**Project Report**

**CSE 3200: System Development Project**

**Face Recognition Home Security System**

****

**Project Team**

**Sourav Mandol (Roll - 1607044)**

**Sumon Mia (Roll - 1607045)**

**Supervised by**

**Dr. Sk. Mohammad Masudul Ahsan**

**Professor**

**Dept. of Computer Science and Engineering**

**Khulna University of Engineering & Technology**

**INTRODUCTION**

**Background:**

Face recognition is a biometric technology that goes beyond just detecting human faces in an image or video. It goes a bit further to determine whose face it is. A face recognition system works by taking an image of a face and predicting whether the face matches another face stored in a dataset (or whether a face in one image matches a face in another).

Face Recognition home security system is a security system that detects a face whether the guy is known or unknown. After detecting the person, it will open the door or unlock any security.

**PURPOSE**

**High Accuracy Rates:**

Most face recognition software being used in smart home locks can accurately determine whether the person trying to gain access to a home matches the database of people who have been authorized to enter. That said, some software programs are more accurate than others, especially when it comes to detecting faces from various angles and recognizing faces of different nationalities.

**Automation:**

Once you’ve set up your smart home security system and granted access to the people you want to allow into your home, the system will automatically let those people in when it detects a match. No need for you to answer the doorbell, or to respond to a request to enter.

Facial recognition technology is here to stay, and will only get better with time and also better from biometric verification.

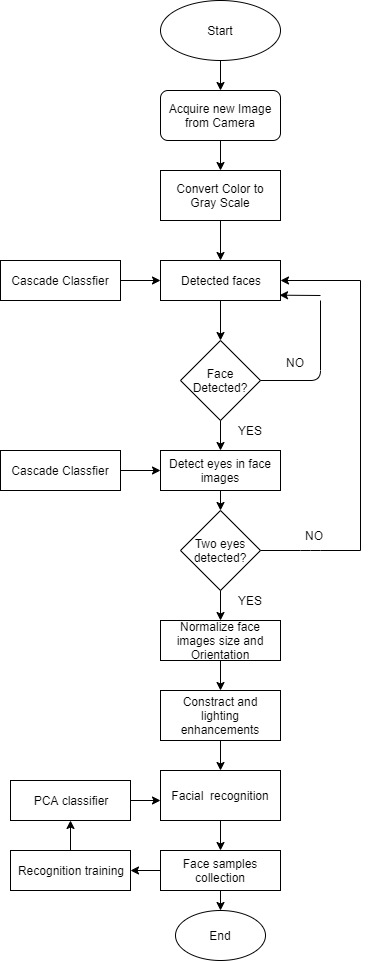
**Faster processing:**

The process of recognizing a face takes a second or less — and this is incredibly beneficial for the companies. In the era of constant cyber attacks and advanced hacking tools, companies need a technology that would be both secure and fast.

**OBJECTIVES**

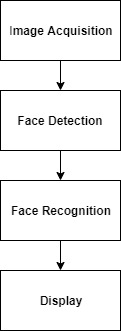
1. To developing an locking system based on face recognition
2. Security
3. IOT based project
4. User Friendly Interface connecting with Security
5. Detect as many faces as possible in the training images
6. Minimum detection of non-process and Multiple detects
7. To Digitalize Home and Office.
8. Image Acquisition

**Design (Data Flow Diagram)**

****

**Fig(1:) Flow Chart of Face Detection Home Security System**

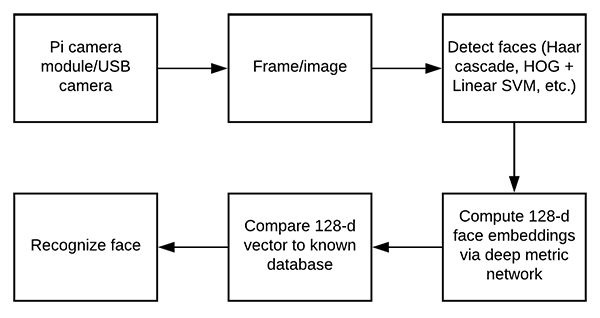
**METHODOLOGY**



**Fig(2): Block Diagram of Methodology of Face Recognition**

In this project we need four types of files for detecting and recognizing face from our dataset .These are the files-

1. encode\_faces.py : This file will find faces in our dataset and encode them into 128-d vectors.
2. encodings.pickle : Our face encodings (128-d vectors, one for each face) are stored in this pickle file.
3. haarcascade\_frontalface\_default.xml : In order to detect and localize faces in frames we rely on OpenCV’s pre-trained Haar cascade file.
4. pi\_face\_recognition.py : This is our main execution script. We’re going to review it later in this post so you understand the code and what’s **going on under the hood. From there feel free to hack it up for your own project purposes.**

****

**Fig(3):Block diagram of computing face recognition embedding**

Face detection performance is a key issue, so techniques for dealing with non‐frontal face detection are discussed. Subspace modeling and learning‐based dimension reduction methods are fundamental to many current face recognition techniques. Face recognition has merits of both high accuracy and low intrusive, so it has drawn the attention of the researches in various fields from psychology, image processing to computer vision.

**Implementation**

For implementing this project we need some equipments.These are given below:

*Hardware*   *Software*

1)Anaconda Environment

2)Pyqt5 (designer)

3)python IDLE

4)Open\_CV

5)Tesnsor Flow

6)Skew Learn library

7)Dependencies libraries for Face\_recognition, camera module,CNN.

1. Web cam/pi camera
2. Mouse and keyboard joined with

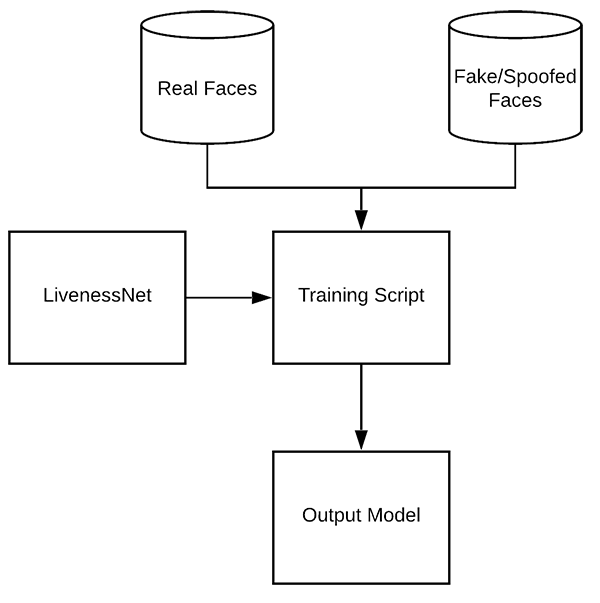
the respected operating system

Dataset Training Method: we have used a deep neural network to compute a 128-d vector that will quantify each face in the dataset .From a fixed path(for our project it is face\_data/member name) we gather images and then we’ll detect vector points and stored them in encodings.pickle file.

After training our machine successfully we have used these points for detecting our known faces.

Face Detecting Method: In order to detect and localize faces in frames we have relied on OpenCv’s pre trained Haar Cascase file located in our directoty file.

Liveness detecting Method: First of all we will input an image or video to our project pipeline . Then we have detected the liveness of our input from our own train CNN. If the input image in real then the project will try to recognize the face from our own made dataset.

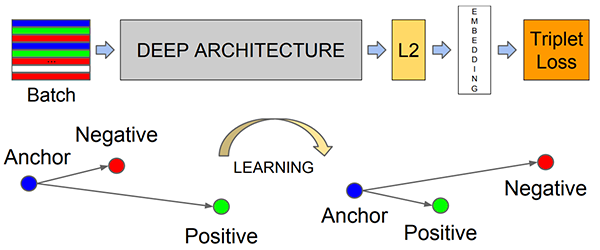


**Fig(4):Block diagram of detecting liveness**

Summary of implementation:

We have applied haar cascade to find/detect the face object and applied face detection to recognize face in the image after detecting the liveness of the images. Optionally, we can compute facial landmarks, enabling us to preprocess and align the face.

Face alignment, as the name suggests, is the process of (1) identifying the geometric structure of the faces and (2) attempting to obtain a canonical alignment of the face based on translation, rotation, and scale.

While optional, face alignment has been demonstrated to increase face recognition accuracy in some pipelines.

**Fig(5): How the deep learning face recognition model computes the face embedding.**

We apply here deep learning and OpenCV together:

i. Detect faces

ii. Compute 128-d face embeddings to quantify a face

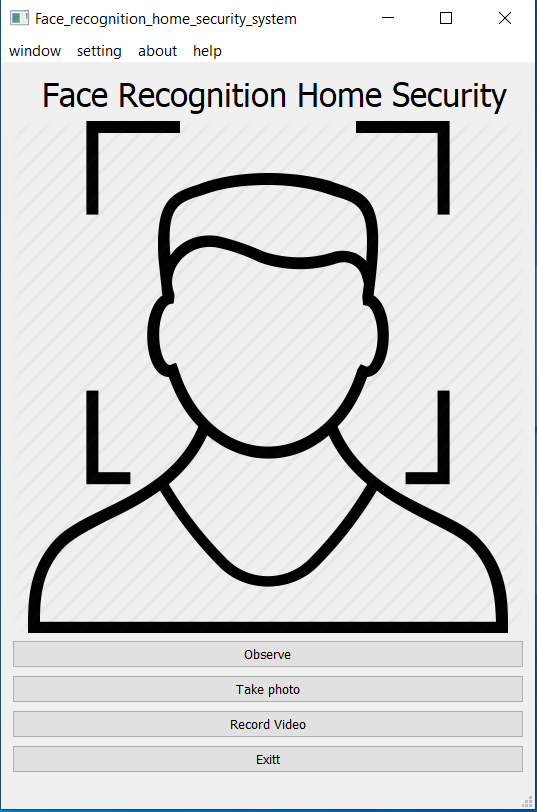
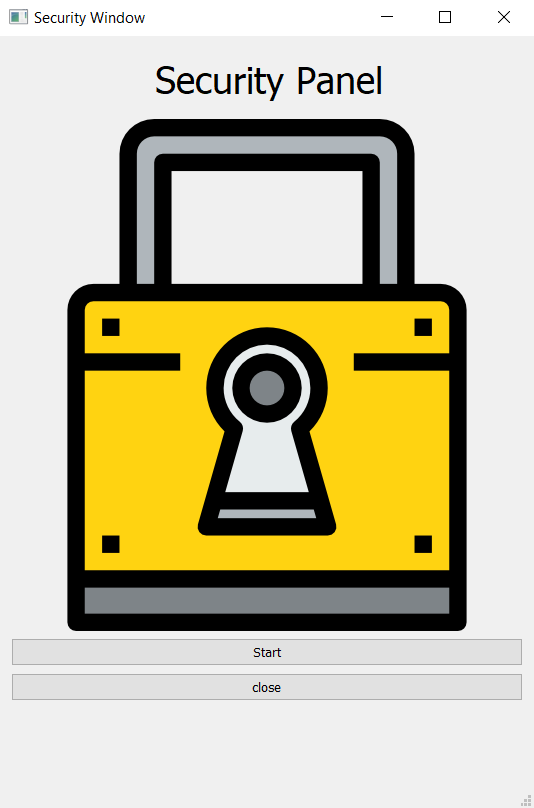
iii. Train a Support Vector Machine (SVM) on top of the embeddings

iv. Detecting faces’ liveness and recognizing faces in images and video streams

v. All of these tasks will be accomplished with OpenCV, enabling us to obtain a “pure” OpenCV face recognition pipeline.

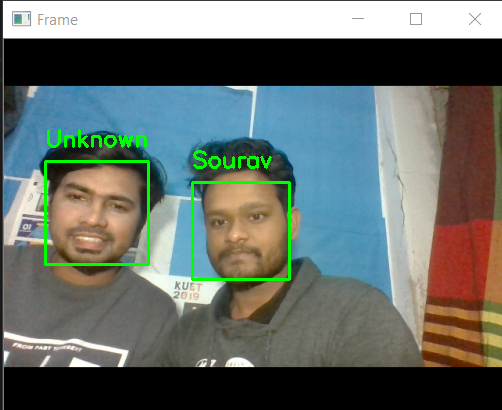
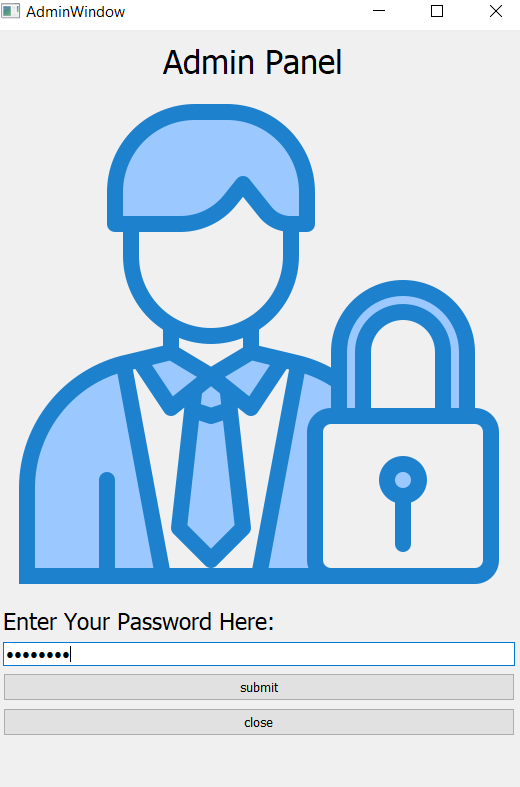
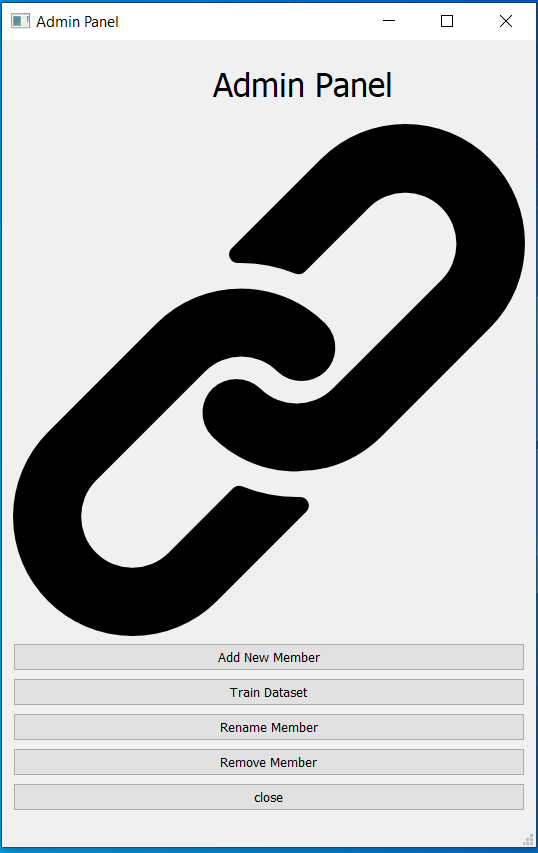
**Graphical User Interface (GUI)**

The user interface has all the options needed for the administration and other debugging purpose so that, we do not need to edit code for any management.

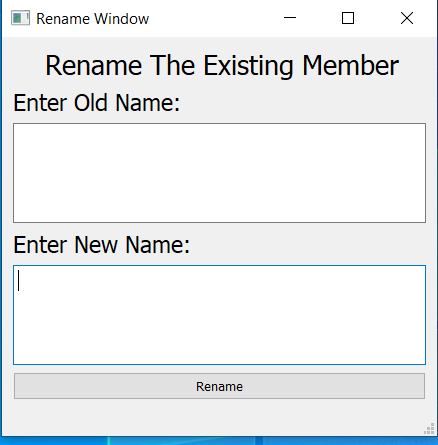


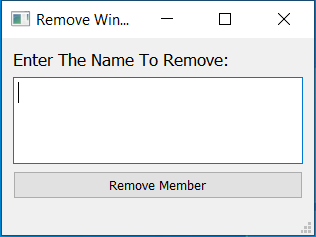
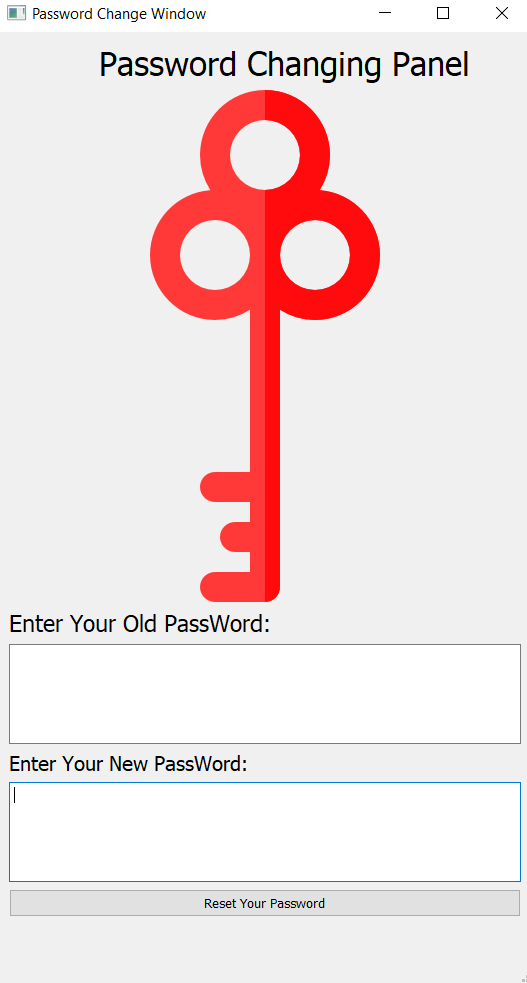
**Fig(6): Home Window Fig(7): Security Window**

After Clicking ‘**Security or Press Ctrl+A’** from Home window, then security window will open. Click ‘Start’ from security panel, then it will open a camera window which recognizes faces. If the face is in dataset, then it will show his/her name, else show unknown on his/her face.

**Fig(8): Recognise Face Fig(9): Admin Panel (Login)**

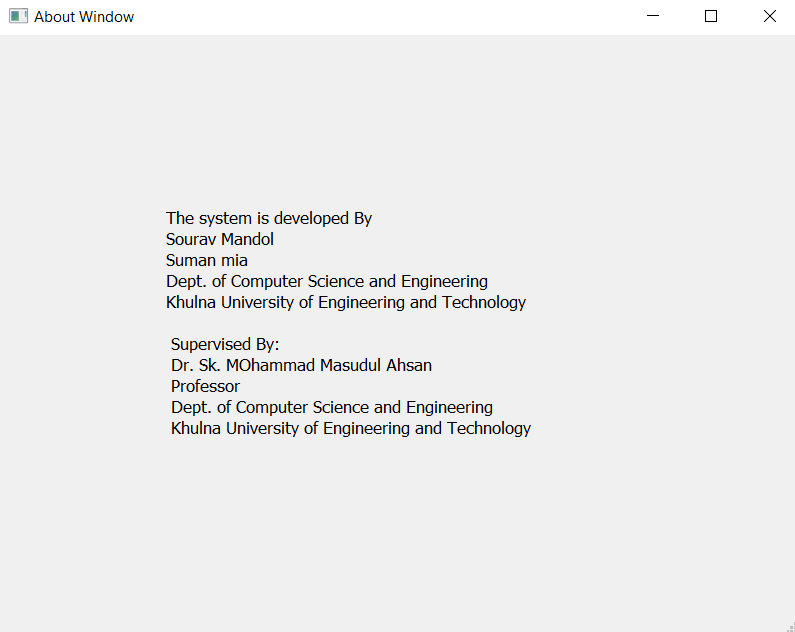
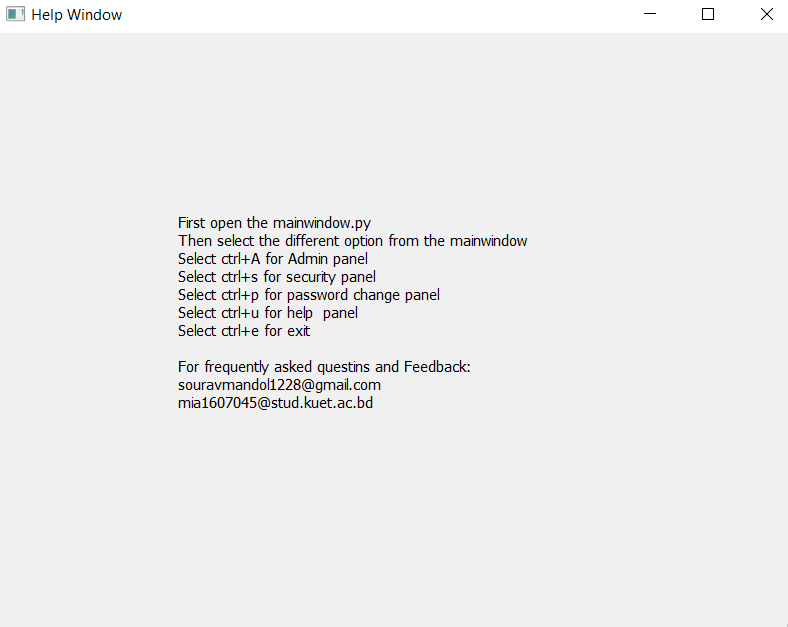
**Fig(10): Admin Section (Fully Accessible for Admin)**

To entering the admin section, need a valid password .If the password is wrong, it will show wrong password else it will go to new admin page where admin can add any member, rename any member if the person exists in database nor it will show “Member Not Found”.

 **Fig(11): Add member** **Fig(12): Rename Member**

Fig(13): Remove Member Fig(14): Password Change

Admin can remove any member from database and also can change his password.



**Fig(15): About Window Fig(16) Help Window**

A user friendly graphical interface is design for this system. Any user can easily use this system without any trouble. To make this interface, we used designer from pyqt5. A major advantage of GUIs is that they make computer operation more intuitive, and thus easier to learn and use.

**Key and Actions**

|  |  |
| --- | --- |
| **Key** | **Action** |
| Ctrl+A | Admin Panel |
| Ctrl+S | Security Panel |
| Ctrl+P | Password Change Panel |
| Ctrl+U | Help Panel |
| Ctrl+E | Exit |
| shoot | Picture Capture |
| Q | Capture Window Exit |

**Conclusion and Recommendation**

**Challenges:**

The designed algorithm was effectively able to detect the different type of faces specified on this project and recognize those faces which are known or unknown. The system is able to detect an object in both high and low light. The system has completely user friendly graphical interface (GUI). Every customer can able to use this system so easily even this customer don’t know how to use computer can able to user this. The system has more efficiency because it detects a customer within six (6) images putting on the dataset. Its accuracy is almost 90%. If an eyeglasses person is trained in system, the system can recognize his faces.

**Limitations**

If bright light appears behind object, the system takes some times to detect it but when the system detects the faces it’s easily recognize it. We also use liveness detection to make the security more strong. But livesness detection isn’t as strong as we want. But for average security, it works good. A person who put eyeglass in front camera but who is already trained without eyeglass, the system is slow to recognize him.

**Future Activities**

In future, this system will be more fast and accurate. Liveness detection will be stronger and perfectly recognize real or fake person. Also, face will easily detect doesn’t matter the person will put eyeglasses or not in future. In every light combination, the system is able recognize object face accurately.

**References**

[1] ‘Design of face detection and recognition system for smart home security application’ , by Dwi Ana Ratna Wati , Dika Abadianto, Department of Electrical Engineering, Universitas Islam Indonesia, Yogyakarta, Indonesia.

[2]’ Facial recognition system’ by http://www.pyimagesearch

[3] ‘Face Recognition System and It's Application’, by Xuehong Tian Published in: 2009 First International Conference on Information Science and Engineering

[4] ‘Research on Face Recognition Based on Embedded System’, by Hong Zhao, Xi-Jun Liang, and Peng Yang, School of Computer and Communication, Lanzhou University of Technology, Lanzhou 730050, China

[5] ‘Image-based Face Detection and Recognition:“State of the Art” ’,by Faizan Ahmad, Aaima Najam and Zeeshan Ahmed, Department of Computer Science & Engineering, Beijing University of Aeronautics & Astronautics