

# Smart Security System based on Frontal Face Detection Method and Android Mobile



Inspiring Excellence

**Supervisor: Dr. Jia Uddin**

Mahjabeen Khan 14101272

*Department of Computer Science and Engineering,*

*BRAC University.*

**Submitted on: 14<sup>th</sup> December 2016**

## **DECLARATION**

I hereby declare that this report is done by my work and effort and that it has not been submitted anywhere for any degree. Materials of all the contents provided here is totally based on my own labor dedicated for the completion of the thesis. Other sources of information have been found by other researcher are mentioned in the reference section.

**Signature of Supervisor**

**Signature of Author**

---

Dr. Jia Uddin

---

Mahjabeen Khan

## **ACKNOWLEDGEMENTS**

My deepest sense of thankfulness to Almighty ALLAH, the creator and the owner of this universe, the most merciful, beneficent and the most gracious, who provided me guidance, strength and abilities to complete this research.

I would like to express my sincere gratitude to my advisor, Dr. Jia Uddin, for his patience, motivation, and guidance throughout this research. This work would not have been completed without his help, support, and practically infinite supply of comments and ideas in uncountable occasions. I am also thankful to the BRAC University Faculty Staffs of the Computer and Communication Engineering, who have been a light of guidance for us in the whole study period.

Finally, I would like to express my sincere gratefulness with my parents, my family, and my friends, who always supported and encourage me.

# CONTENTS

<b>DECLARATION</b> .....	2
<b>ACKNOWLEDGEMENTS</b> .....	3
<b>CONTENTS</b> .....	4
<b>LIST OF FIGURES</b> .....	6
<b>LIST OF TABLES</b> .....	7
<b>ABSTRACT</b> .....	8
<b>CHAPTER 01: INTRODUCTION</b>	
1.1 Motivations.....	9
1.2 Contribution Summary.....	10
1.3 Thesis Orientation.....	10
<b>CHAPTER 02: BACKGROUND INFORMATION</b>	
2.1 System Architecture.....	11
2.2 Choice of Aurdino.....	12
2.3 System connection with Bluetooth.....	12
2.4 Algorithm.....	13
2.4.1 Local Binary Patterns Histograms (LBPH) Recognizer Algorithm.....	13
<b>CHAPTER 03: METHODS AND IMPLEMENTATION DETAIL</b>	
3.1 Introduction.....	16
3.2 Process Execution.....	16
3.3 The Door Lock.....	18
3.4 Establishing Bluetooth Connection.....	18
3.4.1 Bluetooth Module.....	18

3.4.2	Android Phone with Bluetooth and Lock It Door App.....	19
3.4.3	Definite range for Bluetooth connectivity.....	19
3.4.4	Password.....	19
3.4.5	Connectivity.....	22
3.5	Face Detection Method.....	22
3.5.1	Process Execution Steps.....	22
3.5.2	Database of Images.....	23
3.5.3	Connecting the Camera.....	24
3.5.4	Inputting Modules.....	25
3.5.5	Training Recognizer for Face Detection.....	25
3.5.6	Applying Local Binary Patterns Histograms (LBPH) Recognizer algorithm.....	25
3.5.7	Extracting Features from the Images.....	26
3.5.8	Displaying Result as Confidence.....	26
<b>CHAPTER 04: EXPERIMENTAL RESULTS</b>		
4.1	Introduction.....	27
4.2	Establishing Bluetooth Connection.....	27
4.2.1	Bluetooth Signal Strength.....	27
4.2.2	Match of Passwords.....	28
4.3	Face Detection	
4.3.1	Comparison and Result.....	30
4.3.2	Confidence as Accuracy.....	30
<b>CHAPTER 05: CONCLUSIONS AND FUTURE WORKS</b>		
5.1	Concluding Remarks .....	32
5.2	Future Works .....	32
5.2.1	System Connectivity.....	32
5.2.2	User Detection.....	32
<b>REFERENCES.....</b>		<b>33</b>

# LIST OF FIGURES

Figure 2.1: System Architecture of Bluetooth Module.....	11
Figure 2.2: Grained details in Image.....	14
Figure 3.1: Work Flow of the System.....	17
Figure 3.2: An Electric Door Lock and a 12V Battery.....	18
Figure 3.3: The Bluetooth password is set at QWER in the system.....	12
Figure 3.4: The password if found correct, the loop finishes and if not it breaks the loop and continues to ask for the password again.....	20
Figure 3.5: Database of Photos stored in the System.....	21
Figure 3.6: A USB HD Camera.....	23
Figure 4.1: The Android app is not connected so the lock is closed.....	24
Figure 4.2: The Android app is connected and the lock opens.....	29
Figure: 4.3: Results after matching with accuracy using python.....	31

## LIST OF TABLES

Table 2.1: Comparison between Arduino Uno and Arduino Mega.....	12
Table 4.1: Signal Strength for Different Locations.....	31

## **ABSTRACT**

Security nowadays is a very important issue so Smart Security System based on Frontal Face Detection Method and Android Mobile was conceived with the idea that it will offer protection. This system consists two parts where the first is the Bluetooth connectivity via Android phone to an electric door lock forwarded by face detection system via camera. It is done by OpenCv in Python using Local Binary Patterns Histograms (LBPH) Recognizer algorithm.

The Bluetooth is connected with an app on the Android phone. The app will seek password that is saved in the system. If there is a match in the password, the process move forwards towards face detection program.

The recognizer is trained earlier with the images stored in the database. As a face appears in front of the camera, the system compares it with those photos in the database. LBPH works by characterizing the local patterns in each location in the image and thus it analyzes the image. The system will decide whether to restrict or allow any person depending on the comparison result.

Experimental results show that the proposed method exhibited 100% accuracy for a tested dataset.



# CHAPTER 01

## INTRODUCTION

In this era of time, security issues are given utmost priority as the situations around places aren't suitable often. Every business owner strives to keep their employees, assets, and office space as safe as possible and same goes for homes of people. A number of incidents occurred recently in this country due to lack of safety that caused a stir among people. Crimes have increased rapidly due to lack of security measures. The growing crime rates across cities reflect the bitter reality. Many people overlook, ignore, and underestimate the need of taking appropriate home security measures. A burglary or theft can lead to devastating consequences, both emotionally and financially. While the financial loss may be recoverable, the trauma inflicted on the family may last forever.

There are several ways to help increase the security at doors; one of the most effective is to install a security system that is simple as well as protective. Smart Security System is thus proposed viewing the need of an advanced security system.

### **1.1 Motivations**

Understanding the necessity of security system this idea has been proposed. This system is structured in a way so that maximum security is ensured. The door locks now used in most places can be easily opened with master key but this system consists of special features which require conditions to open a door lock. A step towards technology making life simpler and secure is what brought up the motivation to build it. The System consists of two sections where each has a criteria or condition to fulfill. As these days everyone has a smart phone so this system won't be difficult to use. With a help of a simple smart phone this system can be executed.

## **1.2 Contribution Summary**

The summary of the main contributions is as follows:

- Arduino Uno has been used with Bluetooth module to establish the connection with Bluetooth of Android phone.
- An electric door lock that requires 12V to be operated.
- The face detection program is done in OpenCV using Python using Local Binary Patterns Histograms (LBPH) Recognizer algorithm.
- By using Local Binary Patterns Histograms (LBPH) Recognizer algorithm the whole process become faster in execution time as this algorithm makes sure that each face is trained separately giving a complete accuracy.

## **1.3 Thesis Orientation**

The rest of the thesis is organized as follows:

- Chapter 02 includes the necessary background information of the system explaining Bluetooth connectivity and face detection method.
- Chapter 03 presents the methods and implementation details for the system.
- Chapter 04 demonstrates the experimental results and comparison.
- Chapter 05 concludes the thesis and states the future research directions.

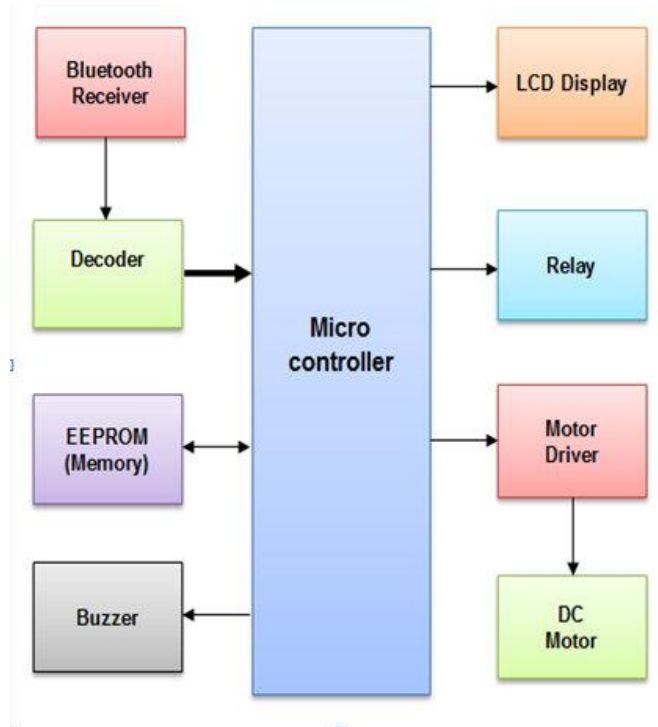
# CHAPTER 02

## BACKGROUND INFORMATION

The security system consists of two different parts so their architectures are separate and different. The Bluetooth Connection consists most of the hardware section so it has the architecture set properly.

### 2.1 System Architecture

The System architecture of the Bluetooth Connection consists of different parts which are shown in figure 2.1. Here Microcontroller is connected with Decoder, Memory, Buzzer, LCD Display, Relay and Motor Driver. Here, the Bluetooth Receiver is connected with the Decoder that makes it able to reach the Microcontroller. The DC Motor is then attached to the Motor Driver.



**Figure 2.1:** System Architecture of Bluetooth Module

## 2.2 Choice of Aurdino

Arduino Uno has been chosen for this system as Arduino uno supports the connection of Bluetooth whereas Arduino Mega despite of having more program space does not allow establishing Bluetooth Connection via Bluetooth Module. Comparison between Arduino Uno and Arduino Mega is shown in the Table 2.1 below. The choice of choosing an arduino is a major decision in establishing a

**Table 2.1:** Comparison between Arduino Uno and Arduino Mega

<b>Comparison</b>	
<b>Arduino Uno</b>	<b>Arduino Mega</b>
Allows Bluetooth Connection	Does Not Allow Bluetooth Connection
Runs most programs	Does not Run most programs
Very compact	Not very compact
Support Wi-Fi	Do Not Support Wi-Fi
Space 32KB Program Space	Space 256 KB Program Space

## 2.3 System Connection with Bluetooth

The former idea was to establish the connection with the system with a Wi-Fi but viewing pros and cons of both Bluetooth and Wi-Fi, it has been later decided that the connectivity of the system with that of the device will be done using Bluetooth Module. This is because the major con of Wi-Fi is that it needs internet connectivity. In most cases Bluetooth acts better than Wi-Fi in terms of connection. The Connectivity speed although is much greater in Wi-Fi but there can be loss of connection due to absence of internet, so Bluetooth has been chosen in the system connection.

## 2.4 Algorithm

OpenCV 2.4 now comes with the Face Recognizer class for face recognition.

The currently available algorithms are:

- Eigenfaces
- Fisherfaces
- Local Binary Patterns Histograms

In this system the Local Binary Patterns Histograms Face Recognizer Algorithm is used as it seems to have efficient way to detect face rather than the rest of the algorithms.

### 2.4.1 Local Binary Patterns Histograms (LBPH) Recognizer Algorithm

LBPH analyzes each face in the training set separately and independently. In LBPH each image is analyzed independently, while the Eigenfaces method looks at the dataset as a whole. The LBPH method is somewhat simpler, in the sense that we characterize each image in the dataset locally; and when a new unknown image is provided, same analysis is performed on it and compares the result to each of the images in the dataset. The way to analyze the images is by characterizing the local patterns in each location in the image. LBPH method for face recognition works better in different environments and light conditions, however, it will depend on the

training and testing data sets. Around 10 different images of a person's face are needed in order to be able to recognize him/her.

LBP operator can be described as:

$$LBP(x_c, y_c) = \sum_{p=0}^{P-1} 2^p s(i_p - i_c) \dots \dots \dots (1)$$

Here,

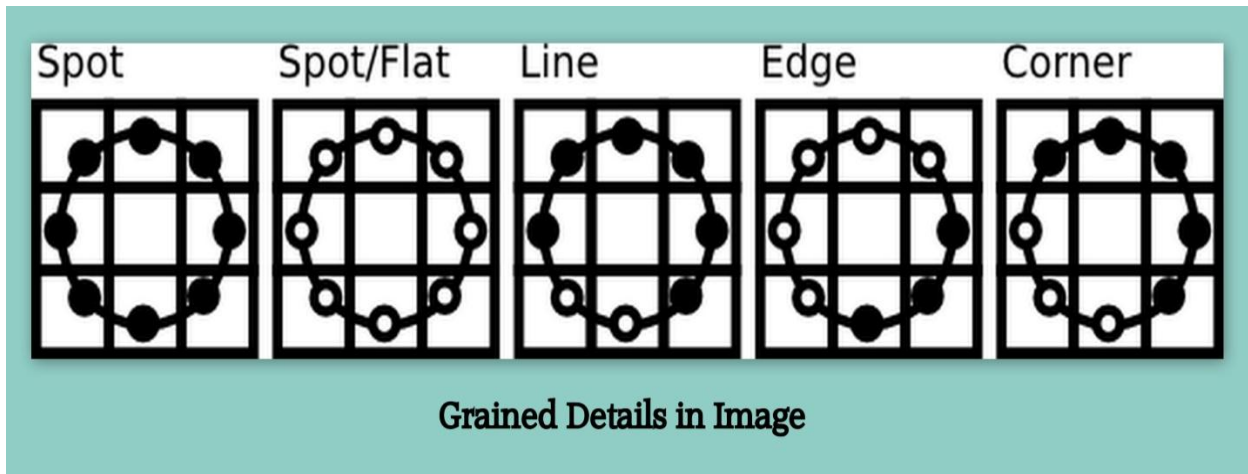
$(x_c, y_c)$  - the central pixel with intensity

$i_p$  and  $i_c$  - the intensity of the neighbor pixel

s, the sign function defined as:

$$s(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{else} \end{cases} \dots \dots \dots (2)$$

This description enables to capture very fine grained details in images. To capture the following neighborhoods, the idea is to align an arbitrary number of neighbors on a circle with a variable radius.



**Figure 2.2:** Grained details in Image

For a given Point  $(x_c, y_c)$  the position of the neighbor  $(x_p, y_p)$   $p \in P$  can be calculated by the following equations:

$$x_p = x_c + R \cos\left(\frac{2\pi p}{P}\right) \dots \dots \dots (3)$$

$$y_p = y_c - R \sin\left(\frac{2\pi p}{P}\right) \dots \dots \dots (4)$$

Where R is the radius of the circle and P is the number of sample points.

# CHAPTER 3

## METHODS AND IMPLEMENTATION DETAIL

### 3.1 Introduction

Smart Security System based on Frontal Face Detection Method and Android Mobile consists two main parts where both are linked together. The system is structured in a way where the second part of the system can be accessed only if the requirement of the first part is fulfilled. As this system is quite large so the steps to execute the complete process is quite a many.

### 3.2 Process Execution

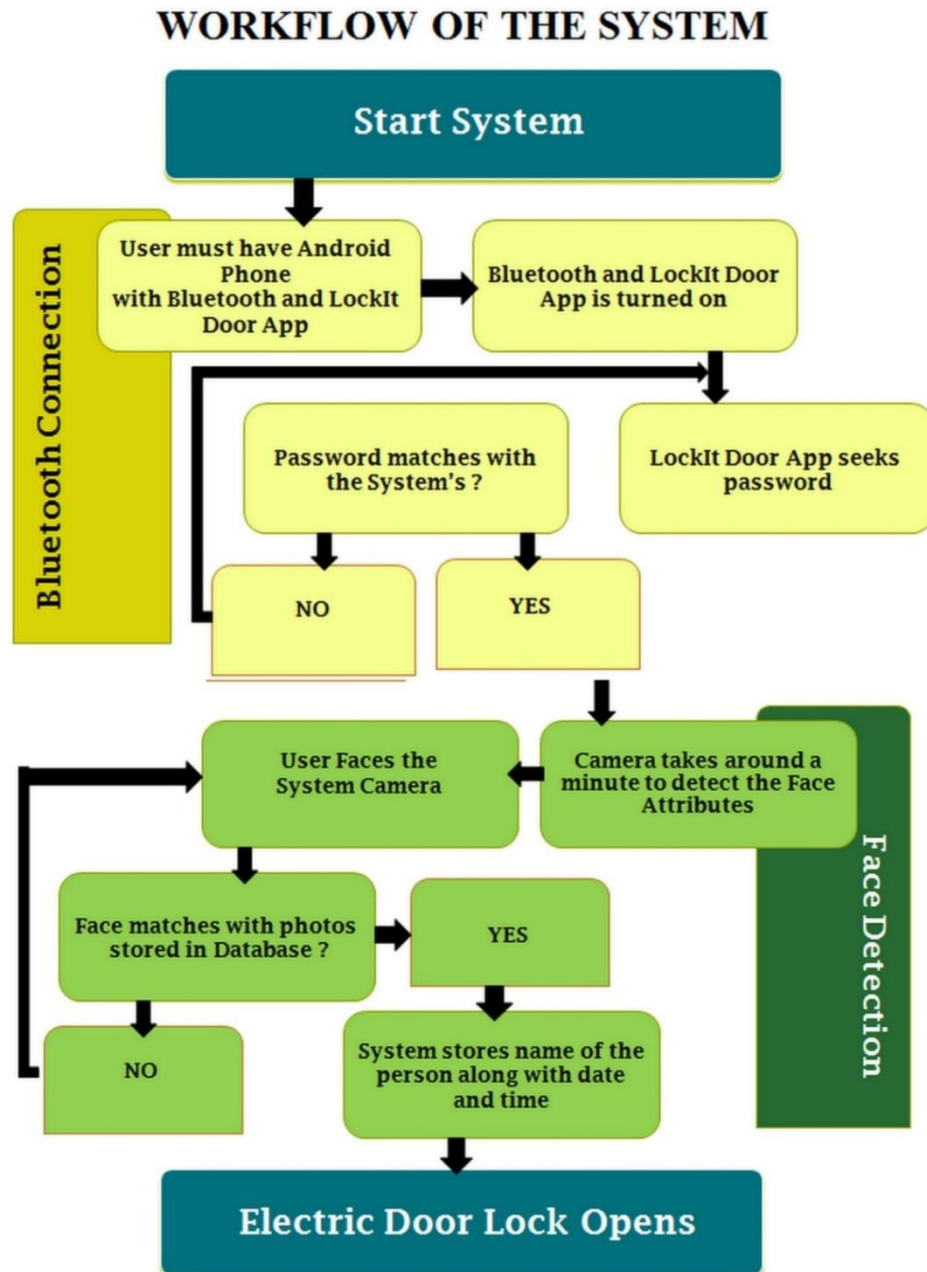
Figure 3.1 demonstrates a detailed implementation of the proposed model. It demonstrates how the algorithm is set up. Firstly, the Bluetooth Connection has to be established which requires the user to have an Android phone with a Bluetooth and Lock It Door App installed. Turning on both of them, the app will seek a password. This password has to be the one that is saved in the system. If the user provides the correct password to the app, there will be a confirmation of the password being matched. As the password matches, the process proceeds to the second part of the system, which is the Face Detection part. On the other hand if the password fails to match with that of the System's, there will be no connection established and the Lock It Door App will ask for the password again.

The second part of the system is the Face Detection part where the user has to face the camera attached to the system. This camera will take a minute or two to detect the face of the person. As the camera finds the face of the person, it will extract features of the face. The extracted features are then compared with those of the image of faces stored in the database.

If the face of the user matches with those saved in the system, the door lock will open. As it opens the process ends and user gets in. The door lock will remain open for 30 seconds and



after that the door lock will close. If the face does not match with those in the system, user has to face the camera again and wait for his face detection process to be executed again.



**Figure 3.1:** Work Flow of the System

### 3.3 The Door Lock

The door lock should be an electronic door lock which is now available at many shops. The lock also requires 12V of battery to be operated. The type of electric door lock and the battery used in this system is shown in figure 3.2.



**Figure 3.2:** An Electric Door Lock and a 12V Battery

### 3.4 Establishing Bluetooth Connection

To establish the Bluetooth Connection, there are few things that should be present and kept in mind to execute the process. They are discussed in the following sections.

#### 3.4.1 Bluetooth Module

The system needs a Bluetooth module that needs to be connected with the Bluetooth of the Android phone.

### **3.4.2 Android Phone with Bluetooth and Lock It Door App**

To get connected with the system, the user should have an Android Phone that has Bluetooth and Lock It Door App. The App seeks a password to establish the Bluetooth Connection. The password should be the one that is saved in the system.

### **3.4.3 Definite range for Bluetooth connectivity**

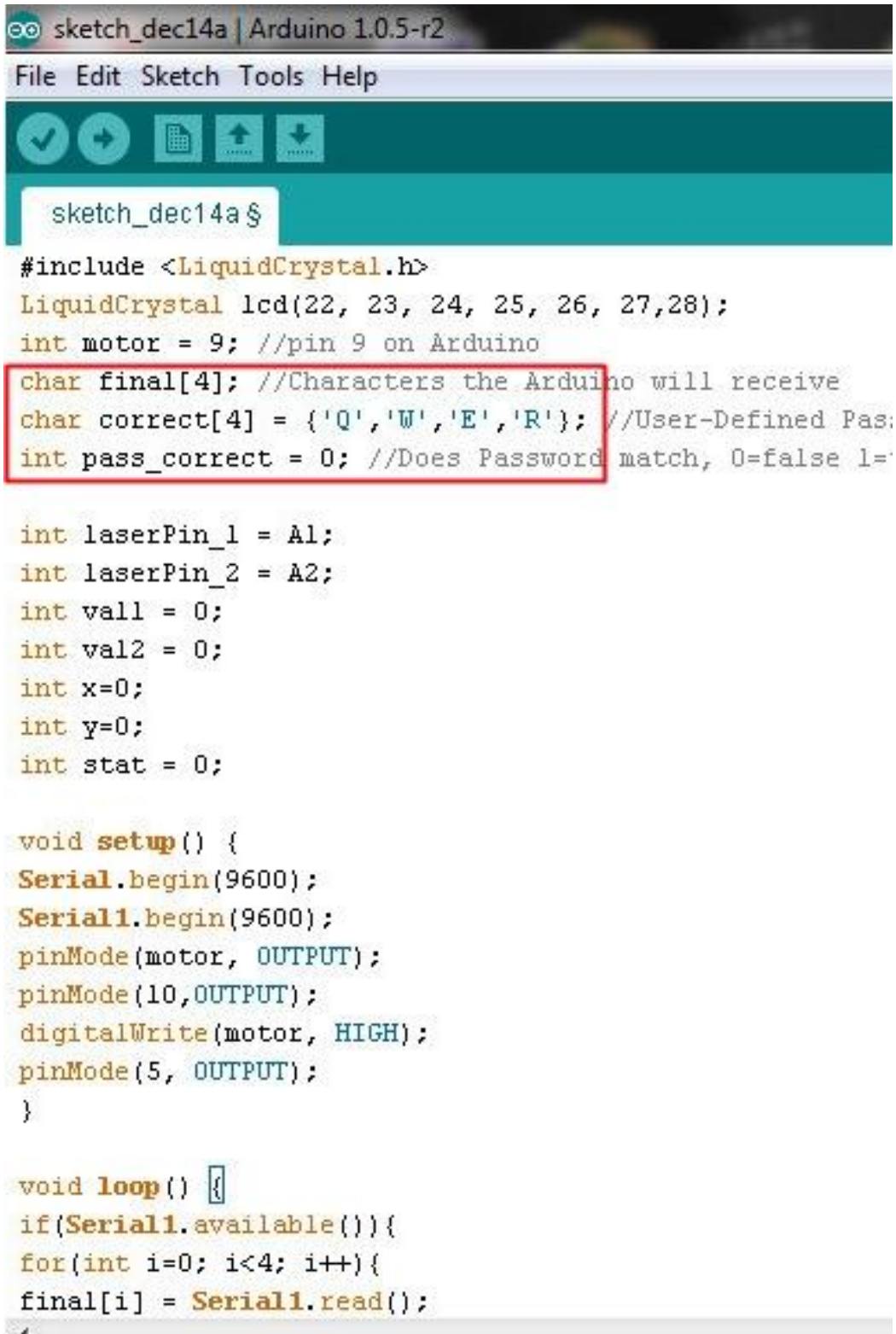
There is a specific limit for the user to be close enough to get connected with the Bluetooth of the system. If the range is too much there is a possibility that the user may not be able to get connected with the system and will cause disconnection. To get connected with the system the user must be in a specific distance that is close enough to the Bluetooth on the door lock. Without being in a definite range, Bluetooth connection is not possible.

### **3.4.4 Password**

The user when wants to get connected with the Bluetooth of the door lock, the Lock It Door app will seek a password for establishing the connection. The system will have a password saved in it and so the user should give that exact same password in the app. As the password matches, the process of Bluetooth connection is completed. There will be a confirmation message in the app once it gets the correct password. If the password does not match with that of the system, user will be asked to give the password again.

Figure 3.2 shows that the Bluetooth password is set as QWER in the system. This is done in the Arduino software and the image is captured from the code executed in the software. The red portion of the code tells that the password is of 4 characters and they are 'Q', 'W', 'E', 'R'. It also says that if the password matches then it will be 1.

Figure 3.3 shows that if the password is found correct, the loop finishes and if not it breaks the loop and continues to ask for the password again. The red portion here is the loop for the password check. If the password is matched, the loop is over whereas for incorrect password the loop breaks and it starts from the beginning of the loop where it again seeks password.



```
sketch_dec14a | Arduino 1.0.5-r2
File Edit Sketch Tools Help

sketch_dec14a $
#include <LiquidCrystal.h>
LiquidCrystal lcd(22, 23, 24, 25, 26, 27,28);
int motor = 9; //pin 9 on Arduino
char final[4]; //Characters the Arduino will receive
char correct[4] = {'Q','W','E','R'}; //User-Defined Pas:
int pass_correct = 0; //Does Password match, 0=false 1=

int laserPin_1 = A1;
int laserPin_2 = A2;
int val1 = 0;
int val2 = 0;
int x=0;
int y=0;
int stat = 0;

void setup() {
  Serial.begin(9600);
  Serial1.begin(9600);
  pinMode(motor, OUTPUT);
  pinMode(10,OUTPUT);
  digitalWrite(motor, HIGH);
  pinMode(5, OUTPUT);
}

void loop() {
  if(Serial1.available()){
    for(int i=0; i<4; i++){
      final[i] = Serial1.read();
    }
  }
}
```

**Figure 3.3:** The Bluetooth password is set at QWER in the system



```
sketch_dec14a | Arduino 1.0.5-r2
File Edit Sketch Tools Help

sketch_dec14a $
pinMode(motor, OUTPUT);
pinMode(10,OUTPUT);
digitalWrite(motor, HIGH);
pinMode(5, OUTPUT);
}

void loop() {
  if(Serial1.available()){
    for(int i=0; i<4; i++){
      final[i] = Serial1.read();
      delay(50);
      Serial.print(final[i]);
    }

    Serial1.flush();
    for(int i=0; i<4; i++){
      if(final[i]==correct[i]){
        pass_correct = 1;
      }
      else{
        pass_correct = 0;
        break;
      }
    }
  }
}
```

**Figure 3.4:** The password if found correct, the loop finishes and if not it breaks the loop and continues to ask for the password again

### **3.4.5 Connectivity**

The Bluetooth connectivity depends on all the mentioned above sections and thus if all these are done in correct manner, the Bluetooth connection can be established properly. If there is any sort of problem in Bluetooth connection then it can be understood that the above criteria were not completed or carried out accordingly. Establishing the Bluetooth connection properly leads to the second part of the system, the face detection.

### **3.5 Face Detection Method**

This part of the system begins if the previous part of establishing Bluetooth connection is successful. The user has to face the camera that is connected with the system. It will take around a minute for the camera to detect the face attributes of the user. After a minute if the face attributes are extracted by the system. The system will match them with the photos of the faces stored in the database. The faces stored in the database have their name and identity saved in the database. If there is a match found then the system will store the name with date and time of the specific day when the match was found. This helps to keep the information of the users entering and leaving the place. After storing, as the processes are successfully completed, the electric door lock opens and the person can enter the room. The electric door lock will remain open for about 20-30 seconds; this timing is fixed in the system. If there no match found with the face appearing in front of the camera with that of photos stored in the database then it goes back to the segment where the user has to face the camera for a minute to detect his/her face attributes to extract.

#### **3.5.1 Process Execution Steps**

The Face detection method is a major part of the system as this completes the entire process. This method has few sections as well to be implemented properly and they are discussed below.

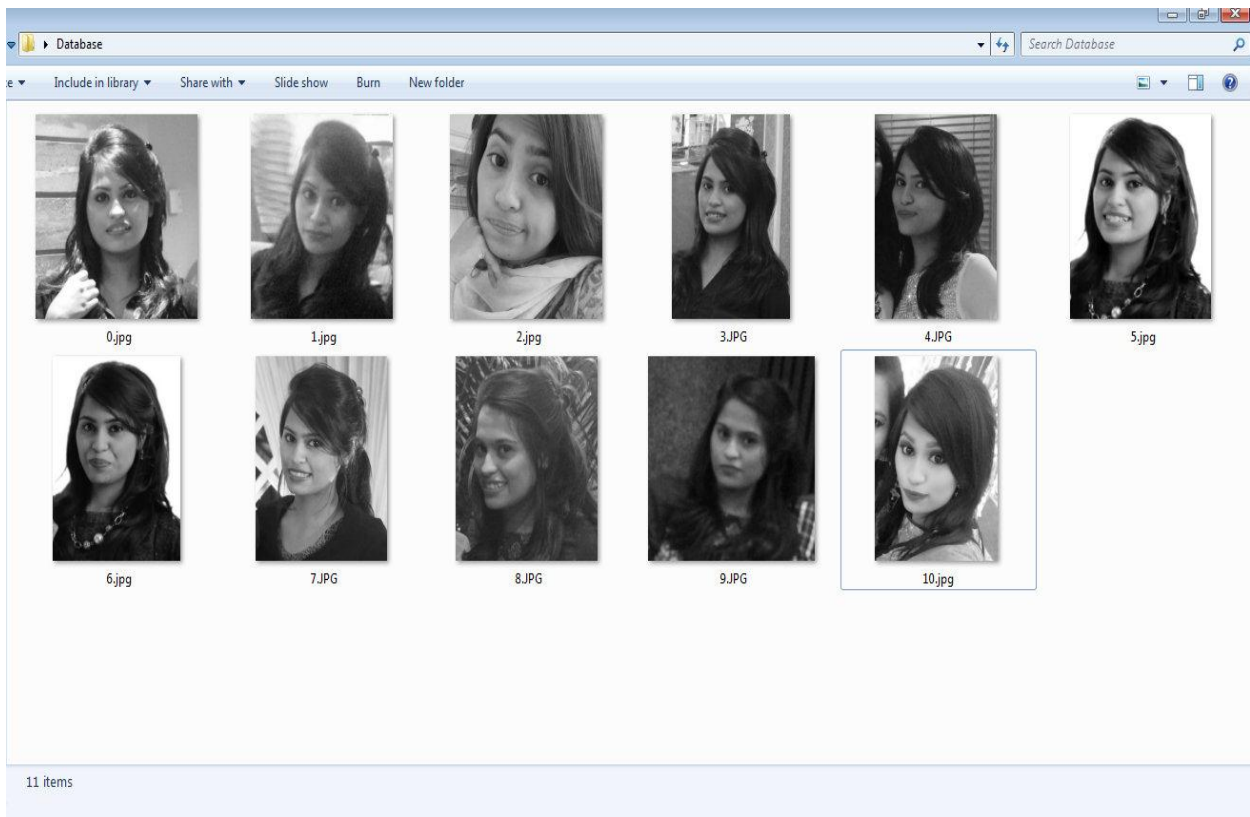
The whole process for training the recognizer can be divided in the following major steps -

- The first step is to find the database of faces with several images for each individual person.
- The next step is to detect faces that are stored in the database images and use them to train the face recognizer.

- The last step is to test the face recognizer so that it is trained enough to recognize faces.

### 3.5.2 Database of Images

The user of this system must have a database of images of him stored in the system which will compare with his face when appeared in front of the camera. The Figure 3.5 below shows the number of image stored in the database for comparison. These images belong to a specific person in different expression that helps to extract the features of face in different ways. The features being different help the recognizer to have most of the expression a human can come up in front of the camera. The number of images here are 11 so all the features of 11 images are stored in arrays.



**Figure 3.5:** Database of Photos stored in the System

### 3.5.3 Connecting the Camera

The camera used in the system can be any USB HD camera that needs to be properly connected with the system for execution. It should be also kept in mind that the megapixel of the camera is good enough to detect a face properly. The kind of camera used in this system is shown in figure 3.6.



**Figure 3.6:** A USB HD Camera



### 3.5.4 Inputting Modules

The next step is to import the modules: CV2, OS, Image and Numphy

- cv2 - For face detection and recognition the OpenCV module and contains the functions.
- os - First, this module is used to extract the image names in the database directory. From these names individual number, which will be used as a label for the face in that image is extracted. This module will be used to maneuver with image and directory names.
- Image – It will use Image module from Python Imaging Library (PIL) to read the image in grayscale format. Since, the dataset images are in gif format and as of now, OpenCV does not support gif format.
- numpy - The images will be stored in numpy arrays.

### 3.5.5 Training Recognizer for Face Detection

First, an image database has been created, containing faces of the users with whom the face on the camera can be matched. In each image, the individual has a different facial expression. For example, there will be several images for the first individual. The database will use these images of each individual to training the recognizer. The algorithm used in this program is Local Binary Patterns Histograms (LBPH). In LBPH each images is analyzed independently. The LBPH method is simpler as it characterizes each image in the dataset locally. When a new unknown image is provided through camera, same analysis is performed on it and compares the result to each of the images stored in the dataset. By characterizing the local patterns in each location in the image and thus it analyzes the image.

### 3.5.6 Applying Local Binary Patterns Histograms (LBPH) Recognizer algorithm

To create the face recognizer object, functions like FaceRecognizer.train trains the recognizer and FaceRecognizer.predict recognizes a face. Here the Local Binary Patterns Histograms Face Recognizer algorithm is used.

The function which prepares the training set has a function called `get_images_and_labels` which takes the absolute path to the image database as input argument and returns tuple of 2 lists, one containing the detected faces and the other containing the corresponding label for that face.

After preparing the training set, the `get_images_and_labels` function with the path of the database directory is passed. This path has to be the absolute path. This function returns the features of the images and labels or captions of the images which will be used to train the face recognizer afterwards. To perform the training the `FaceRecognizer.train` function is used. It requires 2 arguments; the first is the features which in this case are the images of faces and second is the corresponding labels assigned to the images which in this case are the individual number that are extracted from the image names.

### **3.5.7 Extracting Features from the Images**

The first step is to detect the face in each image. As it gets the region of interest (ROI) containing the face in the image, it will use it for training the recognizer. For the purpose of face detection, the Haar Cascade provided by OpenCV is used. The Haar cascades that come with OpenCV are located in the directory of OpenCV installation. For detecting the face, `haarcascade_frontalface_default.xml` is used. The cascade is loaded using the module called `cv2`. Then the function called `CascadeClassifier` function takes the path to the cascade xml file where it is copied in the current working directory, to use the relative path.

### **3.5.8 Displaying Result as Confidence**

As the detection of faces is done accordingly, the result of the detection displays in Python in a way where the confidence of detection is mentioned. The confidence tells how much accurate is the face detected of that human being. The less the confidence found, the more the accuracy is.

# CHAPTER 4

## EXPERIMENTAL RESULTS

### 4.1 Introduction

The experimental result that has been obtained from both the parts i.e. Bluetooth Connection and Face Detection Method is been discussed in the following sections. The results of both the sections gave complete accuracy however many detail conditions have been understood from these result that marks up a lot of changes occurring due to various approaches. The following sections describes what changes and results do the system provides when the experiment is executed.

### 4.2 Establishing Bluetooth Connection

To establish the Bluetooth Connection Properly, the criteria mentioned earlier are to be maintained. If everything is done accurately, the results will be affirmative. The sections below tell the result of those various criteria to establish the Bluetooth Connection.

#### 4.2.1 Bluetooth signal strength

There are two (2) different types of measurements have been done; (i) indoor and, (ii) outdoor with obstacle area and non-obstacle area. As shown in Table 1, it is found that, for indoor (obstacle) the maximum distance that the receiver can detect the RF signal from the Smartphone is around 10 meters and for (non-obstacle) area is 20m. For outdoor (obstacle) the maximum distance that the receiver can detect the RF signal from the Smartphone is around 5 meters and for (non-obstacle) area is 15m.

So it can be said that for non-obstacle area both indoor and outdoor, the system can communicate between input and output at a distance of at least 20 m. Therefore for an obstacle area the effectiveness of the connectivity is reduced up to 20%.

**Table 4.1:** Signal Strength for Different Locations

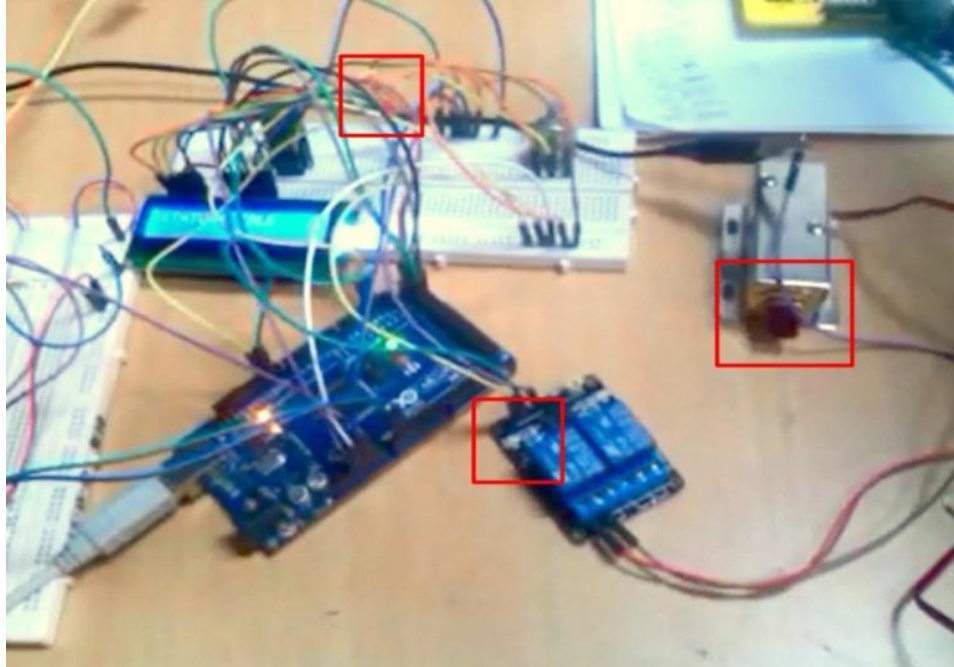
AREA					
INDOOR			OUTDOOR		
Distance	Obstacle	Non-Obstacle	Distance	Obstacle	Non-Obstacle
5m	Connected	Connected	5m	Connected	Connected
10m	Connected	Connected	10m	Disconnected	Connected
15m	Disconnected	Connected	15m	Disconnected	Connected
20m	Disconnected	Connected	20m	Disconnected	Disconnected
25m	Disconnected	Disconnected	25m	Disconnected	Disconnected

#### 4.2.2 Match of Passwords

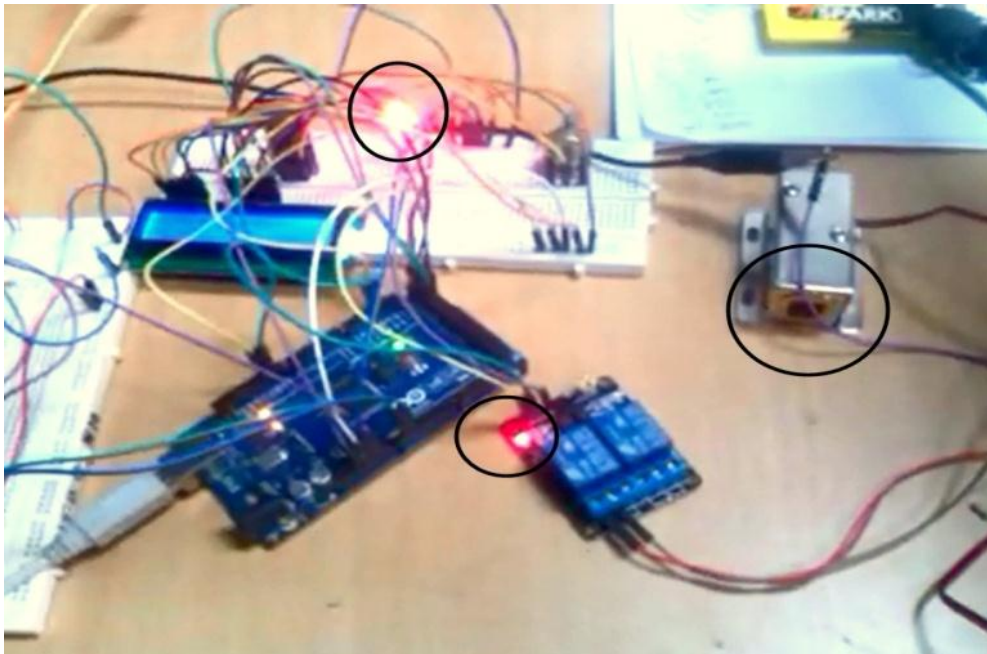
The password provided in the system should be the same that the user has to give in the app so the correct password gave a result of connectivity where as the incorrect password did not connect the Android system with the device leading the user to give the password again. This occurs until and unless the user provides the correct password.

Figure 4.1 shows that the android app is not connected so the lock is closed. It is indicated with red square boxes in the figure where there is no lights turned on making the system inactive to continue any process.

Figure 4.2 shows that the android app is connected with the help of correct password and the door lock opens. This shows that the Bluetooth connection was successful. The black circles in the figure shows the lights in the system in various places like Bluetooth module, relay and Arduino Uno are turned on. Thus the lock is opened being connected with the device.



**Figure 4.1:** The Android app is not connected so the lock is closed



**Figure 4.2:** The Android app is connected and the lock opens

## 4.3 Face Detection

### 4.3.1 Comparison and Result

The testing of face detection is done in a simple manner where the photos of the user are stored in the database. The system when turned on, at first the recognizer should be trained where it can get all the features of the face of the user to compare with those of the photos in database. After the recognizer has been trained, the system can be asked if the face on the camera matches with the photos in database. If the photo matches then the name that is mentioned in the database will show up and if not the system will mention that the face was not recognizable.

Recognizer tests the result by using the images with .sad extension. As done in the `get_images_and_labels` function, all the image names with the .sad extension in the `image_paths` list are appended. It reads in grayscale format and detects faces in it for all images in the list. Having the region of interest (ROI) containing the faces, the ROI is passed to the function called `FaceRecognizer.predict` function which will assign it a label. Here the confidence of the result will also be mentioned. The label is an integer that is one of the individual numbers that is assigned to the faces earlier and it is stored in `nbr_predicted`. The more the value of confidence variable is, the less the recognizer has confidence in the recognition. For example image 11 has confidence of 16.235 and image 10 has confidence of 6.585 so it can be said that image 10 is close to the sample image provided for detection. A confidence value of 0.0 is a perfect recognition.

### 4.3.2 Confidence as Accuracy

In figure 4.3 the python command displays the result after the match of images done by OpenCv. The accuracy of image 8 gives confidence of 0.0 which means that image 8 has a perfect recognition. From this figure it is seen that the Face Recognizer was able to recognize all the faces correctly with an accuracy rate that can determine its confidence.

```
C:\Python27\python.exe
Python 2.7.10 <default, May 23 2015, 09:44:00> [MSC v.1500 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>

User:~/Desktop/FaceRegognition$ facerecognizer.py
Init done
Opengl support available
0 is Correctly Recognized with confidence 25.2105434172
3 is Correctly Recognized with confidence 3.1152210172
7 is Correctly Recognized with confidence 17.6805434170
9 is Correctly Recognized with confidence 50.5104369132
2 is Correctly Recognized with confidence 88.820544172
11 is Correctly Recognized with confidence 16.235434572
2 is Correctly Recognized with confidence 14.654341712
8 is Correctly Recognized with confidence 0.0
6 is Correctly Recognized with confidence 33.8554347297
4 is Correctly Recognized with confidence 2.6585517258
10 is Correctly Recognized with confidence 6.5854341629
1 is Correctly Recognized with confidence 11.98854341784
5 is Correctly Recognized with confidence 18.2054341233
12 is Correctly Recognized with confidence 45.3675434542
User:~/Desktop/FaceRegognition$
```

**Figure 4.3:** Results after matching with accuracy using python

# CHAPTER 5

## CONCLUSION AND FUTURE WORKS

### 5.1 Concluding Remarks

In this thesis, security system based on Bluetooth connectivity and face detection is proposed. The Bluetooth connectivity requires a particular password which is tested with other several passwords and the result comes negative unless that particular password is given. Followed by the establishment of the Bluetooth connection, the face detection part comes where the test result was able to give a complete accuracy. Face detection method was done using Local Binary Patterns Histograms (LBPH) Recognizer algorithm. This algorithm analyzes each face in the training set separately and independently and thus is the optimum choice.

This system is a mass solution to the common masses due to its affordability and lack of training or expertise required so instead of spending money for a costly unit of surveillance this is a better choice. As such, it can be regarded as an elegant as well as a practical solution keeping the budget in mind.

### 5.2 Future Works

The following sections consists potential future directions based on the results of this thesis.

#### 5.2.1 System Connectivity

Bluetooth has been used to connect with the system by using an Android phone but there are many other ways that can help in the system connectivity such as WiFi, ZigBee, Voice and Multimedia etc.

#### 5.2.2 User Detection

The face detection has been done here to detect the user of the system but in future to have more confirmation of the person, other parts of the body can also be detected.



## REFERENCES

- [1] E.Hjelmas, and B.K.Low, "Face detection: A survey", Computer Vision and Image Understanding, Vol. 83, No. 3, Sept. 2001, pp. 236-274
- [2] A.Abdallah, M, Abou El-Nasr, and A. Lynn Abbott, "A New Face Detection Technique using 2D DCT and Self Organizing Feature Map" in Proc. Of World Academy of science, Engineering and technology, Vol. 21, May 2007, pp. 15-19
- [3] J. Nagi, "Design of an Efficient High-speed Face Recognition System", Department of Electrical and Electronics Engineering, College of Engineering, Universiti Tenaga Nasional, March 2007.
- [4] Chi-Chen Raxle Wang and Jenn-Jier James Lien. Adaboost learning for human detection based on histograms of oriented gradients. IEEE Asian Conf. Computer (ACCV), 2007.
- [5] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features", Proceeding of International Conference on The Vision and Pattern Recognition (CVPR), Kauai, HI, USA, 2000.
- [6] N. Anukrishnan, B. Ramya, S. Mohan, Design and development of car ignition access control system based on face recognition technique, SAS TECH Journal 9 (2) (2010)63-70
- [7] Stapathy, A. and Das, D.P., "A system for remote operation of devices: Helpful for elderly and disabled people" in Proc. Of IEEE International Conf on Advanced Electronic Systems, pp. 350-353, 2013.
- [8] Kuang-Yow Lian, Sung-Jung Hsiao and Wen-Tsai Sung, "Home Safety Handwriting Pattern Recognition System" in Proc. of IEEE International Conf on Cognitive Informatics and Cognitive Computing, pp. 477-483, 2012.
- [9] Y. Wang, J. Lu and N. Ge, "Priority-based adaptive frequency hopping for Bluetooth in multi-piconet environments - IEEE Xplore Document", Ieeexplore.ieee.org, 2013.
- [10] Fahim Slauddin and Tarif Riyad Rahman. "A Fuzzy based low-cost monitoring module built with raspberry pi – python – java architecture". International Conference on Smart Sensors and Application (ICSSA), 2015.
- [11] Open Source Computer Vision Library, "The OpenCV Tutorial" [https://docs.opencv.org/2.4/opencv\\_tutorials.pdf](https://docs.opencv.org/2.4/opencv_tutorials.pdf)
- [12] C. L. Huang and S. H. Jeng, "A model-based hand gesture recognition system, "Machine Vision and Appl., vol.12,Pp.243-258, 2001

- [13] J. Davis and M. Shah, "Visual gesture recognition," Proc. Of IEEE on Vision, Image and Signal Processing, Vol.141, pp.101
- [14] A. Carzaniga and A. L. Wolf, "Forwarding in a content-based network," in IN SIGCOMM, 2003, pp. 163-174.
- [15] Bai Yunfei, Wang ping and Sun pan, "The Design and Application of Bluetooth Communications Module," Oigital communication World, 2006, pp. 023-024.
- [16] C. Gomez and J. Paradells, "Wireless Home Automation Networks: A Survey of Architectures and Technologies," IEEE Commun. Mag., vol. 48, no. 6, June 2010, pp. 92–101.
- [17] S. Kamath, "Measuring Bluetooth Low Energy Power Consumption," Application Note AN092 (Texas Instruments), Oct. 2010.
- [18] "The internet engineering task force (ietf)," <http://www.ietf.org/>, accessed: 21 February 2013.
- [19] F. Guidec and Y. Maheo, "Opportunistic content-based dissemination in disconnected mobile ad hoc networks," in Mobile Ubiquitous Computing, Systems, Services and Technologies, 2007. UBIKOM '07. International Conference on, Nov 2007, pp. 49–54.
- [20] D. Johnson, Y. Hu, and D. Maltz, "The dynamic source routing protocol (dsr) for mobile ad hoc networks for ipv4," <http://tools.ietf.org/html/rfc4728>, 2007, accessed: 31 March 2012.
- [21] C. Bisdikian, "An Overview of the Bluetooth Wireless Technology," IEEE Commun. Mag., vol. 39, no. 12, Dec. 2001, pp. 86–94.
- [22] J. Hui and D. Culler, "Extending IP to Low-Power Wireless Personal Area Networks," IEEE Internet Computing, July 2008, pp. 37–45.
- [23] Robert C. Schultz, Robert W. Ives, "Biometric Data Acquisition using MATLAB GUIs," 35th ASEE/IEEE Frontiers in Education Conference, Oct 19-22, 2005.
- [24] Michal Coras, "Perspective Methods of Biometric Human Identification," Inst. of Telecomm., Univ. of Technol. & Life Sci., Bydgoszcz, Poland, 25-27 Sept. 2008.
- [25] "Smart, the next wave of bluetooth," <http://www.abiresearch.com/research/product/1013429-smartthe-next-wave-of-bluetooth/>, accessed: 1 February 2013.
- [26] R. Chand and P. Felber, "A scalable protocol for content-based routing in overlay networks," in Network Computing and Applications, 2003. NCA 2003. Second IEEE International Symposium on, April 2003, pp. 123–130.
- [27] A. Shukla and N. Tyagi, "A new route maintenance in dynamic source routing protocol," in Wireless Pervasive Computing, 2006 1st International Symposium on, Jan. 2006, pp. 4

[28] XianZhong Tian, YongGang Miao and TongSen HU, "Maximum Likelihood Estimation Based on Time Synchronization Algorithm for Wireless Sensor Networks," 2009 ISECS International Colloquium on Computing, Communication, Control, and Management Proceedings Vol.4, 2009

[29] Wang Hong jie, Wang Jin gang. "Realization of drivers for Bluetooth protocol stack based on Windows CE," Electronic Measurement Technology, 2006, pp. 100-101.

[30] CHU Hong yu, "The Design and application of Embedded communication Controller Based on ARM," Control & Automation, 2005, p. 79.