Analysis on Face Recognition based on five different viewpoint of face images using MTCNN and FaceNet

By

Al-imran
15201007
Baniamin Shams
15301030
Faysal Islam Nasim
15201051

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science and Engineering

Bachelor in Computer Science
Department of Computer Science and Engineering
Brac University
August 2019

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Al-imran
15201007

Baniamin Shams
15301030

Faysal Islam Nasim
15201051

Approval

The thesis titled "Analysis on Face Recognition based on five different viewpoint of face images using MTCNN and FaceNet" submitted by

1. Al-imran (15201007)
2. Baniamin Shams (15301030)
3. Faysal Islam Nasim (15201051)

Of Summer, 2019 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science and Engineering on August 7, 2019.

Examining Committee:

Supervisor:
(Member)

Dr. Muhammad Iqbal Hossain
Assistant Professor
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam
Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Dr. Mahbubul Alam Majumdar
Professor and Chairperson
Department of Computer Science and Engineering
Brac University

Abstract

Despite significant recent achievements in the field of face recognition, implementing proper face recognition system by training enough data is still a problem because not everyone has enough photos that we can use to train. The objective of this paper is to implement a proper face recognition system which can successfully recognize known and unknown person by feeding only five phase images for each person into training dataset. In this field, accuracy and speed of identification is the main issue. There are at least two reasons for the importance behind the research of face recognition which has recently received significant attention, especially during the past several years. The first is the wide range of commercial and law enforcement applications, and the second is the availability of feasible technologies after 30 years of research. In this paper, we present a review of the most successful existing method FaceNet for face recognition technology and how we can use it successfully even though we don't have enough data to train and to encourage researchers to embark on this topic. A brief on general information on this topic is also included to compose an overall review. This review is written by investigating past and ongoing studies done by other researchers related to the same subject.

Keywords: Face Recognition; Machine Learning; FaceNet; MTCNN; Convolutional Neural Network

Acknowledgement

Firstly, all praise to the Great Allah for whom our thesis have been completed without any major interruption.
Secondly, to our supervisor Dr. Muhammad Iqbal Hossain for his kind support and advice in our work. He helped us whenever we needed help.
And finally to our parents without their throughout support it may not be possible. With their kind support and prayer we are now on the verge of our graduation.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Over the past few years face recognition has been one of the most popular research area and most successful application of image analysis and understanding Because of the demand for better security. A large number of face recognition [1] techniques use huge amount of image data to train the model to successfully recognize known person. If new persons are added into that particular dataset, then full training process has to be done again and it's a problem. That is why we chose to use FaceNet [2] method to overcome this situation. In this method we don't have to train the whole dataset again, we just have to train a classifier for the new added images. So this paper is aimed to be a complete face recognition system for any organization that wanted to implement which is easy to build, there's no need to collect huge amount of dataset for each people for training purpose to make the system effective.

Face recognition systems are also perfect enhancement of existing security system because it is totally non-intrusive and effective system without bothering the person in any way. In order to use any kind of face recognition technique in any organization, at first, we have to collect significant amount of face images for each person in that particular organization. Then by feeding that data into training dataset we can implement face recognition system that can successfully recognize known and unknown person. Accuracy and speed are major issues here. The more data we feed into to train, the more accuracy we gain but in maximum case it's not possible to get that significant amount of data to achieve the recognition accuracy we want. The main purpose of this paper is to overcome this scenario by implementing face recognition system using MTCNN [3] and FaceNet by training only five different phase images for each person and show the accuracy based on the number of training images, here we increase the number of images by duplicating same five phased images again and again. In this thesis, we also represented how we can successfully recognize the known person using MTCNN and FaceNet by feeding only one front image into training dataset.

The remainder of this thesis is organized as follows, Chapter 2 presents related works in this field, Chapter 3 presents the overall methodology and system workflow, Chapter 4 presents methods of collection and processing of dataset. Experimental setup is also displayed in this chapter. Chapter 5 presents the result and analysis of this approach and finally Chapter 6 concludes the paper with the discussion of further improvement on this development.

Chapter 2

Related Work

Face recognition is one of the most researched computer vision problems over last two decades. In the literature [4] [5] [6] [7] there are many algorithms used to solve face recognition problem. For example – Eigenface Algorithm (1991), Local Binary Patterns Histograms (LBPH) (1996) Fisherface Algorithm (1997), Scale Invariant Feature Transform (SIFT) (1999), Speed Up Robust Features (SURF) (2006), Compact Binary Face Descriptor (CBFD) (2015), Facial Trait Code (FTC), Ensemble Voting Algorithm (EVA), etc. Most of the algorithms are successful with accuracy rate over 90%. Among these algorithms PCA has 77%-84% accuracy, FTC has 81%-90% accuracy and LBP has 92%-97% accuracy.

At past most of the face recognition projects or research or works were detecting and recognizing a face only. They did not do any other experiment or did not use the system in any other project. Ahmed ElSayed, Ausif Mahmood and Tarek Sobh worked on "Effect of Super Resolution on High Dimensional Features for Unsupervised Face Recognition in the Wild"[8] to improve the efficiency of face recognition on low resolution and blurred image using Image Super Resolution algorithm. Reza Shoja Ghiass, Ognjen Arandjelovic Hakim Bendada and Xavier Maldague worked on "Infrared Face Recognition: A Literature Review"[9] in order to review various proposed approach to over the low accuracy in face recognition due to presence of illumination, pose, facial disguise, expression changes etc. Guo-Hui He, Bin Zhu, and Juan-Ying Gan tried to solve face position problems using 3D face model in "Pose-Varied Face Recognition Based on 3-D Face Model"[10]. In "A Geometrical-Model-Based Face Recognition"[11] Yea-Shuan Huang and Suen-Yu Chen presented a two-stage face recognition method to produce better result than other face recognition methods. In the first stage they used Local Vector Pattern (LVP) with a weighting mechanism and in the second Stage they used Bilateral Recognition (BR) approach. In "Face Image Assessment Learned with Objective and Relative Face Image Qualities for Improved Face Recognition"[12] Hyung-Il Kim, Seung Ho Lee and Yong Man Ro proposed a learning based face image assessment to improve the accuracy of face recognition. In their work they introduced two face image quality which are Objective Face Image Quality (OFIQ) and Relative Face Image Quality (RFIQ) to filter the training and testing dataset and remove useless images from the dataset. In all of these papers they tried to improve the efficiency. Vinay A, Desanur Naveen Reddy, Abhishek C. Sharma, Daksha S, N S Bhargav, Kiran MK, KNB Murthy and Natrajan S worked on "G-CNN and F-CNN: Two CNN Based Architectures for Face Recognition"[13] to obtain higher accuracy in face recognition. In 2017 Musab Coskun, Aysegul Ucar, Ozal Uildirim and Yakup Demir developed a CNN architecture in their work "Face Recognition Based on Convolutional Neural Network"[14] to get a high accuracy in face recognition. In most of these papers, the Authors tried different algorithms to get high accuracy in face recognition. On the other hand, in this paper we will discuss about the variables or dependencies of face recognition. We will analyze how the accuracy depends on different variables in face recognition system.

Chapter 3

Methodology and System Workflow

Before Describing the methodology of our system we need to explain some major components of Face recognition to get a faster outcome with higher accuracy. They are face detection, Encoding Faces using FaceNet, Classification and prediction. Later on we have discussed how the system works with flowcharts.

3.1 Methodology

3.1.1 Face Detection using MTCNN

Face detection [15] is the first and foremost step in any face recognition system. We first need to determine whether human faces appear in the given image or not and where is detected faces are located at. At this moment there exists various reliable method for face detection accurately. Among them we used MTCNN - Multi-task convolutional neural network, which combines face detection with face key detection. It is based on cascade framework. This method is proposed by Kaipeng Zhang et al. in their paper 'Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks', IEEE Signal Processing Letters, Volume: 23 Issue: 10. The overall structure can be divided into P-Net, R-Net, and O-Net which is three convolution network and they are able to outperform many face-detection benchmarks while retaining real-time performance.

The proposed framework consists of three stages to perform face detection and facial landmark detection simultaneously. In first stage, it will propose several candidate windows quickly through a shallow CNN [16]. After that, the second network will refine the windows to reject a large number of non-faces windows through a more complex CNN. Finally, it uses a more powerful CNN [17] to refine the result and output five facial landmarks positions. Given an image, this method use image pyramid so that they have the image in multiple scale. Then the image is given as input to the following three-stage cascaded framework:

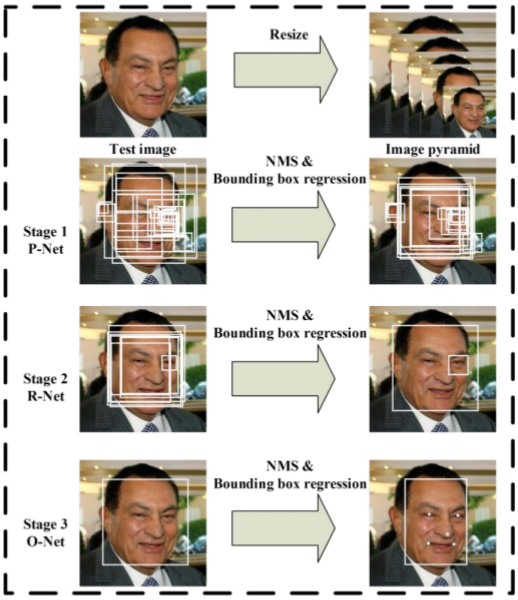

Figure 3.1: Pipeline of MTCNN method

At first we have to pass an image as a input to the program. In this model we have to first create a image pyramid. In order to detect faces in different size of images. The purpose of to create different copies of same images is to search for different sized faces within the same image. Now for each copy we have a 12 x 12 stage 1 kernel that will go through every part of the entire image to scan for faces. It starts in the top left corner, basically a section of the image from (0,0) to (12,12). This portion of the image is fed by P-Net which is known by proposal network. P-Net then returns the coordinates of a bounding box if it detects a face. Then, it will repeat the same process with sections (0+2a,0+2b) to (12+2a, 12+2b), shifting the 12 x 12 kernel 2 pixels right or down at a time. The shift of 2 pixels is known as the stride which actually means how many pixels the kernel moves by every time. Having a 2 stride helps to reduce the complexity of computation without sacrificing the accuracy significantly. The faces in most of the images are larger than 2 pixels. So the probability of that the kernel will miss a face because it shifted 2 pixel is very low. The other benefit is that the machine in which the code is running will have a quarter of the amount of operations to compute, making the program run faster and with less memory. Each kernel would be smaller relative to a large image so that it would be able to find smaller faces in the larger image. Similarly, the kernel would be bigger relative to a smaller sized image so that it would be able to find bigger faces in the smaller sized image. Moreover , after gathering the output from P-Net we need to parse the P-Net output to get a list of confidence levels for each bounding box and delete the boxes with lower confidence because the network is more confident about some boxes compared to others. After that, there must be lot of bounding boxes still left and a lot them overlap. The NMS method which is known as a Non-Maximum Suppression [18] helps to reduce the number of bounding boxes. In this particular code most of the windows are filtered out.

In the stage 2 we consider those images which may contain only a part of a face peeking in from the side of the frame. In that case, the network may return a bounding box that is partly out of the frame. To overcome this situation for every

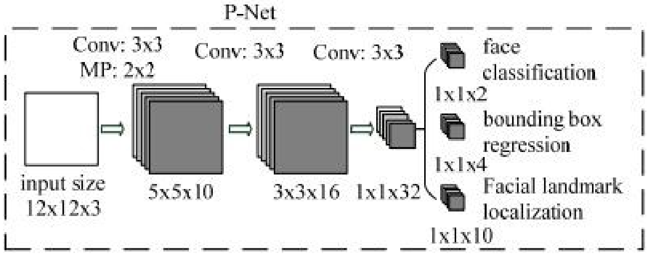

Figure 3.2: P-net

bounding box we create an array of the same size and copy the pixel values of the image which is in the bounding box to the new array. If the bounding box is out of bounds, we only copy the portion of the image in the bounding box to the new array and fill in everything else with a zero. This process of filling arrays with 0s is known as padding. After padding the bounding boxes, we resize them to 24 x 24 pixels and normalize them into values between -1 to 1. Now we can feed these bounding boxes to R-Net which is known as Refine Network and gather its output. R-Net's output is similar to P-Net's output. It includes the coordinates of the new, more accurate bounding boxes and also the confidence level of each of these bounding boxes. In this stage we also get rid of the boxes with lower confidence, and perform Non-Maximum Suppression on every box to further eliminate redundant boxes. Since the coordinates of these new bounding boxes are based on the P-Net bounding boxes, we need to convert them to the standard coordinates and then reshape the bounding boxes to a square to be passed on to O-Net.

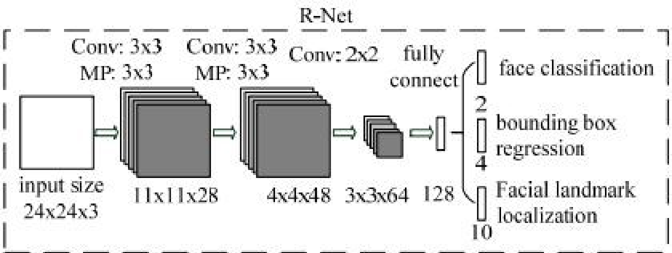

Figure 3.3: R-net

In the last stage before we can pass in the bounding boxes from R-Net, we have to first pad any boxes that are out-of-bounds. Then we have to resize the boxes to 48 x 48 pixels so that we can pass in the bounding boxes into O-Net which is known as Output Network. The output of O-Net is different from output of P-Net and R-Net. O-Net provides 3 outputs, the coordinates of the bounding box (out[0]), the coordinates of the 5 facial landmarks (out[1]), and the confidence level of each box (out[2]). Then once again we have to get rid of the boxes with lower confidence levels, and standardize both the bounding box coordinates and the facial landmark coordinates. Finally, by applying Non-Maximum Suppression again we get the facial features of the person which is basically five facial landmark positions.

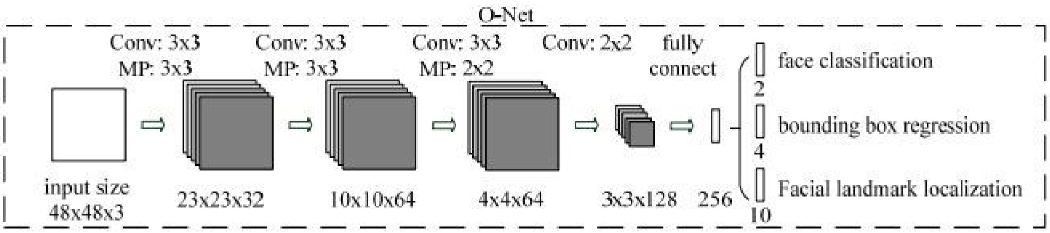

Figure 3.4: O-net

It is mentioned in the paper that they are using the following loss function in their network:

Cross-Entropy Loss [19]: this loss is used to perform face classification for the proposed regions

$$L_i^{det} = -(y_i^{det}\log(p_i) + (1 - y_i^{det})(1 - \log(p_i)))$$

Euclidean Loss [20]: this loss is used to perform bounding-box regression and facial landmark regression

$$L_i^{box} = \left\| \hat{y}_i^{box} - y_i^{box} \right\|_2^2$$

$$L_i^{landmark} = \left\| \hat{y}_i^{landmark} - y_i^{landmark} \right\|_2^2$$

Multi Source Training Loss: Some of input images not only contain face images but also some background images like objects etc. In that case not all loss will be used, for example when training images that will not contain faces, only face detection loss will be used and others will be set as 0. The overall learning target is formulated as:

$$\min \sum_{i=1}^{N} \sum_{j \in \{det,box,landmark\}} \alpha_j \beta_i^j L_i^j$$

These are some test results of five facial features shown in the (figure 3.5)

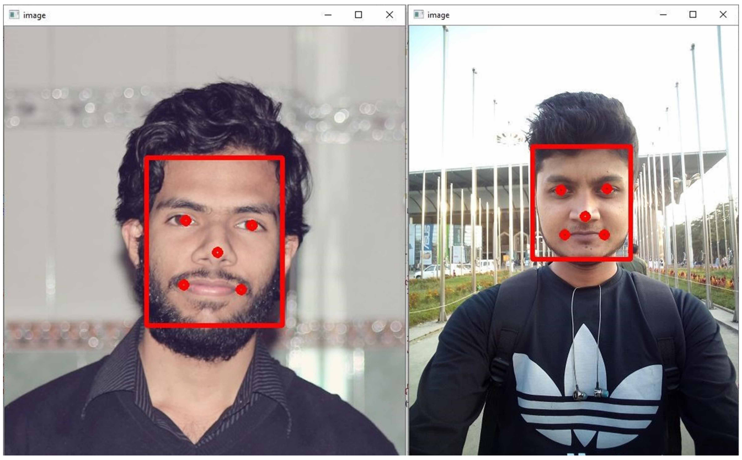

Figure 3.5: Five facial feature

3.1.2 Encoding Faces Using FaceNet

Now for the face recognition and clustering process we used Facenet which is a system of deep learning architecture consisting of convolutional layers based on GoogleNet inspired inception models. Facenet return 128 dimensional vector embeddings for each face. Once these embeddings are created then procedures like face recognition and verification can be done utilising these embeddings as features of faces. Facenet inserts these embeddings into a feature space so that the squared distance between all faces, regardless of the imaging conditions, of the same person, is small, whereas the squared distance between a pair of face images from distinct characters is large. In 2015, Google researchers introduced FaceNet.Like as word embedding FaceNet transforms the face into 128 dimensional Euclidian space.After training the FaceNet model with triplet loss for different persons faces to find the differences and similarities between same and different persons images, we can cluster faces efficiently with this model.After creating the embedding vector space using standard techniques we can easily recognize, verify and cluster faces.In this model faces of same person has low Euclidian distance and faces of different person has high distance. In the paper they described that the network consists of a batch input layer and a deep CNN. Then they used L2 normalization to get the face embedding which is followed by the triplet loss during training. Triplet Loss mainly increase the distance between anchor and the negative of a different identity and decrease the distance between and anchor and a positive having the same identity.In their paper, different types of architectures are used and explored. Their trade-offs in more detail in the experimental section. The parameters and Flops are their practical differences. The best model may be different depending on the application. After the FaceNet model is trained, the embedding for the face can be created by feeding into the model. To compare multiple images, we have to create embedding for all the images by feeding them separately through the model. Then by using above formula we can find the distance which will be lower value for same person's faces and higher value for different person's faces.

$$d(\mathbf{p},\mathbf{q}) = d(\mathbf{q},\mathbf{p}) = \sqrt{(q_1-p_1)^2 + (q_2-p_2)^2 + \cdots + (q_n-p_n)^2}$$
$$= \sqrt{\sum_{i=1}^{n}(q_i-p_i)^2}.$$

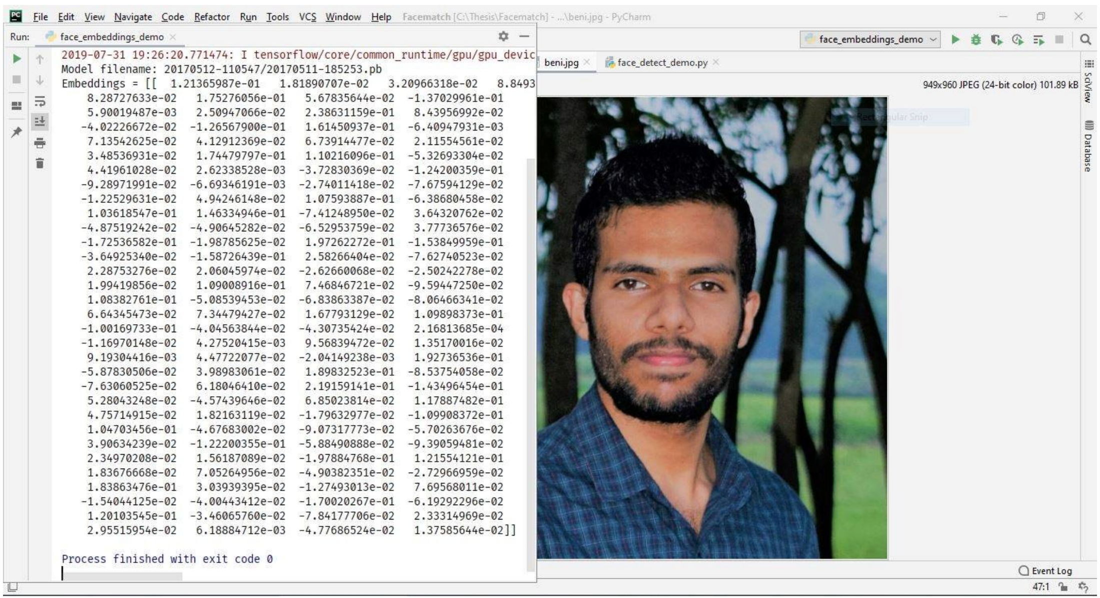

Figure 3.6: 128 dimensional vector embeddings for each face

3.1.3 Triplet Loss

This system introduces a new loss function named triplet loss. The reason behind using triplet loss is that it encourages all the images of one identity to be projected into a single point in the embedding space and for this reason it minimizes the distance between images of the same identity and maximizes the distance between the face images of different characters. Triplet loss reduces the differences between Anchor and Positive and also increases the distance between Anchor and Negative. Anchor is a targeted person and Positive is other images of that person where

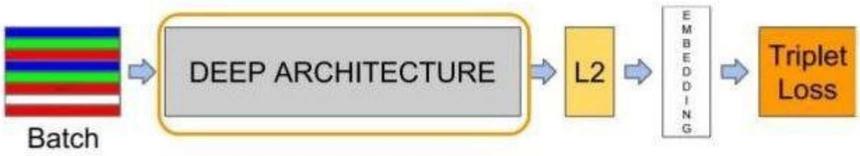

Figure 3.7: Facenet Architecture Block Diagram

Negative is images of other persons. So, in order to training a model to classify we can use triplet loss to minimize the distance between images of same person and also maximize the distance between the images of different person. In FaceNet all the images a processed encoded into 128 dimensional vector which contains the coordinates. Using Triplet Loss function we can identify the images of same person and images of different person only with the distance. If the distance between two images is low then we can say that they are images of same person. On the other hand, if the distance is high then we can say that the images belongs to different persons.

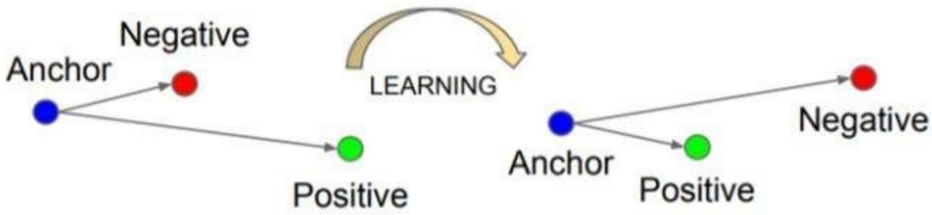

Figure 3.8: Tripless loss training

3.1.4 Train Classifier

This facenet architecture is trained with a dataset of very large number of faces belonging to numerous classes. Here different class means different person. The best part of facenet is we can use it to create embeddings for our own custom dataset by training an SVM (Support Vector Machine) [21] to use these embeddings obtained from different faces from our own dataset and do the classification. Everytime we have a new person's face being added to our dataset, we just need to add another class and train the final classifier rather than training the entire FaceNet model.

3.2 System Workflow

3.2.1 Training Process

For the training purpose our proposed system takes five angle images as input and then detect faces from those images by using MTCNN method. When it detects face successfully in the image then it crops the images into 182 x 182 pixel size containing only face. Then the data are fed into FaceNet model to create 128 unique features for each images called face embeddings. Since we are using pre trained FaceNet model, so we have to train a classifier to train our newly added images. In this system we are using SVM classifier. Shown in (figure 3.9)

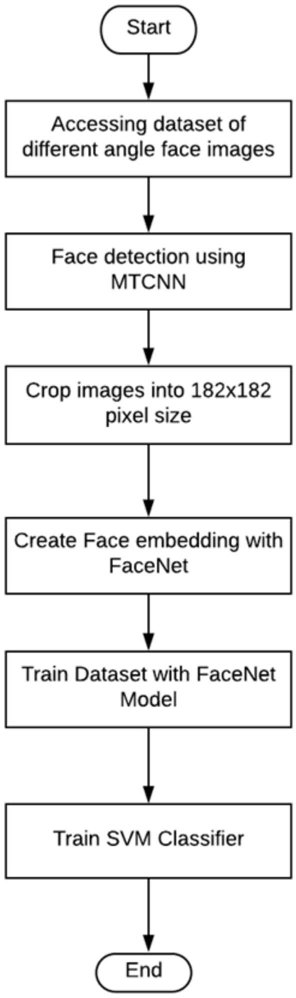

Figure 3.9: Training Process

3.2.2 Recognition Process

In the face recognition process our system starts with input from live video or still images. After getting video or images it reads the frames with OpenCV library. From the frames the system detect faces using MTCNN. After face detection it creates embedding points for faces using FaceNet. After creating the embedding points the system classifies the result and match the result with training data set. If the matching satisfies the threshold then it shows the person's name and percentage matched with the person's face saved in database. If the result do not satisfy the threshold then it shows that the person is unknown and store the face along with time for further use. Shown in (figure 3.10)

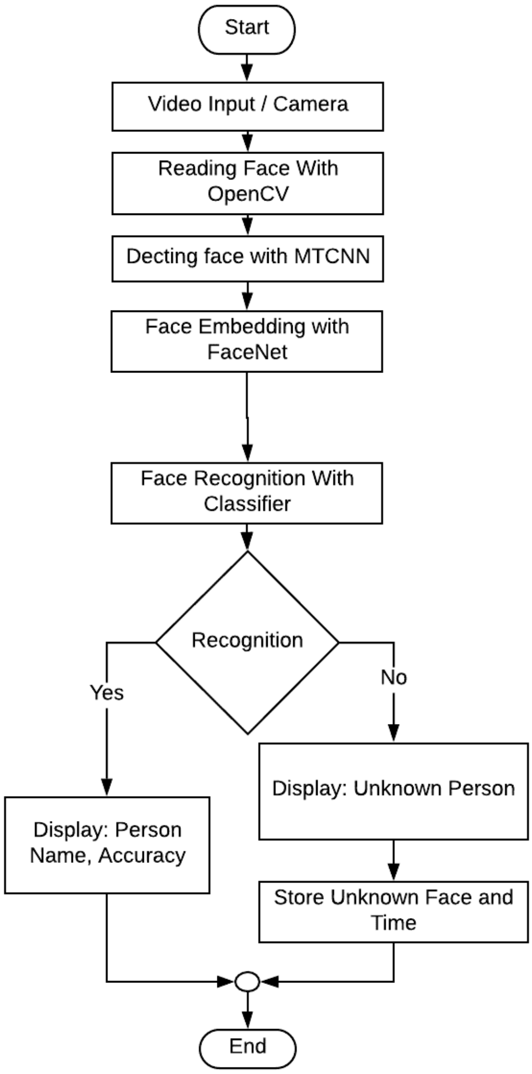

Figure 3.10: Recognition Process

Chapter 4

Dataset and Experimental Setup

4.1 Dataset

We used our own collection of datasets which contains 50 people's image collections. In this dataset each of the person has five different angle images where each of the image size is approximately between 3 to 5 MB. Shown in (Figure 4.1).

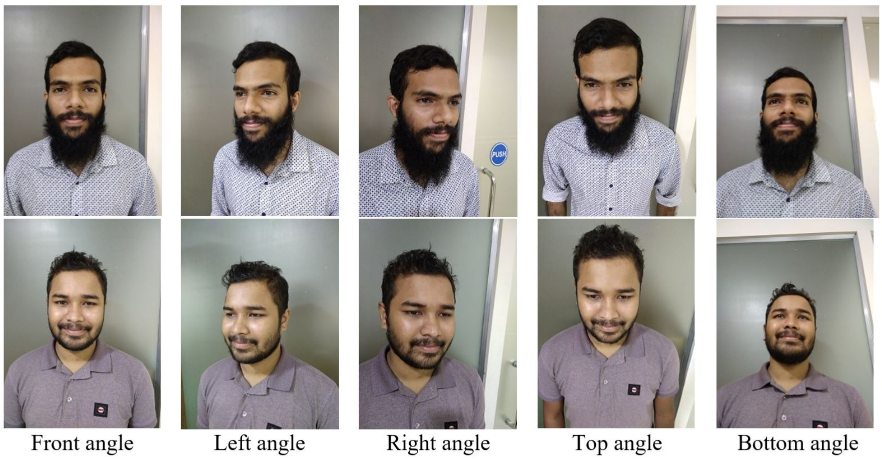

Figure 4.1: five angle image dataset

Later on we preprocessed our image dataset and arranged collection of only faces. Here the resulation of cropped faces are 182x182 pixels. Shown in (Figure 4.2)

In order to collect data for our system we used different places. We have collected our data from School, College and University students of different ages. We captured the images in random weather condition and with average quality camera. Images with high quality camera show higher accuracy but we wanted to explore what happens if we use average quality camera because if we set our system as a security system then there will be average quality CCTV camera. Instead of collecting images from internet or any other sources we collected images from primary source because our proposed system required images of 5 different angle which is not possible to get from any other sources. As the dataset was not available and we had to collect raw data manually, for this reason our dataset is small. We managed to collect data from 50 different persons. Although it's a small number, it is enough to test our system and find accurate result.

Figure 4.2: five angle cropped face image dataset

4.2 Experimental Setup

A personal laptop with specifications of 8 GB ram running on Intel core-i7 with a CPU speed of 2.5 GHz, 64 bits windows 10 with a webcam and 1 TB of hard disk space is used to train and test the program. Python is the programming language and PyCharm IDE is used to run the program. The program is completely based on deep learning neural network and implemented using Tensorflow framework. In order to execute the program these four python libraries are required-

- Tensorflow (1.4.0)
- Scipy (0.17.0)
- Scikit-learn (0.19.1)
- Opencv (2.4.9.1)

TensorFlow is an open source library for numerical computation and large-scale machine learning. TensorFlow bundles together a slew of machine learning and deep learning (aka neural networking) models and algorithms and makes them useful by way of a common metaphor. It uses Python to provide a convenient front-end API for building applications with the framework, while executing those applications in high-performance C++.

SciPy is an Open Source Python-based library, which is used in mathematics, scientific computing, Engineering, and technical computing. Scikit-learn is a library in Python that provides many unsupervised and supervised learning algorithms. The functionality that scikit-learn provides include Regression, Classification, Clustering, Model Selection, preprocessing. OpenCV (Open Source Computer Vision) is a library of programming functions mainly aimed at real-time computer vision. It is mainly used to do all the operation related to Images.

Chapter 5

Result and Analysis

Result and Analysis: In our system we used different numbers of images of 50 people for training model. Then we tested the model with different images. From the result 5 persons result analysis is given here. 2 persons are chosen from high accuracy, 2 persons are chosen from medium accuracy and 1 person is chosen from low accuracy. The person from low accuracy is tested with ae completely different looking images than the training dataset. We use completely different images to test the model in order to show that the system can still identify the person successfully though the images matching point is very low. We have done 3 types of analysis to show how the accuracy varies. At first we created the training model with only frontal images of 50 person. In this case, we run the analysis for multiple time with different number of images in training set. At first we created the training model with only 10 frontal images per person. Then 30 images per person. Finally we used 300 images per person to create the training model. After completing the analysis with frontal images we run the same analysis with image from 3 angles (Front, Left and Right) for every person to train the data set. In this analysis also we use the same technique which is we train model several time with 10 images per person to 300 images per person. In this case the accuracy of testing images is much higher than the frontal face analysis. After finishing the 3 angles face analysis we run another analysis with images from 5 angles (Front, Left, Right, Top and Bottom) per person. In this case also we trained the model several time with different number of images and tested the model. In this case the result is higher than frontal and 3 angles face analysis. Now we will see analysis of different types below:

5.1 Individual Analysis and Result

Here, Analysis for Sohan Based on only Front angle image in training period is given below:

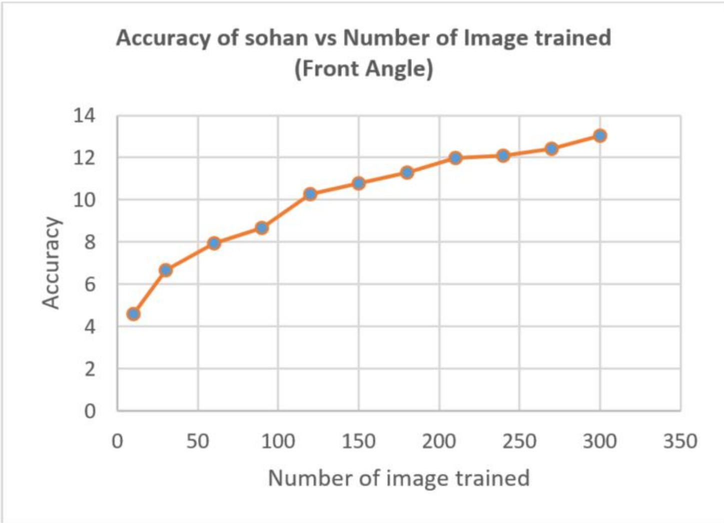

Figure 5.1: Line graph on accuracy of Sohan for front angle image training

Figure 5.2: Sohan's front angle training image and test image

Number of image Trained	Accuracy %
10	4.59
30	6.66
60	7.92
90	8.65
120	10.26
150	10.78
180	11.3
210	11.97
240	12.1
270	12.4
300	13.03

Table 5.1: Accuracy table of Sohan on front angle image training.

In this case, we can see that only for the front angle image the accuracy increased from 4.59% to 13.03% from the amount of trained image 10 to 300 which is not that much good.

Here, Analysis for Sohan Based on three angle image in training period is given below:

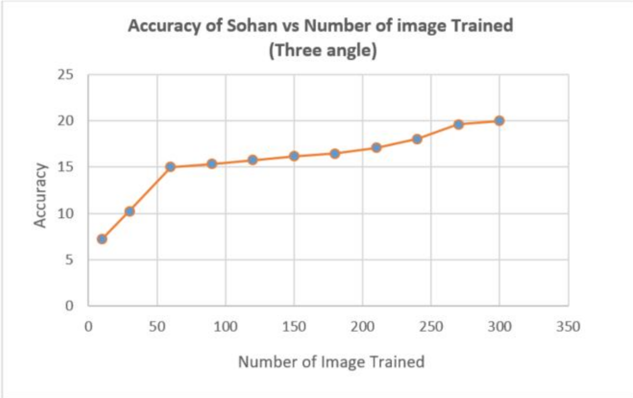

Figure 5.3: Line graph on accuracy of Sohan for three angle image training

Figure 5.4: Sohan's three angle training image and test image

Number of image Trained	Accuracy %
9	7.27
30	10.25
60	15.02
90	15.35
120	15.77
150	16.18
180	16.46
210	17.07
240	18.05
270	19.62
300	19.99

Table 5.2: Accuracy table of Sohan on three angle image training.

In this case, we can see that for the three angle image the accuracy increased from 7.27% to 19.99% from the amount of trained image 10 to 300 which is improvement than the front angle accuracy.

Here, Analysis for Sohan Based on five angle image in training period is given below:

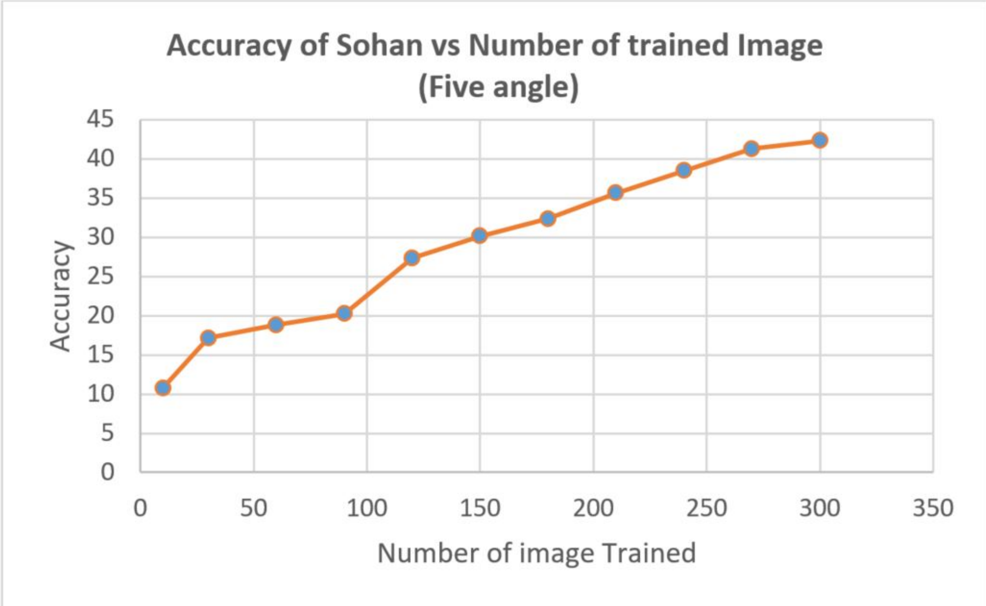

Figure 5.5: Line graph on accuracy of Sohan for five angle image training

Figure 5.6: Sohan's five angle training image and test image

Number of image Trained	Accuracy %
10	10.76
30	17.16
60	18.82
90	20.24
120	27.34
150	30.15
180	32.36
210	35.64
240	38.49
270	41.31
300	42.34

Table 5.3: Accuracy table of Sohan on five angle image training.

In this case, we can see that for the five angle image the accuracy increased from 10.76% to 42.34% from the amount of trained image 10 to 300 which is better than the front angle accuracy and three angle accuracy.

Here, Analysis for Farhan Based on front angle image in training period is given below:

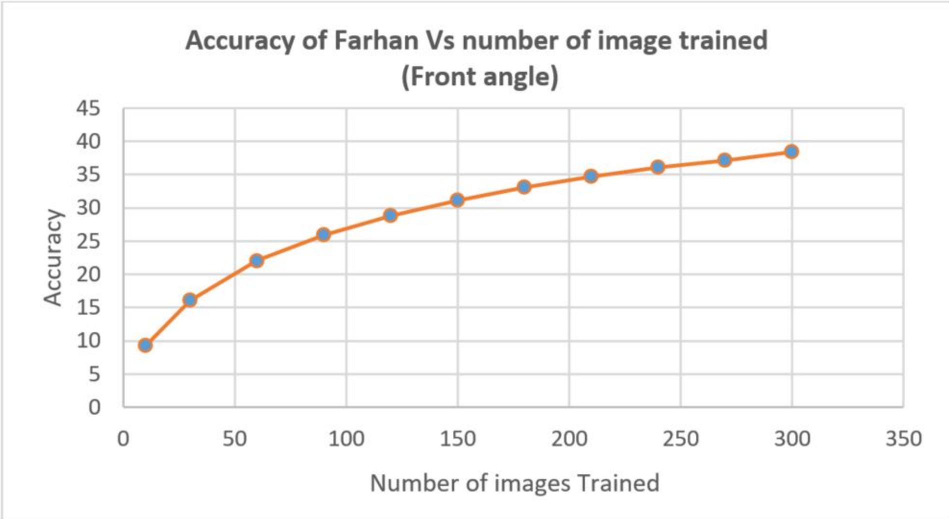

Figure 5.7: Line graph on accuracy of Farhan for front angle image training

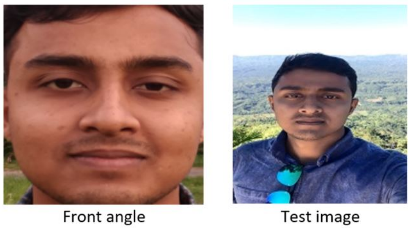

Figure 5.8: Farhan's front angle training image and test image

Number of image Trained	Accuracy %
10	9.24
30	16.12
60	22.05
90	25.96
120	28.86
150	31.17
180	33.07
210	34.68
240	36.12
270	37.1
300	38.39

Table 5.4: Accuracy table of Farhan on front angle image training.

In this case, we can see that only for the front angle image the accuracy increased from 9.24% to 38.39% from the amount of trained image 10 to 300 which is not that much good.

Here, Analysis for Farhan Based on three angle image in training period is given below:

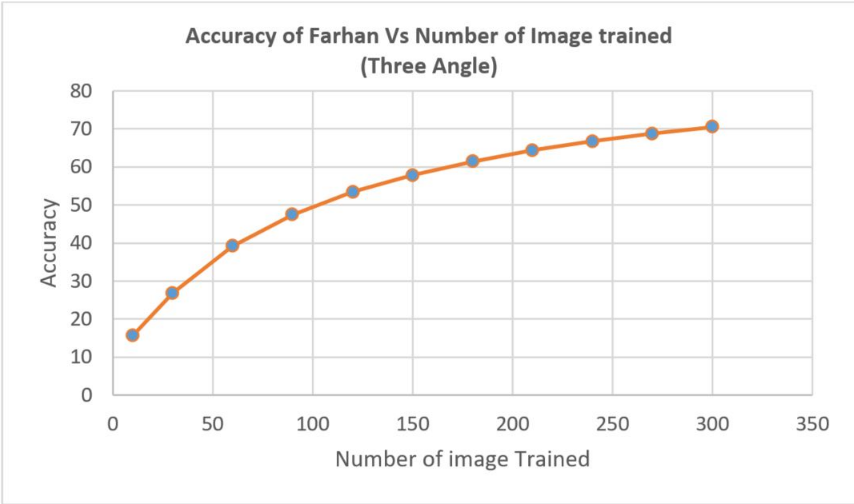

Figure 5.9: Line graph on accuracy of Farhan for three angle image training

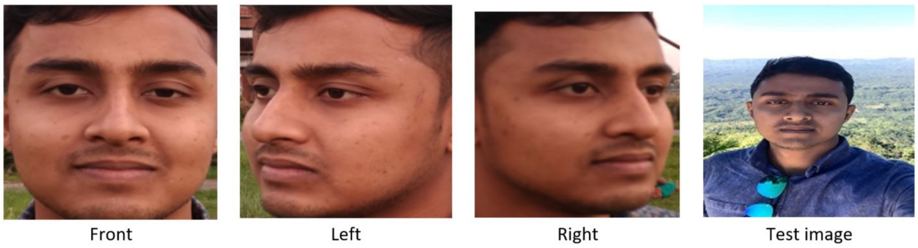

Figure 5.10: Farhan's three angle training image and test image

Number of image Trained	Accuracy %
9	15.59
30	26.87
60	39.17
90	47.48
120	53.43
150	57.94
180	61.49
210	64.37
240	66.76
270	68.79
300	70.53

Table 5.5: Accuracy table of Farhan on three angle image training.

In this case, we can see that for the three angle image the accuracy increased from 15.59% to 70.53% from the amount of trained image 10 to 300 which is much improvement than the front angle accuracy.

Here, Analysis for Farhan Based on five angle image in training period is given below:

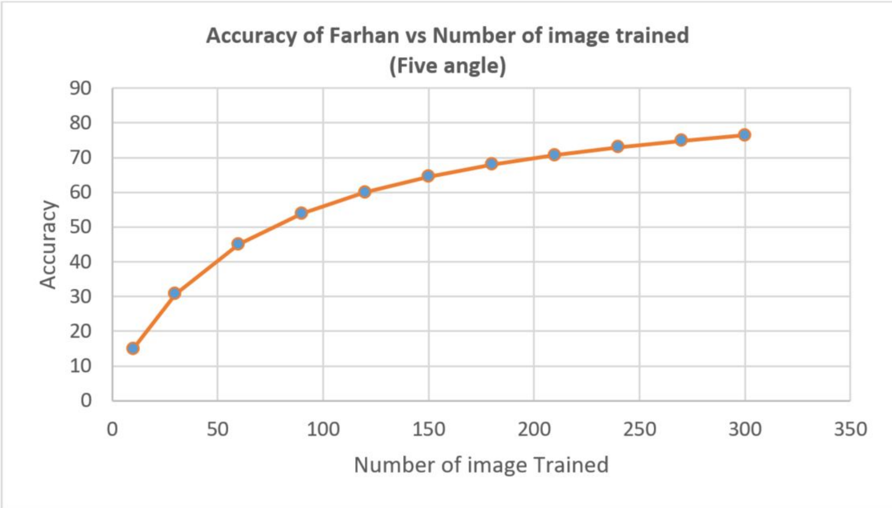

Figure 5.11: Line graph on accuracy of Farhan for five angle image training

Figure 5.12: Farhan's five angle training image and test image

Number of image Trained	Accuracy %
10	14.94
30	30.71
60	45.03
90	53.94
120	60.04
150	64.54
180	67.99
210	70.74
240	73
270	74.87
300	76.45

Table 5.6: Accuracy table of Farhan on five angle image training.

In this case, we can see that for the five angle image the accuracy increased from 14.94% to 76.45% from the amount of trained image 10 to 300 which is better than the front angle accuracy and three angle accuracy.

Here, Analysis for Ashikh Based on front angle image in training period is given below:

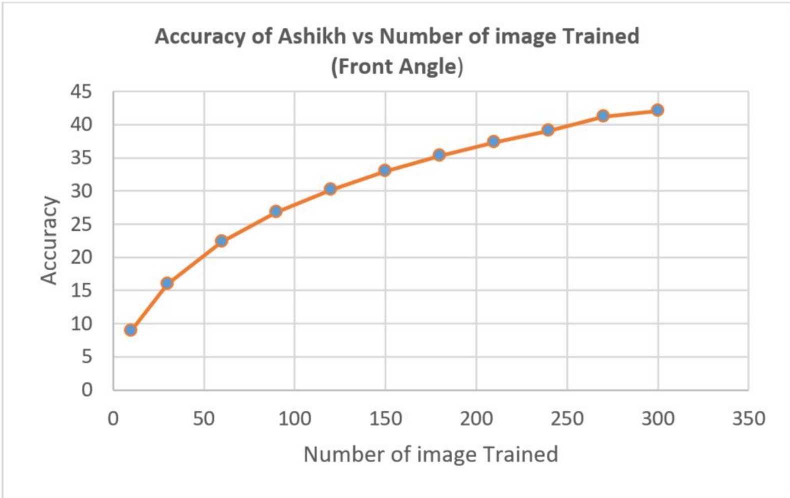

Figure 5.13: Line graph on accuracy of Ashikh for front angle image training

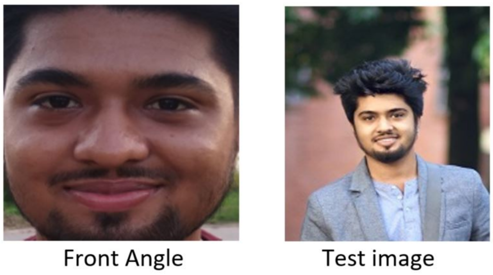

Figure 5.14: Ashikh's front angle training image and test image

Number of image Trained	Accuracy %
10	8.99
30	15.96
60	22.35
90	26.79
120	30.21
150	32.99
180	35.33
210	37.34
240	39.1
270	41.2
300	42.09

Table 5.7: Accuracy table of Ashikh on front angle image training.

In this case, we can see that only for the front angle image the accuracy increased from 8.99% to 42.09% from the amount of trained image 10 to 300 which is not that much good.

Here, Analysis for Ashikh Based on three angle image in training period is given below:

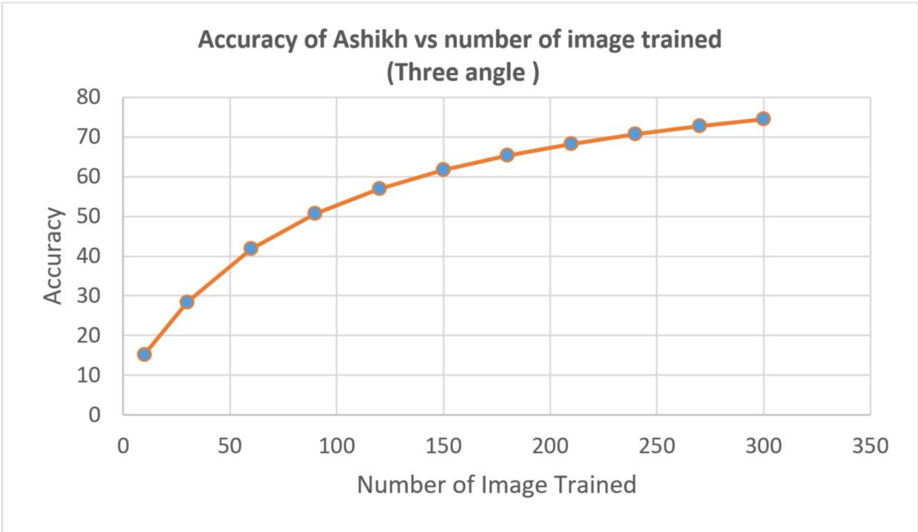

Figure 5.15: Line graph on accuracy of Ashikh for three angle image training

Figure 5.16: Ashikh's three angle training image and test image

Number of image Trained	Accuracy %
9	15.23
30	28.43
60	41.81
90	50.72
120	56.98
150	61.68
180	65.34
210	68.28
240	70.7
270	72.73
300	74.46

Table 5.8: Accuracy table of Ashikh on three angle image training.

In this case, we can see that for the three angle image the accuracy increased from 15.23% to 74.46% from the amount of trained image 10 to 300 which is much improvement than the front angle accuracy.

Here, Analysis for Ashikh Based on five angle image in training period is given below:

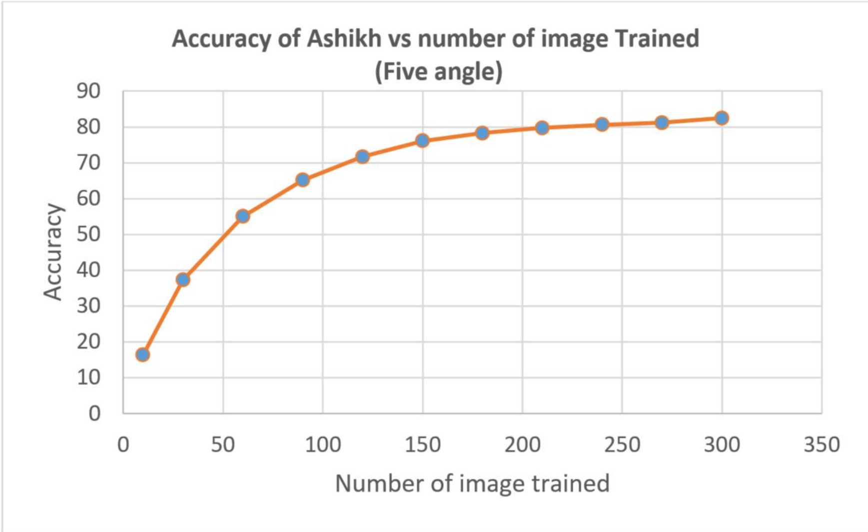

Figure 5.17: Line graph on accuracy of Ashikh for five angle image training

Figure 5.18: Ashikh's five angle training image and test image

Number of image Trained	Accuracy %
10	16.41
30	37.29
60	55.03
90	65.18
120	71.64
150	76.06
180	78.28
210	79.71
240	80.62
270	81.16
300	82.42

Table 5.9: Accuracy table of Ashikh on five angle image training.

In this case, we can see that for the five angle image the accuracy increased from 16.41% to 82.42% from the amount of trained image 10 to 300 which is better than the front angle accuracy and three angle accuracy.

Here, Analysis for Sujon Based on front angle image in training period is given below:

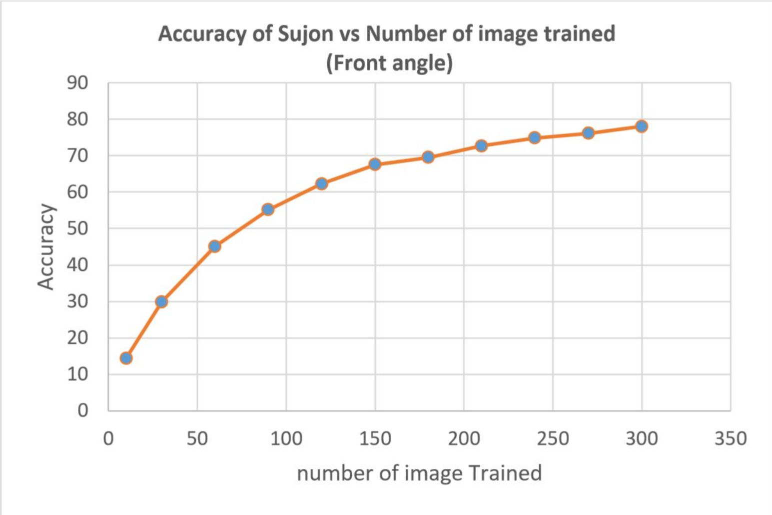

Figure 5.19: Line graph on accuracy of Sujon for front angle image training.

Figure 5.20: Sujon's front angle training image and test image

Number of image Trained	Accuracy %
10	14.37
30	29.88
60	45.08
90	55.16
120	62.27
150	67.52
180	69.53
210	72.67
240	74.83
270	76.12
300	77.98

Table 5.10: Accuracy table of Sujon on front angle image training

In this case, we can see that only for the front angle image the accuracy increased from 14.37% to 77.98% from the amount of trained image 10 to 300 which is good in this case as test image is good.

Here, Analysis for Sujon Based on three angle image in training period is given below:

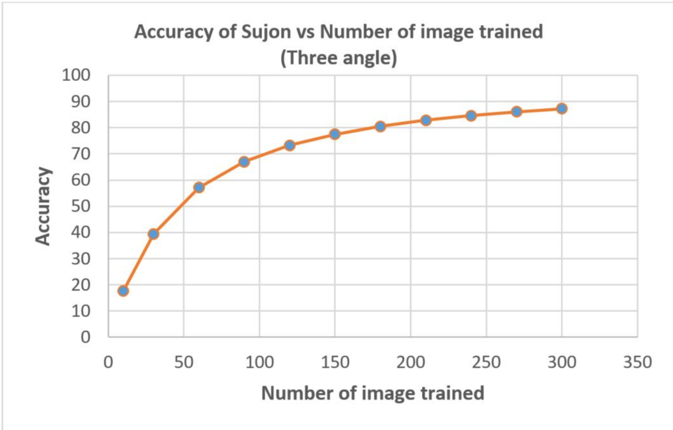

Figure 5.21: Line graph on accuracy of Sujon for three angle image training

Figure 5.22: Sujon's three angle training image and test image

Number of image Trained	Accuracy %
9	17.7
30	39.41
60	57.14
90	67.01
120	73.21
150	77.44
180	80.5
210	82.8
240	84.6
270	86.05
300	87.24

Table 5.11: Accuracy table of Sujon on three angle image training

In this case, we can see that for the three angle image the accuracy increased from 17.7% to 87.24% from the amount of trained image 10 to 300 which is much improvement than the front angle accuracy.

Here, Analysis for Sujon Based on five angle image in training period is given below:

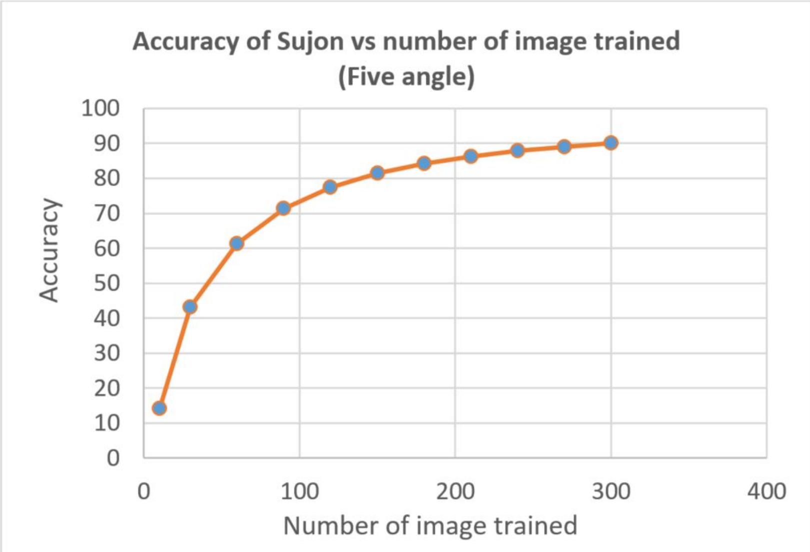

Figure 5.23: Line graph on accuracy of Sujon for five angle image training

Figure 5.24: Sujon's five angle training image and test image

Number of image Trained	Accuracy %
10	14.24
30	43.17
60	61.37
90	71.37
120	77.41
150	81.38
180	84.17
210	86.23
240	87.8
270	89.06
300	90.07

Table 5.12: Accuracy table of Sujon on five angle image training

In this case, we can see that for the five angle image the accuracy increased from 14.24% to 90.07% from the amount of trained image 10 to 300 which is better than the front angle accuracy and three angle accuracy.

Here, Analysis for Suvo Based on front angle image in training period is given below:

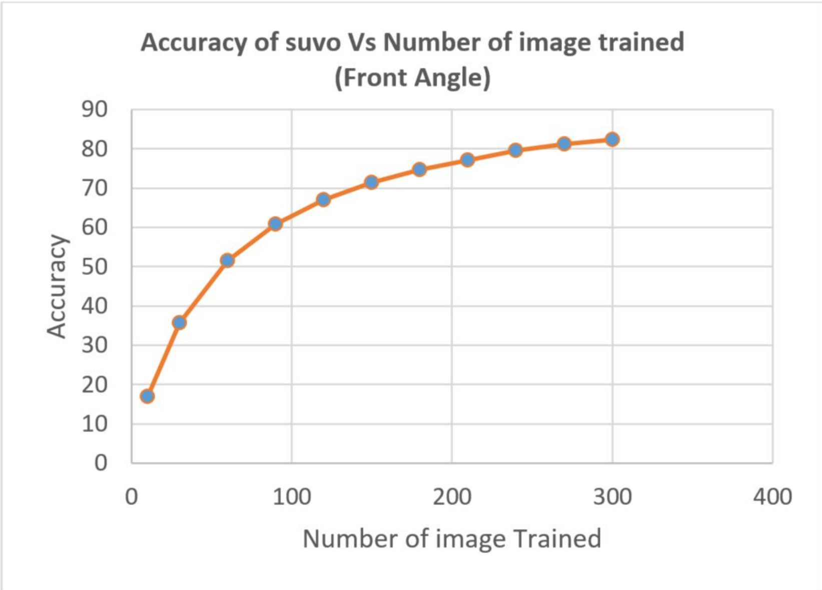

Figure 5.25: Line graph on accuracy of Suvo for front angle image training

Figure 5.26: Suvo's front angle training image and test image

Number of image Trained	Accuracy %
10	17.02
30	35.67
60	51.52
90	60.84
120	66.98
150	71.34
180	74.61
210	77.15
240	79.56
270	81.2
300	82.25

Table 5.13: Accuracy table of Suvo on front angle image training

In this case, we can see that only for the front angle image the accuracy increased from 17.02% to 82.25% from the amount of trained image 10 to 300 which is good in this case as test image is good.

Here, Analysis for Suvo Based on three angle image in training period is given below:

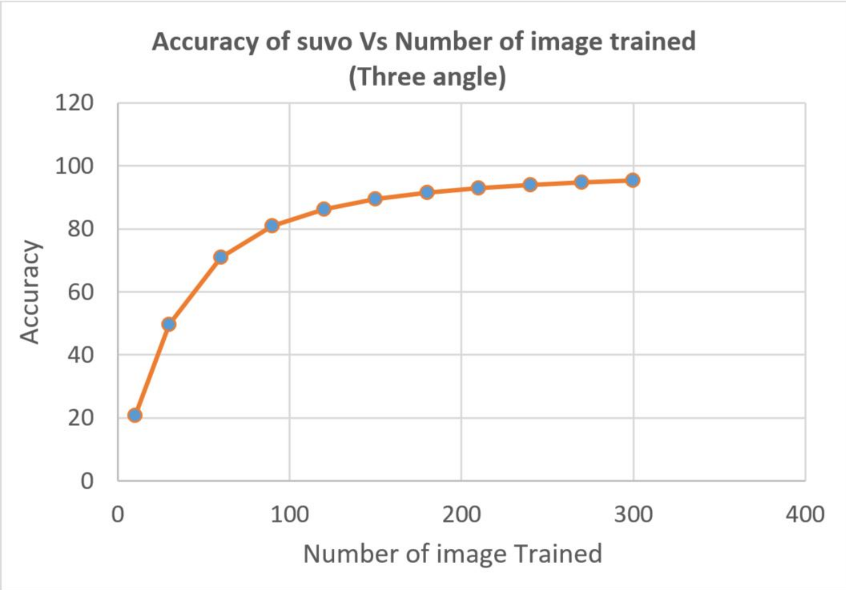

Figure 5.27: Line graph on accuracy of Suvo for three angle image training

Figure 5.28: Suvo's three angle training image and test image

Number of image Trained	Accuracy %
9	20.82
30	49.58
60	71.04
90	80.93
120	86.22
150	89.39
180	91.46
210	92.9
240	93.96
270	94.75
300	95.38

Table 5.14: Accuracy table of Suvo on three angle image training

In this case, we can see that for the three angle image the accuracy increased from 20.82% to 95.38% from the amount of trained image 10 to 300 which is improvement than the front angle accuracy.

Here, Analysis for Suvo Based on five angle image in training period is given below:

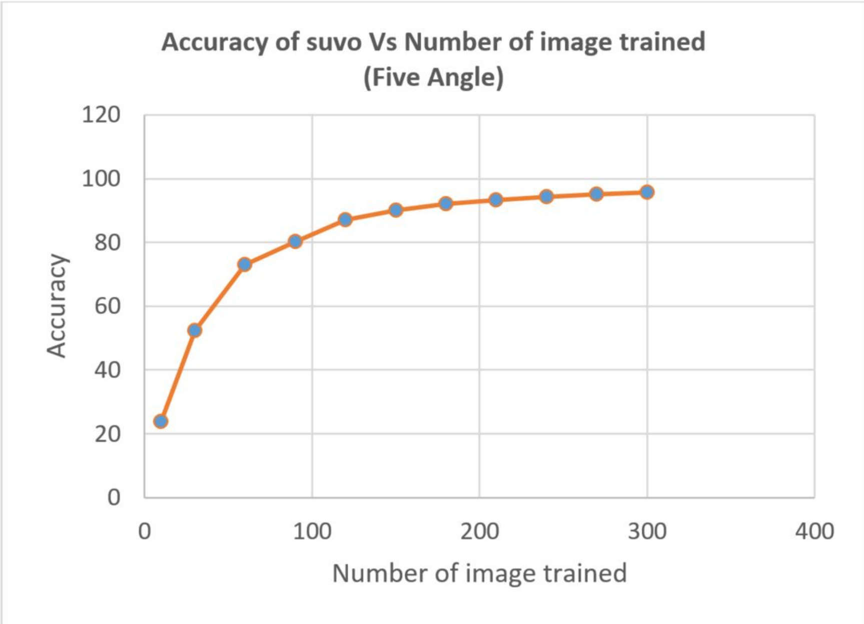

Figure 5.29: Line graph on accuracy of Suvo for five angle image training

Figure 5.30: Suvo's five angle training image and test image

Number of image Trained	Accuracy %
10	23.76
30	52.37
60	72.97
90	80.23
120	87.14
150	90.07
180	91.98
210	93.31
240	94.31
270	95.05
300	95.62

Table 5.15: Accuracy table of Suvo on five angle image training

In this case, we can see that for the five angle image the accuracy increased from 23.76% to 95.62% from the amount of trained image 10 to 300 which is better than the front angle accuracy and three angle accuracy.

5.2 Combined Analysis and Result

Now if we combine all those five person's accuracy of front angle in one chart we can see the below chart and see the effect of increasing number of images in the training period. This clearly indicates that if we increase the number of same images in training period, the accuracy will increase. However, in figure 5.31 we can see that sohan's overall accuracy is much less than the other 4 person's accuracy. This happened because we have intentionally used bad testing image for sohan to understand how much impact appears in accuracy when bad images will arise in the system . Again we can also see that there are also significant accuracy deference between Ashikh, Farhan and Suvo, Sujon. This happened because we have used average testing image for Ashikh, Farhan and good testing image for Suvo , Sujon.

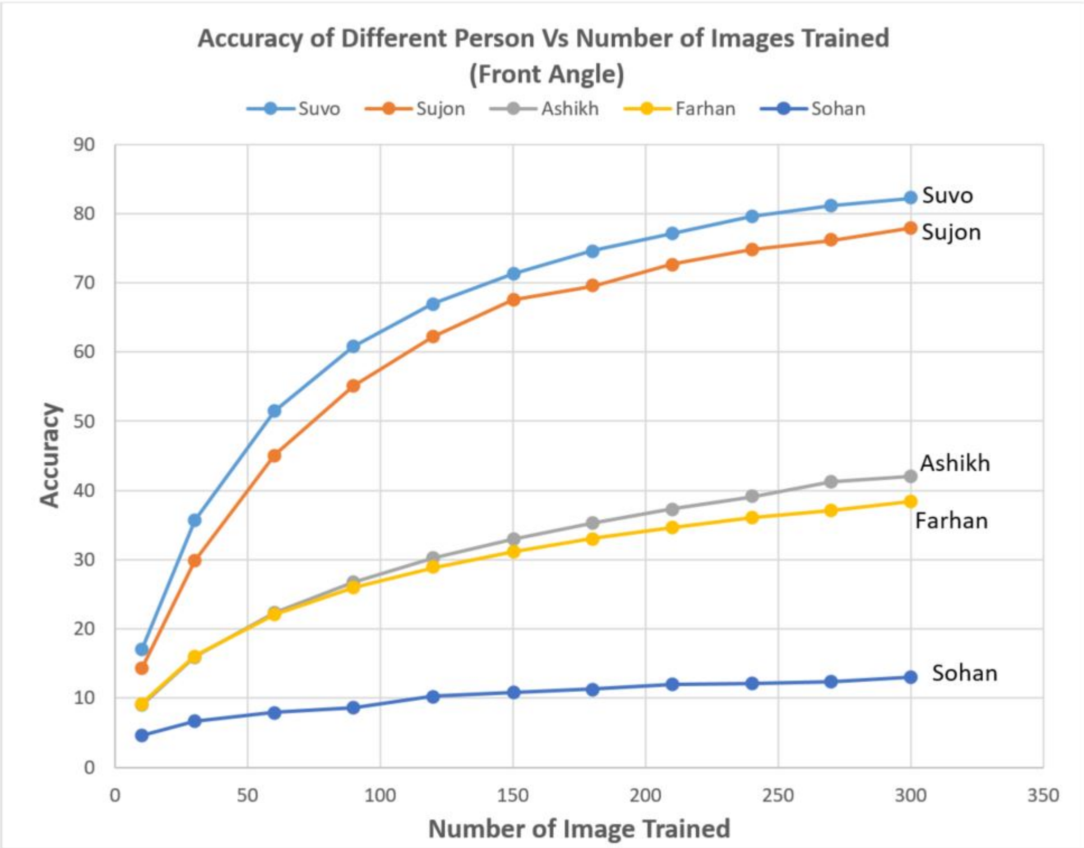

Figure 5.31: Combined five person's Accuracy on front angle image training

Again if we combine those five persons accuracy of three angle we can see in figure 5.32 the accuracy increases than the front angle accuracy at a large scale especially for Ashikh and Farhan. So we can conclude, three angle image in training period give much better accuracy than the front angle image. However, in again we can see that sohan's overall accuracy is much less than the other 4 person's accuracy. The reason is same as before for sohan which is intentionally using bad testing image. On the other hand except sohan other 4 person's accuracy increased very well compared to front angle accuracy as during the training process left and right angle images has been added along with the front angle image. On top of that in between these 4 person's accuracy varies because their training image quality are different from each other.

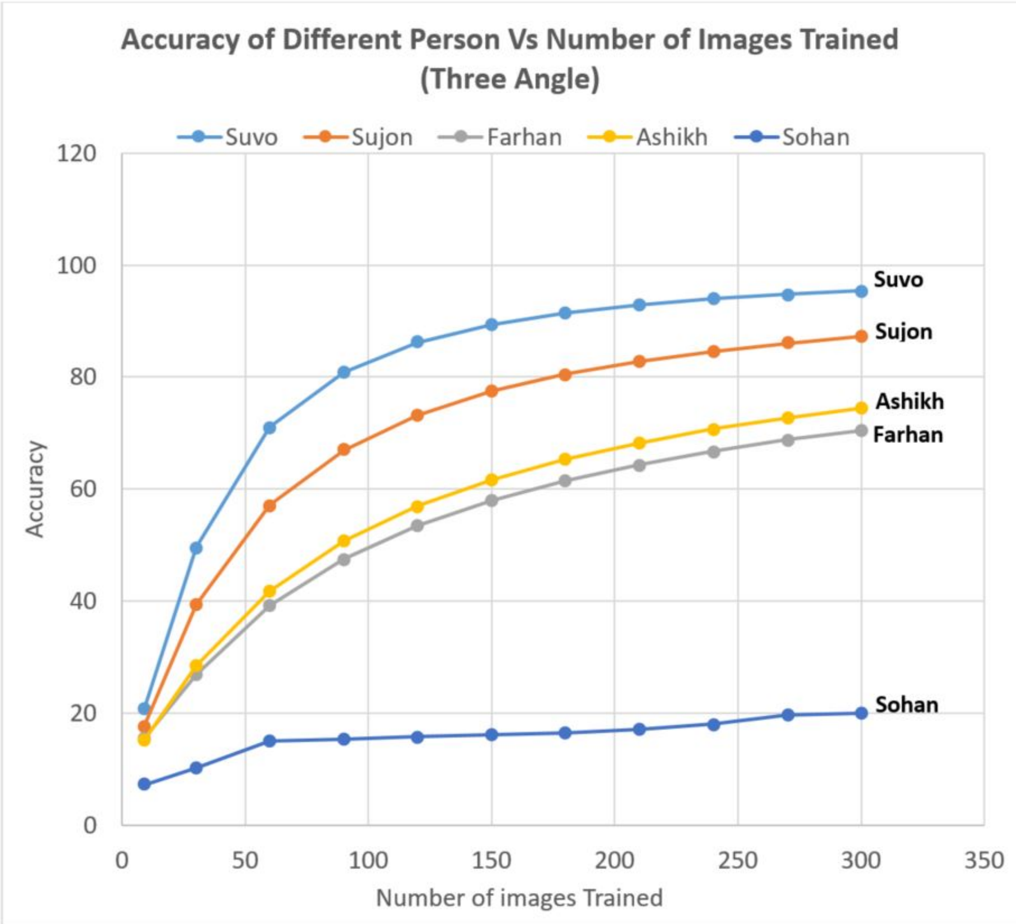

Figure 5.32: Combined five person's Accuracy on three angle image training

Now we will observe the all five person's accuracy based on five angle in the training process shown in figure 5.33. Here, we can see except sohan all other fours person reached to a standard accuracy. As sohan's testing image was chosen bad image to analyze the difference between good and average image testing. So sohan's lower accuracy can be ignored. Again in addition, except sohan other 4 person's accuracy increased more compared to three angle accuracy as during the training process top and bottom angle images has been added along with the front , left and right angle image of each person. On top of that in between these 4 person's accuracy varies because of the same reason as we have used different quality testing image for each person in the training period. So, it seems that in this case training five angle images leads the model to reach standard accuracy for every person.

Figure 5.33: Combined five person's Accuracy on Five angle image training

5.3 Average Result

Now we will see how the average accuracy changes based on the different angle of images with the increase of number of image trained for all 50 persons in our dataset. Firstly, 10 images have been used in training model, it gives 13.66% accuracy for only front faced images in training dataset. Accuracy is 14.06% if we use images from 3 –front, left and right – angle for every person. Similarly, when we increase the number of images in training dataset accuracy also increases. For instance, for 30 images accuracy are 19.08%, 26.13% and 28.27% respectively for front faced images in training data set, for 3 angle faces in training dataset and for 5 angle faces in training dataset for every person. Thus this way the accuracy increases with the increase of trained image amount. However, the accuracy doesn't increase that much when the trained image amount is more than 300. Here, for 300 images the accuracy for front face, 3 angle faces and 5 angle faces are respectively 45%, 64.87% and 68.08% which are very close to the accuracy of previous sample size which is 270 images per person. So, we can conclude on this analysis that around 300 images for a person will give effective result while training model for this system. Shown in figure 5.34.

Figure 5.34: Average accuracy of all persons on different angle viewpoint

5.4 Threshold

Finally here comes the most crucial part , threshold value. If the system gets the input image containing faces that are not in the database, then it should display unknown person recognized as output. However, the system calculate the euclidean distance for input image respect to persons that are in the current database, and then calculate probability distribution value for each person from that input image. In that case, the person which gets the highest value is the output of the system, but we don't want this output, we want to display output like "unknown person". That is why threshold value is so important in this scenario. We tested few images to identify the threshold value so that system can recognized unknown person for faces which are not in the current database. The following 3 images contain 3 different person which are not in the database , but it shows 3 different person name with 0.13 , 0.15 , 0.12 values respectively which is very low. Currently the default threshold value is 0.1. shown in figure 5.35.

Clearly we can see that now with 0.2 threshold value system can tell us unknown person successfully which are not in our dataset. shown in figure 5.36.

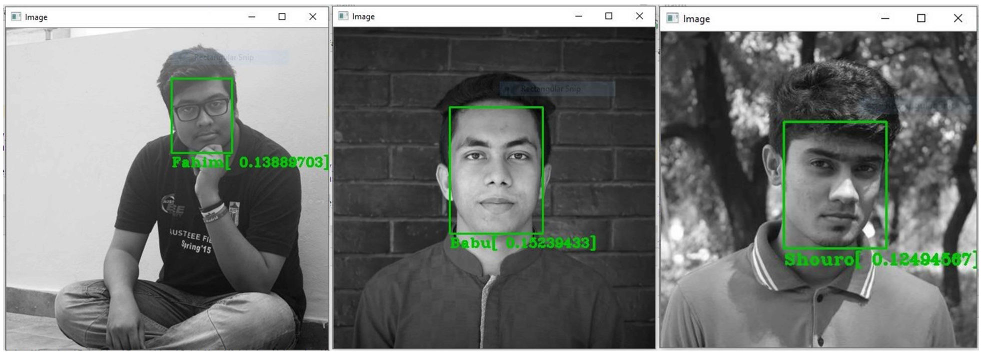

Figure 5.35: without threshold wrong prediction

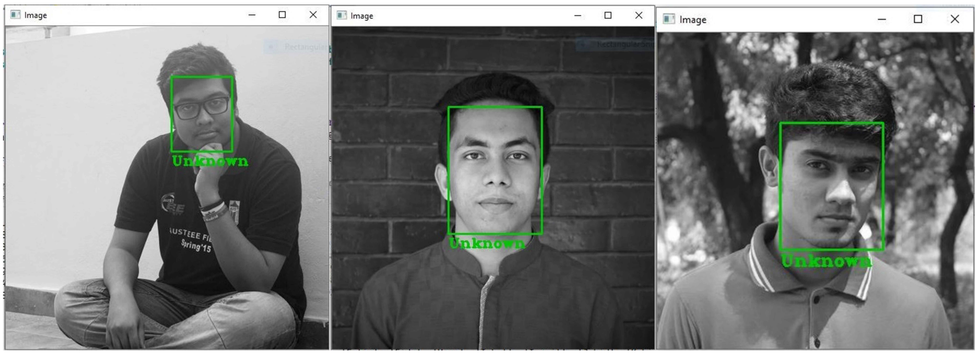

Figure 5.36: with threshold correct prediction

After giving a threshold we tested randomly many photos and also with videos where everything worked fine. For example, figure 5.37 shows randomly tested image where it was successful to identify the appropriate class that was trained earlier. This predicts Sujon and Beni with 89.13% and 91.02% accuracy respectively.

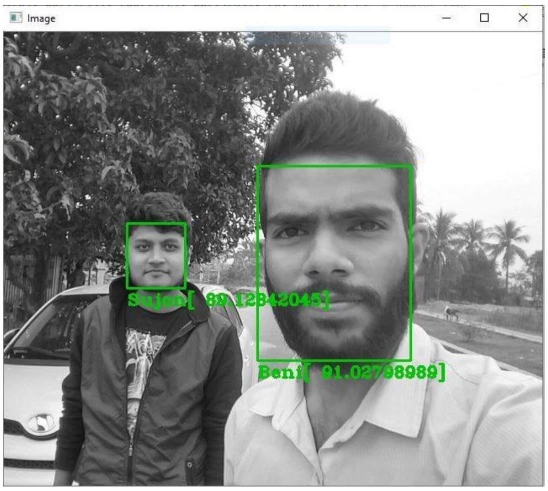

Figure 5.37: Randomly Tested image(Sujon: 89.13% , Beni: 91.02%)

5.5 Comparison

Here, in this section we have showed some comparison among some papers for face recognition. Based on Complexity , dataset per person and working area the comparison has been made to understand the overall face recognition system.

Title	Effect of Super Resolution on High Dimensional Features for Unsupervised Face Recognition in the Wild	Pose-Varied Face Recognition Based on 3-D Face Model	A Geometrical-Model-Based Face Recognition	FaceNet: A Unified Embedding for Face Recognition and Clustering	Analysis on Face Recognition based on five different viewpoint of face images using MTCNN and FaceNet
Year	2017	2004	2015	2015	2019
Complexity	High	Very High	High	Medium	Medium
Dataset per person	N/A	N/A	N/A	Around 20 images	5 Images
Working Area	Low resolution and Blurred Images	3D Face	LVP BR	Face Embedding Points	Images from 5 angle viewpoint

Table 5.16: Comparison with other face recognition system

Chapter 6

Conclusion

Face recognition is both challenging and important recognition technique. Among all the biometric techniques, face recognition approach possesses one great advantage, which is its user-friendliness (or non-intrusiveness). In this paper, we have given an introductory survey for the face recognition technology. This paper proposed FaceNet based face recognition system by training five viewpoint images only by duplicating those same five viewpoint images again and again. We have covered issues such as face alignment and accuracy vs training data for face recognition, factors that may affect the performance of the recognizer. We hope this paper can provide the readers with a better understanding of face recognition and helps to create more options and better opportunities in face recognizing techniques. This method is our preliminary attempt and the pre-trained FaceNet model is used. In the future, we look forward to developing our own powerful neural network for face recognition by training very low amount of face images.

References

[1] C. Liu, "The development trend of evaluating face-recognition technology", in *2014 International Conference on Mechatronics and Control (ICMC)*, Jul. 2014, pp. 1540–1544. DOI: 10.1109/ICMC.2014.7231817.

[2] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering", in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 815–823. DOI: 10.1109/CVPR.2015.7298682.

[3] J. Xiang and G. Zhu, "Joint face detection and facial expression recognition with mtcnn", in *2017 4th International Conference on Information Science and Control Engineering (ICISCE)*, Jul. 2017, pp. 424–427. DOI: 10.1109/ICISCE.2017.95.

[4] J. Dhamija, T. Choudhury, P. Kumar, and Y. S. Rathore, "An advancement towards efficient face recognition using live video feed: "for the future"", in *2017 3rd International Conference on Computational Intelligence and Networks (CINE)*, Oct. 2017, pp. 53–56. DOI: 10.1109/CINE.2017.21.

[5] P. Jaturawat and M. Phankokkruad, "An evaluation of face recognition algorithms and accuracy based on video in unconstrained factors", in *2016 6th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, Nov. 2016, pp. 240–244. DOI: 10.1109/ICCSCE.2016.7893578.

[6] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection", in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, Jun. 2005, 886–893 vol. 1. DOI: 10.1109/CVPR.2005.177.

[7] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees", in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2014, pp. 1867–1874. DOI: 10.1109/CVPR.2014.241.

[8] A. ElSayed, A. Mahmood, and T. Sobh, "Effect of super resolution on high dimensional features for unsupervised face recognition in the wild", Oct. 2017, pp. 1–5. DOI: 10.1109/AIPR.2017.8457967.

[9] R. S. Ghiass, O. Arandjelović, H. Bendada, and X. Maldague, "Infrared face recognition: A literature review", in *The 2013 International Joint Conference on Neural Networks (IJCNN)*, Aug. 2013, pp. 1–10. DOI: 10.1109/IJCNN.2013.6707096.

[10] Z.-J. Liu, X.-Z. Li, and M. Wu, "Pose-varied face recognition based on the 3d model", vol. 31, pp. 218–222, Apr. 2004.

[11] Y. Huang and S. Chen, "A geometrical-model-based face recognition", in *2015 IEEE International Conference on Image Processing (ICIP)*, Sep. 2015, pp. 3106–3110. DOI: 10.1109/ICIP.2015.7351375.

[12] H. Kim, S. H. Lee, and Y. M. Ro, "Face image assessment learned with objective and relative face image qualities for improved face recognition", in *2015 IEEE International Conference on Image Processing (ICIP)*, Sep. 2015, pp. 4027–4031. DOI: 10.1109/ICIP.2015.7351562.

[13] A. Vinay, D. N. Reddy, A. C. Sharma, S. Daksha, N. S. Bhargav, M. K. Kiran, K. N. B. Murthy, and S. Natrajan, "G-cnn and f-cnn: Two cnn based architectures for face recognition", in *2017 International Conference on Big Data Analytics and Computational Intelligence (ICBDAC)*, Mar. 2017, pp. 23–28. DOI: 10.1109/ICBDACI.2017.8070803.

[14] S. Guo, S. Chen, and Y. Li, "Face recognition based on convolutional neural network and support vector machine", in *2016 IEEE International Conference on Information and Automation (ICIA)*, Aug. 2016, pp. 1787–1792. DOI: 10.1109/ICInfA.2016.7832107.

[15] K. Dang and S. Sharma, "Review and comparison of face detection algorithms", in *2017 7th International Conference on Cloud Computing, Data Science Engineering - Confluence*, Jan. 2017, pp. 629–633. DOI: 10.1109/CONFLUENCE.2017.7943228.

[16] X. Liang, X. Cai, L. Li, and Y. Chen, "Facial feature extraction method based on shallow and deep fusion cnn", in *2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, Jul. 2017, pp. 50–53. DOI: 10.1109/FSKD.2017.8393320.

[17] H. Yanagisawa, T. Yamashita, and H. Watanabe, "A study on object detection method from manga images using cnn", in *2018 International Workshop on Advanced Image Technology (IWAIT)*, Jan. 2018, pp. 1–4. DOI: 10.1109/IWAIT.2018.8369633.

[18] E. Zaytseva and J. Vitrià, "A search based approach to non maximum suppression in face detection", in *2012 19th IEEE International Conference on Image Processing*, Sep. 2012, pp. 1469–1472. DOI: 10.1109/ICIP.2012.6467148.

[19] J. Cao, Z. Su, L. Yu, D. Chang, X. Li, and Z. Ma, "Softmax cross entropy loss with unbiased decision boundary for image classification", in *2018 Chinese Automation Congress (CAC)*, Nov. 2018, pp. 2028–2032. DOI: 10.1109/CAC.2018.8623242.

[20] M. D. Malkauthekar, "Analysis of euclidean distance and manhattan distance measure in face recognition", in *Third International Conference on Computational Intelligence and Information Technology (CIIT 2013)*, Oct. 2013, pp. 503–507. DOI: 10.1049/cp.2013.2636.

[21] Zhao Lihong, Song Ying, Zhu Yushi, Zhang Cheng, and Zheng Yi, "Face recognition based on multi-class svm", in *2009 Chinese Control and Decision Conference*, Jun. 2009, pp. 5871–5873. DOI: 10.1109/CCDC.2009.5195250.