

SMART DOOR LOCK SYSTEM USING FACIAL RECOGNITION

by

N Vedanjali-19BEE1159

Mahima Rajesh-19BEE1188

Pappula Rajasri- 19BEE1118

A project report submitted to

VAITHILINGAM C.

SCHOOL OF ELECTRICAL AND ELECTRONICS ENGINEERING

in partial fulfilment of the requirements for the course of

ECE3501– IOT FUNDAMENTALS

in

B.Tech. ELECTRICAL AND ELECTRONICS ENGINEERING



VIT CHENNAI

Vandalur – Kelambakkam Road

Chennai – 600127

DECEMBER 2021

INDEX:

S. No	Description	Page No.
1	Bonafide Certificate	3
2	Acknowledgements	4
3	Abstract	5
4	Introduction	6
5	Hardware	7
	5.1 Hardware Components	7
	5.2 Hardware Assembly and Configuration	7
	5.3 Connections	8
6	Block Diagram	8
	6.1 Working Procedure	9
7	Results and Output	10
8	Conclusion	13
9	Future Scope and Applications	13
10	Video Link of the Project	13
11	Appendix	14
	11.1 Arduino Code	14
	11.2 Python Code for collecting data of face	15
	11.3 Python Code for facial recognition	16
12	References	21

1. BONAFIDE CERTIFICATE

Certified that this project report entitled “**SMART DOOR LOCK SYSTEM USING FACIAL RECOGNITION**” is a bonafide work of **N VEDANJALI** (19BEE1159), **MAHIMA RAJESH** (19BEE1188) and **PAPPULA RAJASRI** (19BEE1118) who carried out the Project work under my supervision and guidance.

DR. VAITHILINGAM C

Associate Professor

School of Electrical and Electronics Engineering (SELECT),

VIT University, Chennai

Chennai – 600127.

2. ACKNOWLEDGEMENT

We express our thanks to our Programme Chair **Dr. Chendur Kumaran R** for his support throughout the course of this project.

We wish to express our sincere thanks and deep sense of gratitude to our project guide, **Dr. Vaithilingam C**, School of Electrical and Electronics Engineering, for her consistent encouragement and valuable guidance offered to us in a pleasant manner throughout the course of the project work.

We also take this opportunity to thank all the faculty of the school for their support and their wisdom imparted to us throughout the course.

We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

N VEDANJALI

MAHIMA RAJESH

PAPPULA RAJASRI

3. ABSTRACT

A smart door lock system using facial recognition is safer, more efficient than traditional locks. In contrast to traditional locks, this innovative product can identify and decide on its own whether or not you are authorized to enter your house or office. The benefits of using this particular product outweigh the downsides due to its effectiveness in protecting people from human mistakes. Only those faces that are registered under the system can open the lock and enter into the premises, while all others will be blocked for unauthorized entry.

This project is a smart IoT based face recognition system which firstly recognizes the face of the person near the door and then compares with the already uploaded faces stored in the data base of the system. If a person is detected then the door would open and welcomes them. If an unknown person enters, the owner would be alerted by message.

Our aim is to build a cost-effective smart door lock system using facial recognition which can be used to provide touch less access systems and increase the security of the building. With COVID-19 protocols being enforced in every public place today, this model can be used in verification of identity.

4. INTRODUCTION

A webcam, arduino and a servo motor are used in this project to implement the facial recognition process. This model is capable of performing all the facial recognition stages such as face detection, features extraction and face recognition using OpenCV libraries. OpenCV is an image and video processing library and is highly used for image and video analysis and face recognition. Open CV uses machine learning algorithms to search for faces within a picture. Further, Python is used to control the Arduino with the combination of Open CV.

We have used a python code to start the webcam and obtain the picture, if the picture matches with the database up to 83% a character is sent to the Arduino. The Arduino then moves the servo motor for few seconds to open the door.

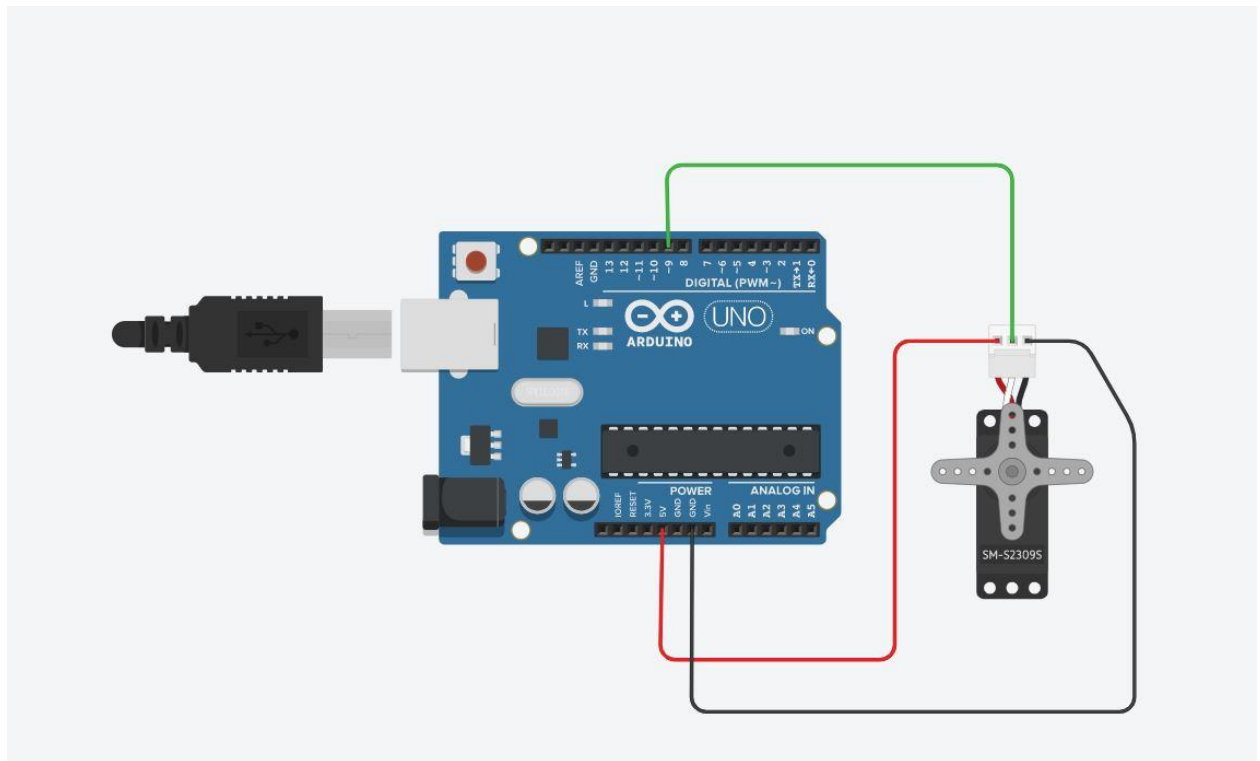
C language is used to code the Arduino which is responsible for the movement of the door as per the requirement. Overall, our aim is to create a smart automation system based on IoT using a python code which can be used in both home and public spaces for safety purposes.

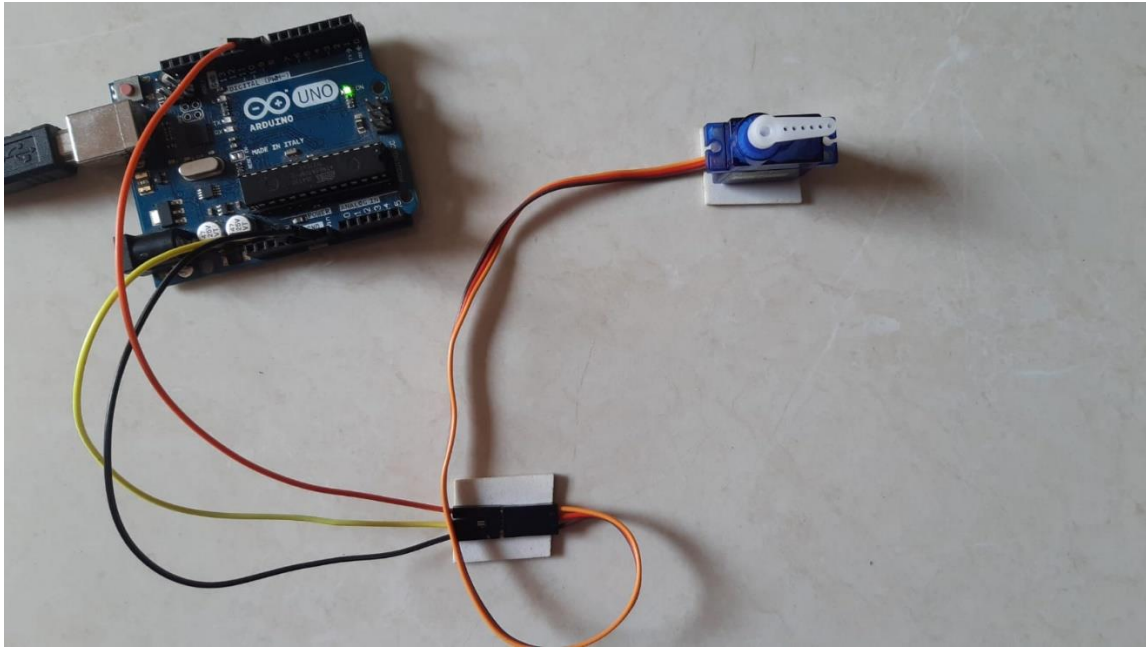
5. HARDWARE

5.1 HARDWARE COMPONENTS

Component Name	Specification
Arduino Uno	R3 CH340G ATmega328p
Servo Motor	SM-S2309S
Jumper wires	-

5.2 HARDWARE ASSEMBLY AND CONFIGURATION

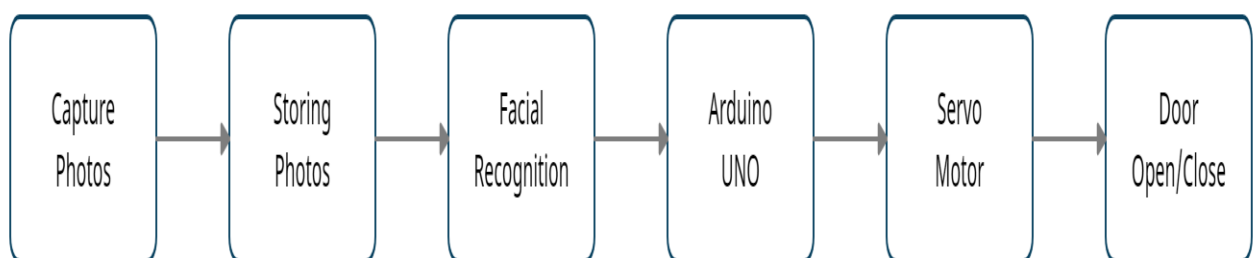




5.3 CONNECTIONS

- The Servo Motor Signal Pin will be Connected to the arduino pin number 9.
- The 5V pin will be connected to Arduino 5V pin and Ground pin will be connected to Arduino Ground pin.
- Arduino must be connected to PC through Arduino Cable.

6. BLOCK DIAGRAM

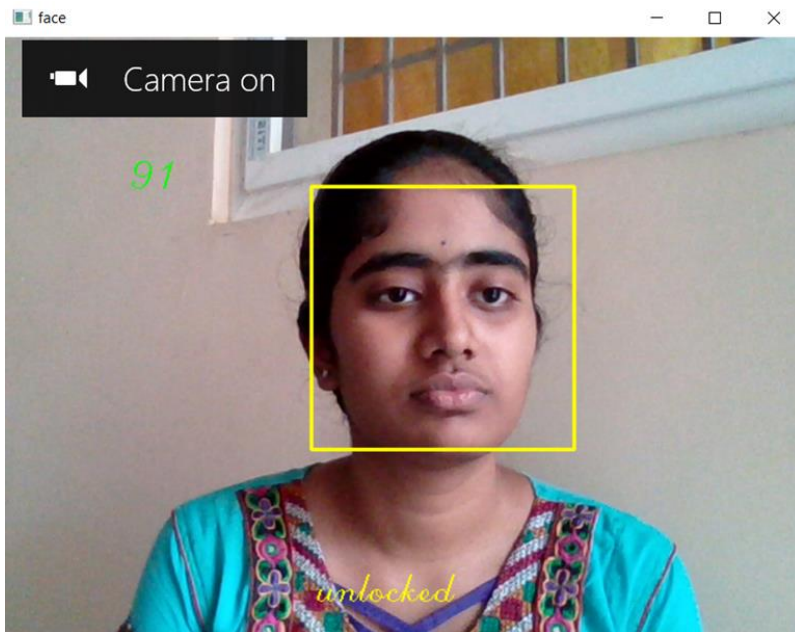


6.1 WORKING PROCEDURE

- At first the python code will be written on any of the python IDE for collecting sample pictures and for facial recognition.
- Connect the Arduino Uno with the servo motor as specified above.
- Then Arduino code will be uploaded to Arduino Uno board using Arduino IDE
- Later run the 1st python code for collecting the sample pictures of user face.
- Then run the 2nd python code and it will start the webcam and take pictures,
 - If the picture match with database up to 83% then it will display “unlocked” and door opens for 5 seconds with audio message of “Face recognition complete. It is matching with database welcome, mam. Door is opening for 5 seconds.”, and after 5 seconds the door closes with audio message of “Door is closing”. This happens only when “a” character is sent to the Arduino.
 - If the picture does not match with the data-base then it will display “locked” and the door doesn’t open but one can hear audio message “Face is not matching please try again”.
 - If the face is not found in the webcam then it displays “face not found” and the door doesn’t open but one can hear audio message of “Face not found please try again”.

7. RESULTS AND OUTPUTS

Face recognized:

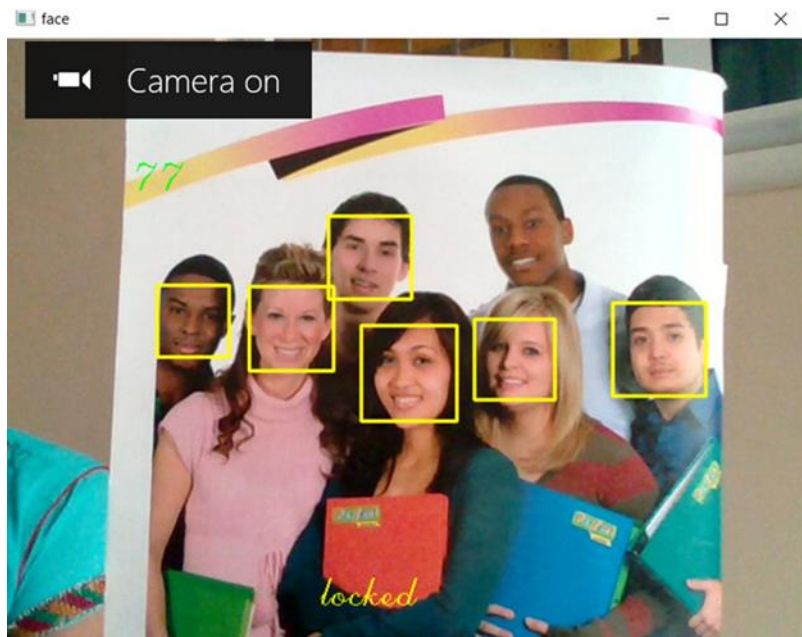


Door opens:



Audio message: Face recognition complete. It is matching with database. Welcome, mam door is opening for 5 seconds. Door is closing.

Face not recognized:



Door is closed:



Audio message: Face is not matching please try again.

Face not found:



Door is closed:



Audio message: Face not found please try again.

8. CONCLUSION

Smart door lock system are very useful for the houses which gives out an immense security to the house. Here it is done by facial recognition which will be even more helpful as it is not even required to touch by hands. These smart door lock systems using facial recognition is being even more useful in this pandemic situation. To be noted point that these are very less in usage around in many areas.

This project has been done using the python code that includes open cv and nymph libraries for storing of photos that are to be captured by the user and then a python code is done using pytsx3, open cv, nymph libraries for recognition of face and to unlock or lock the door via display and voice control. A code is also done for opening and closing of servo motor using the arduino.

This is a cost-effective smart door lock system using facial recognition which can be used to provide touch less access systems and increase the security of the building. It can be used in offices, banks and institutions for verification of identity.

9. FUTURE SCOPE AND APPLICATIONS

With the COVID-19 protocols in place, this facial recognition system can be used in offices, banks and institutions for verification of identity.

The system can also be very effective in preventing theft of valuables. It can be programmed to recognize the items you never want anyone to steal and so determine who deserves access and who doesn't.

Features like face mask detection of the person entering the premises can also be added to this project. In colleges and universities this project can combined with a security check to detect the if face mask is placed accurately or not.

We can further attach a WIFI module which will enable multiple devices live streaming.

10.VIDEO LINK OF THE PROJECT

<https://drive.google.com/file/d/1wkm-w5p-lcVi7nsgfxdjVRsKTW72r77X/view>

11. APPENDIX

11.1 ARDUINO CODE

```
#include<Servo.h>

Servo myservo;

char d;

int pos;

void setup() {

  Serial.begin(9600);

  pinMode(5,OUTPUT);

  myservo.attach(9);

  myservo.write(0);

}

void loop() {

  if(Serial.available())

  {

    d=Serial.read();

  }

  if(d=='a')

  {

    Serial.print(d);

    delay(300);

    for(pos=0;pos<=90;pos+=5)

    { myservo.write(pos);

      delay(20);

    }

  }

}
```

```

delay(5000);

for(pos=90;pos>=0;pos-=5)
{
myservo.write(pos);
delay(20);
}
}

d;
}

```

11.2 PYTHON CODE FOR COLLECTING THE DATA OF FACE:

```

import cv2

import numpy as np

face_classifier = cv2.CascadeClassifier('C:/Users/Girish Sai N/Desktop/FACELOCKING-DOOR-USING-PYTHON-AND-ARDUINO-PROGRAMING-master/requirements/haarcascade_frontalface_default.xml')

def face_extractor(img):

    gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)

    faces = face_classifier.detectMultiScale(gray,1.3,5)

    if faces is ():

        return None

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

        cropped_face = img[y:y+h,x:x+w]

    return cropped_face

cap = cv2.VideoCapture(0)

count = 0

while True:

    ret,frame = cap.read()

```

```

    if face_extractor(frame) is not None:
        count += 1
        face = cv2.resize(face_extractor(frame),(200,200))
        face = cv2.cvtColor(face,cv2.COLOR_BGR2GRAY)
        file_name_path = 'C:/Users/Girish Sai N/Desktop/Training_pictures/'+str(count)+'.jpg'
cv2.imwrite(file_name_path,face)

        cv2.putText(face,str(count),(50,50),cv2.FONT_HERSHEY_COMPLEX,1,(0,255,0),2)
        cv2.imshow("face cropper",face)
    else:
        print('face not found')
        pass
    if cv2.waitKey(1)==13 or count==500:
        break
cap.release()
cv2.destroyAllWindows()
print("collecting samples complete")

```

11.3 PYTHON CODE FOR FACIAL REGONITION

```

import cv2
import numpy as np
from os import listdir
from os.path import isfile,join
import serial
import time
import pyttsx3
q=1
x=0

```



```

c=0
m=0
d=0
while q<=2:
    data_path = 'C:/Users/Girish Sai N/Desktop/Training_pictures/'
    onlyfiles = [f for f in listdir(data_path) if isfile(join(data_path,f))]
    Training_data, Lebel = [],[]
    for i , files in enumerate(onlyfiles):
        image_path = data_path + onlyfiles[i]
        images = cv2.imread(image_path,cv2.IMREAD_GRAYSCALE)
        Training_data.append(np.asarray(images, dtype = np.uint8))
        Lebel.append(i)

    Lebel = np.asarray(Lebel, dtype = np.int32)
    model = cv2.face.LBPHFaceRecognizer_create()
    model.train(np.asarray(Training_data),np.asarray(Lebel))
    print("training complete")

    q+=1

face_classifier = cv2.CascadeClassifier('C:/Users/Girish Sai N/Desktop/FACELOCKING-
DOOR-USING-PYTHON-AND-ARDUINO-PROGRAMING-
master/requirements/haarcascade_frontalface_default.xml')

def speak(audio):
    engine.say(audio)
    engine.runAndWait()

engine = pyttsx3.init('sapi5')
voices=engine.getProperty('voices')
engine.setProperty("voice",voices[0].id)
engine.setProperty("rate",140)

```

```

engine.setProperty("volume",1000)

def face_detector(img, size= 0.5):

    gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)

    faces = face_classifier.detectMultiScale(gray,1.3,5)

    if faces ==():

        return img,[]

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

        cv2.rectangle(img, (x,y),(x+w,y+h),(0,255,255),2)

        roi = img[y:y+h, x:x+w]

        roi = cv2.resize(roi,(200,200))


    return img,roi


cap = cv2.VideoCapture(0)

while True:

    ret, frame = cap.read()


    image, face = face_detector(frame)


    try:

        face = cv2.cvtColor(face,cv2.COLOR_BGR2GRAY)

        result= model.predict(face)

        if result[1]<500:


            confidence = int((1-(result[1])/300)*100)

            display_string = str(confidence)

```

```

        cv2.putText(image,
display_string,(100,120),cv2.FONT_HERSHEY_SCRIPT_COMPLEX,1,(0,255,0))

if confidence>=83:

    cv2.putText(image,"unlocked",(250,450),cv2.FONT_HERSHEY_SCRIPT_COMPLE
X,1,(0,255,255))

    cv2.imshow('face',image)

    x+=1

else:

    cv2.putText(image,"locked",(250,450),cv2.FONT_HERSHEY_SCRIPT_COMPLEX,
1,(0,255,255))

    cv2.imshow('face',image)

    c+=1

except:

    cv2.putText(image,"Face not
found",(250,450),cv2.FONT_HERSHEY_SCRIPT_COMPLEX,1,(0,255,255))

    cv2.imshow('face',image)

    d+=1

    pass


if cv2.waitKey(1)==13 or x==10 or c==30 or d==20:

    break

cap.release()

cv2.destroyAllWindows()

if x>=5:

    m=1

    ard = serial.Serial('com4' ,9600)

    time.sleep(2)

    var = 'a'

```

```
c=var.encode()

speak("Face recognition complete..it is matching with database...welcome..mam....Door is
opening for 5 seconds")

ard.write(c)

time.sleep(4)

elif c==30:

    speak("face is not matching..please try again")

elif d==20:

    speak("face is not found please try again ")

if m==1:

    speak("door is closing")
```

12. REFERENCES

1. Jayakumar A.J.K., Muthulakshmi S. (2018) Raspberry Pi-Based Surveillance System with IoT. In: Thalmann D., Subhashini N., Mohanaprasad K., Murugan M. (eds) Intelligent Embedded Systems. Lecture Notes in Electrical Engineering, vol 492. Springer, Singapore.
2. P. Bhatia, S. Rajput, S. Pathak and S. Prasad, "IOT based facial recognition system for home security using LBPH algorithm," *2018 3rd International Conference on Inventive Computation Technologies (ICICT)*, 2018, pp. 191-193, doi: 10.1109/ICICT43934.2018.9034420.
3. R., Manoj & Biradar, Rekha & R., Raju & A., Sharad. (2020). Smart Home Security System using Iot, Face Recognition and Raspberry Pi. *International Journal of Computer Applications*. 176. 45-47. 10.5120/ijca2020920105.
4. P. B. Balla and K. T. Jadhao, "IoT Based Facial Recognition Security System," *2018 International Conference on Smart City and Emerging Technology (ICSCET)*, 2018, pp. 1-4, doi: 10.1109/ICSCET.2018.8537344.
5. C. M. Blanco Muñoz, F. Gómez Cruz and J. S. Jimenez Valero, "Software architecture for the application of facial recognition techniques through IoT devices," *2020 Congreso Internacional de Innovación y Tendencias en Ingeniería (CONIITI)*, 2020, pp. 1-5, doi: 10.1109/CONIITI51147.2020.9240416.