Face Recognition Using Deep Learning

CS4090 Project Final Report

Submitted by

Bastian Jose	**B150343CS**
Jishnu L	**B150266CS**
Kevin George	**B150444CS**
Mohammed Basil A.K	**B150451CS**

Under the Guidance of
Dr.Jayprakash P

तमसो मा ज्योतिर्गमय

Department of Computer Science and Engineering
National Institute of Technology Calicut
Calicut, Kerala, India - 673 601

April 25, 2019

NATIONAL INSTITUTE OF TECHNOLOGY
CALICUT, KERALA, INDIA - 673 601

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

2019

CERTIFICATE

Certified that this is a bonafide record of the project work titled

FACE RECOGNITION USING DEEP LEARNING

done by

Bastian Jose

Jishnu L

Kevin George

Mohammed Basil A.K

of eighth semester B. Tech in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering of the National Institute of Technology Calicut

Project Guide
Dr.Jayprakash P
Associate Professor

Head of Department
Dr. Saleena N
Associate Professor

DECLARATION

I hereby declare that the project titled, **Face Recognition Using Deep Learning**, is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or any other institute of higher learning, except where due acknowledgement and reference has been made in the text.

Place :
Date :

Signature :
Name :
Reg. No. :

Abstract

In this project, we aim to use Computer vision to identify faces using deep learning techniques. The aim is to build a face recognition based attendance system that outputs names of faces if stored in the database, given an input video stream and to measure accuracy using confusion matrix.

ACKNOWLEDGEMENT

We took much effort in the completion of this project. However, this would not have been possible without the help and guidance of many individuals.

We owe our gratitude to our project guide Dr.Jayprakash., who took interest on our project work and guided us.

Contents

List of Figures

Chapter 1

Introduction

Face recognition is a challenging, ever-improving field which has made giant leaps over the past few years. Face recognition technology can be used for building authentication systems, identification of each individual from CCTV, advertising, healthcare, and many others. Face recognition involves both face identification and face verification.

Previous face recognition algorithms made use of thermal readings, 3D mapping of a face, biometric features.etc. But we make use of an algorithm that far outperforms existing systems by forming relations between RGB values of pixels[1]. The system uses the simple vector form of representation using deep learning framework for each face. But still performs much better than systems that use biometric identification, as this system isn't influenced by factors like lighting, expression, and occlusion[1]. In the algorithm, each face is represented is a vector in an n-dimensional plane. And the similarity between faces is measured using the distance between these faces[2]. The input is as a live feed from the webcam and is compared with stored face data.

Chapter 2

Literature Survey

An early example of a face recognition system was by Kohonen (1989)[3], who demonstrated that a simple neural net could perform face recognition for well aligned, normalized face images. The type of network he employed computed a face description by approximating the eigenvectors also known as 'eigenfaces'. Feature selection from face was implemented using PCA or Karhunen-Loeve method. This method was used by Sirovich and Kirby (1990)[5] to get a low-dimensional representation of facial images . They showed that a set of basic features can be produced by analyzing a collection of face images. But poor discriminating power within the class and large computation power are problems of using PCA. SVM was used for face recognition successfully after applying a feature extraction method[7]. SVMs were implemented using a decomposition algorithm that guarantees global optimality and also can be trained over very large datasets. But SVM cannot be applied when the feature defining samples have missing entries.Turk and Pentland (1991)[4] also demonstrated that face detection followed by face recognition could achieve reliable, real-time recognition of faces in a constrained environment. This demonstration that simple, real-time pattern recognition techniques could be combined to create a useful system sparked an explosion of interest in the topic of face recognition. With the advent of

Artificial neural networks using feed forward learning algorithms[8] and usage of CNN[9](convolutional neural networks) more powerful face recognition systems that can approximate complex decision surfaces much more readily were built.

Chapter 3

Face Recognition

3.1 Problem Definition

Build a deep learning model that identifies faces from a visual input stream.

3.2 Dataset

Images were collected using redmi note 4 front camera.It was then preprocessed and added to the dataset.

3.2.1 Training Set

Training set consists of faces of each individual in the database, the number of images of each individual in the dataset can be increased. When adding a user it adds 5 images of that particular individual to the dataset. We can update the dataset by logging in, which adds another 5 images of that person logged in. When adding user a directory for that user is created in which we can add images from any source which is then processed and added to the dataset. There is a total of 4 individuals in the training set at present.

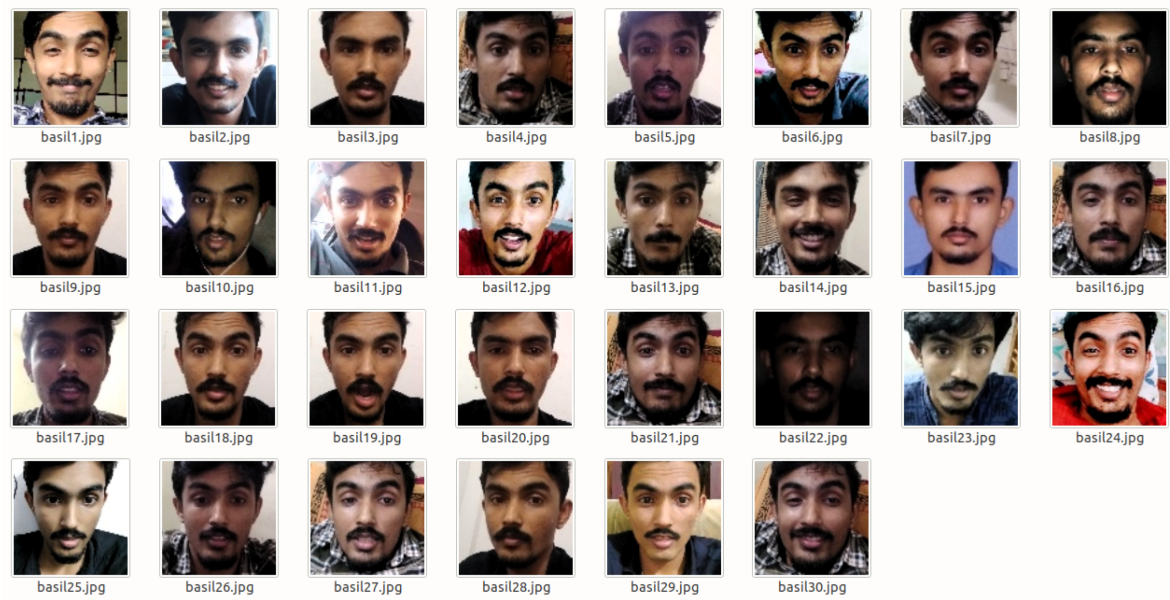


Figure 3.1: Sample dataset of an individual

3.2.2 Test Set

The test set consists of 10 image instances of each individual in the database and a total of 30 images instances of individuals not in the database. There is a total of 4 individuals in the database so there are 4*10+30=70 faces in the test set.

Chapter 4

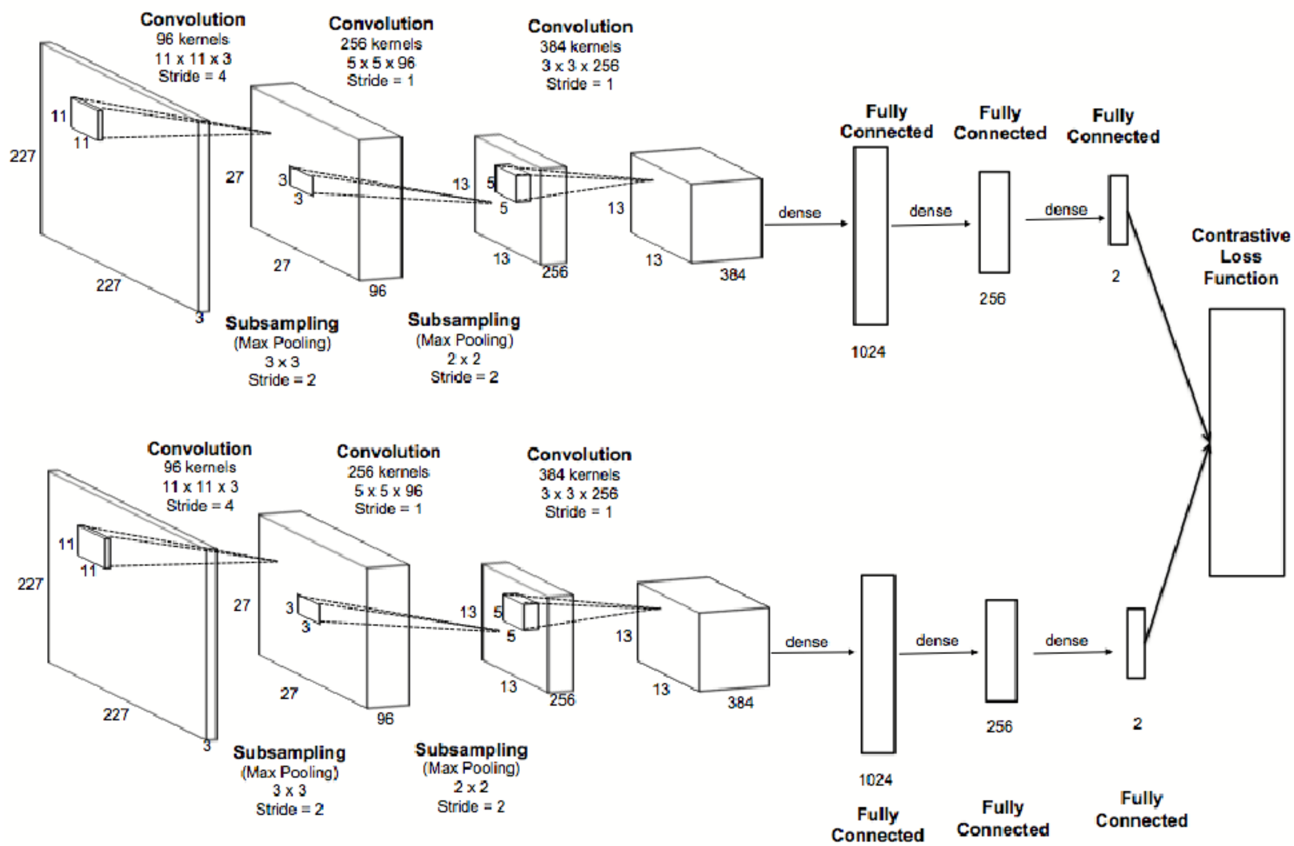
Methodology

Face Recognition problem involves 3 phases detection, align, feature extraction and classify.

4.1 Detection and Align

4.1.1 Siamese Network

Siamese networks[1] are subnetworks that are identical, have identical architecture and share the same weights.
The main function of siamese networks is in comparing between outputs of these subnetworks. The inputs can be anything from numerical data to images. The face recogntion problem makes use of siamese networks to compare the distance between images or the similarity between 2 faces.

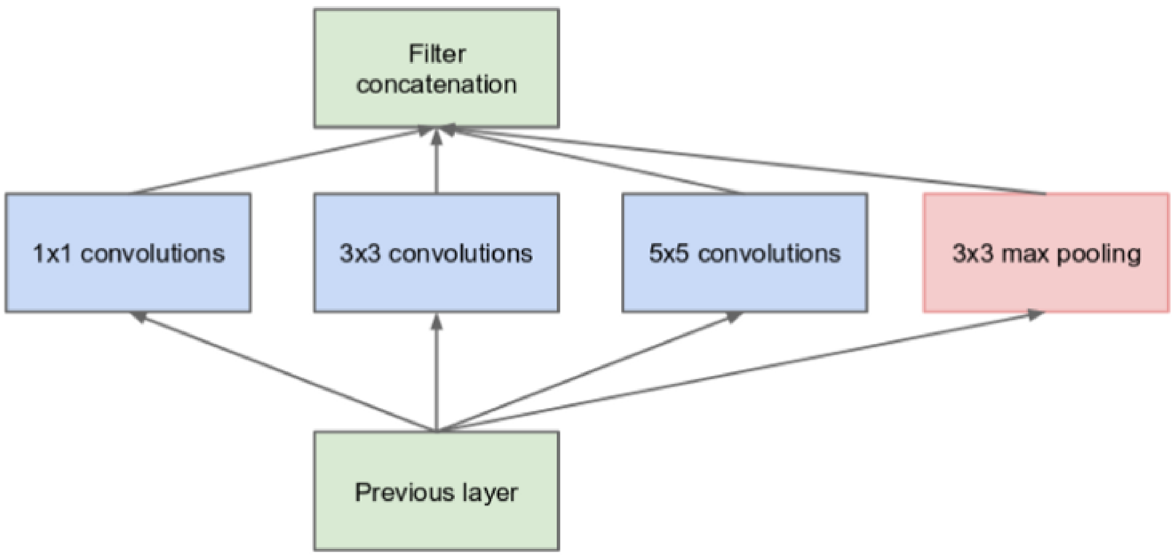


[1](Y. Taigman et al.,2014)

Figure 4.1: Siamese Network

4.1.2 Inception v1 Network

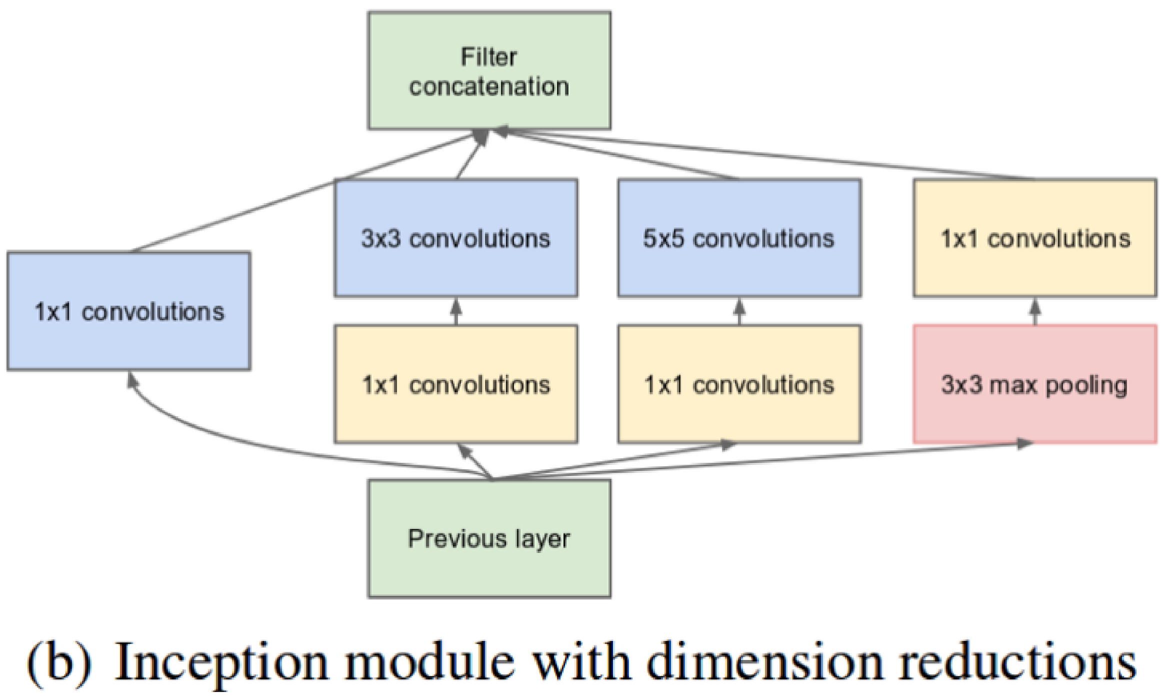
The Inception v1[6] network compensates for the many disadvantages of deep CNN. It makes use of different sized kernels, speeding up computation using 1x1 convolution.



(a) Inception module, naïve version

[6](Szegedy et al.,2015)

Figure 4.2: Inception Module Naive version



[6](Szegedy et al.,2015)

Figure 4.3: Inception Module with Dimension Reductions

4.1.3 Transfer Learning

Transfer learning[10] and domain adaptation refer to the situation where what has been learned in one setting is exploited to improve generalization in another setting.

4.1.4 Triplet Loss Function

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

[2](F. Schroff et al.,2015)

Figure 4.4: Triplet Loss Function

Triple Loss Function[2] ensures that the difference between vector forms of images is indeed a representation of similarity between faces. Once this embedding has been produced, then the aforementioned tasks become straight forward: face verification simply involves thresholding the distance between the two embeddings; recognition becomes a k-NN classification problem, and clustering can be achieved using off-the-shelf techniques such as k-means or agglomerative

4.2 Implementation

The implementation of the project was done using django. Django is a Python-based free and open-source web framework, which follows the model-view-template architectural pattern.
The implementation was done such that it can be hosted in a local network and android devices in that network can use this for face recognition and attendance of the individuals in the database.

4.2.1 Requirements

Django: It is a Python-based free and open-source web framework, which follows the model-view-template (MVT) architectural pattern. In this model is sqlite3, template is html,css and javascript and view is python.
Python libraries: Numpy , dlib, tensorflow, keras, imutils and opencv are the major libraries used.
Other: For the project to work in an android device we require a software that converts the device camera to ip camera.We use IP Webcam app for this purpose.

4.2.2 Functionalities

1.Add User

2.Update Face

3.Face detection from a single frame

4.Attendance

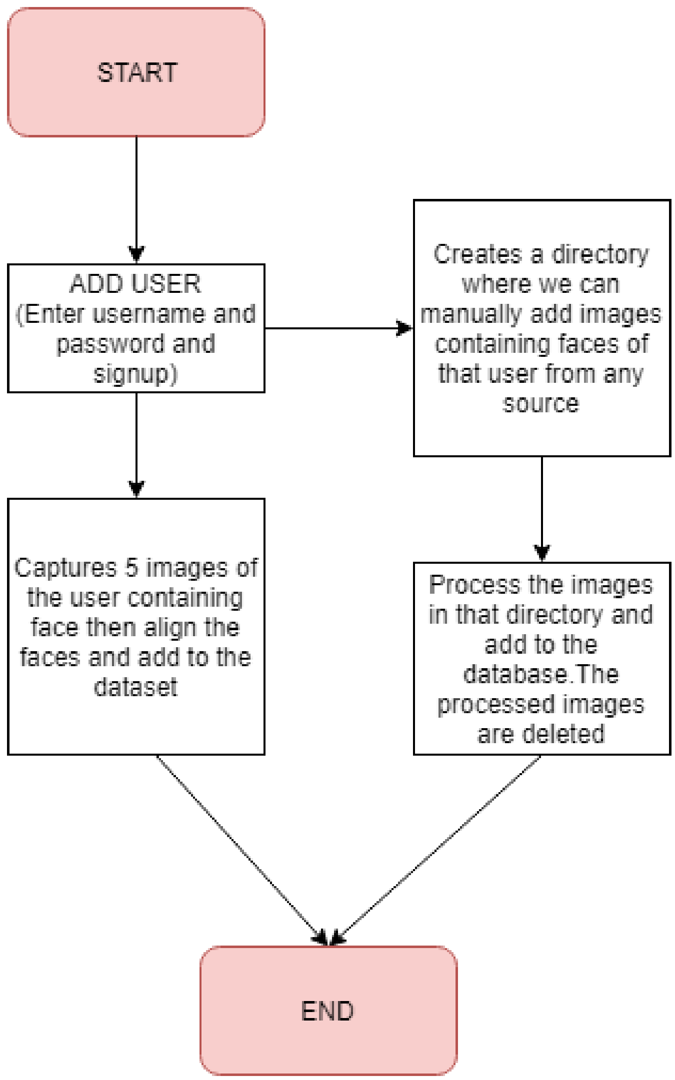


Figure 4.5: Add User

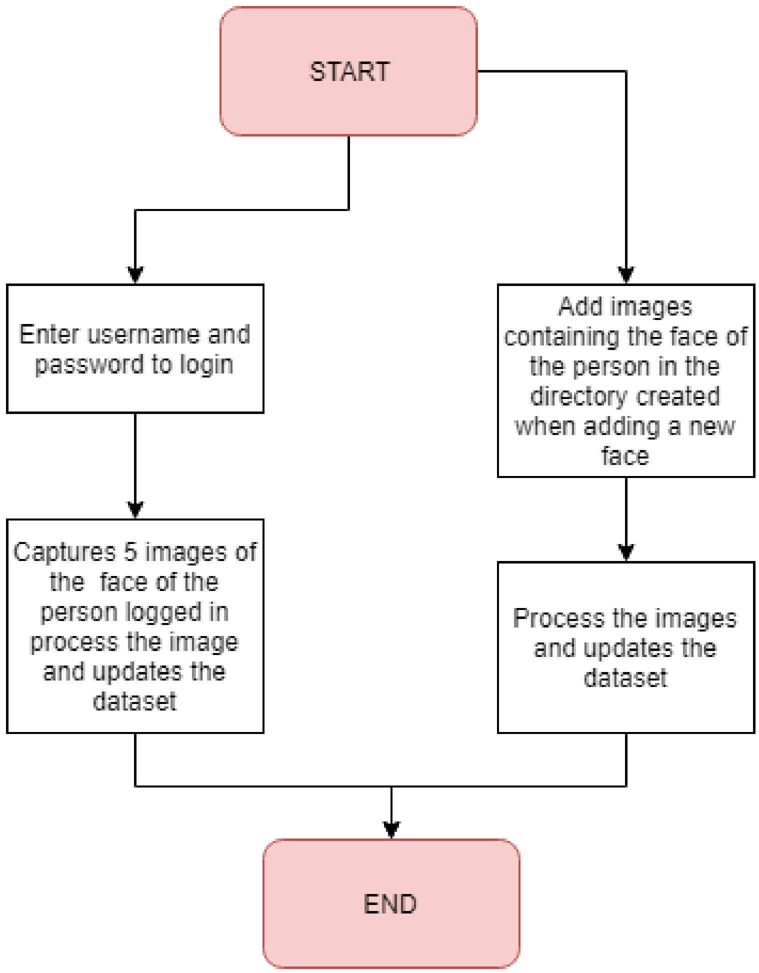


Figure 4.6: Update Face

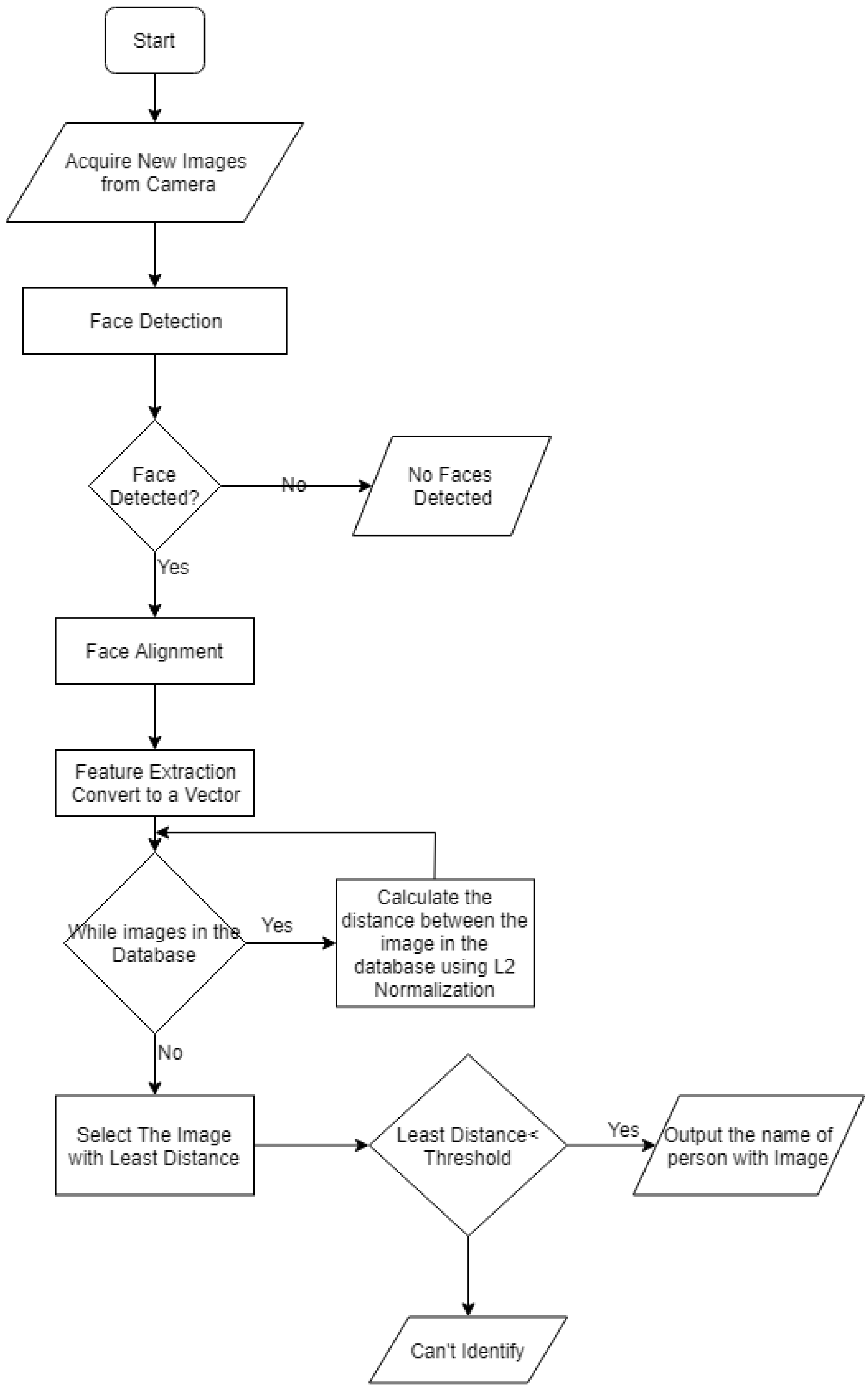


Figure 4.7: Face detection from a single frame

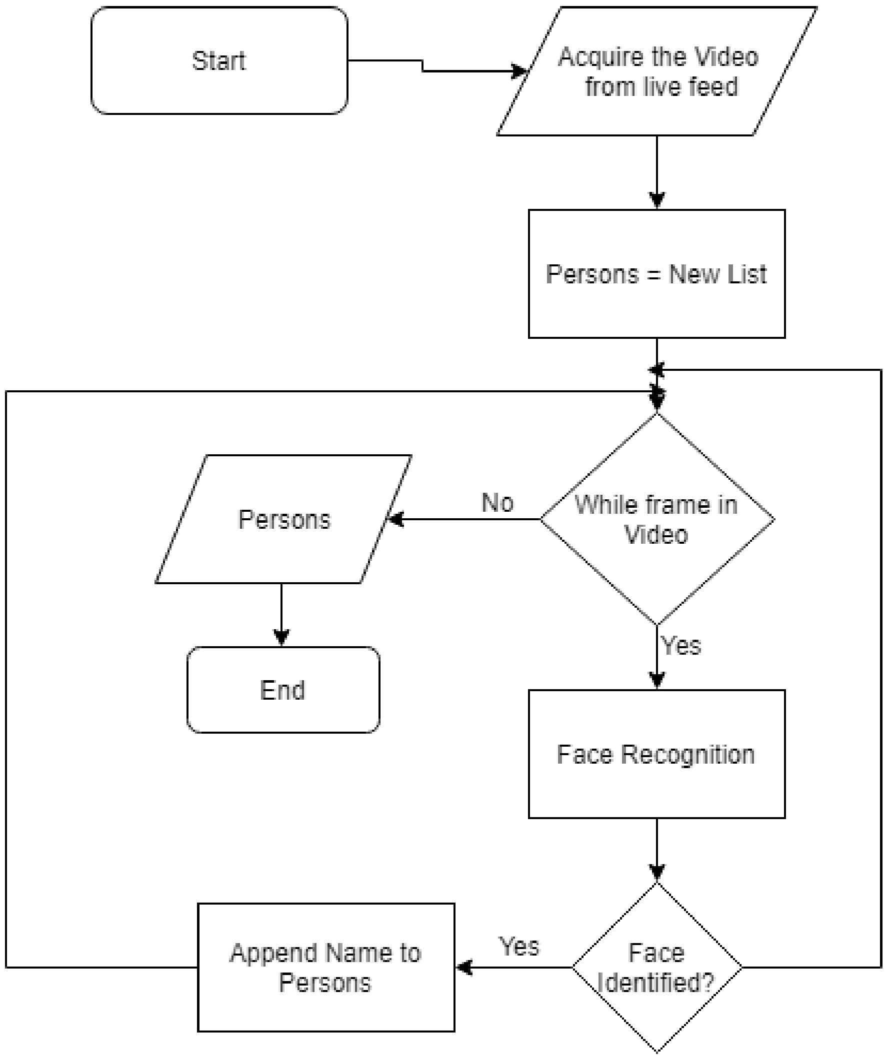


Figure 4.8: Attendance

Chapter 5

Results

5.1 Test Cases

Bastian:10 instances
Basil:10 instances
Kevin:10 instances
Jishnu:10 instances
Unknown:30 instances

5.2 Observations

Table 5.1: Observation

	Bastian(P)	Basil(P)	Kevin(P)	Jishnu(P)	Unknown(P)
Bastian(A)	7	3	0	0	0
Basil(A)	0	10	0	0	0
Kevin(A)	0	1	9	0	0
Jishnu(A)	0	2	1	4	3
Unknown(A)	1	4	3	0	22

Table 5.2: Confusion Matrix

	True Positive	**True Negative**
Predicted Positive	30	15
Predicted Negative	3	22

Table 5.3: Measures of accuracy

Measure	Value	Derivation
Accuracy	0.7429	ACC = (TP + TN) / (P + N)
Sensitivity	0.9091	TPR = TP / (TP + FN)
Specificity	0.5946	SPC = TN / (FP + TN)
Precision	0.6667	PPV = TP / (TP + FP)
F1 Score	0.7692	F1 = 2TP / (2TP + FP + FN)

Table 5.4: Dataset vs Accuracy

	Dataset	**Accuracy**
Bastian	30	70%
Basil	62	100%
Kevin	60	90%
Jishnu	6	40%

Table 5.5: Observation for 2 individuals away from camera

	Bastian(P)	Kevin(P)	Other(P)
Bastian(A)	6	0	4
Kevin(A)	0	8	2

Table 5.6: Observation for 2 individuals near camera

	Bastian(P)	Kevin(P)	Other(P)
Bastian(A)	7	0	3
Kevin(A)	0	9	1

Table 5.7: Distance vs Accuracy

	Near	Away
Bastian	70%	60%
Kevin	90%	80%

Table 5.8: Observation for 2 individuals for camera 1(used to create dataset)

	Bastian(P)	Kevin(P)	Other(P)
Bastian(A)	7	0	3
Kevin(A)	0	9	1

Table 5.9: Observation for 2 individuals for camera 2

	Bastian(P)	Kevin(P)	Other(P)
Bastian(A)	4	0	6
Kevin(A)	0	7	3

Table 5.10: Camera vs Accuracy

	Camera 1	Camera 2
Bastian	70%	40%
Kevin	90%	70%

5.3 Analysis

Few images were required for training and only RGB values are required for feature extraction.
With increase in number of images of an individual in the dataset the accuracy for that particular individual increases.
Images from same type of camera used to create dataset gave better accuracy measure.
With increase in distance from camera accuracy decreases.

5.4 Examples

5.4.1 Single Frame

Input: Image acquired from the camera
Sample Outputs:

Figure 5.1: Sample output 1

Figure 5.2: Sample output 2

5.4.2 Video

Input: Video stream from the camera
Sample output:

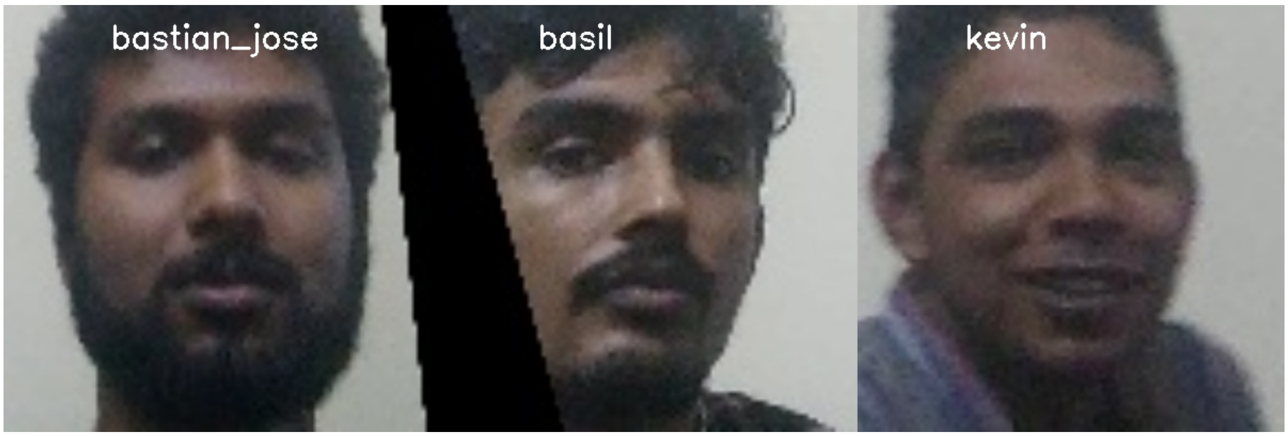


Figure 5.3: Sample output

Chapter 6

Conclusion and Future work

6.1 Future Work

Live Updation of Attendance Register

Instead of having to process the complete video after recording,the frames are send for processing while recording for the live updation of attendance register.

Device wide support

The software only works on the host device and the clients which runs on android. Improvement of the system would be to have support across multiple platforms.

Module for Uploading photos

When a user signs up, snapshot of user is taken using the device camera which is saved in the database. Administrator can manually copy the other photos of the user to the corresponding folder in the database. A module can be implemented where a user can upload their photos themselves to the server.

References

[1] Y. Taigman, M. Yang, M. Ranzato and L. Wolf *DeepFace: Closing the Gap to Human-Level Performance in Face Verification*, 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, 2014, pp. 1701-1708.

[2] F. Schroff, D. Kalenichenko and J. Philbin *FaceNet: A unified embedding for face recognition and clustering*, 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, 2015, pp. 815-823.

[3] T. Kohonen *Self-organization and Associative Memory*, Springer-Verlag, Berlin, 1989.

[4] M. Turk and A. Pentland *Eigenfaces for recognition*, J. Cog. Neuroscience, vol. 3, no. 1, pp. 71-86, 1991.

[5] M. Kirby and L. Sirovich *Application of the karhunen-loeve procedure for the characterization of human faces*, IEEE Pattern Analysis and Machine Intelligence, vol. 12, no. 1, pp. 103-108, 1990.

[6] C. Szegedy et al. *Going deeper with convolutions*, 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, 2015, pp. 1-9.

[7] E. Osuna, R. Freund and F. Girosit *Training support vector machines: an application to face detection* Proceedings of IEEE Computer Society

Conference on Computer Vision and Pattern Recognition, San Juan, Puerto Rico, USA, 1997, pp. 130-136.

[8] Xiaoguang Li and S. Areibi *A hard- ware/software co-design approach for face recog- nition* Proceedings. The 16th International Conference on Microelectronics, 2004. ICM 2004., Tunis, Tunisia, 2004, pp. 55-58.

[9] A. Bouzerdoum *Classification and function approximation using feed-forward shunting in- hibitory artificial neural networks*, Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Per- spectives for the New Millennium, Como, Italy, 2000, pp. 613-618 vol.6.

[10] Pan, Sinno Jialin and Qiang Yang. *A Survey on Transfer Learning.* IEEE Transactions on Knowledge and Data Engineering 22 (2010): 1345-1359.