SMART DOOR UNLOCKING

A DISSERTATION

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE AWARD OF DEGREE
OF

MASTER OF TECHNOLOGY
IN
COMPUTER SCIENCE ENGINEERING

Submitted by:

KARTHIK K M

2K17/CSE/06

Under the supervision of

Dr. RAJESH KUMAR YADAV
(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

JUNE, 2019

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi - 110042

CANDIDATE'S DECLARATION

I, Karthik K M , Roll No. 2K17/CSE/06 student of M.Tech (Computer Science and Engineering), hereby declare that the project Dissertation titled **"SMART DOOR UNLOCKING"** which is submitted by me to the Department of Computer Science & Engineering, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of and Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: Delhi	Karthik K M
Date:	2K17/CSE/06

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi - 110042

CERTIFICATE

I hereby certify that the Project Dissertation titled **"SMART DOOR UNLOCKING"** which is submitted by Karthik K M, 2K17/CSE/06 Department of Computer Science & Engineering, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree of Master of Technology, is a record of the project work carried out by the students under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere

Place: Delhi

Date:

Dr. R.K. Yadav
SUPERVISOR
Assistant Professor
Department of Computer Science & Engineering
Delhi Technological University

ACKNOWLEDGEMENT

The success of this project requires the assistance and input of numerous people and the organisation. I am grateful to everyone who helped in shaping the result of the project.

I express my sincere thanks to **Dr. R.K. Yadav**, my project guide, for providing me with the opportunity to undertake this project under his guidance. I was always supported and encouraged by him to realise that the learning process is more important than the final outcome. I am very indebted to the faculties for their guidance, continuous monitoring and motivation during all the progress evaluations to complete my work. They helped me throughout with new ideas, provided information necessary and pushed me to complete the work.

I also thank all my fellow students and my family for their continued support.

KARTHIK K M

ABSTRACT

Internet of Things (loT) has become one of the promising technologies used to connect, manage and control smart objects connected to the Internet. IoT's main goal is to smarter and more meaningful management and control of physical objects around us and improving life quality by offering economic, secure and entertaining livelihoods. In the development of smart cities, smart homes play a key role among many IoT applications. Various tech giants have provided various devices like Google home, Amazon echo which allows you to control lighting in your home, play music, read news and various other features which are controllable with voice. Smart Home also helps physically challenged people in their day to day life activities. Smart door unlocking is a step towards the make of smart homes which allows to automatically unlock door for known people and also allowing user to open door remotely through smartphone. People face various problem with door unlocking like lost key, no keys and thus making a person to wait for someone else to open the door or waiting for someone to arrive with keys if door is locked, approach the door if someone has arrived. Disabled people living alone will have more issues if someone comes and they have to open the door. Therefore a smart door that operates automatically which is secured and convenient way of door unlocking that uses face recognition and eye blink detection has been proposed.

CONTENTS

Candidates's Declaration i

Certificate ii

Acknowledgement iii

Abstract iv

Contents v

List of Figures vii

List of Tables ix

List Of Abbreviations x

CHAPTER 1 INTRODUCTION 1

1.1 Internet of Things 1

1.1.1 Applications of IoT 2

1.1.2 Challenges in IoT 4

1.1.3 Architecture of IoT 6

1.2 Smart Homes 7

1.2.1 Layered architecture of Smart Home System 8

CHAPTER 2 RELATED WORK 10

CHAPTER 3 PROPOSED WORK 13

3.1 Problem statement 13

3.2 Proposed solution 13

3.3 Summary of proposed solution 17

CHAPTER 4 WORKING AND ANALYSIS 19

4.1 Face recognition 19

4.1.1 Face detection algorithm 20

4.1.2 Face matching algorithm 26

4.2 Face training 28

4.3 Face matching 30

4.4 Eye blink detection 31

CHAPTER 5 IMPLEMENTATION 33

5.1 Face recognition system 33

5.1.1 Face training 34

5.1.2 Face matching and eye blink detection 34

5.2 Android 35

5.3 Server 37

CHAPTER 6 RESULT ANALYSIS 41

6.1 Face Recognition 41

6.2 Eye blink detection 42

CONCLUSION AND FUTURE SCOPE 44

REFERENCES 46

LIST OF FIGURES

1.1 Applications of IoT 4

1.2 IoT architecture 6

1.3 Smart Home concept 7

1.4 Layered architecture of Smart Home System 9

3.1 Basic flow of the proposed algorithm 14

3.2 Face spoofing detection flow 15

3.3 Smartphone user authentication 16

4.1 Different Haar features 20

4.2 Example of Haar 21

4.3 LBP Feature calculation I 21

4.4 LBP Feature calculation II 22

4.5 Simple Neural Network 23

4.6 Deep Neural Network 23

4.7 Analysis of DNN with different confidence factor 26

4.8 People for training the system 27

4.9 One shot learning concept 28

4.10 Working of face training 29

4.11 Face matching process using dlib 30

4.12 Facial indicators linked with the eye 32

5.1 Implementation of system flow 33

5.2 Face training working 34

5.3 Facial landmarks 35

5.4 Implementation of blink detection based on EAR 35

5.5 Implementation of authentication for android user 36

5.6 REST API - Success response 37

5.7 REST API - Error response 38

5.8 REST API - Visitor 38

5.9 FCM success message after sending notification 39

5.10 Example of FCM 40

6.1 Face recognition result 42

6.2 Result of blink detection without spectacles 43

6.3 Result of blink detection with spectacles 43

LIST OF TABLES

2.1 Comparison of different communication approaches 11

4.1 Working of algorithms on different orientation 24

4.2 Working of algorithms on different lighting condition 25

4.3 Working of algorithms on random photographs 26

4.4 Face matching algorithms comparison 28

LIST OF ABBREVIATIONS

1. IoT: Internet of Things
2. LFW : Labeled Faces in the Wild
3. LBP : Local Binary Pattern
4. DNN : Deep Neural Network
5. JSON : Java Script Object Notation
6. EAR : Eye Aspect Ratio
7. NN : Neural Network
8. NFC : Near Field Communication
9. GSM : Global System for Mobile Communications
10. SMS : Short Message Service
11. REST : Representational State Transfer
12. DRF : Django Rest Framework
13. API : Application Programming Interface
14. FCM : Firebase Cloud Messaging

CHAPTER 1

INTRODUCTION

1.1 Internet of Things

Internet is a marvellous thing and it gives us all kinds of advantages, which were previously not possible. Think about your mobile phone before it became a smartphone. Now, you can read every book, look into any movie, listen to all the songs in the hand palm. In reality, the Internet of Things is a simple concept, it means to take and connect all things in the world with the internet. IoT is a network of millions of devices devices equipped with some kind of sensors that gather some data and these data are transmitted over the network. The things in IoT refers to any device like sensor present in automobile, a heart monitor carried by a person, cameras getting live feed of animals in forest. These things are then connected to the internet i.e they have been assigned an IP address and the data data collected by these devices are then transferred over the network. Based on these data various actions can be performed. It is currently one of the emerging topics nowadays.

All the things that are connected to Internet can be classified into three categories:

1. Things that gather and transmit information

This include sensors like air quality sensor, temperature sensor, IR sensor and many more. These sensors gather information from their environment and send it forward. Example, farmers can get information regarding soil moisture and can decide accordingly to water right amount of water to crops

2. Things which obtain information and act accordingly

The sensors receives some information and act accordingly. For example, car receives signal from keys and the door opens, printer receives information and prints. There are numerous examples, but the real IoT emerges when things can perform both above.

3. Things that does both

Taking the farming example, the sensors gather information of soil moisture and tell farmers the amount of water needed to crops. Rather than involving a farmer the system can automatically decide the amount and provide the water itself. Extending it a step further, if the system receives information regarding weather, then it would also know if it's going to rain or not and can decide to water the crops or not.

1.1.1 Applications of IoT

With the increase in demand of IoT, it is predicted the count of the connected devices will reach approx. 24 billion by 2020. Following are some of the areas where IoT is being used to carry out the task efficiently:

1. Smart Home

Living in a home that acts smartly where we can control appliances remotely through smartphone or appliances take actions on their own based on the environment for example turning the lights on/off based on the people present in the room, turning on the air conditioner remotely in a hot summer weather, monitoring of the appliances.

2. Wearables

Wearable devices are equipped with different sensors that collect data and this data is processed to provide user with meaningful information. These wearable generate information mainly related to fitness, entertainment and health.

3. Connected cars

A car connected to the internet where the onboard sensors gather data to provide comfort, optimise operation, maintenance. For example a sensor monitors the pressure in the tyre and it alerts user if pressure below certain threshold.

4. Industrial IoT

The Industrial Internet of Things (IIoT), combines machines, people at work and advanced analytics. It is the network of devices that are connected to each other leading to systems, which like never before can monitor, collect, analyse and provide new outputs. These insights can then assist industrial companies in making beneficial business decisions.

5. Agriculture

IoT in the field of agriculture leads to the development of smart farming. It is used to detect soil dampness and complements, control the use of water for the development of plants and decide on custom-made compost. The data generated by the sensors are used by the farmers to improve the profitability. IoT-based smart farming is highly efficient when compared with the conventional approach.

6. Smart retail

Use of IoT in shops would give a chance to retailers improve the experience of consumers for in-store shopping. Retailers can use Beacon technology to interact with smartphones for providing better service to the consumers. They may improve their profit by tracking the users path and modifying the layout of the store accordingly.

7. Medical

IoT in the field of medical aims at providing a healthier life to people by wearing devices connected to internet that collects different data. This data can then be used in the analysis of health and provide strategies to fight illness.

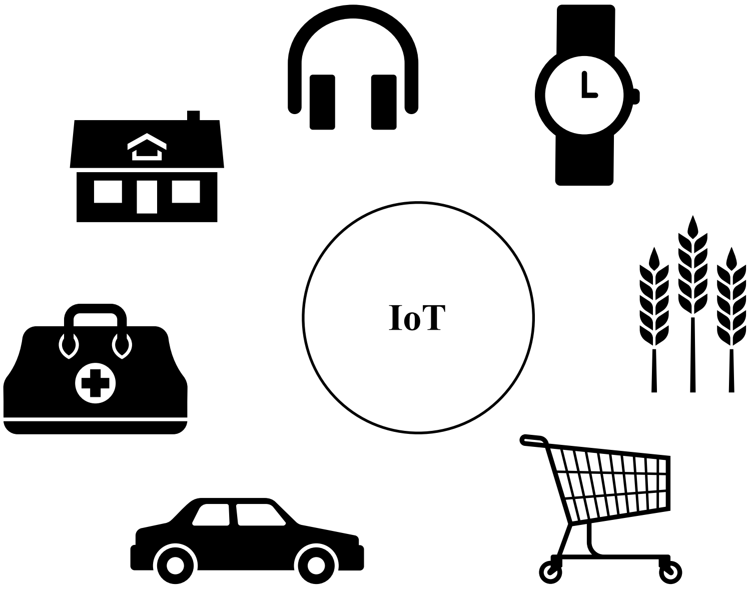


Figure 1.1 : Applications of IoT

1.1.2 Challenges in IoT

IoT is a quite new industry concept and presents them with an enormous chance to flourish in this world of digital transformation. The IoT stands for companies in many ways, but the basic concept stays the same; data collection, data analyzing, and then reengineering processes and benefit realization insights are provided. The challenges to be faced by the IoT industry are:

1. Security

The increased number of connecting devices allows security vulnerabilities and poorly designed devices to be exploited which can expose user data to theft by not providing sufficient protection for data streams and in some cases endangering the safety and health of people (implanted, Internet-enabled medical devices and hacking cars).

2. Privacy

The IoT poses unique privacy challenges which go beyond the existing data protection issues. Much of that is because we integrate devices without us knowingly using them in our environments.

3. Standards

Sometimes developers design products that are operative on the Internet without standards to guide manufacturers, without taking into consideration their impact. If they are not designed and configured properly, such devices can negatively affect their networking resources and the broader Internet.

4. Regulation

Legal issues related to IoT devices cover the transnational flow of data; conflicts between law enforcement and civil rights; policies on the retention of information and destruction; and legal liability for unauthorized use, breaches of security or privacy weaknesses.

5. Development

The wide range of IoT problems will not be exclusive to industrialized countries. The IoT has a major promise in terms of social and economic advantages for the developing and emerging economies.

1.1.3 Architecture of IoT

The architecture of IoT varies, depending on the type of solution we want to build. IoT as a technology consists mainly of four main components, which are structured around an architecture.

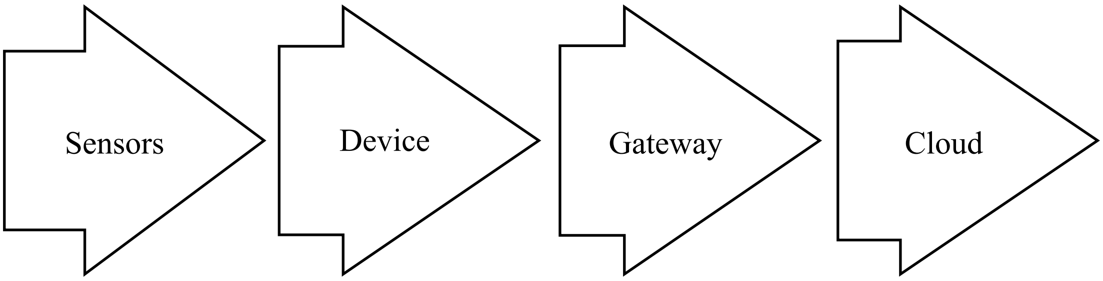


Figure 1.2 : IoT Architecture

Stage 1:

Sensors collect and transform data from the environment or object under measurement.

Stage 2:

The data generated by the sensors is collected by a device placed at a close proximity to these sensors. This data is processed/digitised for further processing

Stage 3:

It includes the network layer through which data is transferred to the cloud.

Stage 4:

Stage 3 data is transmitted to a cloud, where more powerful information technology systems can analysis, manage and save data. The final processed data is further used by various applications as needed

1.2 Smart Home

Having a home when every thing is controllable through a tap of button or through voice would be a dream for every person. With great advancement and research done in the field of IoT this is nowadays possible. Currently tech giants like Google, Amazons have products like Google Home and Amazon echo which can control home lighting and provide news, music, various other information on the command of person voice.

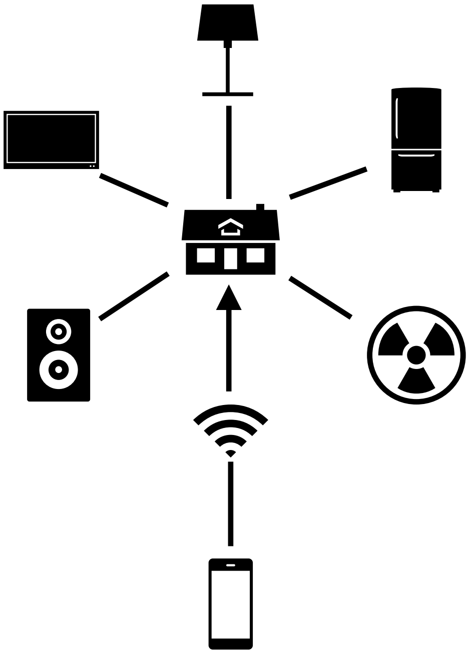


Figure 1.3 : Smart Home concept

Smart Home is an application of Internet of Things (IoT) where devices interact with other devices present and take actions on their own or these devices are controllable through smart phone, smart watch or through voice and thus making life of people easier. Some examples of smart home involve turning the lights on/off based on the people present in the room, turning on the air conditioner remotely in a hot summer weather, monitoring of the appliances and many more. Nowadays people are surrounded by world of automation where they get the work done automatically with little involvement.

A smart home concept is shown in Figure 1.3 where different appliances present in home are controllable through a smartphone. Smart Home is about well being, security and comfort at your finger-tips. It helps to deal with the regular issues which people generally face daily while working in their home. For example, we often forget to turn off our home appliances which is a concern for energy management. A smart home device allows us to control all the appliances of home through smartphone or any gadget with internet connectivity from anyplace around the globe.

1.2.1 Layered architecture of Smart Home

A layered architecture is presented for Smart Home in [5][6]. The system is divided into three layers sensing layer, network layer, and application layer (Figure 1.4). Sensing layer is consist of all the sensors and is responsible for collecting data from all appliances present in home and this data is then sent to the second layer that is network layer. Network layer depicts the internet through which data sensing layer is sent to application layer. In application layer there are different applications for different purposes which processes the data according to their needs, for example getting live feed of home directly to our smart home, getting health reports.

The concept of smart home would be a great assistance for disabled people. Disabled people completely living alone will definitely have more issues than one living with some non disabled person. These people face different issues in their everyday life. One of the issue that include in door unlocking, the person has to go physically to open the door.

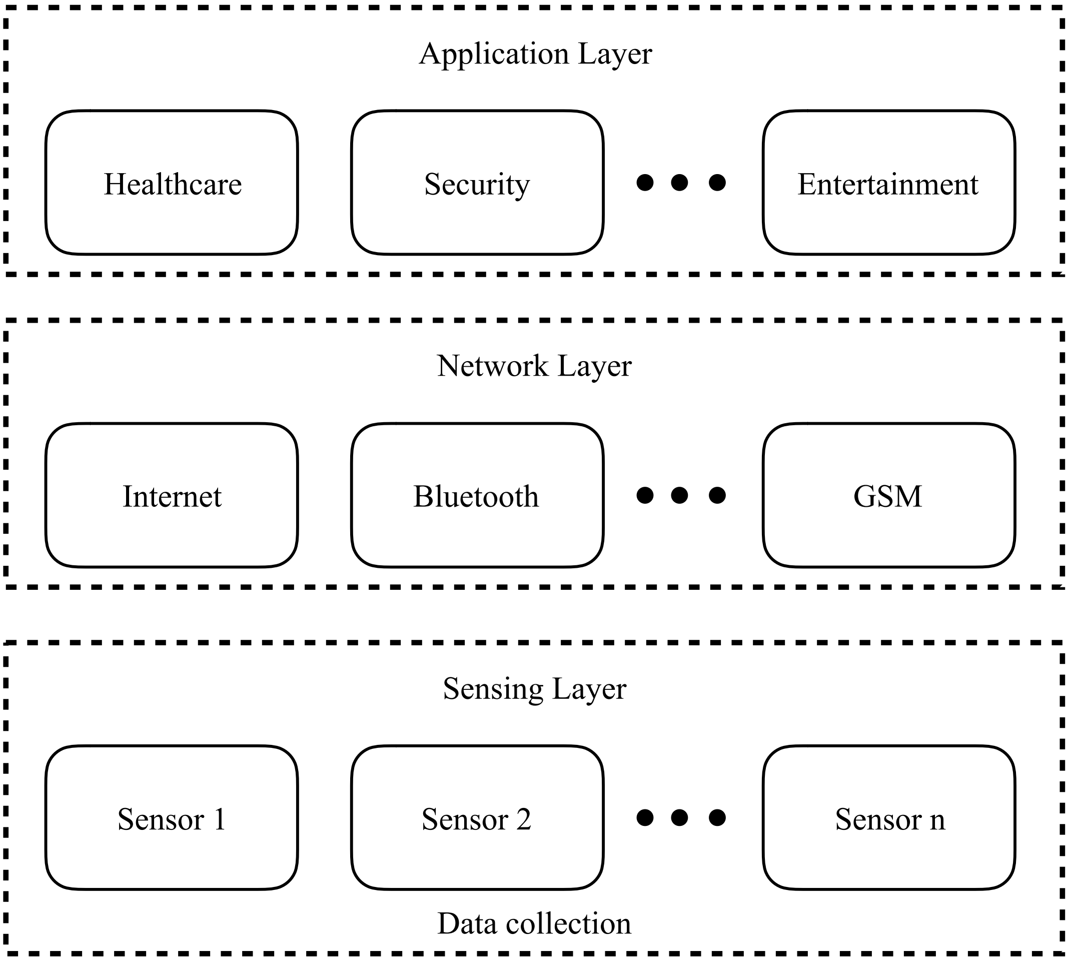


Figure 1.4 : Layered architecture of Smart Home System

The structure of the report is as follows:

Section 2 : Gives a summary of related work

Section 3 : Explains the problem statement and proposes a solution

Section 4 : Provides analysis of different algorithm useful for implementing the solution

Section 5 : Implementation of the proposed solution

Section 6 : Analysis of the result obtained

Section 7 : Provides conclusion and future scope of the work

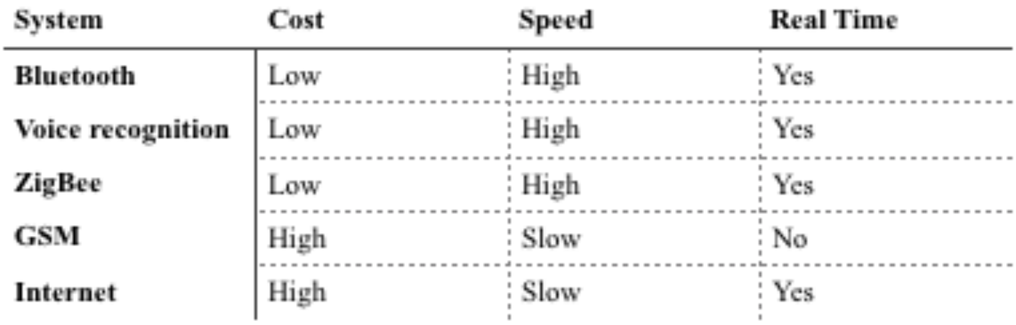
CHAPTER 2

RELATED WORK

The IoT (Internet of Things) is an integration of various technologies that enable social services to be improved utilizing smart sensors and smart objects. Smart devices can be accessed and operated at any moment and from any location via IP (Internet Protocol) connectivity[20]. Because of the large data collection and processing, data security (Availability, Integrity, Privacy) is particularly common. Attacks may be classified as passive or active[21]. While passive attacks involve data robbery or subversion of privacy, active attacks concern the destruction of data in the network.

In a smart home, home safety is one of the major categories[15]. The same study predicts that smart homes will develop in distributed smart devices, interacting through some form of wireless network with other smart devices. In intelligent home safety, the identification of visitors is essential and numerous approaches are actively studied. A smart home system can be categorised in two categories based on controlling : locally controlled and remotely controlled. A locally controlled system involves controlling appliances while staying at home though bluetooth [7][23][24] [25][26], NFC [31][32] and remotely controlled involves controlling through WiFi [8][27][28][29], GSM[30].

Various approaches have been suggested for the implementation of smart home system. In [12] , comparison is presented on various technologies such as



System	Cost	Speed	Real Time
Bluetooth	Low	High	Yes
Voice recognition	Low	High	Yes
ZigBee	Low	High	Yes
GSM	High	Slow	No
Internet	High	Slow	Yes

Table 2.1 : Comparison of different communication approaches

GSM, Bluetooth, ZigBee and Wi-fi which are used as a mean of communication between user and smart home.

Each of the technologies has its own pros and cons and depending on the application the best suitable approach can be chosen. Bluetooth proves to be the best for short range communication. Voice recognition would be a helpful mean of communication for disabled people.

A SMS based solution is proposed in [11] which uses SMS for exchanging data over a GSM network. To exchange data through SMS will incur additional cost and might be expensive also an SMS approach lacks in the area of Graphical User Interface (GUI) which makes this approach not user friendly.

A door unlocking mechanism is proposed in [9] provides with two approaches for unlocking door : speech command and pin input. It is a locally controlled door unlocking mechanism which uses bluetooth for sending and receiving data from smartphone to micro controller and vice versa. If an unknown person arrives then the user needs to always approach the door to check the identity of the user which can be handled by adding a camera to the system which captures the face and send it to the users smartphone.

A system based on face recognition [10] uses OTP to provide security to the system. In this system irrespective of the person is authorised or unauthorised the user receives a notification about the person arrival and has to unlock the door after OTP authentication. A face spoofing techniques like 3D mask can be applied to the breach the system.

Today, facial recognition systems have high accuracy rates and shown in [13] as a feasible method for secured door unlocking, but lacked automatic capability for image capture. Because visitors need to frame their head in an appropriate position close to the camera, and there would be some inconvenience to visitors who are very tall or very short. A high configured camera with a wide angle lens is possible, but would incur cost. Because of these issues, voice recognition for the visitor is used by another interesting approach which is cost effective in [14]. Environmental noise is the main problems associated with this method.

A protocol was developed to transmit visual data to the home owner for a manual video identification [16]. In the system, devices were connected to a server and that streams the video to the home owner wirelessly. The fingerprint of visitors and the RFID tag card are used for biometric identification[17]. Interestingly, it made a central server for home automation of the door phone identity system. However, manual identification as well as intrusive identification do not offer visitor and homeowner comfort and convenience. Future, smart door systems should become an automatic identification system, as well as an intrusive identification system.

Face recognition recognition proves to be a reliable and efficient approach for authentication but can be hacked with face spoofing [19][22]. The following section gives the algorithms used for achieving smart door unlocking with anti spoofing technique which is easy, efficient, secured and reliable.

CHAPTER 3

PROPOSED WORK

3.1 Problem statement

People face various problem with door unlocking like lost key, no keys and thus making a person to wait for someone else to open the door or waiting for someone to arrive with keys if door is locked, approach the door if someone has arrived. Disabled people living alone will have more issues if someone comes and they have to open the door. Therefore a smart door that operates automatically has been proposed.

3.2 Proposed Solution

A solution that is easy to use and reliable for the users needs to be developed. Face Recognition is used a the principal technology for developing such system. Face Recognition is currently one of the promising field of research because of its demand in everyday life. Technology leaders have made great advancement with high accuracy in this field. Face ID technology developed by Apple Inc. is being used for unlocking device, making payments, tracking facial expression for Animoji. Facebook uses the face recognition technology for tagging friends in photos. Google Photos allow us to search photos by the people with great accuracy. The basic flow chart of the system is shown in Figure 3.1

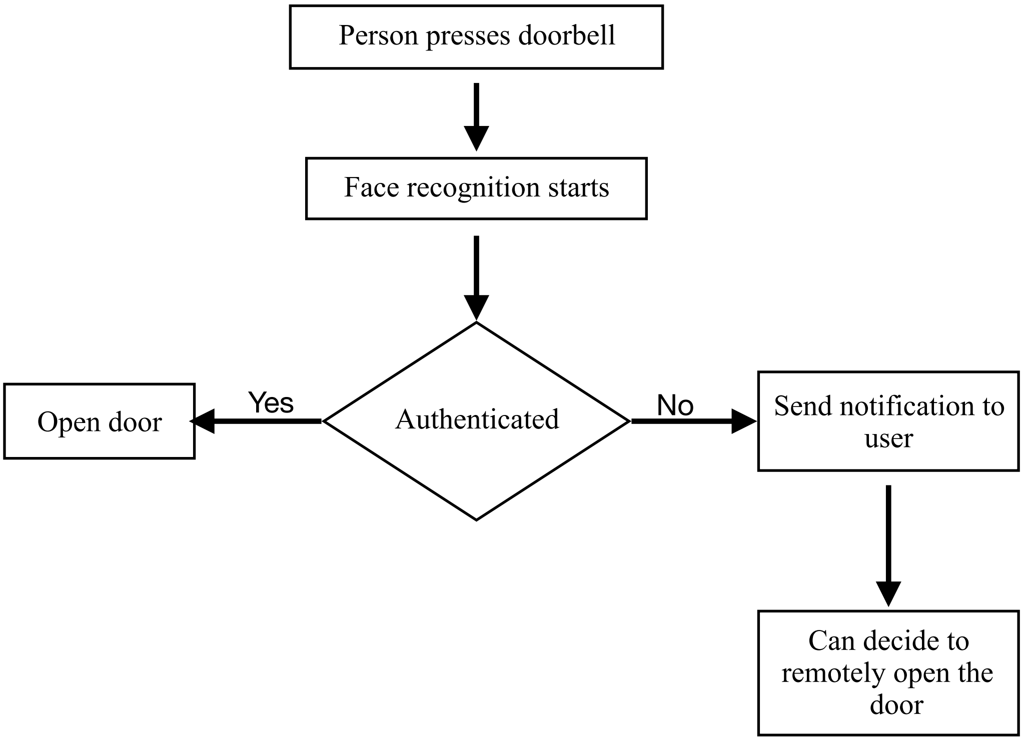


Figure 3.1 : Basic flow of the proposed algorithm

When a person presses doorbell, face recognition starts and the face is matched with the authorised database. If a person is authorised the the door will open automatically else a notification will go to a authorised person with the picture of person visited and the user can decide to open the door remotely.

The proposed system sounds easy and convenient for a user but lacks a security of face spoofing and authentication at smartphone side.

Face spoofing[19][22] is done mainly in 3 ways:

1. Photo attack : In this a picture of an authorised user (in this case say owner of house) is presented to the camera. The camera processes the photograph and gives a success result. This is the most common attack used by an intruder.

2. Video attack : If any system requires not only just face recognition but some activity to be performed the the intruder can capture a video of the user and present it to the system.

3. 3D Mask attack : A mask is worn be an intruder

To overcome this problem and detect face spoofing a solution of eye blinking is proposed. The authentication process of the system is shown in Figure 3.2. When an image is fed into the system, it checks with the existing images in the database.

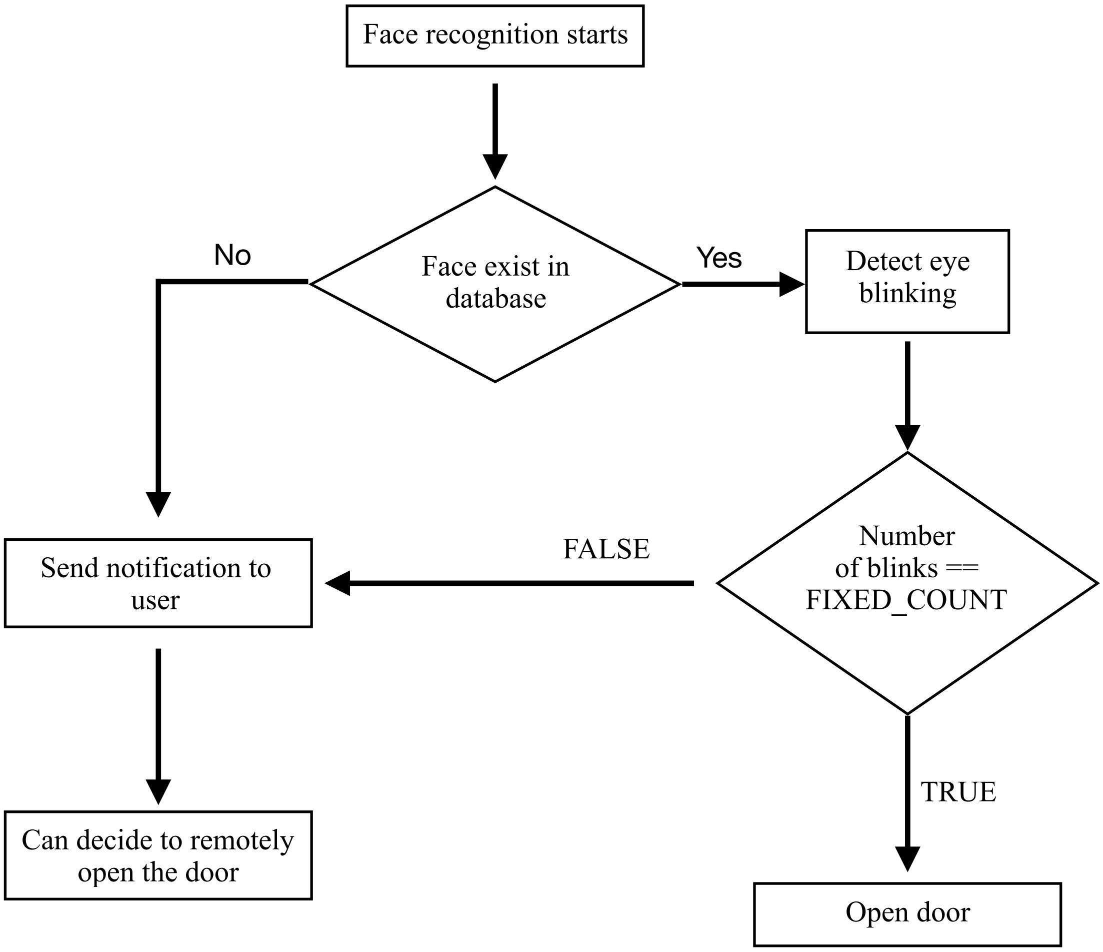

Face recognition starts
No
Face exist in database
Yes
Detect eye blinking
Send notification to user
FALSE
Number of blinks == FIXED_COUNT
TRUE
Can decide to remotely open the door
Open door

Figure 3.2 : Face spoofing detection flow

1. If the image of person doesn't exist in the database then a notification is sent to user phone with the picture of visitor and the user can decide to open the door remotely

2. If the image of person matches i.e authorised then the person need to blink his/her eye for FIXED_COUNT (say 4). If the number of blinks is exactly equal to

the FIXED_COUNT then the door will be opened else a notification would be sent to the user.

For enhanced security the value of FIXED_COUNT can be changed daily and its value can be sent to user as a notification

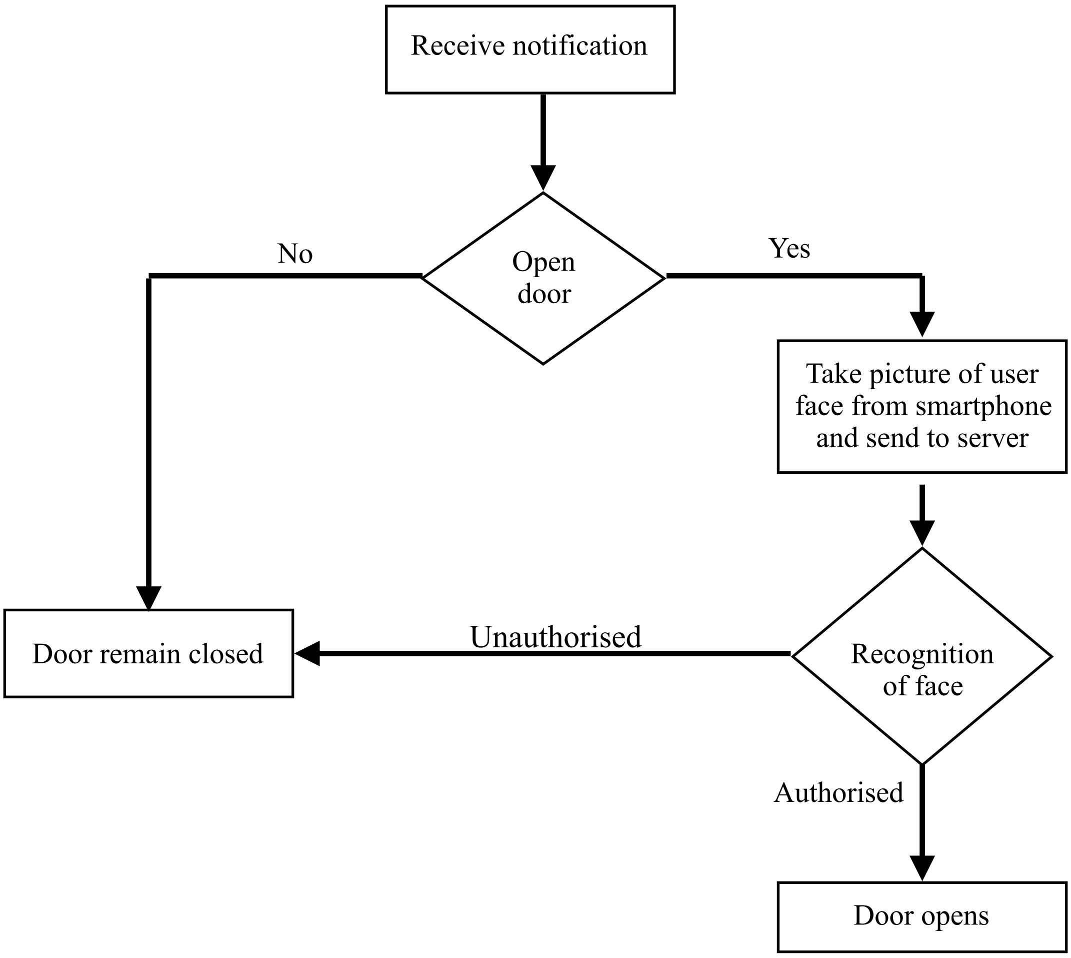


Figure 3.3 : Smartphone user authentication

With the proposed authentication system the face spoofing can easily be detected:

1. Photo attack : As person needs to blink eye, therefore this attack is not possible

2. Video/ 3D mask attack : The user need to blink his/her eyes for certain count to authorise himself/herself. A video of eye blinking can be made and provided to the system but the number of blinks needs to be exact (neither extra nor less). So

this count is rated as a PASSWORD here which is known only to the authorised users.

When a user receives notification if an unauthorised user visits then he has to decide to open or not open the door. The person with the smartphone can be someone else and not the actual authorised person who is allowed to take action regarding unlocking door. How can the user with the smartphone be authenticated. The flow for authenticating user with smartphone is given in Figure 3.3.

The full proposed system consists of one one user acting as a super user who decides to open or not open the door when a unauthorised person visits. The face of the super user is stored in the server and when the person selects to open the door then application asks the user to capture his photo and then this photo is sent to the server where the face recognition takes place. If the authentication is successful then the door gets unlocked else the user receives an error message.

There can be multiple other ways of authentication like PIN, fingerprint/face unlocking in the smartphone. The PIN can be forgotten or if someone gets access to the pin then he can unlock the door, the fingerprint/face stored in the smartphone can be of multiple people, therefore anyone can unlock the door. The proposed system for authentication at smartphone proves to be secured and reliable.

3.3 Summary of proposed solution

The proposed system sounds to be secured, smart and reliable, this section provides step wise process for the system.

I. Visitor arrives

II. Face recognition system starts

III. If person face exist in database then goto step IV else goto step VII

IV. Eye blink detection system starts

V. If number of blinks equals pre stored count then sends notification and goto step VI else goto step VII. If pre stored count is say 4 and the rate of blinking is 1 blink/sec then number of blinks detected at the end of 4 seconds is checked with pre stored count

VI. Opens the door and END

VII. User receives notification and if user selects Yes then goto step VIII else END

VIII. User captures his face and send to server. If the server response is successful the door opens

The super user receives notification in both cases i.e authorised and unauthorised. So for any reason if an intruder gets access then the user will receive notification and can take appropriate measures. But this case would not occur as the count of blinks which acts as a PASSWORD is with the super user and other authorised people only.

The problem statement stated earlier at the beginning of the chapter like lost key, locate for keys to open the door, making a person to wait for someone else to open the door or waiting for someone to arrive with keys if door is locked if the person owner/authorised, approach the door if someone has arrived. Disabled people will have more problem with the issue addressed latter. All the problems addressed are easily solved by the solution proposed.

CHAPTER 4

WORKING AND ANALYSIS

The two main concepts int the proposed solution are Face recognition and Eye blink Detection. To develop an efficient solution efficient and reliable algorithm needs to be used.

4.1 Face Recognition

The face recognition involves a three step process for detecting a person:

1. Face detection
2. Face matching : Matching the detected face with the faces available in the database

Before these steps a pre training step is involved in which the database is created with the faces of authorised person at different angles, and this database is used to match with the input face.

With great advancement in Machine Learning, lot of open source libraries are available for research, developing and testing such as Tensorflow, Keras, Sciki-learn and many more. The prototype for the proposed solution is built with the help of Python, OpenCV and dlib. These libraries provide some pre trained models for face recognition with accuracy as high upto 99%.

The following section gives analysis and comparison of different algorithm available in OpenCV for face recognition

4.1.1 Face Detection Algorithm

OpenCV provides three different algorithms for face detection and they are HAAR Cascade, LBP Cascade and DNN. The working is analysed by using LFW and FEI face dataset based on orientation, illumination and some random pictures. But before going to the analysis of these algorithms let us study the basic details behind the working of these algorithms.

Haar Cascade Classifier

It is a machine learning algorithm for object detection created by Paul Viola and Michael Jones [18] using cascade of simple features. The algorithm requires lots of images which are classified as positive and negative images for training. Then the features are extracted using Haar features (Figure 4.1). The Haar cascade algorithm is applied on black and white images.

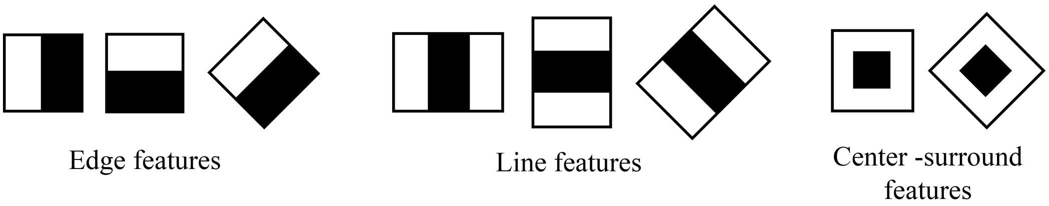


Figure 4.1 : Different Haar features

A single feature is calculated by subtracting the sum of white pixels from sum of black pixels. These windows shown in Figure 4.1 is applied over the image in all possible location and sizes and hence providing lot of features and most of them will be irrelevant. For example (Figure 4.2), the top row presents two features - first focuses on the fact that eyes are darker than nose and cheeks and second feature focuses on the face eyes are darker than the nose. The best features is obtained using Adaboost. After obtaining the best features from Adaboost, now rather than applying

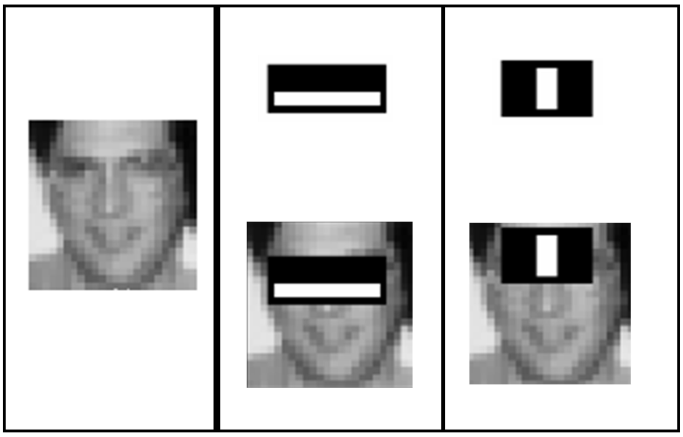


Figure 4.2 : Example of Haar

all the feature say 600 features to an image and classify it as face or non-face, it groups the features into different stages of classifier hence the name cascade classifier. If at any stage a window fails then it doesn't check for remaining features.

LBP Cascade Classifier

Local Binary Pattern classifier extracts features from hundreds of training images which are classified as positive and negative for forming a feature vector which is used to classify face or non-face. These features are calculated using a 3X3 window i.e 9 pixels which is traversed through the input image having the window centre pixel coinciding with each pixel of the input image. During each iteration the centre pixel value is compared with every neighbouring pixel and for each neighbour pixel that is greater or equal to the centre pixel the value is set to 1 else to 0. Then the value is read clockwise forming a binary number and this binary number is then

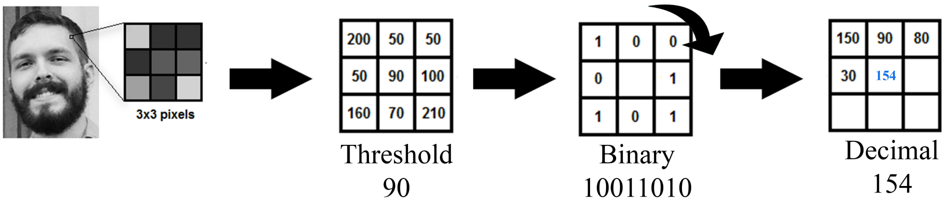


Figure 4.3 : LBP Feature calculation I

converted to decimal number which is the new value for the centre pixel. With this all the pixels will be updated with a new value. (Figure 4.3)

Then image is segmented into regions and for each region their histogram is calculated. Finally all the histograms are concatenated to from one feature vector of the image. (Figure 4.4)

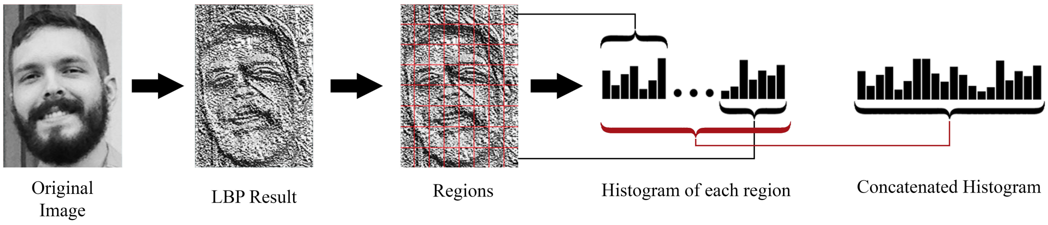


Figure 4.4 : LBP Feature calculation II

Deep Neural Network(DNN)

Deep Neural Network is a deep learning which have multiple layers between the input and output layer of artificial neural network. A simple neural network consist of one input layer, one output layer and almost one hidden layer (Figure 4.5). A network comprising of multiple hidden layers is known as a deep neural network (Figure 4.6). The number of input layer would be equal to number of pixels of the input image and each input is fed into every node of first hidden layer. The output of one layer acts as the input of next layer. The output through each node is calculated using the formula:

$$output_j = \Phi(w_o + \sum w_i * x_i) \tag{4.1}$$

where:

w_0 = bias

j = i+1 i.e output of layer j depends on previous layer output

ø = activation function (Decides whether a node should be activated or not).

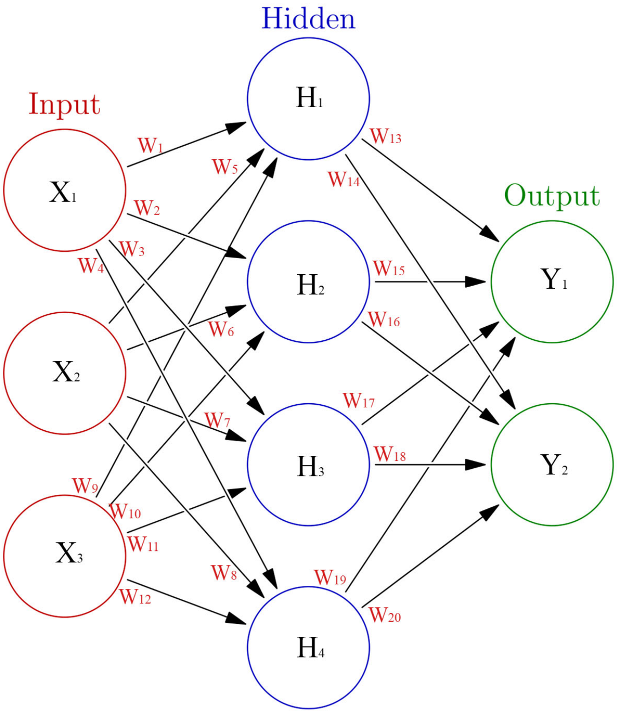


Figure 4.5 : Simple Neural Network

The number of nodes in output layer for detecting faces would be 1 and its output will be either TRUE (1) or FALSE (0). It requires a large set of labelled data with positive and negative images. Initially random weights are assigned to the full network and after the first iteration occurs from input to output the output is compared with the required output and then the process is run backwards modifying the weights using Back propagation algorithm. This process continues with all the thousands of images and in the end we get a weight matrix which is used on a testing image to get the output.

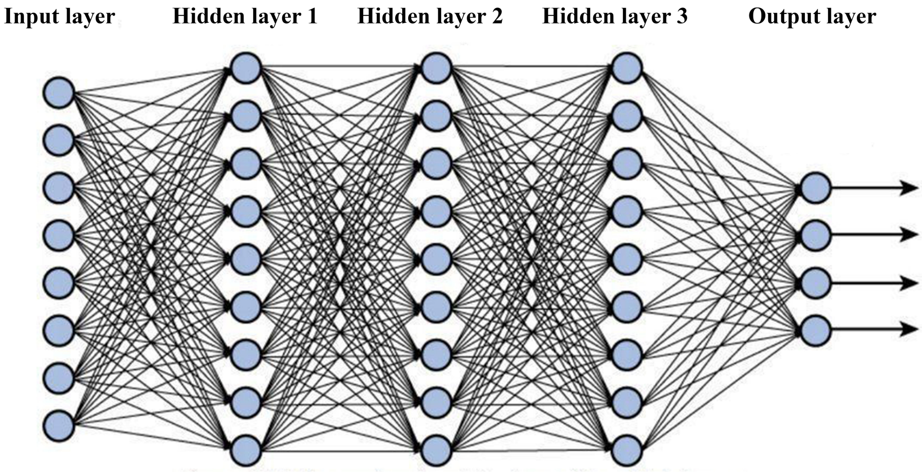


Figure 4.6 : Deep Neural Network

Analysis done on various angles of face starting from +90 to -90 is shown in Table 4.1 The result shows the DNN method gives the best result detecting face at every angle, then is the HAAR not able to detect face at 90 degree angles and last is the LBP only detecting face at within range of -45 to +45

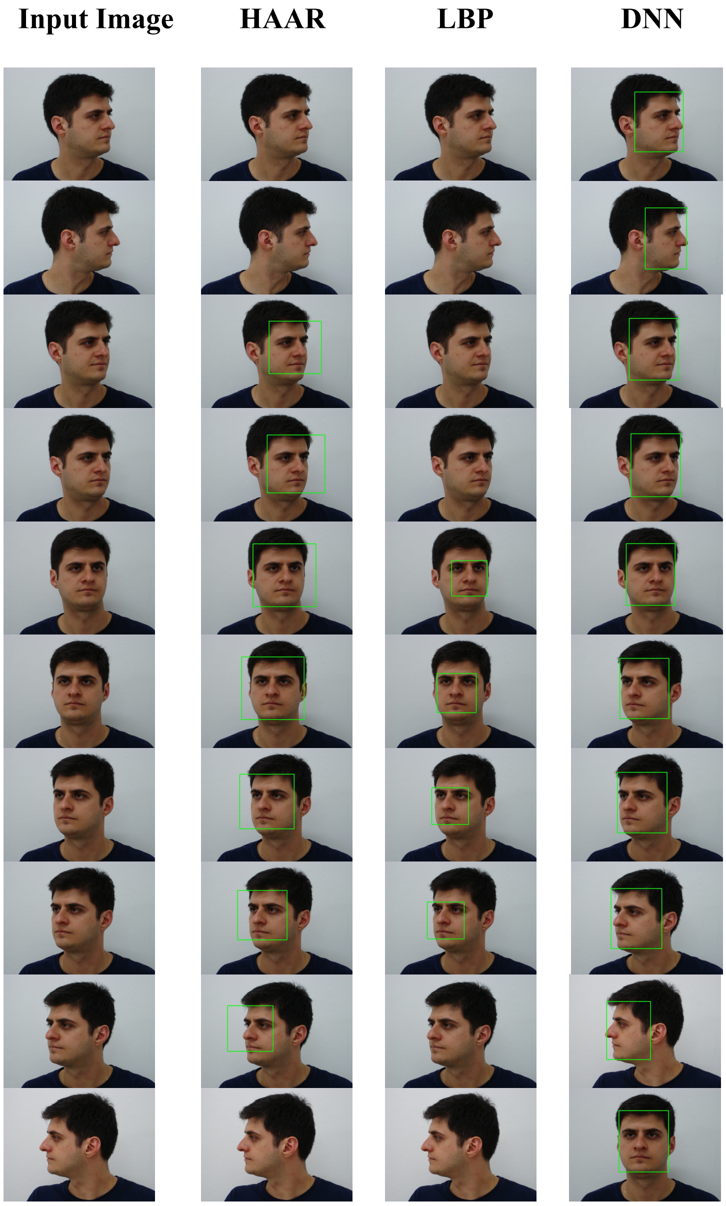


Table 4.1 : Working of algorithms on different orientation

Table 4.2 shows the analysis after running the algorithm on different lighting conditions. The result shows that LBP performs better among all three, woking on even low lighting condition whereas HAAR and DNN perform equally.

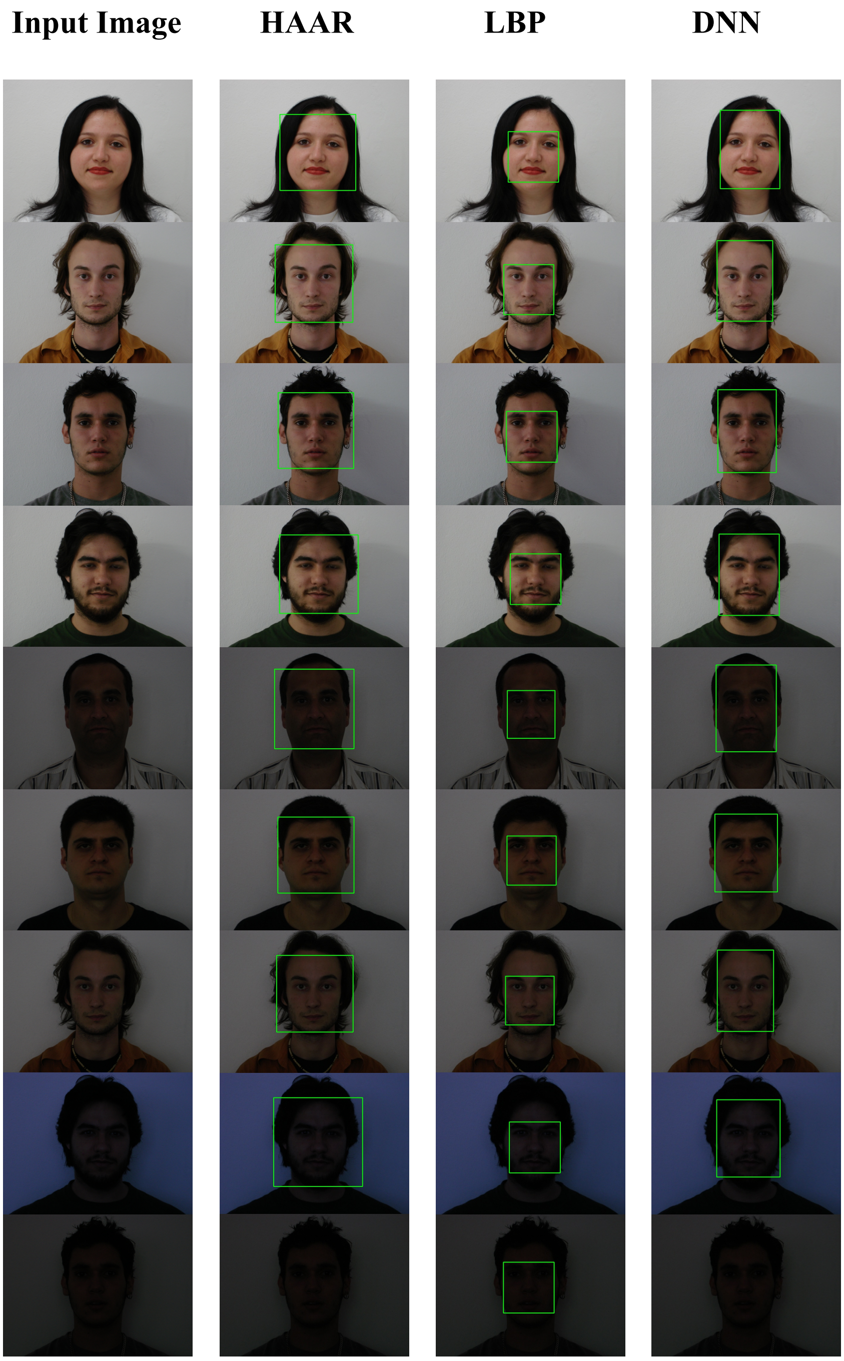


Table 4.2 : Working of algorithms on different lighting condition

Table 4.3 shows the result of the algorithm on random pictures of people. The highest accuracy is shown by DNN then is HAAR and last is LBP. DNN is able to accurately detect multiple faces in an image if it has, which is not seen in the case of HAAR and LBP. Although in some cases DNN gave false positive result. DNN provides a confidence factor which tell how confident it is that the region detected is a face. The confidence factor is in range of [0.0 , 1.0]. Figure 4.7 shows the working of DNN with different confidence factors. The result shows that confidence factor and false positive are inversely proportional. The result of DNN algorithm in Table 4.3 was obtained with confidence factor > 0.8.

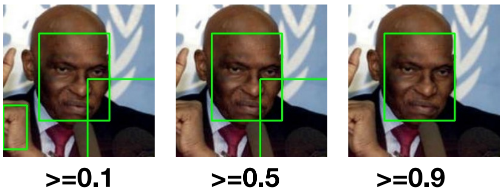


Figure 4.7 : Analysis of DNN with different confidence factor

After combining all the result it can be seen that the most accurate result is shown by DNN by detecting face in every angle and in low lighting condition. It is also able to detect multiple faces accurately. With confidence factor set to >=0.9 it is able to work well with least false negative result.

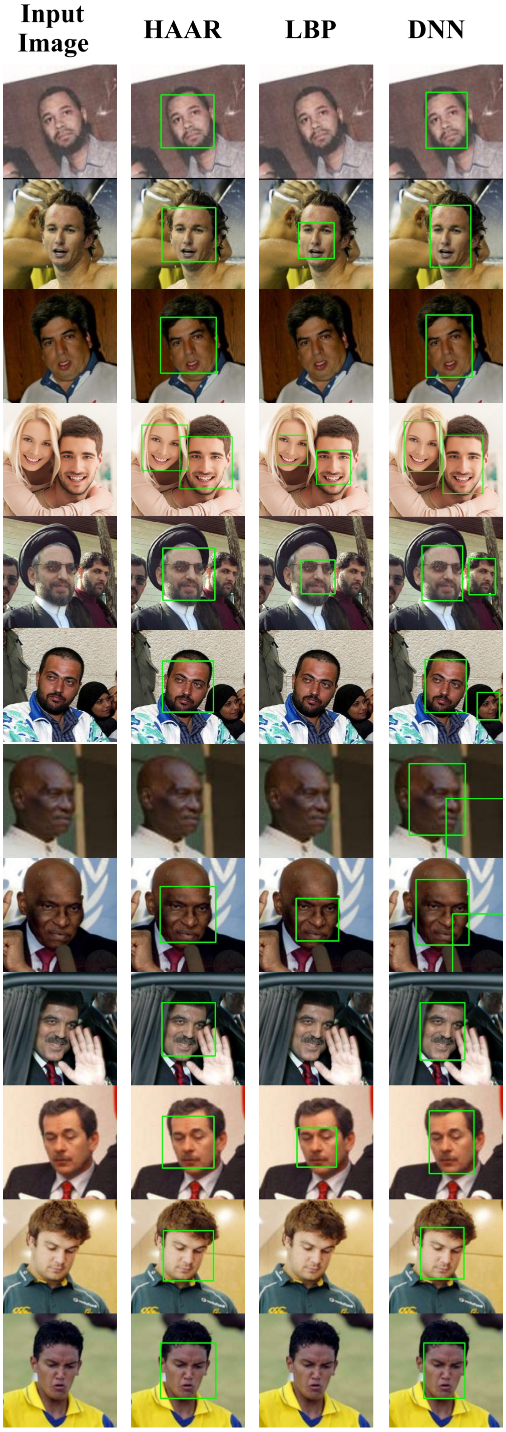


Table 4.3 : Working of algorithms on random photographs

4.1.2 Face Matching Algorithm

The working of face matchings is analysed with LFW dataset, taking 10 images of each user for training purpose. The system is trained with 4 user with 10 images each i.e. total of 40 images and for testing mixture of positive and negative test images are taken. OpenCV provides a confidence factor and a label (predicted face) for all the three algorithms of face matching. The confidence factor is the distance to the label. A confidence factor of 0 is a perfect match i.e accuracy is inversely proportional to confidence factor.

Figure 4.8 shows the people with which the system is trained. The training dataset include 10 different faces of same person. After training the system and verifying the accuracy of the trained system, 2 different images of each person is taken and along with it some negative images are also taken. The result of recognition step is shown in Table 4.4. OpenCV returns two values after prediction - person to which it has closely matched and the confidence value i.e. how confident it is to the matched person. The result shows that the working of LBPH works better than that of other two. As OpenCV provides prediction for every cases, therefore a threshold needs to be set to detect positive and negative results. Looking at the result a threshold of 80 i.e. confidence value below 80 will be considered as positive and rest as negative would provide a better system.

Figure 4.8 : People for training the system

Although LBPH provided better result but it has mix of confidence values making it an unreliable method for prediction. Since with the progress in Machine Learning, Neural Network have proved in providing better result for carrying out face matching task. The same data set is analysed by using dlib library which provides face matching using NN and the result it shown is 100% accuracy for the given set of test images in Figure 4.8. The dlib library provides one shot learning for face matching i.e for making a system to learn a person face only one picture is needed. The model is pre trained with million of pictures and provides weight of the model after training on the data set. Rather than training our own model and generating weights this model gives a vector of 128 real valued numbers for the input face (Figure 4.9). So we need to store this vector of 128 real valued numbers and for each test image we generate a vector and compare with the stored vector.

This one shot learning concept is used in the implementation of the prototype of face training and face matching.

4.2 Face Training

Face training i.e add a person face to a list of authorised person, is the first step of this prototype. In this step image of person is fed to the dlib library one after the other and the vector generated by dlib is stored as a JSON file in the following format (Figure 4.10):

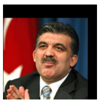
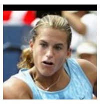
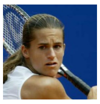
Input Image	LBPH Face	Fisher Face	Eigen Face
	Correct 89.03	Wrong 920.45	Wrong 3494.72
	Correct 80.71	Correct 58.66	Correct 3064.34
	Correct 77.30	Correct 563.29	Correct 2914.38
	Correct 80.59	Correct 427.05	Correct 2899.34
	Correct 82.10	Correct 227.85	Correct 3205.65
	Correct 78.57	Correct 467.84	Correct 2850.90
	Correct 96.02	Correct 167.24	Correct 3926.15
	Correct 0.0	Correct 0.0	Correct 0.0
	Wrong 91.52	Wrong 874.40	Wrong 3824.51
	Wrong 88.44	Wrong 530.79	Wrong 5268.31

Table 4.4. Face matching algorithms comparison

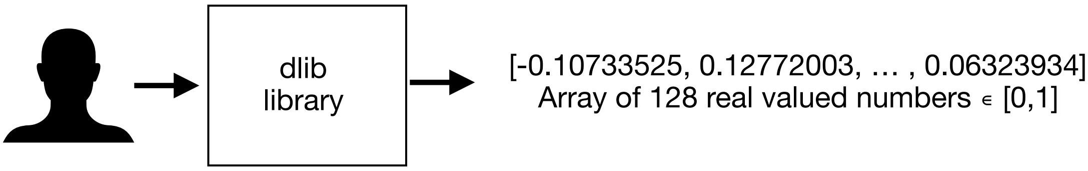


Figure 4.9 : One short learning concept

```
{
        “encodings” : [ [ encoding 1 ], [ encoding 2 ], … , [ encoding n ] ],
        “names” : [ [ name 1 ], [ name 2 ], … , [ name n ] ]
}
```

Here encoding 1 is the vector of 128 real valued numbers of person 1 and name 1 is the name of person 1. Whenever a new face needs to be trained (added to the list of authorised users), the image needs to be passed to dlib library and append the encoding and name in the existing JSON.

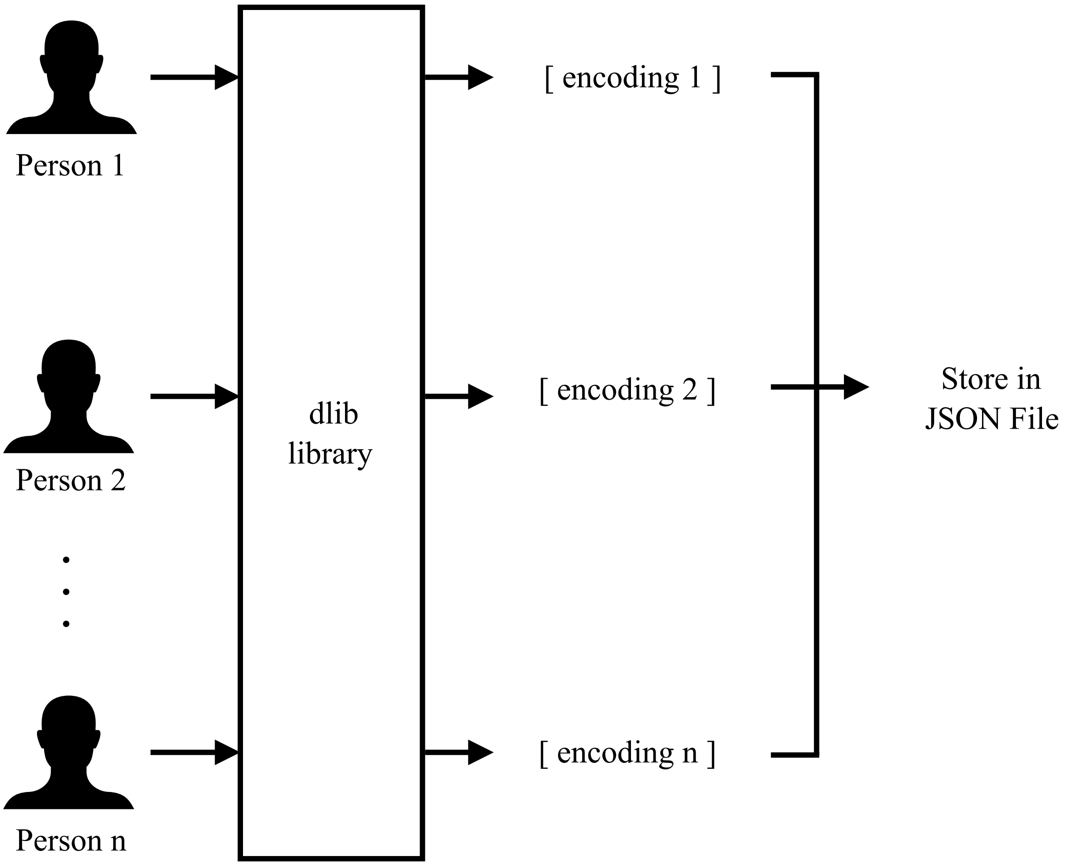


Figure 4.10 : Working of face training

4.3 Face Matching

Once the face training step is completed and the JSON file is created, the next step is to use the file to match the encoding with the person who arrives at the door. The flow of face matching includes calculation of encoding when a person arrives at the door, then this encoding is matched with all the encodings present in the JSON file that is created during face training process. The encoding for a single person will be different every time due to the fact that a very slight difference (say orientation) in face will change the full encoding. A simple matching the elements of vector would not be possible for getting the result, therefore we need to calculate how closely two vectors (one input and the other from JSON file) are related to each other or the

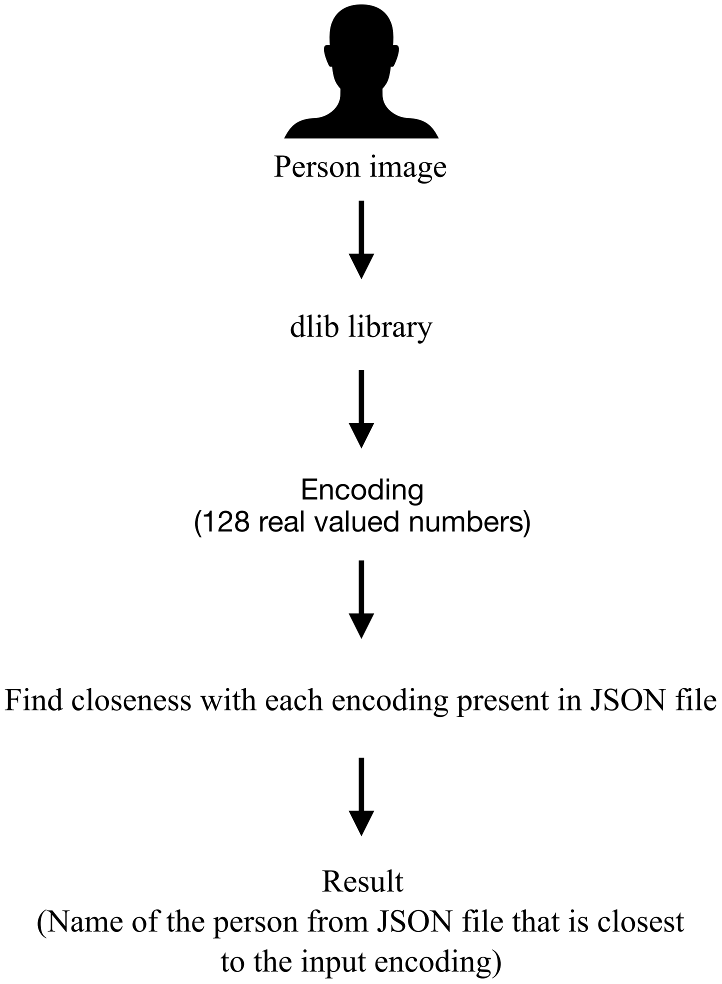

Person image
dlib library
Encoding
(128 real valued numbers)
Find closeness with each encoding present in JSON file
Result
(Name of the person from JSON file that is closest
to the input encoding)

Figure 4.11 : Face matching process using dlib

distance between the two vectors should be minimal. The flow of face matching is shown in Figure 4.11.

The calculation of distance between the two vectors is done by calculating the Euclidean norm which is given by

$$||v1 - v2|| = \sqrt{\sum_{1}^{128}(v1 - v2)^2} \tag{4.2}$$

This norm would return the distance between the two vectors (one is the vector of the input image and the other is from the stored JSON). The norm is calculated for each of the encoding stored and the distance with value <= 0.5 is returned.

4.4 Eye blink detection

Detection of eye blinking is done by calculating Eye Aspect Ratio (EAR) [1]. Though there are some conventional image processing methods for eye blinking detection like template matching for every frame to detect state of eye as opened or closed [2], [3] provides a method of intensity vertical projection which calculates the intensity of every row if a image, [4] proposes a model for tracking eyelid for blink detection, the EAR method proves to be fast, simple and straightforward ratio calculation of distances between facial indicators for the eye.

$$EAR = \frac{||p2 - p6|| + ||p3 - p5||}{2\,||p1 - p4||} \tag{4.3}$$

Figure 4.12 shows different facial indicators for the eye which are used for calculating EAR. The value of EAR approaches zero as eye closes. The blinking is done synchronously by both eyes, therefore average is taken of both eyes EAR.

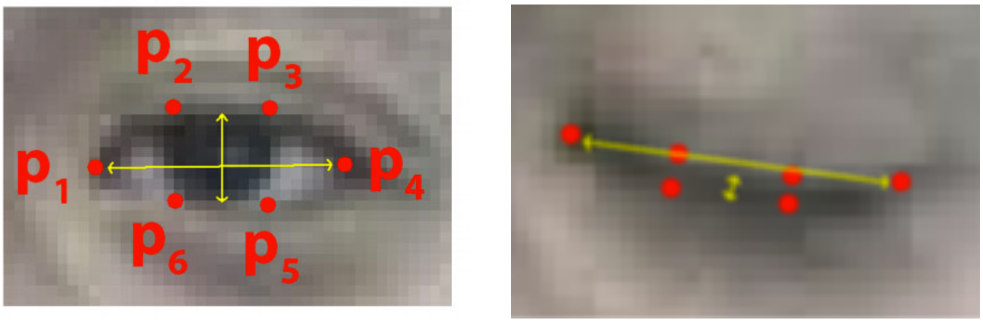


Figure 4.12 : Facial indicators linked with the eye [1]

To get these indicators i.e points on the eye dlibs facial landmark detector is used. The dlibs facial landmark provides with the six landmark points as shown in Figure 4.12. On the basis of these points EAR is calculated. When eye closes the EAR value approaches to zero. Therefore, if value is zero for some consecutive frames of images then the eye blink has occurred.

The working of blink detection on the basis of EAR works well but if a person wears spectacles then it gives false result. Therefore, a more accurate eye blink detection will be a part of future scope for this project.

The two main concept behind the proposed system are face recognition and eye blink detection. This section provided the analysis of various algorithms and came to a conclusion of an algorithm which would give best result and in the following section working and what all technologies used to implement these algorithms is described.

CHAPTER 5

IMPLEMENTATION

The implementation of the proposed work is done with the help of Python, OpenCV, dlib, Android IDE, Firebase Cloud Messaging, Django Rest Framework. Python, OpenCV and lib are used for computer vision tasks like face recognition and eye blink detection. Android IDE is used for developing the application used by the super user for receiving notification and taking actions. The notification on the smart phone are received through Firebase Cloud Messaging and Django Rest Framework is used to create RESTful APIs for interaction between client(Android, Door with face recognition system installed) and server (Figure 5.1).

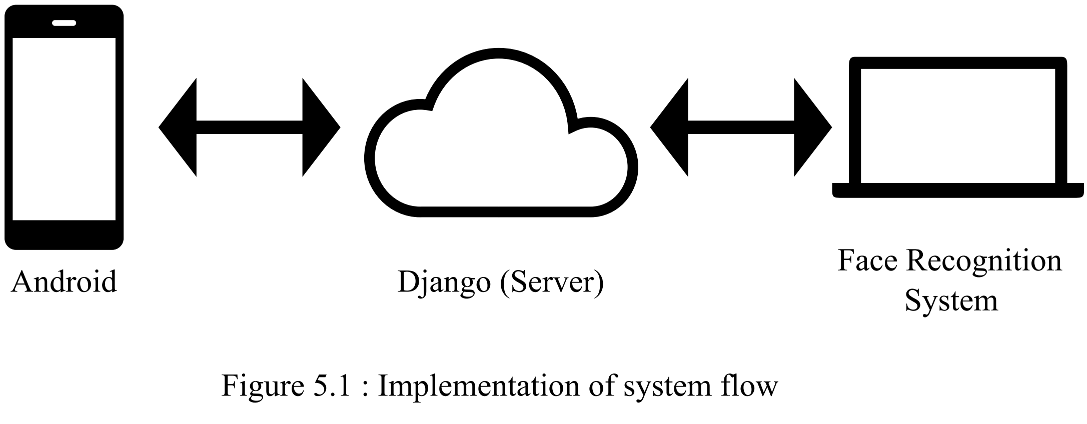


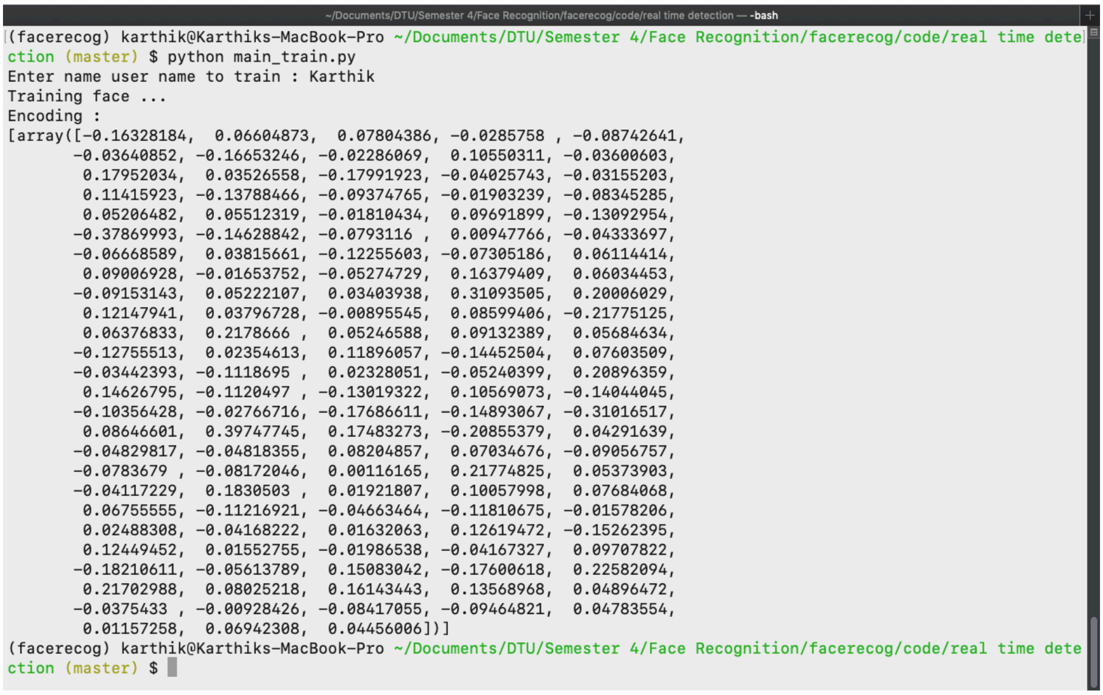
Figure 5.1 : Implementation of system flow

5.1 Face Recognition System

This system is installed at the door and it runs when a user presses the door bell. The system is built using Python, OpenCV and dlib.

5.1.1 Face Training

This is the first step of building the database of authenticated users. The system asks user to enter name. After entering the name it captures the person image through the camera and stores the name and face encoding in a file. For each new authenticated user the file is appended with the new values. The working is shown in Figure 5.2 where after training a face it gives a vector of 128 real valued numbers which is stored along with name in a JSON file.



~/Documents/DTU/Semester 4/Face Recognition/facerecog/code/real time detection — -bash

```
(facerecog) karthik@Karthiks-MacBook-Pro ~/Documents/DTU/Semester 4/Face Recognition/facerecog/code/real time dete
ction (master) $ python main_train.py
Enter name user name to train : Karthik
Training face ...
Encoding :
[array([-0.16328184,  0.06604873,  0.07804386, -0.0285758 , -0.08742641,
       -0.03640852, -0.16653246, -0.02286069,  0.10550311, -0.03600603,
        0.17952034,  0.03526558, -0.17991923, -0.04025743, -0.03155203,
        0.11415923, -0.13788466, -0.09374765, -0.01903239, -0.08345285,
        0.05206482,  0.05512319, -0.01810434,  0.09691899, -0.13092954,
       -0.37869993, -0.14628842, -0.0793116 ,  0.00947766, -0.04333697,
       -0.06668589,  0.03815661, -0.12255603, -0.07305186,  0.06114414,
        0.09006928, -0.01653752, -0.05274729,  0.16379409,  0.06034453,
       -0.09153143,  0.05222107,  0.03403938,  0.31093505,  0.20006029,
        0.12147941,  0.03796728, -0.00895545,  0.08599406, -0.21775125,
        0.06376833,  0.2178666 ,  0.05246588,  0.09132389,  0.05684634,
       -0.12755513,  0.02354613,  0.11896057, -0.14452504,  0.07603509,
       -0.03442393, -0.1118695 ,  0.02328051, -0.05240399,  0.20896359,
        0.14626795, -0.1120497 , -0.13019322,  0.10569073, -0.14044045,
       -0.10356428, -0.02766716, -0.17686611, -0.14893067, -0.31016517,
        0.08646601,  0.39747745,  0.17483273, -0.20855379,  0.04291639,
       -0.04829817, -0.04818355,  0.08204857,  0.07034676, -0.09056757,
       -0.0783679 , -0.08172046,  0.00116165,  0.21774825,  0.05373903,
       -0.04117229,  0.1830503 ,  0.01921807,  0.10057998,  0.07684068,
        0.06755555, -0.11216921, -0.04663464, -0.11810675, -0.01578206,
        0.02488308, -0.04168222,  0.01632063,  0.12619472, -0.15262395,
        0.12449452,  0.01552755, -0.01986538, -0.04167327,  0.09707822,
       -0.18210611, -0.05613789,  0.15083042, -0.17600618,  0.22582094,
        0.21702988,  0.08025218,  0.16143443,  0.13568968,  0.04896472,
       -0.0375433 , -0.00928426, -0.08417055, -0.09464821,  0.04783554,
        0.01157258,  0.06942308,  0.04456006])]
(facerecog) karthik@Karthiks-MacBook-Pro ~/Documents/DTU/Semester 4/Face Recognition/facerecog/code/real time dete
ction (master) $
```

Figure 5.2 : Face training working

5.1.2 Face Matching and Eye blink detection

Face recognition is implemented using Python, OpenCV and dlib library. The accuracy shown by the lib library is 99%. Dlib library provides with 68 points representing the face landmarks (Figure 5.3) for the face which is used to get the six points highlighted within green box which is used to calculate EAR. The EAR value is calculated for each eye and then the average of both eye is taken for processing and when the EAR is less than threshold of 0.2, then a blink has occurred.

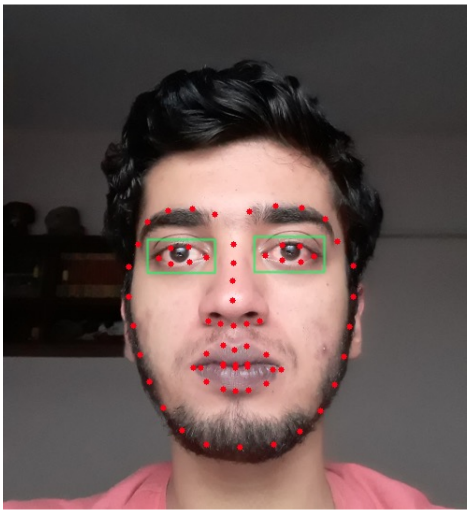
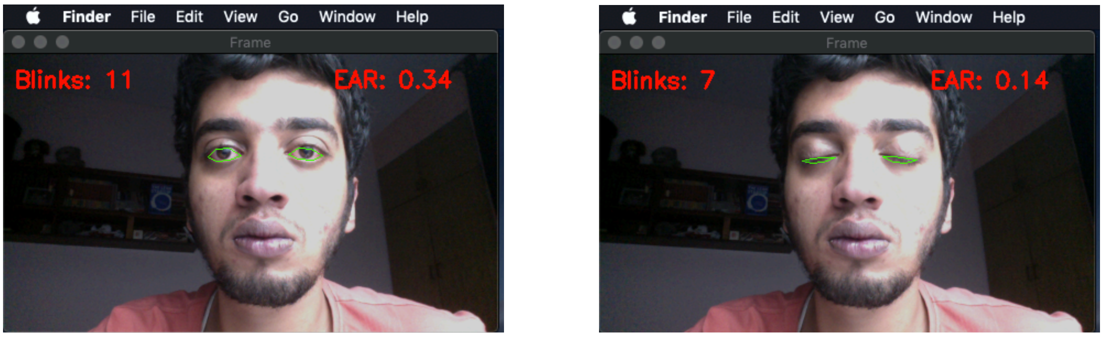


Figure 5.3 : Facial landmarks

The implementation of blink detection based on EAR is shown in Figure 5.4 where it can be seen the EAR value near to zero for closed state and non zero for open state.


Finder File Edit View Go Window Help
Frame
Finder File Edit View Go Window Help
Frame

Blinks: 11 EAR: 0.34

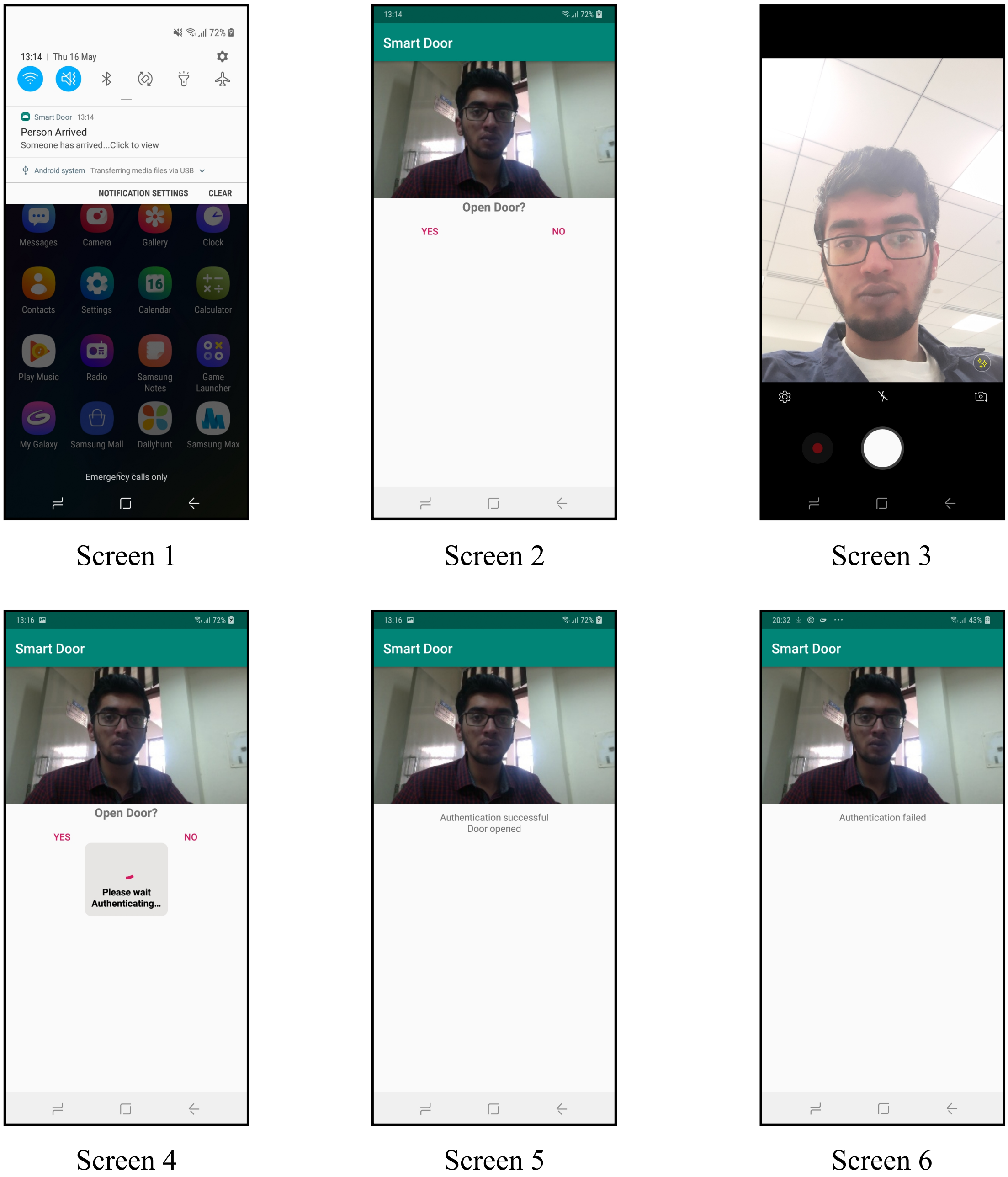
Blinks: 7 EAR: 0.14

Figure 5.4 : Implementation of blink detection based on EAR

Through combining face recognition and blink detection an authentic and reliable system is developed for detecting people. When a user is not recognised then notification goes to user and the actions taken ahead is explained in below section.

5.2 Android

The android application is with the super user and he decides to take action accordingly whenever a notification arrives. The flow of this process is shown in Figure 3. Android implementation for the process described is shown in Figure 5.5.



Smart Door 13:14

Person Arrived

Someone has arrived...Click to view

Android system Transferring media files via USB

Screen 1

Open Door?

Screen 2

Screen 3

Open Door?

Please wait
Authenticating...

Screen 4

Authentication successful
Door opened

Screen 5

Authentication failed

Screen 6

Figure 5.5 : Implementation of authentication for android user

Screen 1 - User receives notification

Screen 2 - On clicking the notification the user is shown with the picture of visitor

Screen 3 - If user selects **Yes** then he has to capture his photo

Screen 4 - The photo is sent to server which authenticates the user

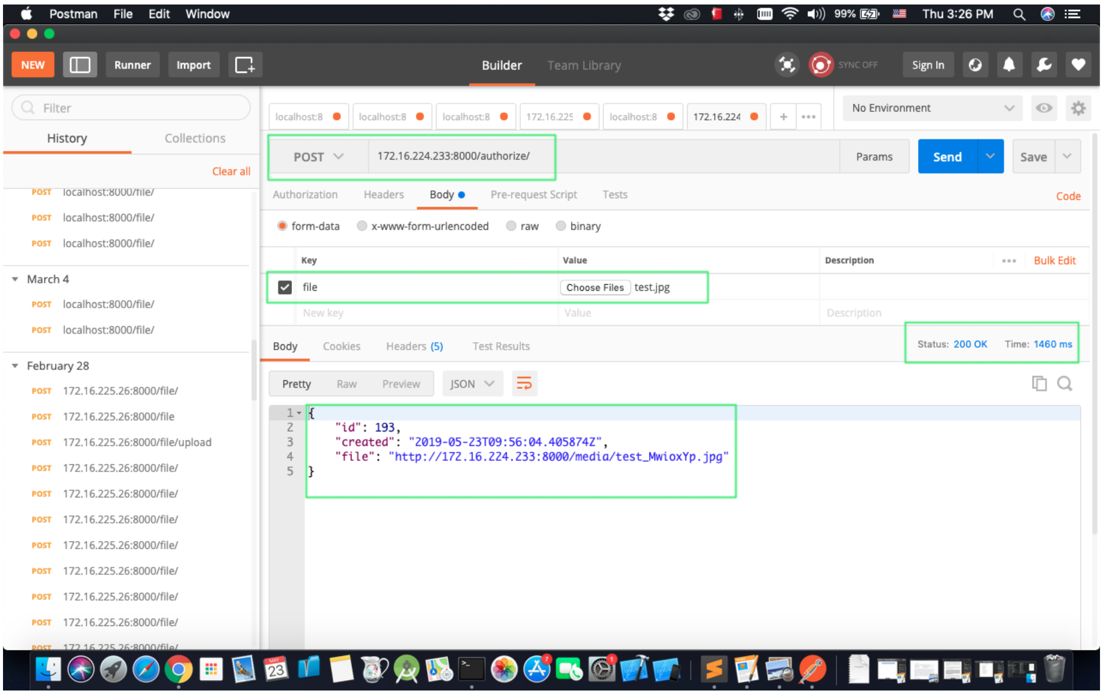
Screen 5 - On success it provides a message saying ‘Authentication successful Door opened’

Screen 6 - If authentication isn’t successful it gives message ‘Authentication failed’

5.3 Server

The server act as middle layer for providing remote access and transfer of data through Internet between Face recognition system and Android smartphone. The server is implemented using Django which is a web framework written in Python. To interact with the server REST APIs are created with help of Django Rest Framework(DRF).

The working of REST API is shown in Figure 5.6 and 5.7. The API shown in the example is http://172.16.224.233:8000/authorise/ which is used to authorise the user with the smartphone. It is a HTTP POST request that accepts the image file int the request parameter and returns a success response (200 OK) in case of authorised user along with JSON data (Figure 5.6) and if the user is unauthorised then it send an error response code (400 Bad Request) and empty JSON data (Figure 5.7)



POST localhost:8000/file/
POST localhost:8000/file/
POST localhost:8000/file/

March 4
POST localhost:8000/file/
POST localhost:8000/file/

February 28
POST 172.16.225.26:8000/file/
POST 172.16.225.26:8000/file
POST 172.16.225.26:8000/file/upload
POST 172.16.225.26:8000/file/
POST 172.16.225.26:8000/file/
POST 172.16.225.26:8000/file/
POST 172.16.225.26:8000/file/
POST 172.16.225.26:8000/file/
POST 172.16.225.26:8000/file/
POST 172.16.225.26:8000/file/
POST 172.16.225.26:8000/file/

POST 172.16.224.233:8000/authorize/

● form-data ○ x-www-form-urlencoded ○ raw ○ binary

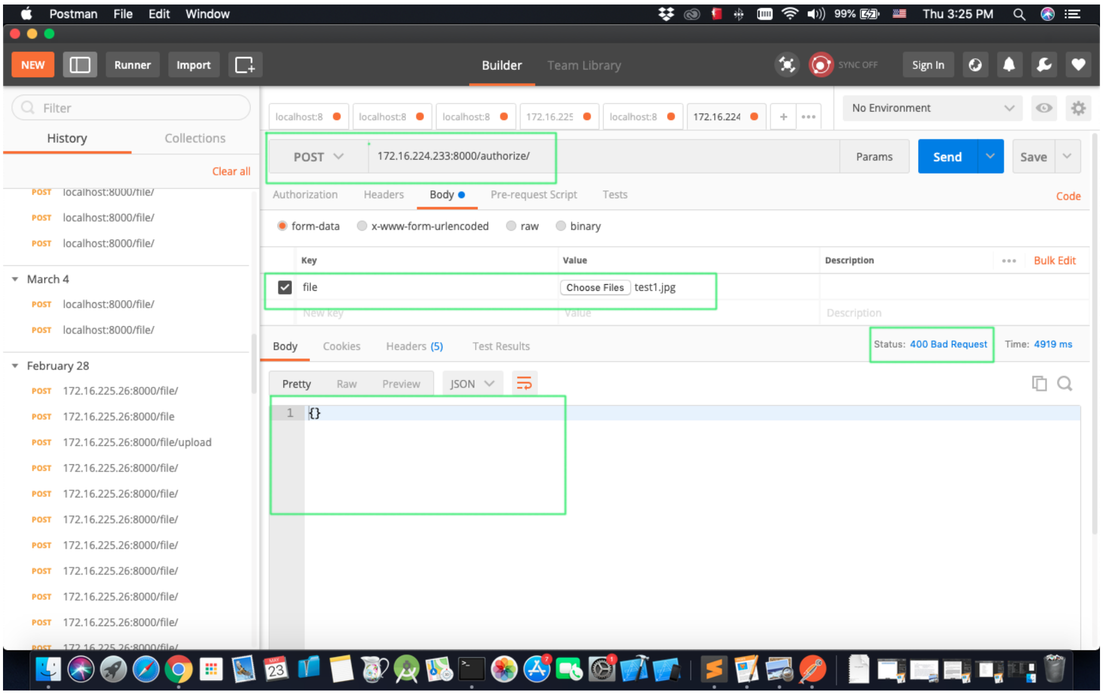
Key	Value
☑ file	test.jpg

Status: 200 OK Time: 1460 ms

JSON

```
{
    "id": 193,
    "created": "2019-05-23T09:56:04.405874Z",
    "file": "http://172.16.224.233:8000/media/test_MwioxYp.jpg"
}
```

Figure 5.6 : REST API - Success response



POST 172.16.224.233:8000/authorize/

Body: ● form-data ○ x-www-form-urlencoded ○ raw ○ binary

Key	Value	Description
☑ file	Choose Files test1.jpg	

Status: 400 Bad Request Time: 4919 ms

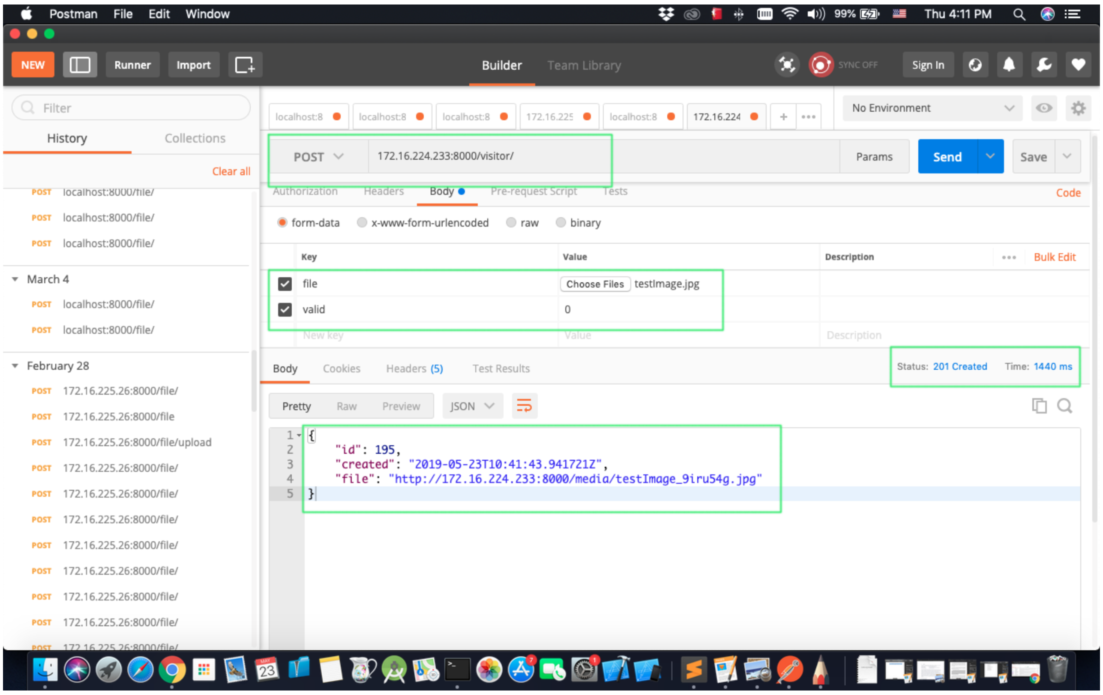
```
{}
```

Postman File Edit Window
Thu 3:25 PM
NEW Runner Import Builder Team Library Sign In
History Collections Clear all
No Environment
Params Send Save
Authorization Headers Body Pre-request Script Tests Code
Body Cookies Headers (5) Test Results
Pretty Raw Preview JSON

Figure 5.7 : REST API - Error response

The API used to notify the server regarding the visitor is a POST API with endpoint http://172.16.224.233:8000/visitor/ (Figure 5.8). It accepts two parameters:

- image of the visitor
- valid field that tells if the visitor is authorised or not (1 - authorised and 0 - unauthorised)



POST 172.16.224.233:8000/visitor/

Body: ● form-data ○ x-www-form-urlencoded ○ raw ○ binary

Key	Value	Description
☑ file	Choose Files testImage.jpg	
☑ valid	0	

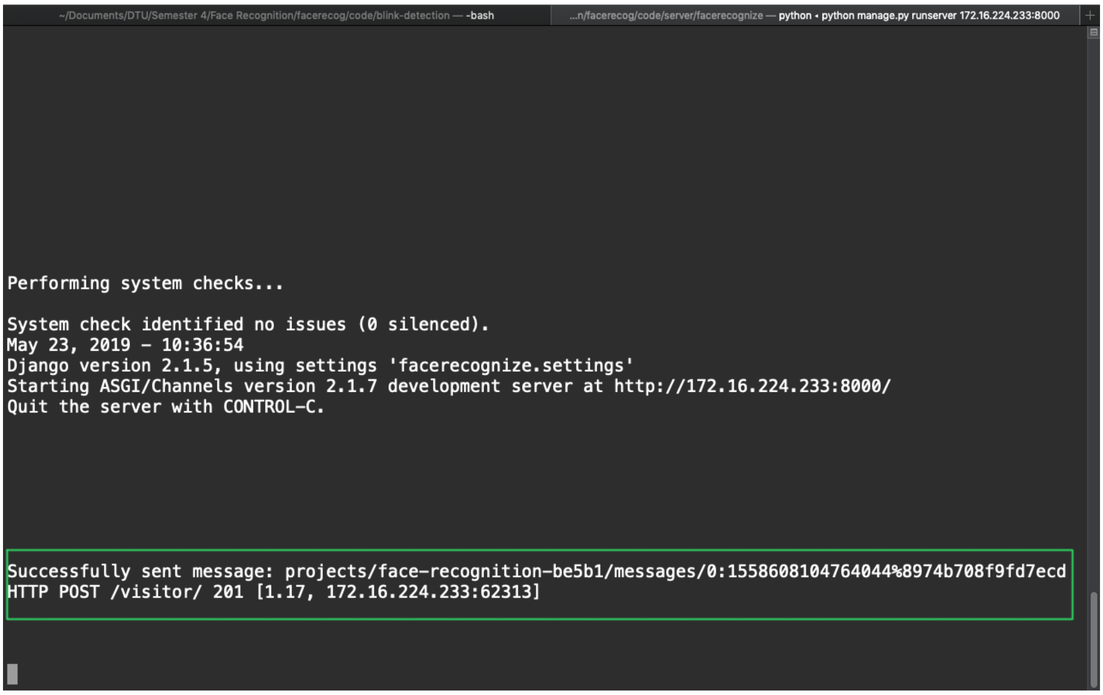
Status: 201 Created Time: 1440 ms

```
{
    "id": 195,
    "created": "2019-05-23T10:41:43.941721Z",
    "file": "http://172.16.224.233:8000/media/testImage_9iru54g.jpg"
}
```

Postman File Edit Window
Thu 4:11 PM
NEW Runner Import Builder Team Library Sign In
History Collections Clear all
No Environment
Params Send Save
Authorization Headers Body Pre-request Script Tests Code
Body Cookies Headers (5) Test Results
Pretty Raw Preview JSON

Figure 5.8 : REST API - Visitor

The response is a JSON data and in case the visitor is unauthorised then the server sends a notification using FCM to the user smartphone. On successfully sending a notification FCM send a success message (Figure 5.9). Each device(smartphone) is uniquely identified by a ID provided by FCM. This ID is stored in the server for sending notification.


~/Documents/DTU/Semester 4/Face Recognition/facerecog/code/blink-detection — -bash
...n/facerecog/code/server/facerecognize — python ◂ python manage.py runserver 172.16.224.233:8000

```
Performing system checks...

System check identified no issues (0 silenced).
May 23, 2019 - 10:36:54
Django version 2.1.5, using settings 'facerecognize.settings'
Starting ASGI/Channels version 2.1.7 development server at http://172.16.224.233:8000/
Quit the server with CONTROL-C.

Successfully sent message: projects/face-recognition-be5b1/messages/0:1558608104764044%8974b708f9fd7ecd
HTTP POST /visitor/ 201 [1.17, 172.16.224.233:62313]
```

Figure 5.9 : FCM success message after sending notification

Firebase Cloud Messaging (FCM)

Firebase Cloud Messaging(FCM) is a cross platform message delivery service provided by Google at free of cost. The data sent to the client is in form of JSON and the maximum payload allowed is 4KB. The format of the payload is:

```
{
        “message” : {
                “token” : “fd2iqbkm77g:APA91bFWphX4QkLtnyOYDLPKtQTp…”,
                “notification”:{
                        “title” : ” Title”,
                        “body” : “Body”
```

```
            },
            “data” :{
                        “key1” : “value1”,
                        “key2”, “value2”
            }
}
```

token : It is unique ID called instance ID provided by FCM to the client to uniquely identify a device. This ID is refreshed after app reinstall or if app data cleared

notification.title : The title of the notification

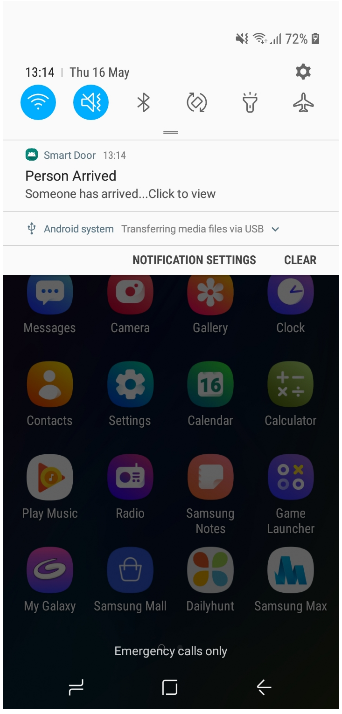
notification.body : The description of the notification

data : It contains custom key/value pairs data that developer needs to process at the client side

In our proposed system the payload data is of the format (Figure 5.10):

```
{
  “message” : {
        “token” : “d2iqbkm77g:APA91bFWphX4QkLtnyOYDLPKtQTp…”,
        “data“  : {
              “url” :<URL of the visitor>
              “message” : “Someone has arrived…
                                    Click to view”
              “valid” : “0”
        }
}
```

- The “Person Arrived” title is static
- The valid field is sent to decide whether to show door open buttons or not
- The url contains the image url which is used to show the image



13:14 | Thu 16 May

Smart Door 13:14

Person Arrived

Someone has arrived...Click to view

Android system Transferring media files via USB

Figure 5.10 : Example of FCM

CHAPTER 6

RESULT ANALYSIS

The main principal algorithms behind the working of proposed solution are face recognition and eye blink detection. Therefore these algorithms should provide accurate result in the building of smart, reliable and convenient system.

6.1 Face Recognition

The face recognition was implemented with the help of dlib library and the accuracy of the library was tested on FEI face dataset. The system has been trained on 10 persons using one shot learning concept (Figure 4.9) and for testing a total of 20 samples were taken with 50:50 ratio for authorised and unauthorised. The result is shown in Figure 6.1. The accuracy shown by the library is 95% for the 20 samples tested. This high accuracy is because of the fact that the library is already trained with billion of sample images and has already generated a weight matrix which is used in computing result. In machine learning the accuracy of the algorithm depends on this weight matrix and this matrix is generated with help of training samples. As the system is trained with billion of samples, therefore it provides a more accurate weight matrix hence a high accuracy result. The working of lib is based on DNN and it requires high computation power hence the face recognition part of the system can be performed at the server side. This would be requiring a good internet connectivity at client side.

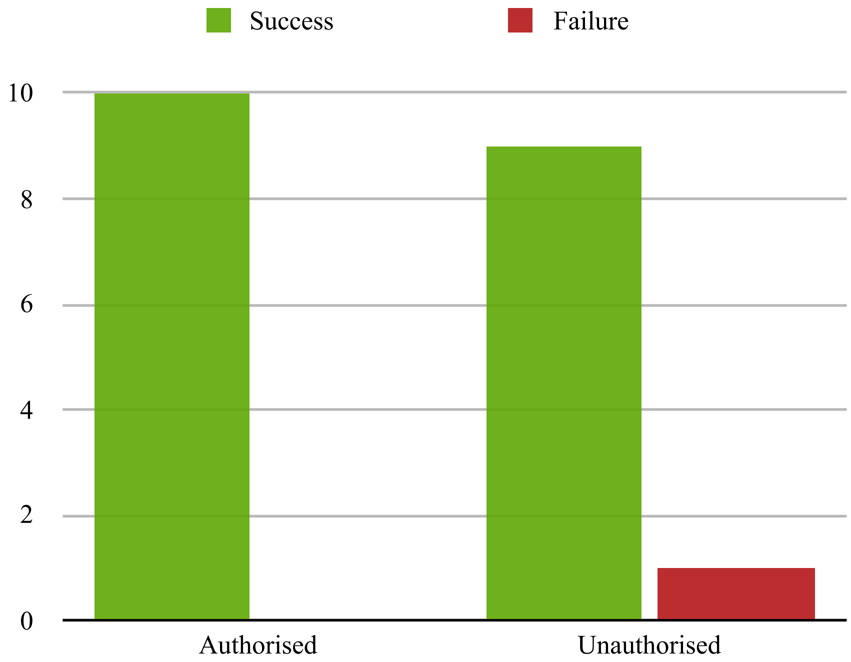


Figure 6.1 : Face recognition result

6.2 Eye blink detection

This method is the second principal algorithm of the proposed system and is used as a security part for preventing face spoofing. The working of blink detection is based on the calculation of EAR. The EAR value is checked with a threshold value, if it is less than the threshold then eye is closed else open. If the eye is closed for three consecutive frames then a blink is detected. The accuracy of blink detection is calculated with different threshold value. For each threshold value 10 blinks is processed and correspondingly the number of blinks the system detects is noted as output. Each blink is performed at an interval of 1 sec. Figure 6.2 shows the result of blink detection. It can be seen through the graph that with the increase in the EAR that system provides with a greater false positive result (>= 0.4). The accurate number of blinks is shown with EAR of 0.2 having the result of 9 blinks. The result was noted without wearing any spectacles, but if the user is wearing spectacles the result is very inaccurate as shown in Figure 6.3 resulting to drawback of the system having constraint of detecting authorised people only without spectacles. The accuracy of the system came out to be 90%. The value of EAR calculated is

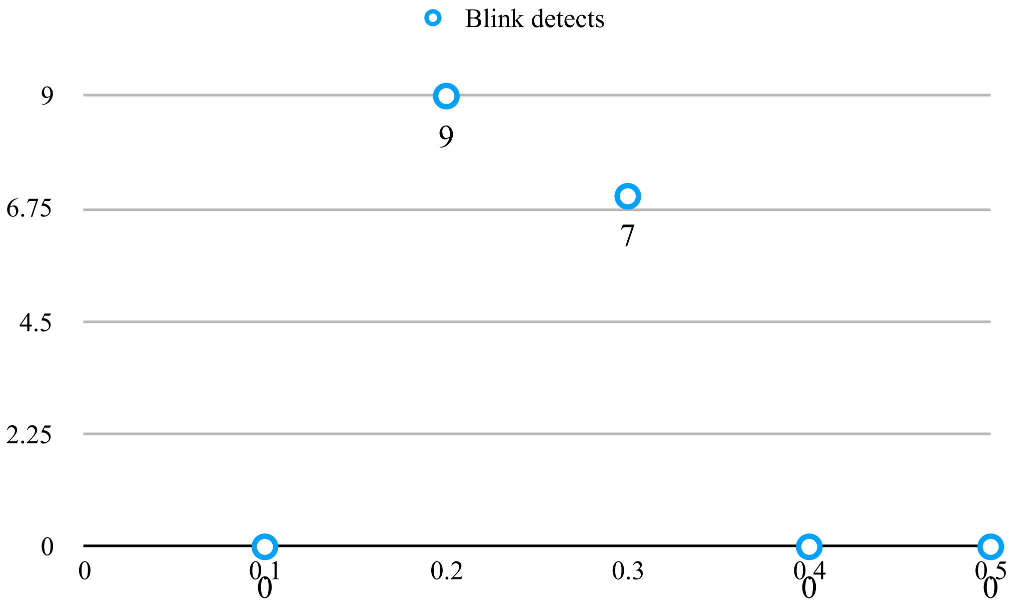


Figure 6.2 : Result of blink detection without spectacles

generally in the range of [0.1, 0.4], therefore higher threshold will make the system always detect a closed eye state so counter value remains 0.

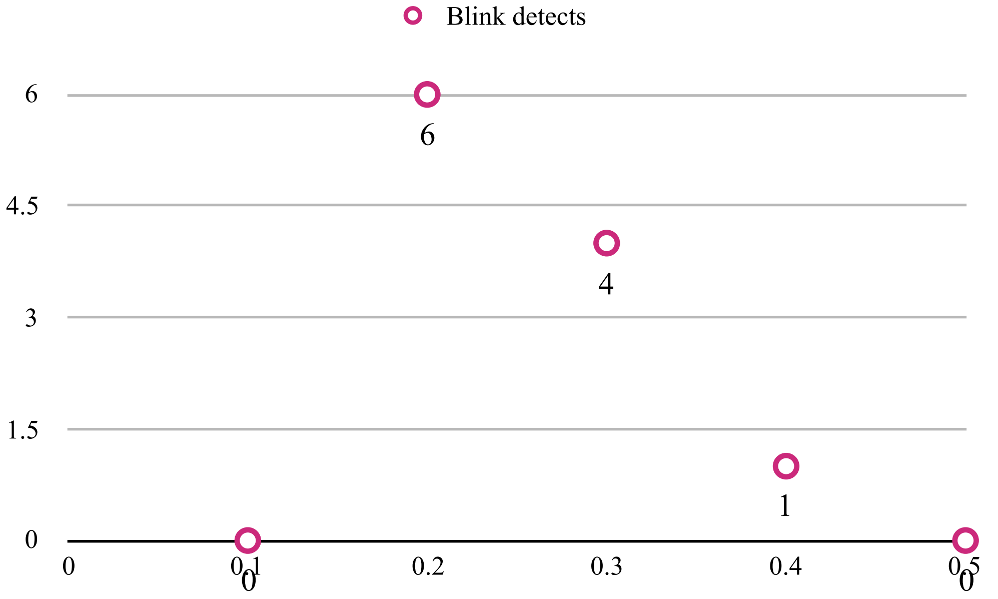


Figure 6.2 : Result of blink detection with spectacles

The accuracy of entire system comes out to be 92.5%. Hence it proves to be an efficient system for smart door unlocking.

CONCLUSION AND FUTURE SCOPE

Internet Of Things (IoT) is one of the current hot topics and lot of research is going around it and new applications are being developed involving IoT for helping users and improving their experiences. Smart door unlocking is a way to make our home smart by opening door on its own if some authorised user has arrived and which also can be controlled through an application which is installed on the authorised user smart home. This door unlocking mechanism will help users overcome various issues that they face involving door unlocking like they forget keys or they have to wait for someone to open the door. Also this application would help physically handicapped people to unlock door remotely through their smart phones.

The proposed solution for Smart door unlocking is simple and secured. It doesn't involve any multiple sensors/hardware, its just involves a camera which captures image and processing is done on the image. The requirement for the proposed solution is a good internet connectivity because when the user receives notification he captures his photo and send to server. If the user has bad network connectivity then the image would take time to get uploaded at the server and then the computation time and lastly the response from the server. If the system takes time then visitor might have gone and the door might get opened later which is unacceptable. The accuracy of the system comes to be same as that of blink detection which is 92.5%.

The working of blink detection works well without a person wearing glasses, but with glasses the dlib facial landmark detection the library doesn't provide accurate results. Therefore providing a better blink detection algorithm with/without glasses is part of future scope. The camera for face recognition would be placed at a certain height on the door thus limiting this approach for people whose height is upto the level of the camera. Also when a person arrives there should be enough lighting for eye detection to work accurately.

To enhance the security the password in blink detection i.e number of times the person needs to blink can be changed randomly to a different value every day. This request can also be requested by user through providing a functionality in the application if he/she feels the value has been leaked. Whenever the value changes a notification would go to the user with the new value. Currently only one super user exist in the system who will have the application. New value of the password will only be available to the super user, thus it will be his/her duty to provide the new value securely to other authorised users. The system can easily be extended for multiple super user.

Whenever a user sense of some suspicious activity outside the door then the user thinks of going to the door and have a look. As a part of future scope a feature of getting a live feed on the user smart phone can be added to the system. This live feed feature will allow user to get outside view without going and opening the door.

REFERENCES

[1] T. Soukupova and J. Cech, “Real-time eye blink detection using facial landmarks,” in 21st Computer Vision Winter Workshop (CVWW2016), 2016

[2] M. Chau and M. Betke. “Real time eye tracking and blink detection with USB cameras”, Technical Report 2005-12, Boston University Computer Science, May 2005

[3] H. Dinh, E. Jovanov, and R. Adhami, “Eye blink detection using intensity vertical projection”, International Multi-Conference on Engineering and Technological Innovation, IMETI 2012

[4] F. Yang, X. Yu, J. Huang, P. Yang, and D. Metaxas, “Robust eyelid tracking for fatigue detection”, ICIP 2012

[5] P. P. Gaikwad, J. P. Gabhane, S. S. Golait, "A Survey based on Smart Homes System Using Internet-of-Things”, International Conference on Computation of Power Information and Communication, 2015

[6] K. Bing, L. Fu, Y. Zhuo and L. Yanlei, "Design of an Internet of Things-based Smart Home System", The 2nd International Conference on Intelligent Control and Information Processing, 2011

[7] N. H. Ismail and E. I. Saadon, "Android-Based Home Door Locks Application via Bluetooth for Disabled People", Proceedings of the International Conference on control System Computing and Engineering Penang IEEE, 2014

[8] X. Mao, K. Li, Z. Zhang, J. Liang, "Design and implementation of a new smart home control system based on internet of things" in Smart Cities Conference, IEEE, 2017

[9] R. D. H. Arifin and R. Samo, “Door Automation System Based on Speech Command and PIN using Android Smartphone”, International Conference on Information and Communications Technology (ICOIACT), 2018

[10] S. Tiwari, S. Thakur, D. Shetty and A. Pandey, “Smart Security: Remotely Controllable Doorlock”, Proceedings of the 2nd International Conference on Inventive Communication and Computational Technologies, 2018

[11] H. ElKamchouchi and A. ElShafee, “Design and Prototype Implementation of SMS Based Home Automation System”, International Conference on Electronics Design, System and Applications (ICEDSA), 2012

[12] M. Asadullah and A. Raza, "An overview of home automation systems", 2nd International Conference on Robotics and Artificial Intelligence (ICRAI), 2016.

[13] H. Hassan, R. A. Bakar, and A. T. F. Mokhtar, "Face recognition based on auto-switching magnetic door lock system using microcontroller," International Conference System Engineering and Technology (ICSET), pp.1-6, Sep. 2012.

[14] I. Kramberger, M. Grasic, and T. Rotovnik, "Door phone embedded system for voice based user identification and verification platform," IEEE Trans. Consumer Electronics, vol.57, no.3, pp.1212-1217, 2011

[15] M. R. Alam, M. B. I. Reaz, and M. A. M. Ali, "A Review of Smart Homes—Past, Present, and Future," IEEE Trans. Systems, Man, and Cybernetics, Part C: Applications and Reviews, vol.42, no.6, pp.1190- 1203, 2012

[16] Y. J. Oh, E. H. Paik, and K. R. Park, "Design of a SIP-based real-time visitor communication and door control architecture using a home gateway," *IEEE Trans. Consumer Electronics*, vol.52, no.4, pp.1256- 1260, 2006

[17] Y. T. Park, P. Sthapit, and J. Y. Pyun, "Smart digital door lock for the home automation”, *TENCON 2009 - 2009 IEEE Region 10 Conference* pp.1-6, 2009

[18] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features”, CVPR, 2001

[19] M. Bagga and B. Singh, ”Spoofing Detection in Face Recognition: A Review”, 3rd International Conference on Computing for Sustainable Global Development (INDIACom), 2016

[20] S. L. Keoh, S. S. Kumar, and H. Tschofenig. “Securing the internet of things: A standardization perspective”, IEEE Internet of Things Journal, 1(3):265–275, 2014

[21] H. Modares, R. Salleh, and A. Moravejosharieh, “Overview of security issues in wireless sensor networks”, 3rd International Conference on Computational Intelligence, Modelling Simulation, 2011

[22] S. Kumar, S. Singh and J. Kumar, “A comparative study on face spoofing attacks”, International Conference on Computing, Communication and Automation (ICCCA), 2017

[23] R. A. Ramlee, M. A. Othman, M. H. Leong, M. M. Ismail and S. S. S. Ranjit, "Smart home system using android application," 2013 International Conference of Information and Communication Technology (ICoICT), Bandung, pp. 277-280, 2013

[24] S. Kumar and S. R. Lee, "Android based smart home system with control via Bluetooth and internet connectivity," The 18th IEEE International Symposium on Consumer Electronics (ISCE 2014), JeJu Island, pp. 1-2, 2014

[25] V. Puri and A. Nayyar, "Real time smart home automation based on PIC microcontroller, Bluetooth and Android technology," 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, pp. 1478-1484, 2016

[26] D. Sullivan, W. Chen and A. Pandya, "Design of remote control of home appliances via Bluetooth and Android smart phones," 2017 IEEE International Conference on Consumer Electronics - Taiwan (ICCE-TW), Taipei, pp. 371-372, 2017

[27] M. Uddin and T. Nadeem, "EnergySniffer: Home energy monitoring system using smart phones”, 8th International Wireless Communications and Mobile Computing Conference (IWCMC), Limassol, , pp. 159-164, 2012

[28] S. Li, J. Li, X. Nie and L. Kong, "Design and Implementation of Smart Home Based on Android," 2015 4th International Conference on Advanced Information Technology and Sensor Application (AITS), Harbin, pp. 32-35, 2015

[29] S. Tang, V. Kalavally, K. Y. Ng and J. Parkkinen, "Development of A Prototype Smart Home Intelligent Lighting Control Architecture Using Sensors Onboard a Mobile Computing System", Energy and Buildings, vol. 138, pp. 368-376, 2017

[30] N. Datta, T. Masud, R. Arefin, A. A. Rimon, M. S. Rahman and B. B. Pathik, "Designing and implementation of an application based electrical circuit for smart home application," 2014 IEEE Student Conference on Research and Development, Batu Ferringhi, pp. 1-5, 2014

[31] P. Vagdevi, D. Nagaraj and G. V. Prasad, “Home: IOT Based Home Automation Using NFC ”, International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2017

[32] N. Chandrakar, S. Kaul, M. Mohan, C. S. Vamsi and K. R. Prabhu, “NFC based profiling of smart home lighting system”, International Conference on Industrial Instrumentation and Control (ICIC), 2015