Review* available at https://github.com/TessFerrandez/research-papers
7) B.Amos, B.Ludwiczuk, M. Satyanarayanan. *OpenFace.* CMU School Of Computer Science, 2016.
8) Davis King. *Dlib: Face Detection and Alignment.*
9) D. Chen, S. Ren, Y. Wei, X. Cao, and J. Sun. Joint cascade face detection and alignment. In Proc. ECCV, 2014.
10) Ian Goodfellow, Yoshua Bengio, Aaron Courville. *Deep Learning* available at www.deeplearningbook.org