Smart and Secure Home Using Facial Recognition System

Rashmi N[1], Khazal Ammaarah[2], Ujwala L[3]

[1,2,3]*Department of Electronics and Communication, BMS Institute of Technology and Management, Avalahalli Yelahanka Bengaluru 560064, Karnataka, India*

***Abstract*--The two important facets of human being is security and surveillance. The most significant feature of any home security system is to be able to detect who is entering or leaving the house. This paper presents a new technique which can prevent theft in highly secure home environment with reduced power consumption and more reliable standalone security device for both security and intruder detection through "face recognition door lock security system". This system is acheived by using Raspberry pi circuit.Each time an individual walks in front of the door, the system recognizes the face and if the face is recorded then the door is unlocked. Otherwise the door does not unlock.**

***Keywords:* Smart home, secure home, Cascade classifiers, Object detectors.**

I. INTRODUCTION

Biometrics is the technical term for body measurements and calculations. It refers to metrics related to human characteristics. Biometrics authentication is used in computer science as a form of identification and access control. It is also used to identify individuals in groups that are under surveillance if they are the person they claim to be. Biometrics verification can be achieved through the property that human trait linked to a person is like a unique data stream. Some solutions are already available in the market such as: passwords or pins have been in use for security. These solutions still have many disadvantages as they fail to provide complete protection leading to theft till date. Instead, facial recognition can be used as they are one's biometric trait. These are distinctive and cannot be altered or stolen easily. The level of security is thereby increased.

This paper is mainly focused on facial recognition aspect. Face is most commonly used biometric to recognize people. Over the decade facial recognition has received extensive attention from researchers and developers due to human activities present in various applications such as security like airport, criminal detection, face tracking and forensics. In comparison to all the other biometrics that are present like palm print, iris or fingerprint, face recognition is comparatively non-intrusive in nature.

The intended project involves two step process; Face detection and Face recognition. The former Face detection involves identification of the face whereas the latter face recognition involves capturing a face and comparing it with the database.

II. SYSTEM DESCRIPTION

The proposed system uses Haar feature-based cascade classifiers for real time face recognition.

Object Detection using Haar feature-based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones in their paper, "Rapid Object Detection using a Boosted Cascade of Simple Features" in 2001. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images. Here we have used it for face detection. Initially, the algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then we need to extract features from it.

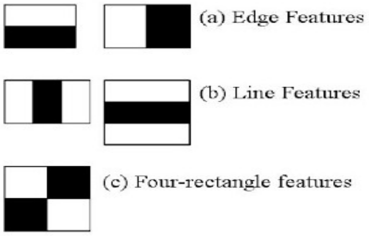


Fig 1. Haar cascade feature extraction

We select the features with minimum error rate, which means they are the features that best classifies the face and non-face images.

III. SYSTEM IMPLEMENTATION

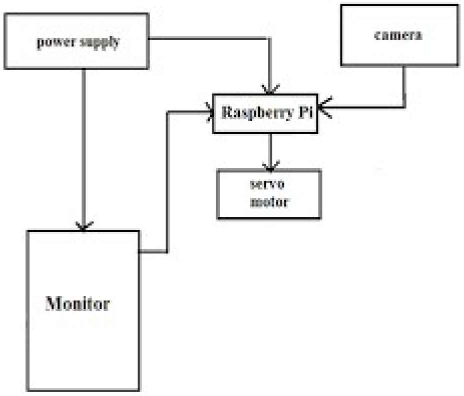


Fig 2. Block Diagram

The following procedure gives the description of the applications used for data gathering, face detection, training and face recognition. We coded our project in Python using OpenCV platform. First stage was setting up the Raspberry pi 3 model b+ by installing raspbian operating system on to the micro- SD card and installing Win32 Disk Imager on the machine. There are three ports for the Raspberry Pi 3

Model B+ a micro USB port, HDMI port, stereo output and composite video ports, which gives even more audio visual options. After plugging in all the cables and plugging in the power link last. The system boots up like so.

The next stage is to install OpenCV. This process can be quite time consuming since many dependencies and prerequisites have to be installed. First thing is to expand the filesystem to include all available space on the micro-SD card and reboot the system to conceive the result. Then update and upgrade any existing packages. We installed some developer tools, including CMake, which helped us configure the OpenCV build process. The OpenCV library comes with a sub-module named highgui which is used to display images on to our screen and build basic GUIs. Many operations inside of OpenCV can be reformed further by installing a few extra dependencies. After installation of complete dependencies we downloaded the OpenCV source code. Before we can begin compiling OpenCV on our Raspberry Pi 3, we need to install, a Python package manager. After this we created the Python virtual environment that we'll use for computer vision development. A virtual environment is a tool used to keep the dependencies required by various projects in isolated environments. Next and last python dependency to install is called Numpy, a Python package used for numerical processing. Finally, we compile and install OpenCV.

We used Raspberry pi camera for face detection. To interface the pi camera with raspberry pi board we connected the camera strip between the TRRS/AV socket and the HDMI connector. By powering up the system the camera module is enabled. Putting in a few commands we were able to snap a picture with the raspberry Pi. Next stage is the face detection. For the face detection system we used Haar-cascades Classifier. Despite the fact that training is required for creating new Haar-cascades, OpenCV has a powerful set of Haar-cascades that we used for this project. Using face-cascades alone led to random objects being identified and thus eye, nose and mouth cascades were incorporated to obtain stable face detection. Haar Cascade is the most common way to detect a face or any object. Using a few lines of code we were able to detect a face, using Python and OpenCV.

Next we had to gather face data (images) of the people to be identified. Executing the Python script and by capturing a few Ids. We gathered all the data required for training. To incorporate a new user or to change the photos for one that already exists we have to run the script each time.

Lastly we take all the user data from the dataset and run the script to train the OpenCV recognizer. This is completed directly by a specific OpenCV function. Now the system is ready to recognize some faces.

The Pi camera captures a face using the recognizer and if this individual had their face captured and trained before then the recognizer will make a prognosis returning its id and an index, this shows how confident the recognizer is with this match.

A servo motor is a type of DC motor that, upon receiving a signal of a certain frequency, can rotate itself to any angle from 0-180 degrees. Its 90 degree position is generally referred to as neutral position, because it can rotate equally in either direction from that point. We interfaced servo motor to the raspberry pi and generated a code suitable for the motor to acts as a door. Therefore when any individual attempts to open the door, the security system detects if the individual is a intruder or not and hence unlocks the door if the face is recognized.

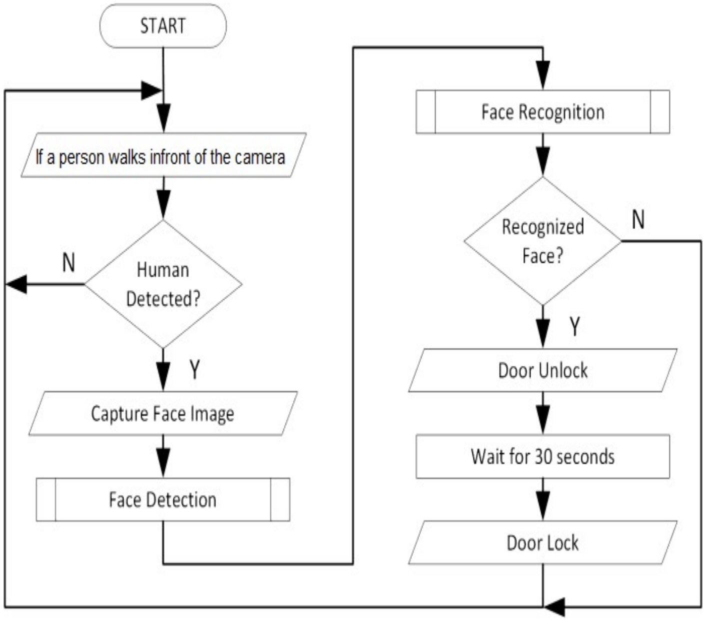
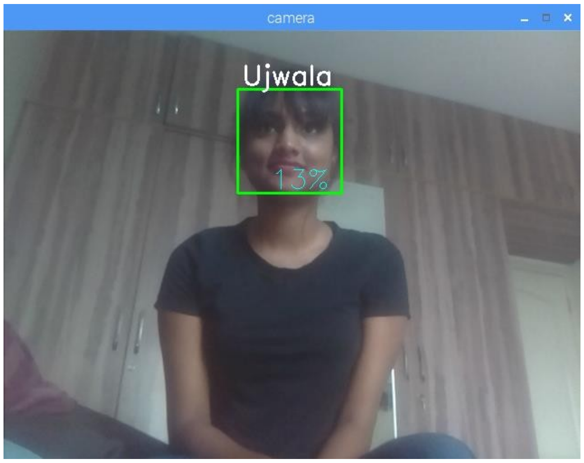
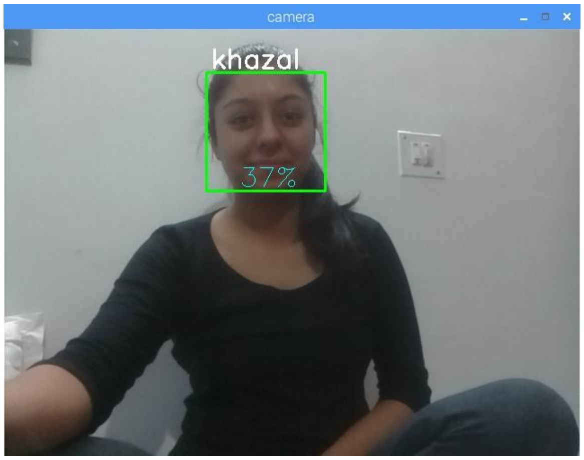


Fig 3. Flow chart

IV. RESULT

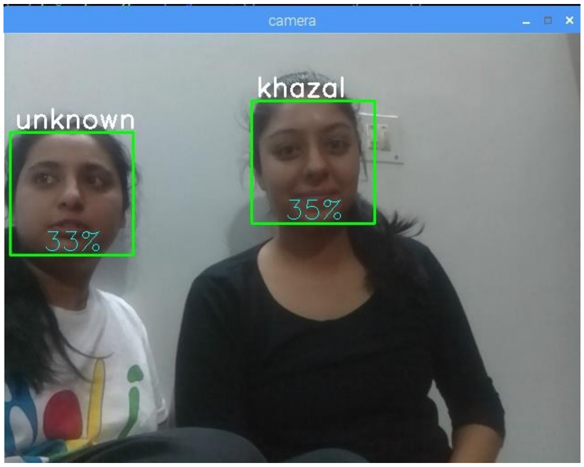
This project is built using a Raspberry pi 3 model b+ with OpenCV Library and coded in Python language. OpenCV was designed for computational efficiency and with a strong focus on real-time applications. After gathering 30 images for capturing the image in the dataset and training the program for the system to store the images. The result obtained are as follows :





Since we captured and trained our faces, the system compared the real time images to the images stored in the dataset. Whenever it detected the authorized person it sent electric pulses to the servo motor making it rotate at a angle of 90 degrees and acting as a door it unlocked.

We also tried it with an intruder whose face was not stored. It showed the following result :



The system detected the unauthorized person as unknown and hence the servo motor did not rotate.

V. CONCLUSION

This project proposed an system for door access control through facial recognition has been presented. The system is achieved and verified by using the enrolled facial images in the database. There are several methods that can be used to achieve this purpose. Some of them include using PCA or eigenfaces. Although there are various other techniques that are present to implement, this mechanism provides a better performance. The Haar Cascade methodology is one among them as it provides an accurate performance.

In this proposed system the user details are fed into the database though face detection, face training and face recognition. Whenever a person was detected by the camera it was compared with the database. If the person is detected the door is unlocked if not, the door remains idle. The result of this project is to provide a higher level of security which was realized with the help of OpenCV platform and Raspberry pi.

REFERENCE

[1]. Paul Viola, Rapid Object Detection using a Boosted Cascade of Simple Features, Conference on computer vision and pattern recognition 2001.

[2]. Marian Stewart Bartlett, Javier R. Movellan and Terrence J. Sejnowski, Face Recognition by Independent Component Analysis Member, IEEE, and Fellow,IEEE2.

[3]. Javier Ruiz-del-Solar and Pablo Navarrete, Eigenspace-based Face Recognition: A comparative study of different approaches, IEEE 2005.

[4]. Maneesh Ayi , Ajay Kamal Ganti , Maheswari Adimulam , Badiganti Karthik , Manisha Banam , G Vimala Kumari, Face Tracking and Recognition Using MATLAB and Arduino, IJRASET, 2017.

[5]. S. M. Kamrul Hasan and Tanvir Ahmed Chowdhury, Face Recognition Using Artificial Neural Networks,RUET 2013.