**CSE 360 : Computer Interfacing**

****

**Project Title:**

Smart biometric door-lock system using facial recognition

*Members:*

|  |  |
| --- | --- |
| ID | Name |
| 17101040 | Piash Paul |
| 17101208 | Shovon Bhowmick |
| 17101235 | Mohammed Shehabul Islam |
| 17101196 | Raihan Momtaz |
| 17101164 | Kamrus Salehin |

**Abstract:**

In today’s world, face recognition is an important part for the purpose of security and surveillance. Hence there is a need for an efficient and cost effective system. We are building a system which would unlock a door using facial recognition. There will be a sonar sensor which will turn on the camera whenever it detects an object in front of it. Then if the person’s image is stored in the computer only then it will unlock the door. When there is not sufficient light, the LDR will turn on the bulb and the same procedure is followed. With the use of the Arduino Uno, we aim at making the system cost effective and easy to use, with high performance.

**Introduction:**

Technology is developed and growing with the usage of different equipment. The trend has moved from fingerprint to face recognition. Nowadays, we prefer a face recognition system for unlocking the door. Facial recognition is widely used in various industries and corporate sectors. In the face recognition approach, a given face is compared with the faces stored in the database so as to recognize the individual. The reason is to find a face in the database, which has the most accuracy with the given face. In the field of biometrics, facial recognition innovation is one of the quickest developing fields.

The steps of proposed work as given below:

1. An object is detected in front of the sonar sensor

2. Camera is turned on captures the video

3. Person’s image/video is sent to the computer

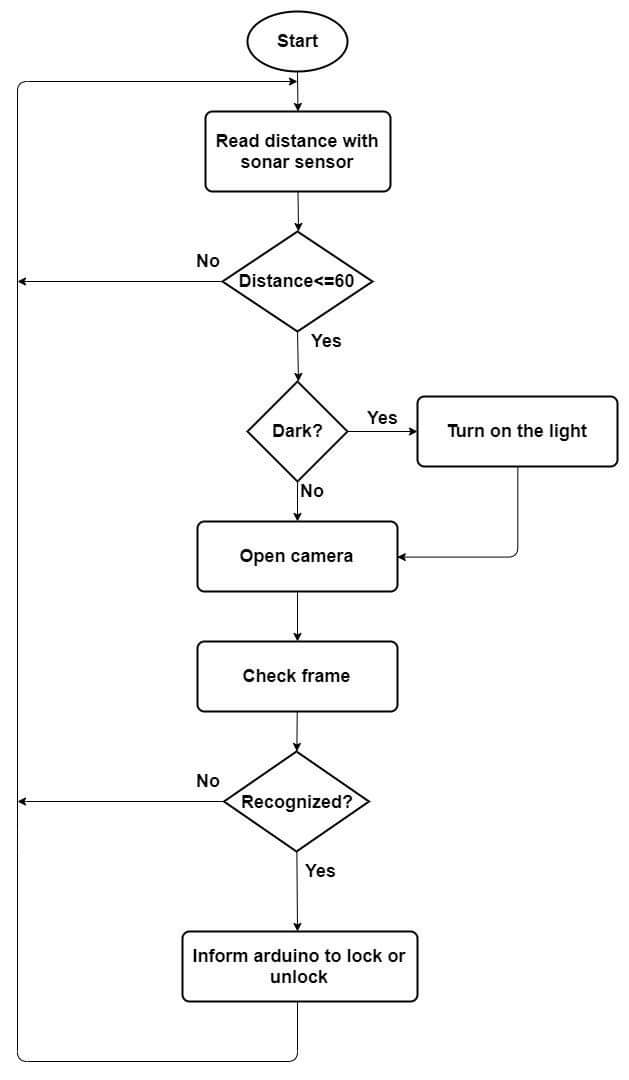
4. Computer checks whether it matches with the saved images and sends signal to Arduino

5. If the image matches, the door unlocks

6. If there is not sufficient light, a bulb is automatically turned on

7. Then the process from step-1 is followed

**FlowChart:**

****

**Literature Review:**

1. Timse,P., Aggarwal, P., Sinha,P., Vora,N., (2014). Face Recognition Based Door Lock System Using Opencv and C# with Remote Access and Security Features. Prathamesh Timse et al Int. Journal of Engineering Research and Applications. 4(4). 52-57. Retrieved from: <https://www.semanticscholar.org/paper/Face-Recognition-Based-Door-Lock-System-Using-and-Timse-Aggarwal/0bc182c11ba3356924513f1bd8a89ba65b121a18>

Title: Face Recognition Based Door Lock System Using Opencv and C# with Remote Access and Security Features

Description: This paper investigates the accuracy and effectiveness of the face detection and reputation algorithms the use of OpenCV and C#. They have used Adaboost algorithm for face detection and PCA for face recognition.

1. Vamsi,T.k., Sai, K.C., Vijayalakshmi M.,(2019).Face recognition based door unlocking system using Raspberry Pi. International Journal of Advance Research, Ideas and Innovations in Technology. 5(2). Retrieved from: <https://www.ijariit.com/manuscripts/v5i2/V5I2-1856.pdf>

Title: Face recognition based door unlocking system using Raspberry Pi

Description: This paper uses face recognition module to capture human images and compares with stored database images. If the image matches with the authorized user, then the system will unlock the door by an electromagnetic lock.

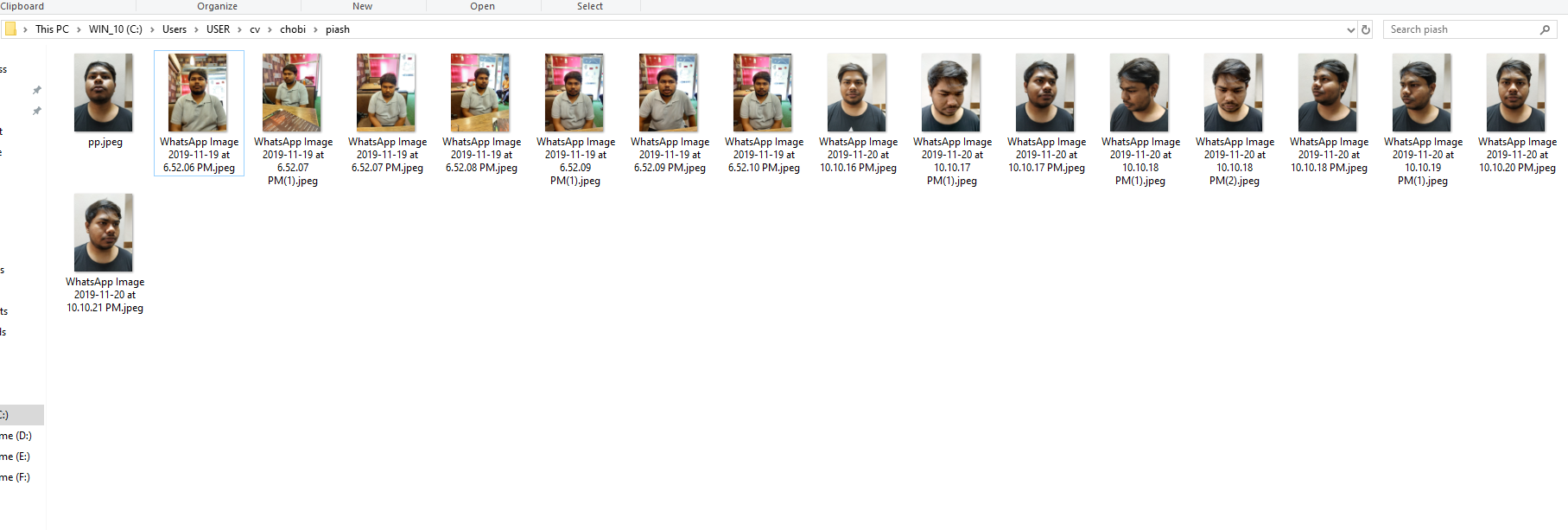
1. Saraf, T., Shukla, K., Balkhande,H., Deshmukh,A., (2018).Automated Door Access Control System Using Face Recognition.International Research Journal of Engineering and Technology (IRJET) .5(4). Retrieved from: <https://www.irjet.net/archives/V5/i4/IRJET-V5I4671.pdf>

Title: Automated Door Access Control System Using Face Recognition.

Description: This project uses a method for automatic door access system using face recognition technique by using python programming and from OpenCV library Haar cascade method. Object Detection using Haar feature-based cascade classifiers is an effective object detection method. Automatic email notification is achieved by sending security alert mail to the users email id.

**Methodology:**

First, we trained the recognizer with the following folder of photos, with help from cascade classifier.



The arduino is constantly taking data from the sonar sensor, to know if someone is present in front of the camera. If at any time, it detects that some object is present, it sends a notification through the serial cable to the computer, where a python script is continuously checking for a data. If the script gets the data it was waiting for, it turns on the camera that is connected to a USB port; and captures a frame. Then it turns the frame into an N-dimensional array, with the help of the numpy library. Then the script matches the new array with the old arrays it got from the training script, and if it can find a match, it sends a signal through the serial cable to the arduino, and the arduino then uses the motor driver to lock or unlock the latch. If there were no match, the whole process is repeated.The arduino is also taking values from the LDR. It has a variable called “state” that keeps track of whether it’s dark outside or lit. If at any given time, it finds that the state is dark and there is someone in front of the camera, the relay is used to turn on the light bulb, which helps the camera to get a better frame. There is also a push switch connected to the arduino, which sends a value “0” if it’s been pressed. The switch is used to control the latch from the inner side from the house. The confidence level used by the python script to make sure about the match is limited, to avoid photos. Due to being still, the photos tend to get a confidence level over 100. Which will be ignored by the script, thus making sure that the system cannot be cheated and it is secured.

**Connections:**

The arduino is connected to the computer with a USB A to a USB B cable, which establishes the serial communication between computer and arduino. A sonar sensor, HC-SR04 is connected to pins 12 and 13 of the arduino. This will help us know if there is any person or object in front of the camera, thus the camera can only be turned on when there is something to scan and save a lot of energy by being turned off the rest of the time. A photoresistor is connected to analog pin A1, through a resistor, which will give us an analog value on the darkness. This will help us to turn on a light if there isn’t enough natural light for the camera to capture proper frames. A relay is connected to pin 8, which will be used to complete the circuit of a light bulb, when needed. A push switch is connected to arduino pin 7, through a resistor, to control the lock manually from the inside. A motor driver is connected to pins 2,3 and 4, which controls the locking mechanism. Finally, a webcam is connected to the computer to capture frames when there’s a person.

**Codes:**

Python Script For Recognizer:

import numpy as np

import cv2

import serial

import time

face\_cascade=cv2.CascadeClassifier('src/cascades/data/haarcascade\_frontalface\_alt2.xml')

recognizer= cv2.face.LBPHFaceRecognizer\_create()

recognizer.read("trainner.yml")

st=0

sx= serial.Serial('COM4',9600,timeout=1)

x=sx.readline().strip()

x=x.decode()

while(True):

x=sx.readline().strip()

x=x.decode()

if x=='5':

cap = cv2.VideoCapture(0)

while (True):

ret, frame= cap.read()

gray= cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

faces=face\_cascade.detectMultiScale(gray, scaleFactor=1.5, minNeighbors=5)

for(x, y, w, h) in faces:

#print(x,y,w,h)

roi\_gray= gray[y:y+h, x:x+w]

roi\_color= gray[y:y+h, x:x+w]

id\_, conf = recognizer.predict(roi\_gray)

if conf >70 and id\_ is 1 and conf <85:

print(id\_,conf)

sx.write('s'.encode())

t = time.localtime()

hour = time.strftime("%H", t)

m = time.strftime("%M", t)

sec = time.strftime("%S", t)

name='C:/Users/USER/cv/people/'

name=name+hour+'-'+m+'-'+sec

name=name+'.png'

cv2.imwrite(name,frame)

st=1

break

if st==1:

st=0

cap.release()

cv2.destroyAllWindows()

Break

Python Script For Training:

import os

import cv2

import numpy as np

from PIL import Image

import pickle

BASE\_DIR= os.path.dirname(os.path.abspath(\_\_file\_\_))

im\_dir=os.path.join(BASE\_DIR, "chobi")

face\_cascade=cv2.CascadeClassifier('src/cascades/data/haarcascade\_frontalface\_alt2.xml')

recognizer= cv2.face.LBPHFaceRecognizer\_create()

x\_t=[]

y\_l=[]

current\_id= 0

label\_ids= {}

for root, dirs, files in os.walk(im\_dir):

for file in files:

if file.endswith("png") or file.endswith("jpg") or file.endswith("jpeg"):

path= os.path.join(root, file)

label= os.path.basename(os.path.dirname(path))

print(label)

if not label in label\_ids:

label\_ids[label]= current\_id

current\_id+=1

id\_=label\_ids[label]

print(label\_ids)

pil\_image= Image.open(path).convert("L")

image\_array= np.array(pil\_image, "uint8")

#print (image\_array)

faces= face\_cascade.detectMultiScale(image\_array, scaleFactor=1.5, minNeighbors=5)

for (x,y,w,h) in faces:

roi= image\_array[y:y+h, x:x+w]

x\_t.append(roi)

y\_l.append(id\_)

with open("labels.pickle", 'wb') as f:

pickle.dump(label\_ids, f)

recognizer.train(x\_t, np.array(y\_l))

recognizer.save("trainner.yml")

C Script For Arduino:

const int trigPin = 12;

const int echoPin = 13;

int l=1;

int ldrPin = A1;

// defines variables

long duration;

String state;

int co=0;

int coo=0;

void setup() {

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

Serial.begin(9600);

pinMode(4,OUTPUT);

pinMode(2,OUTPUT);

pinMode(3,OUTPUT);

pinMode (7,INPUT);

pinMode(8,OUTPUT);

}

void loop() {

if((co>=5&&state=="light")||coo>=10){

Loff();

co=0;

coo=0;

}

int a=analogRead(ldrPin);

if(a>=1019){

state= "dark";

}

else{

state= "light";

}

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);

int dis= duration\*0.034/2;

if(dis<60){

for(int i=0;i<=5;i++)

Serial.println(5);

delay(100);

if(state=="dark"){

Lon();

}

}

else if(dis>100){

coo++;

}

else{

co++;

//Serial.println(co);

}

delay(500);

if(Serial.available()>0){

if(Serial.read()=='s'){

yes();

}

}

int b= digitalRead(7);

if(b==0){

yes();

}

}

void yes(){

if(l==0){

unlock();

l=1;

}

else if(l==1){

lock();

l=0;

}

Loff();

}

void lock(){

digitalWrite(4,LOW);

digitalWrite(3,HIGH);

digitalWrite(2,HIGH);

delay(4000);

digitalWrite(4,LOW);

digitalWrite(3,LOW);

}

void unlock(){

digitalWrite(3,LOW);

digitalWrite(4,HIGH);

digitalWrite(2,HIGH);

delay(4000);

digitalWrite(4,LOW);

digitalWrite(3,LOW);

}

void Lon(){

digitalWrite(8,HIGH);

delay(500);

}

void Loff(){

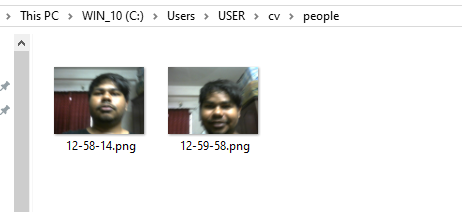
digitalWrite(8,LOW);

delay(500);

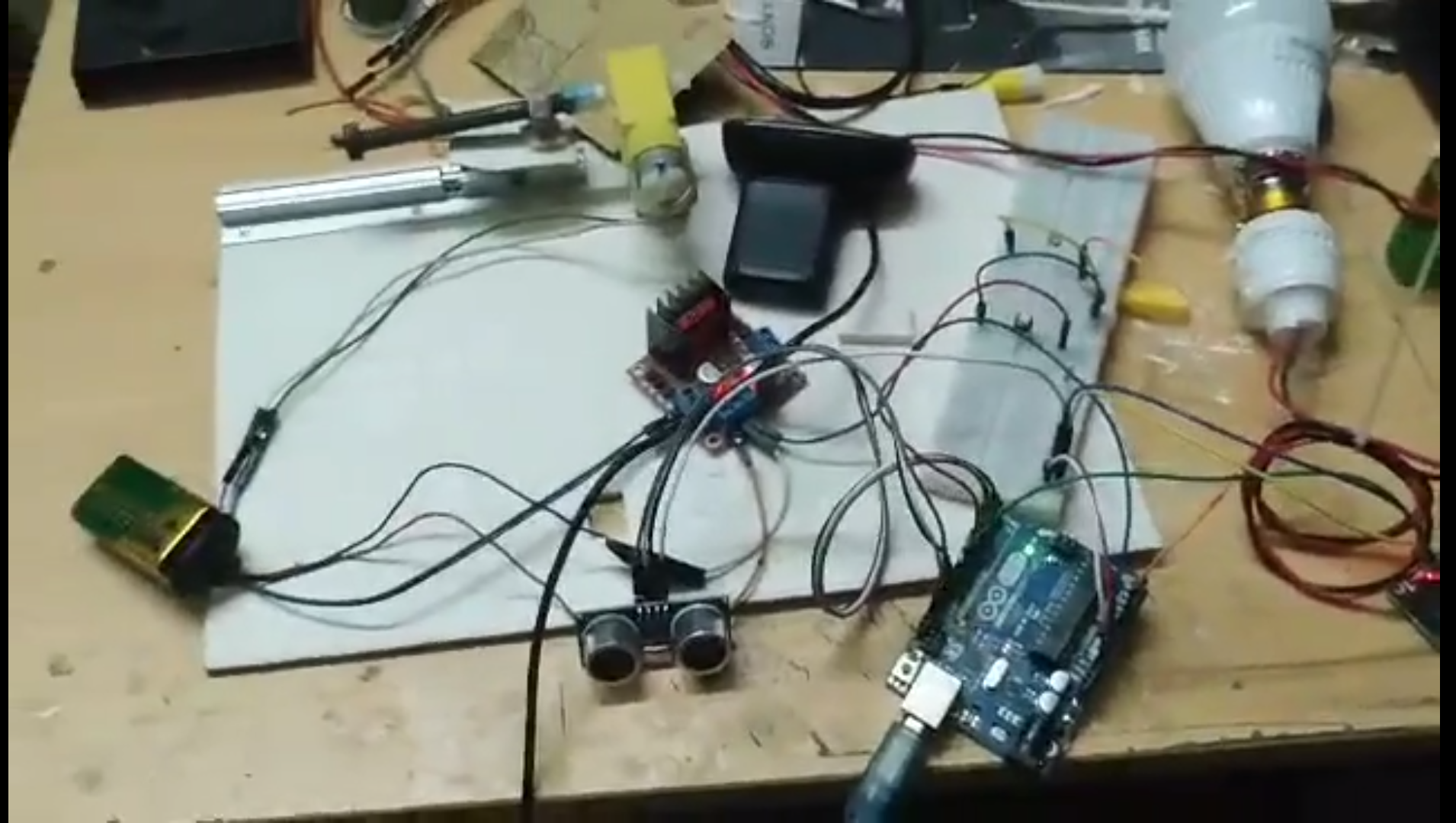
}

**Experimental Result:**

The photo of every person that locks or unlocks the door using the facial recognition system will be saved in a folder for later checking and tracking. The photo is saved with a timestamp, hour, minute and second. From the following photo, we can see that the door was locked at 12:58:14 and unlocked at 12:59:58.



**Experiment Setup:**



|  |  |
| --- | --- |
| **Equipment** | |
| Arduino UNO R3 |  |
| Webcam/Camera |  |
| Sonar Sensor |  |
| L928N Stepper Motor Driver |  |
| Push switch |  |
| LDR 5mm |  |
| Gear Motor |  |
| Relay |  |

**Equipment Cost Analysis:**

|  |  |  |
| --- | --- | --- |
| **No.** | **Name** | **Price** |
| **1** | Arduino UNO R3 | 420 tk |
| **2** | Webcam/Camera | 2500 tk |
| **3** | Sonar Sensor | 100 tk |
| **4** | Push switch | 5 tk |
| **5** | L928N Stepper Motor Driver | 250 tk |
| **6** | LDR 5mm | 5tk |
| **7** | Gear Motor | 5tk |
| 8. | Relay | 85tk |

**Future Work:**

As convenient as our facial recognition system is, there are still sides to work on and improve. And we have already made some prototypes for the upgrades that needs a little tweaking to make the system even better. First of all, with a cable, there is too much hassle to management. We are planning to introduce a server that will be powered with a raspberry pi. We tried pickle with python and also a php server, and we are still trying to decide which one can make it more convenient and simple. Our idea is to wirelessly send the frames to the main computer from the raspberry pi. Even though raspberry pi is a powerful computer on its own, there is no denying that a complete cisc computer will obviously offer a better result for our system. We are also planning to introduce deep learning into the system. The OpenCV library does give us a huge benefit with the simplicity, but the deep learning system will help us make our system work even better and more confidently. There has also been an idea floating to include a door alignment sensor that will detect break-ins and notify the user through a text, using a GSM module. We are planning to connect a small battery with the system that will help the system work even when there is a power outage.

**Conclusion:**

The purpose of the facial recognition system was to make life easier and more secure. Which it does perfectly. It provides us hands-free and secured door-lock system that is hard to cheat and easier to use. Due to being only active when user is present, our system also becomes energy efficient. The use of light bulb and LDR, makes it easier and more convenient. By properly utilizing the confidence, we were able to achieve the ability to detect when someone is trying to cheat the system with a photo. Lastly, due to having all it processing done in a distant server or computer, it doesn’t need any extra processing unit, which brings the cost to a very low price while also bumping up the efficiency. The future upgrade of gsm module and cloud computing is going to help us achieve goals we didn’t even expect for. With proper planning and use of resources, it could be possible to produce and provide this product in our country for everyone.