International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor :6.887
Available at www.ijraset.com
SYNERGY-VIT University, Vellore

Intruder Detection Using Face Recognition and I.O.T services on RASPBERRY PI

Yellanti Venkat Vivek[1], P Chandan[2], Praneeth Komati[3]
[1, 2, 3]*Computer Science and Engineering, Vellore Institute of Technology*

Abstract: ***Now a day's security has become one of the major issue in real world and for each and every method of security there is a loop hole or some hack to surpass the security system and steal or cause damage to the data which has been enclosed with the high security. Present day mechanisms which use cctv footage, which are used to view and track the intruder, has not given the desired results. The footage of the video will increase and the storage capacity is also very high when we implement with the cctv footage's. In order to overcome this we are using IOT devices to alert the system whenever a person comes close to the camera. The quality of video is not sufficient enough to detect the intruder and also has many other issues. In order to overcome this problem we propose a new method by using machine learning and image processing for automatic intruder detection and recognition. As we all know, the advancement of the technology, even biometric, is easily modified to provide the unauthorized entry, therefore we provide the face recognition for detecting the intruders in our system.***

Keywords: Image processing, Machine Learning for training dataset, I.O.T SERVICES, Face Detection, Recognition, Intruder, location on map, security system, Live Stream.

I. INTRODUCTION

Recently, many cases have been filed under thefts. In this paper, we have implemented a new system of intruder detection using face recognition which gives us the detailed image of the intruder and it also detects the faces which we have already kept in our database.

We have also implemented a new system in which if one were to select a face of a person, it locates the person's current location by using our cameras which are installed everywhere possible. As we all know, face detection is very useful in the detection of intruders which has a very large use in security related issues in real world scenarios.

For this, we will first train the machine with known faces which consists of many photographs of the test subjects and we will train it from different angles in order to establish the code needed to recognize the intruder from all directions.

If the intruder has entered into our system then we will get a message from IOT devices and afterwards when we check the intruder we can also trace his location and we will give it to the owner of the system through a map route. If any intruder is recognized by the system, the system will give a message to the owner that an intruder has been detected. If it is a person who is authorized to enter into the system, it will keep a record of all the timings of his/her entry, which will be helpful for future reference and will not deliver a message to the owner.

II. PROPOSED SYSTEM

A. Training

In this stage we create a database using opencv and store all the pictures of the authorized persons in the system. All the images are in stored by using python image library (PIL). At first, a separate folder is maintained for each and every number and in each folder all the images of the person are stored in it. Then, all the images are converted into unique matrix with the pixel values of the image and these are stored in a separate file. For every different image there will be unique matrix with different pixels values. If we add the image under any one folder then we will train that image and convert it into pixels values. We can view the matrix by using different text editors like notepad++ and .yml files are stored in separate folder. These yml files consists the different matrices for different authorized users.

B. Intruder Detection

In the next phase we will detect the faces when a face is approached near the camera. The infrared sensor present near the camera will send a message to the camera and then the camera will start to detect the face. To detect the face we are using

©IJRASET: All Rights are Reserved

LBPH (Local Binary Pattern Histogram) face recognizer and change it into matrix form which consists of many arrays of pixels. Now this separates the images: if he is the authorized users it goes into the database and his activity log when he has entered or leaved the system is recorder. If he is an unauthorized person it goes to the next step.

C. Recognition of Intruder

If any unauthorized person enters into our system then at first the camera scans his image and changes it into matrix form which consists of many arrays of pixels.

Then it searches in the database for any similar face. If no face is found then it sends a message to the owner of the system stating that some intruder has been entered into your system. Then the owner of the system goes into the ubidots and check where the intruder is present, he will get a neat location map where the intruder is present which will make it very easy for the owner to detect and recognize the intruder.

As the security system is very large we can't supervise the entire thing with one system, so we are using the raspberry-pi with Jessi OS.

This raspberry pi is connected to each camera and are installed in for the surveillance. All these raspberry pi's are connected to a single server, If any one camera detects the intruder then it comes to the raspberry pi. Then the owner of the system can view location map of the intruder using ubidots.

III. ALGORITHM USED

We are using LBPH (Local Binary Pattern Histogram)[1] face recognizer to recognize the faces of all the persons (authorized and unauthorized).This is stored in the form of matrix of pixels. These can be solved by using Eigen values and Eigen vectors. We treat the image in the form of matrix as the vector for the verification of the face. The Eigen faces maximizes the total scatter which is very helpful for the identification of the faces, and to preserve some discriminative information, we applied a linear discriminant analysis and optimized.

IV. ARCHITECTURE

Each camera is attached to a raspberry pi. Any type of camera can be used, preferably camera with motion detection or an infra-red sensor can be used for motion detection.

Each raspberry pi is connected to a single server.

This server will have all the trained dataset of authorized person. When motion is detected, the code inside starts recognizing faces. If a face is not recognized, using i.o.t services like Ubidots, we get notification on our mobile saying "intruder has been detected". On the ubidots dash board we find the location where intruder is detected and using motion eye we get live feed.

Functional Architecture: This involves collection of data, training and recognition.

1) *Data collection:* Frontal face data of each individual is collected. Python libraries used here is open cv[2].
2) *Data Training*: Collected data is analysed and then stored. Followed algorithm is Local binary pattern[1] histogram. Python library used here is PIL.
3) *Recognition:* Here the face data is collected from the feed and is matched within the existing data set created during training process. Incase face not recognized, we get notification of intruder. Libraries used here Opencv[2] and Ubidots. Functions used facecascade and eyecascade for recognition process. With help of motion eye we can have live feed also.

Things needed here are ubidots library[4], opencv library[2],PIL library[3] , Python idle, motion eye, dashboard of ubidots on pc.

Hardware Architecture: Camera, infrared sensor, raspberry pi with Jessie installed, camera, active internet connection and pc.

©IJRASET: All Rights are Reserved

V. FLOW CHART

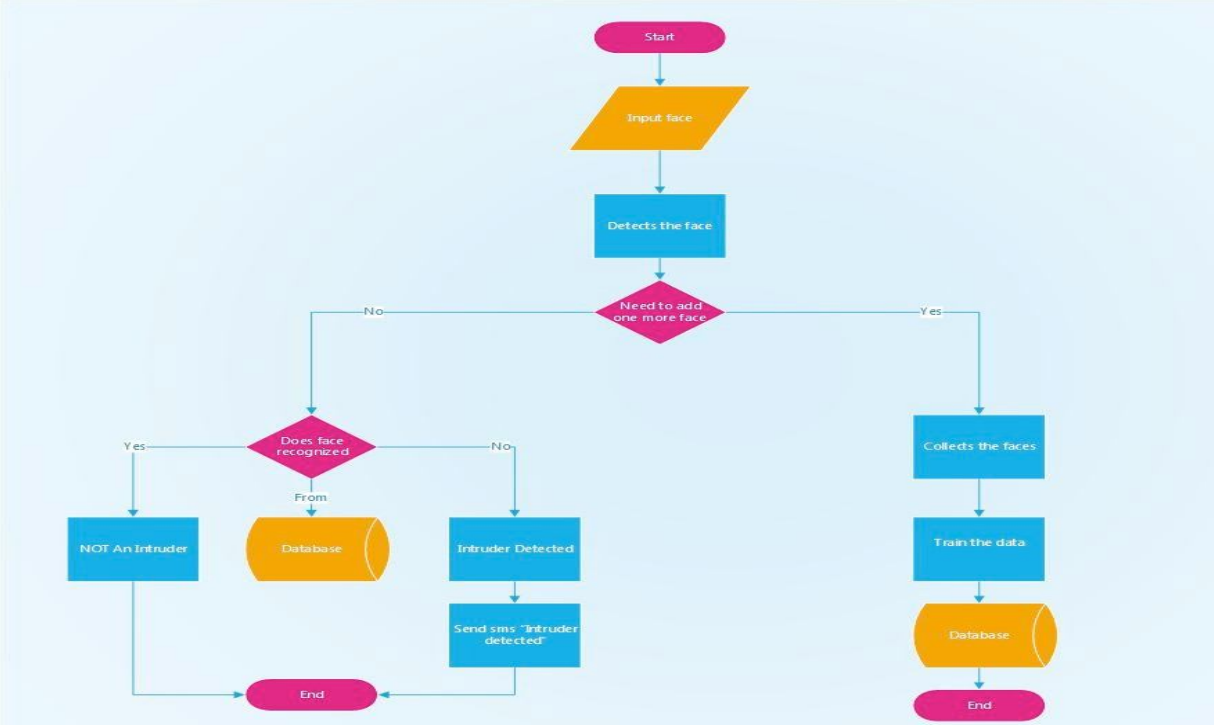

Figure 1: process of the project

VI. FUNCTIONALITIES

At first, face detection is done which means it detects our face with the various objects and then the machine gets trained with our face to recognize it next time. Then it stores our face data in matrix. At the time of recognition, it recognizes if it is similar face. If so, then it says “Not an intruder” and if it is not detected it says “Intruder detected” and sends an sms.

VII. TEST CAS

A. Test case 1

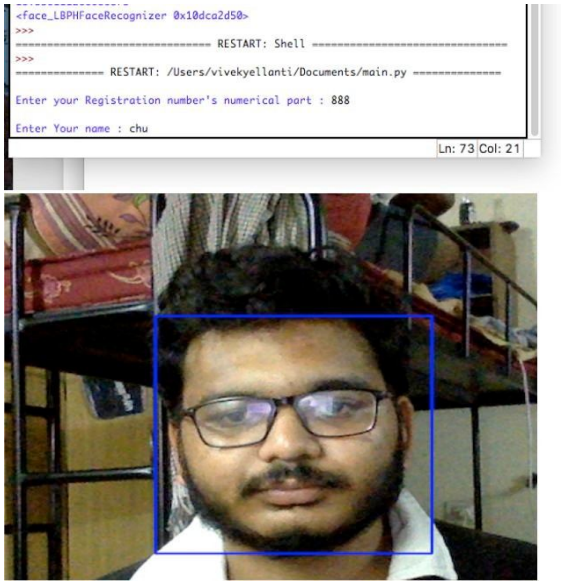

Figure 2 and 3 creating face data set

©IJRASET: All Rights are Reserved

B. Test case 2

Figure 4 training dataset

Figure 5 recognition

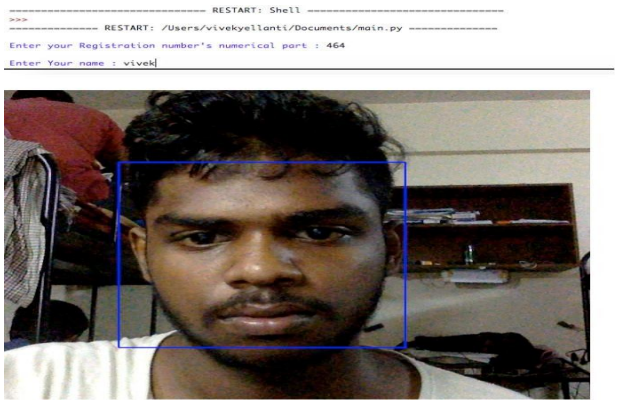

Figure 6 and 7 adding face into data base

Figure 8 recognition above test cases are performed on P.C

C. On Raspberry-pi

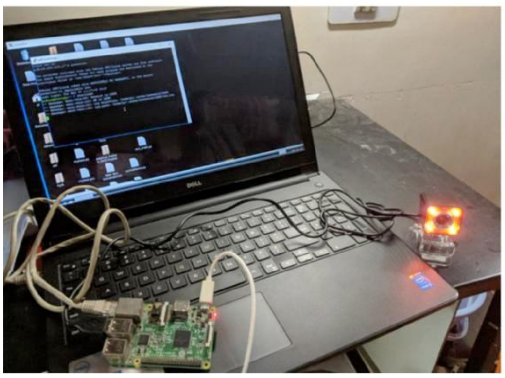

Figure 9 setup on raspberry pi

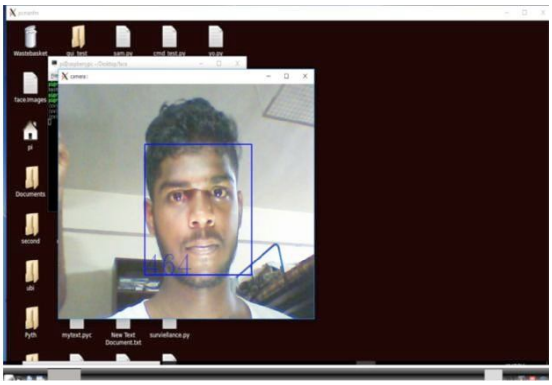

Figure 10 recognition on raspberry pi

©IJRASET: All Rights are Reserved

VIII. OUTCOME

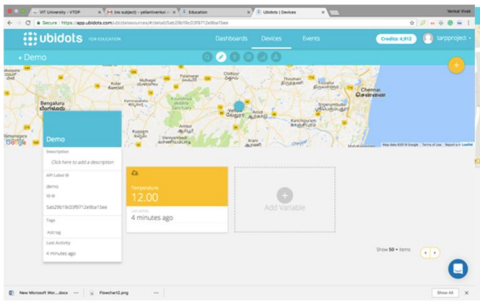

Figure 11 dashboard on ubidots site

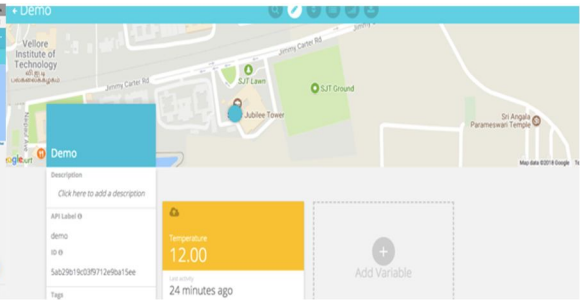

Figure 12 location on the map where the person is detected

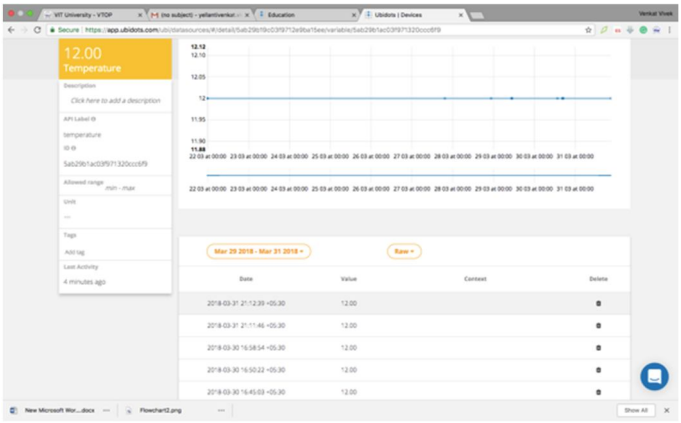

Figure 13 no.of times detected

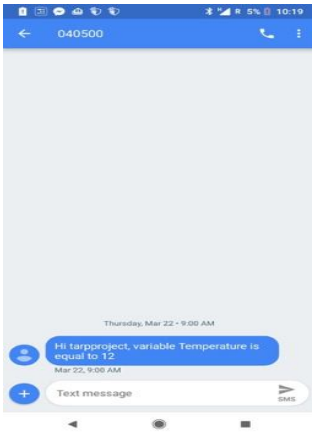

Figure 14 sms allert

IX. CONCLUSION

Prototype for Security system using face detection is successfully implemented and tested. Since it can be implemented on raspberry- pi it can be used for security systems in real time. The test results shown in the paper show that an intruder can be accurately detected and the user will receive an alert and will able to know the place where he/she is detected in real time. This paper shows that by using openCV and Raspberrypi in real time, implementation is possible for small scale industries. Future Work: even more efficient algorithm can be made and implemented for large scale breach detection. Eg. To find criminals in the society.

REFERENCES

[1] Face Recognition using Local Binary Patterns (LBP) By Md. Abdur Rahim, Md. Najmul Hossain, Tanzillah Wahid & Md. Shafiul Azam
[2] https://docs.opencv.org/2.4/index.html
[3] http://effbot.org/imagingbook/pil-index.htm
[4] https://ubidots.com/docs/api/
[5] https://www.pyimagesearch.com/2017/09/04/raspbian-stretch-install-opencv-3-python-on-your-raspberry-pi/
[6] G. Bradski & A. Kaebler (2009), "Learning OpenCV", China: Southeast Univ. Press.
[7] Open Source Computer Vision Library Reference Manual-intel
[8] Gary Bradski & Adrian Kaehler O"Reilly (2008), "Learning OpenCV", O'REILLY Media.

©IJRASET: All Rights are Reserved