Design for Smart Security System using Face Recognition

Rohan Shah[1] Prasad Zende[2] Aishwarya Panhale[3] Chinmay Karnik[4]
[1,2,3,4]Department of Information Technology
[1,2,3,4]A.P. Shah Institute of Technology, Thane, Maharashtra, India

Abstract— This research designs face detection and recognition systems for smart home security application. This system is used for door lock system to open or close door using face recognition. The design is implemented using NodeMCU, Servo Motor and programmed using Arduino IDE. The connection between the camera and NodeMCU controlled Servo Motor is on a wi-fi network. The image of a person is acquired via webcam connected to a device at door. Door is accessed by servo motor through Thingspeak API when face is identified. The face detection system is built based on the histogram analysis, while the face recognition is based on the Local Binary Patterns Histograms (LBPH). The testing is done to examine the performance of the face detection in various change of distance, light intensity, light position angles. The face detection module has good performance in some conditions as distance between the person and the camera is less than 100 cm, person doesn't use accessories that cover part of face, person doesn't use shirt with color similar to skin color, and background color is difference from skin color. While the face recognition system has 80% of accuracy when it is tested using real time image.

Key words: Face Recognition, Home Automation, Face Detection, Security

I. INTRODUCTION

Home security is one of the utmost things that must be considered by the humans as well as in the smart home systems. Currently used home security system, is a conventional home security system, which is a security system with a mechanical system that requires users to open or close the door. This makes the home security weak due to several factors, namely: the ease of duplication of keys, the probability of a lost key or changing hands, and others.

In addition to this, the use of conventional security systems is also considered less effective and efficient. This is because the user is required to open the door by first inserting the key into the lock then turning the key in a certain direction so that the door can be opened. The process of opening the door is long enough, this makes the conventional security system becomes ineffective and inefficient. Therefore, it is needed a home security system which is more effective, efficient and has a high level of security.

The security system of home can be developed by using face recognition method. Face is used as a key to access home. By using real face, the process of opening the door will be more effective and efficient because it just needs to direct a face on the camera, so the camera can identify whether the person is allowed for coming in or not. By using the face, the level of security becomes higher because the face cannot be duplicated as well as changed hands.

The human face has a particular shape that requires some calculations in order to recognize it. Individuals are distinguished by their faces, with which they are being identified. Human can memorize many faces during their life journey and get to know them immediately, even after years. Aging and distractions like glasses, beard or change of skin color may gradually vary face recognition rates. Face identification represents one of the most used types of biometry. It proceeds as follows: Starting with calculating and subtracting specific characteristics, then verifying them with the already existing database, in addition to obtaining a positive correspondence between the compared faces. After getting the face shape details, the system adjusts them by using some algorithm models, finally face images are stored in the database and resolved using other algorithm.

II. LITERATURE REVIEW

Sr No	Author	Methodology	Limitation
1	Dwi Ana Ratna Wati & Dika Abadianto ICITISEE (2017) [1]	MyRIO is used and is connected to computer with wifi Face Detection and Recognition can be implemented using MyRIO as a main controller.	To detect face, it must be positioned at 240 cm or less than that. Distance more than 240 won't be detected. Various accessories can be difficult to detect face. I.e.; slight change in face features.
2	Ayman Ben Thabet & Nidhal Ben Amor (2015) [2].	Raspberry pi board with ARMv7 Cortex-A7 is used with OpenCv library. PCA algorithm is used for face recognition	range of 40 cm to 1 meter of the camera is required.
3	T Archana & T Archana, T. Venugopal (International Conference on Green Computing and Internet of Things, 2015) [3].	Comparison between two face recognition approach in PCA & Template. Along with comparison advantages and important factors of two approaches.	On Frontal view recognition is accepted but factors are: Facial Expression Change in plane Illumination. Rotation of head.

4	Mohamma Djaved R. Mulla & Rohita P. Patil & Dr. S.K. Shah IEEE (2015)[4].	A MATLAB based Principal Component Analysis is used for face Matching decision.	
5	Yashwanth Sai, Vijai Chandra Prasad, Niveditha, Sasipraba, Vigneshwari & S. Gowri (IEEE Conference, 2017)[5].	PCA algorithm is used for face recognition & take the dimensions of face messages and convert to grayscale. High prevalent CCTV cameras for intruders. Uses Raspberry PI & camera modules and sensors alerting users through email or mobile notification SMS 7 generate log of default entry & exits.	An authority is required to be present to do surveillance to watch the activities.

Table 1: Literature Review of Research Papers

III. METHODOLOGY

A. Face detection:

Face detection method is to detect human face using computer technology. Face Detection is the first and essential step for face recognition, and it is used to detect faces in the images. It is used to detect faces in real time for surveillance and tracking of person or objects. Being the first step in face recognition system it is essential to detect face as per the given parameters and measurement to get desired pixels for better performance.

B. Data Gathering:

Extract unique characteristics of face that it can use to differentiate from another person, like eyes, mouth, nose, etc.

C. Data Comparison:

Despite variations in light or expression, it will compare those unique features to all the features of all the people you know.

D. Face Recognition:

For security purpose face recognition system is used to recognize the human who is detected Face recognition is a method of identifying or verifying the identity of an individual using their face. Face recognition systems can be used to identify people in photos, video, or in real-time.

Classical human face recognition systems are divided into three phases first step is preprocessing, which consists of many types of operations, such as image registration, scaling, face normalization, reducing the effect of background noise, detection and resizing, all of which affect the face recognition accuracy. Second phase, which can be achieved by using powerful transformation approaches.

In LBPH algorithm for face recognition these four steps are carried out step by step: The face is detected using video capture using 3x3 pixel matrix. The captured images follow the four parameters like Radius, Neighbors, Grid X, Grid Y. Radius is used to build the circular local binary pattern and represents the radius around the central pixel (set to 1). Neighbor parameter is the number of sample points to build the circular local binary pattern (set to 8 usually).

Grid X gives the number of cells in the horizontal direction as the number of cells increases accuracy of grid is increased. Grid Y gives the number of cells in the vertical direction as the number of cells increases accuracy of grid is increased. The algorithm uses a concept of a sliding window, sliding window refers to an imaginary box that hold the frames on both sender and receiver side, based on the parameter's radius and neighbors. Using Grid X and Grid Y parameter images are divided into multiple grid and the histogram is generated of each region of gray scale image. The Histogram is then used for detecting the face from training dataset.

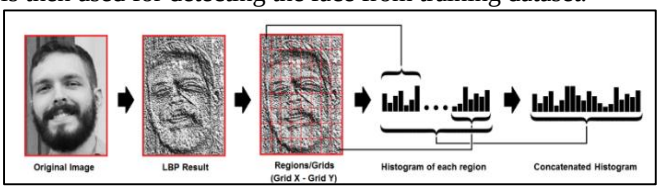

Fig 3.1: Extracting the Histograms

1) Why LBPH over Eigen faces and Fisher Faces?

Both Eigenfaces and Fisher faces are mathematical algorithm that use mathematical formulas to find a value for comparison.

Eigen faces is an algorithm based on principal component analysis (PCA) that is used for dimensionality reduction of an image matrix. If the sample image represents in n-dimensional space, this method uses a linear transform to mapped the original n-dimensional space into m-dimensional feature space, where m < n.

Fisher faces uses a class specific linear method to reduce the image dimensionality and use a classifier to reduce feature space. The class specific method uses for shaping the scatter in order to increase the classification reliability. Both eigenfaces and fisher faces involve creation of matrices and computation of the values in the matrices.

LBPH (Local Binary Pattern Histogram) on the other hand is an algorithm that divides the images into local regions and to each region a pixel is allotted that is labelled with a decimal value. These values are used to make a histogram. These histograms of different images undergo computation for similarity for the classification purpose. Both Eigen and Fisher Face algorithm are component-based algorithm having low efficiency whereas LBPH has pixel-based data generation technique which makes it more efficient as all pixels of the image are considered for computation.

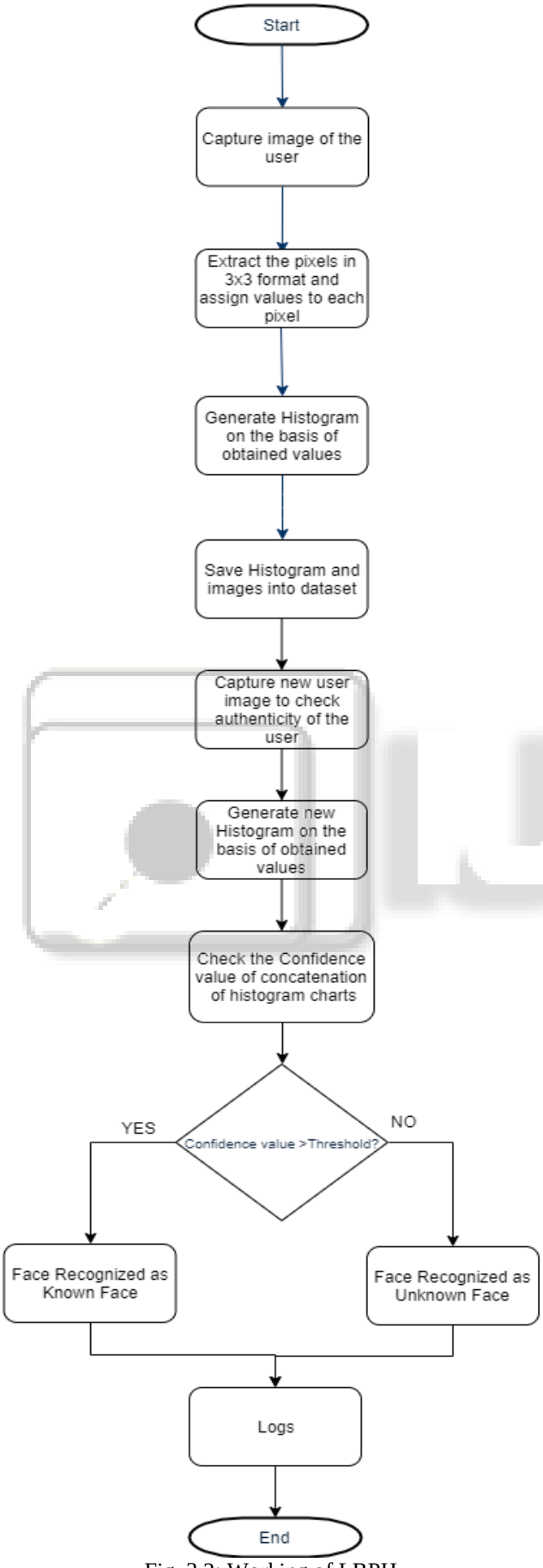

Fig. 3.2: Working of LBPH

IV. WORKING

The setup consists of a Node MCU, a servo motor, a tower bolt, and a 64bit Windows 7 system with a ram of 4.00 GB. Python is used to implement video-based face recognition by using OpenCV library.

Node MCU is to control a servo motor that opens and closes the tower bolt based on a known or unknown face respectively. A Thingspeak server maintains the logs and is responsible for the delivery of data from both ends.

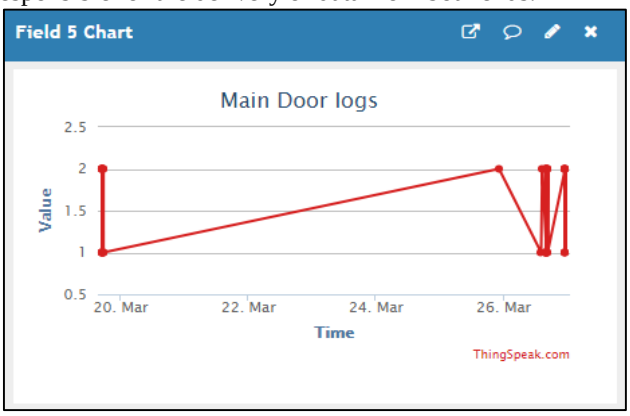

Fig. 4.1: Thingspeak Server Logs

The algorithm behind the Face recognition system is LBPH (Local Binary Pattern Histogram) Analysis. An SQLite3 database is used to store the user details.

The dataset is designed in such a way that it can capture as many images as the user wants where each user is provided with a unique Id that is used as a call function in the system. The database contains images with the four types of facial expressions that are normal face, smile, closed eyes and smirk. Some of the factors to be considered while capturing the images for the dataset are light exposure, Noise and the video resolution. Once the images are captured, they are converted into grey scale images and stored into the dataset as shown in Fig 4.2.

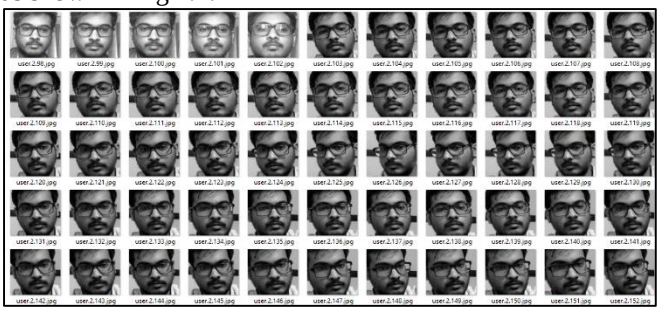

Fig. 4.2: Generated Dataset

For the images to have a unique identification each image is saved with the ID of the user and the count of the image. Once the dataset if ready it is to be trained with respect to a cascade file which helps the system to understand what the objects in the environment are and what are the local values generated in the LBP algorithm. These local values are used to plot a histogram that is used for comparison with the real face as shown in Fig 4.3.

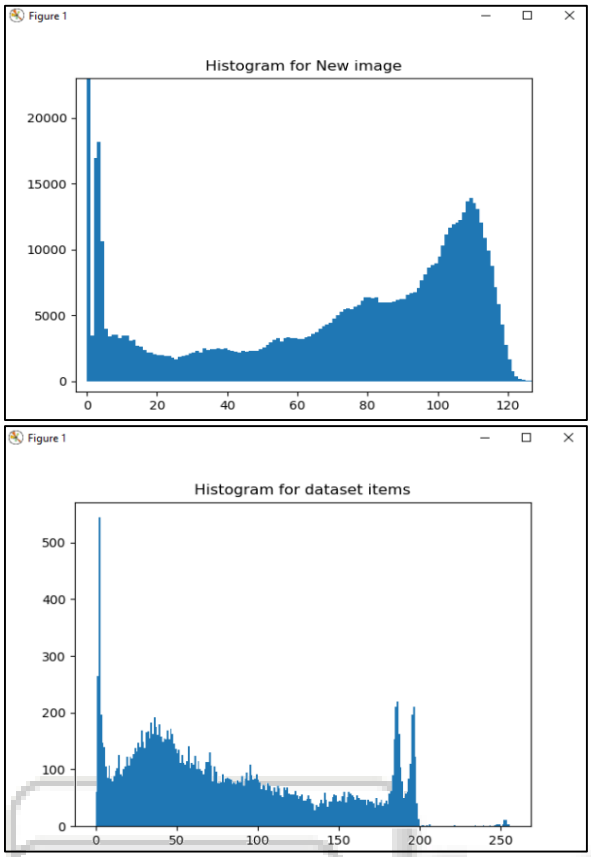

Fig. 4.3: Histograms of Images

When a face is brought in front of the camera an image is captured and converted into gray scale and then compared with the images in the dataset. Here a new histogram is created for the image of the person at the door and is compared with the histogram of the images in the dataset. If the values match approximately the respective name is displayed on the screen and is recognized at a known person. If the person at the door is found out to be an unknown person a normal image of the person is captured and stored in the Unknown dataset with the image name as the current date time. This helps keep the logs of the unauthenticated entry.

Not only does the system recognize the person at the door, the system is also designed to automate security. To automate we have a NodeMCU that controls a servo-motor operating a Tower-bolt shown in Fig 4.4

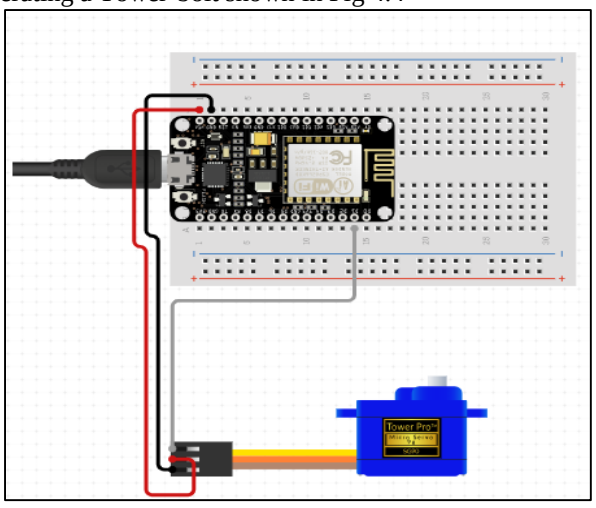

Fig. 4.4: Setup Circuit Diagram

The NodeMCU is connected to the internet using the ESP8266 present on it and it is kept on polling mode with the address as the Thingspeak server. By Polling we say that the NodeMCU keeps on checking the values at the server constantly at a fixed time interval. If the person at the door is recognized the system executes a POST method that updates the server field value as 2 and if the person is unrecognized the method updates the server field value as 1. As the NodeMCU is kept on Polling mode it executes a GET request at fixed time intervals. The NodeMCU is programmed to actuate the servo motor to move its arm to positive 90 degrees on receiving the value 2 from the server. On receiving the field value as 1 it moves its arm to negative 90 degrees. On the movement of the servo motor the Tower-bolt opens and closes making the lock mechanism of the door automated.

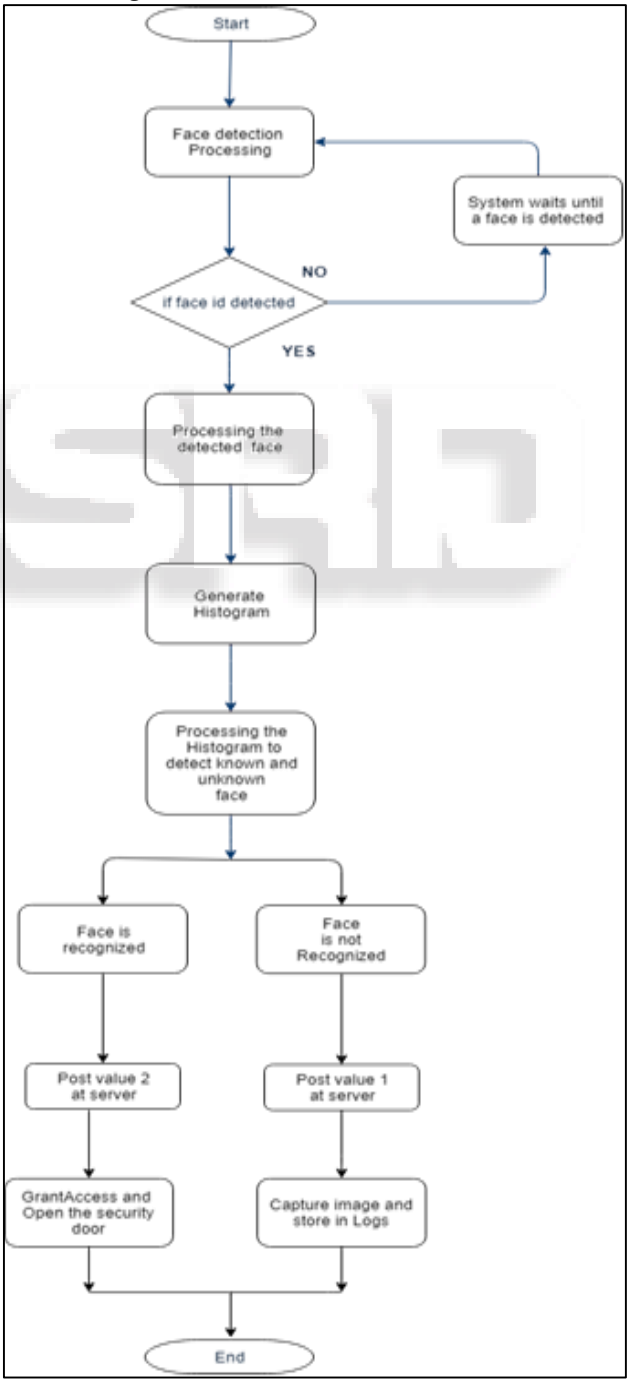

Fig. 4.5: Working of Security Door

V. RESULT

The system made use of a 720p camera to capture over 200 images of a single user, Collectively the images of different user was a total of over 1000 images each captured as a RGB image but processed and saved as Grayscale image. All the images and the cascade files were trained which takes over 15 minutes to complete the training of the system. Once trained the system was run as a Detector where a camera was placed outside the main door. Light illumination did not have much effect on the accuracy if the system as the light fixtures on the door were always of the same intensity. Fig 5.1 is a test case where a known person came in front of the camera.

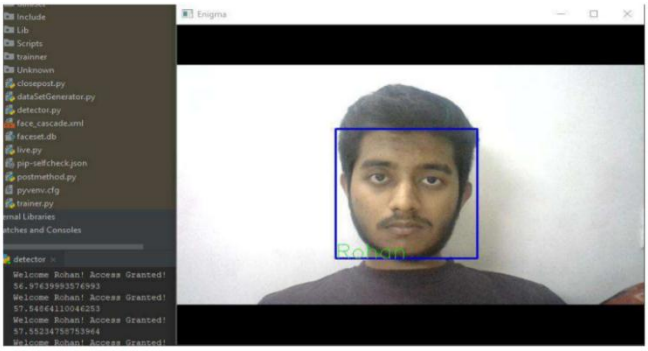

Fig. 5.1: Detection of Face at Door

The algorithm checks for the confidence value each time a face hits up on the camera screen. If the confidence value is greater than the threshold value then the predictor is initiated which predicts the name of the person on the camera. With each computation the confidence value increases as the system keeps on learning constantly. Apart from saving the images of unknown people the system also sometimes fluctuates with different user names during predictions and may also declare the user as unknown.

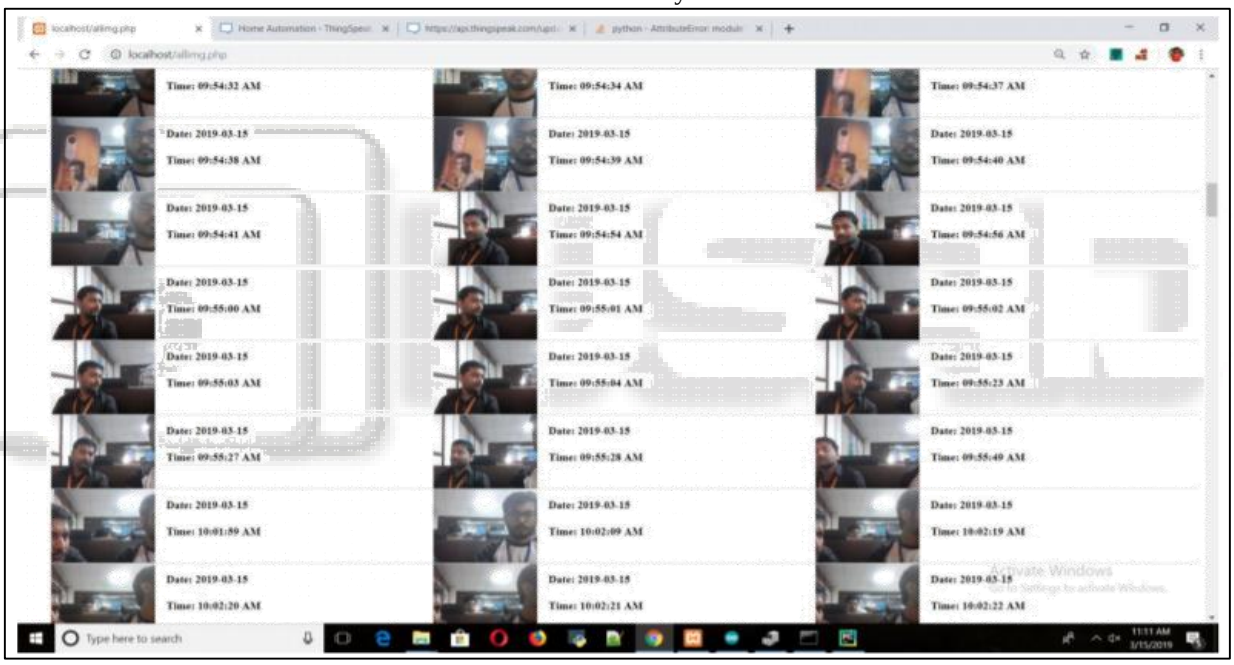

Fig. 5.2: Logs of Unknown Faces

This fluctuation occurs only when there is increased noise in the image or the facial features of the users match till some extent. But it was observed that with more images given to the system as dataset input, with training the system is observed to become more accurate.

The NodeMCU is responsible to open the tower bolt if a known person is detected at the main door. As the NodeMCU is always on polling mode to the server it keeps on checking for the updated value that the face recognition system sends to the server using the POST method mentioned earlier. The server also maintains the logs of known and unknown entries but at some intervals it was observed that there was a delay of not more than 5 seconds or the NodeMCU to receive the updated value from the server. But the Servo motor successfully opens the tower bolt and lets the known individual through the main door making the process smart and automated.

The LBPH algorithm is accurate in both classification and recognition of the input images and the system successfully actuates the NodeMCU controlled servo motor as shown in the Fig 5.3.

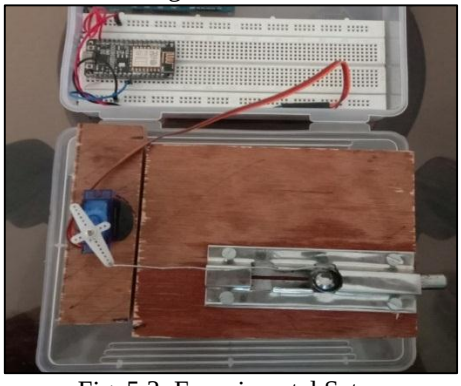

Fig. 5.3: Experimental Setup

VI. CONCLUSION

This paper shows the use of LBPH algorithm on OpenCV to be used as a Smart security system. The outcome that can be derived from the implementation shows that there is a tradeoff that exists between the authenticated recognition rate and the threshold value. With the increase in threshold value, the number of misses in the recognition system begins to decrease that can possibly result in misclassifications.

LBPH algorithm is an accurate and efficient algorithm that can be used for face recognition systems in OpenCV as changes in light illumination and angle orientation has minimal effect on the accuracy. Neural networks can be included in the system to aid the LBPH algorithm in the future which will in turn make the system accurate and more efficient and would reduce the complexity over the system.

ACKNOWLEDGMENT

We would like to thank our respected Guide Prof. Selvin Furtado, Dr. Rahul Ambekar, Prof. Vishal Badgujar and HOD K. B. Deshpande for their encouragement and support. We gratefully acknowledge them for imparting us with valuable basic knowledge of Digital Image Processing and Embedded Systems. We are also thankful to the department of Information Technology, Smt. A. P. Shah Institute of Technology, Thane for providing us infrastructure, facilities and moral support.

REFERENCES

[1] Dwi Ana Ratna Wati, Dika Abadianto, "Design of Face Detection & Recognition System for Smart Home Security Application", International Conferences on Information Technology and Electrical Engineering, IEEE 2017.
[2] Ayman Ben Thabet, Nidhal Ben Amor, "Enhanced Smart Doorbell System Based on Face Recognition", 16th international conference on Sciences and Techniques of Automatic control computer engineering - STA'2015, Monastir, Tunisia, December 21-23, IEEE 2015.
[3] T Archana, T. Venugopal, "Face Recognition: A Template Based Approach", International Conference on Green Computing and Internet of Things, IEEE 2015.
[4] Mohammadjaved R. Mulla, Rohita P. Patil, Dr. S. K. Shah, "Facial Image Based Security System Using PCA", International Conference on Information Processing (ICIP) Vishwakarma Institute of Technology. Dec 16-19, IEEE 2015.
[5] Yashwanth Sai, Vijai Chandra Prasad, Niveditha, Sasipraba, Vigneshwari & S. Gowri, "Low cost automated Facial Recognition system", IEEE 2017.

