

# IoT based Facial Recognition Biometric Attendance

G. Divya zion<sup>1</sup>, D. Govardhan<sup>2</sup>, B. Sravana Kumari<sup>3</sup>, K. vani<sup>4</sup>, K.M. Karthik<sup>5</sup>,  
D. Madhu Sudhan Reddy<sup>6</sup>

<sup>1</sup>Assistant Professor of Computer Science & Engineering, Mother Theresa institute of Engineering & Technology, Palamaner, Andhra Pradesh, India

<sup>2,3,4,5,6</sup>UG Student, Department of Computer Science & Engineering, Mother Theresa Institute of Engineering & Technology, Palamaner, Andhra Pradesh, India

\*\*\*

**ABSTRACT** – Internet of Things(IoT) has seen steady growth over recent years-smart home appliances, personal assistance and many more. The most difficult task in any college / school is attendance marking. In this project, we proposed an automated attendance management system which tackles the predicament of recognition of faces in biometric systems. This model incorporates a web camera that captures input image, an algorithm to detect a face from the input image, encode it and recognize the face and mark the attendance. The system camera captures the image and sends it to the database where faces are recognized from the database and attendance is calculated on basis of it. Thus our proposed system saves time and eliminates the proxy given. Also the attendance would be marked in each 1 hour without disturbing the class and makes students to be present all the periods without bunking. Also our proposed system invokes multiple frame detection to detect and identify multiple humans at a **time**.

**Keywords:** Facial Recognition, Internet of Things(IoT), Image matching, Rasberry pi, Sensor System, Web Camera.

## I. INTRODUCTION

Now in the present generation IoT is becoming popular Technology. The same is true for the field of biometrics where the need for automatic and secure recognition. In order to recognize a person, one commonly looks at faces, which differentiate from one person to another. It's the fastest biometric technology that has one and only purpose to identify human faces. Today facial recognition technology is now being used in a variety of fields because of all excellent benefits it has to offer. Biometric Technology is becoming increasingly popular for attendance tracking. Although fingerprint biometrics is used in many areas, facial recognition time tracking is an alternative biometric solution with many distinct advantages. The devices that are used in the Internet of Things (IoT) are often powered with limited computational resources that are used for tracking attendance.

## II. RELATED WORK:

In existing system, Fingerprint recognition based attendance management system is a running field today, but recognition of individual fingerprint from a set of enrolled fingerprints is a time taking process. Most fingerprint based attendance systems store the finger prints of a user in the fingerprint module database. The fingerprint system does not reveal any data about the original fingerprint of the user.

### Disadvantages:

- Using the finger print scanner, due to the physical contact, diseases can be easily spread.
- It can lead to false acceptances.
- Physically challenged people cannot use this finger print.

## III. PROPOSED SYSTEM:

The aim of our project is to provide a high security system using face recognition on Raspberry Pi board and send an alert to the authorized person, this will increase the security of our project.

The proposed work is as follows:

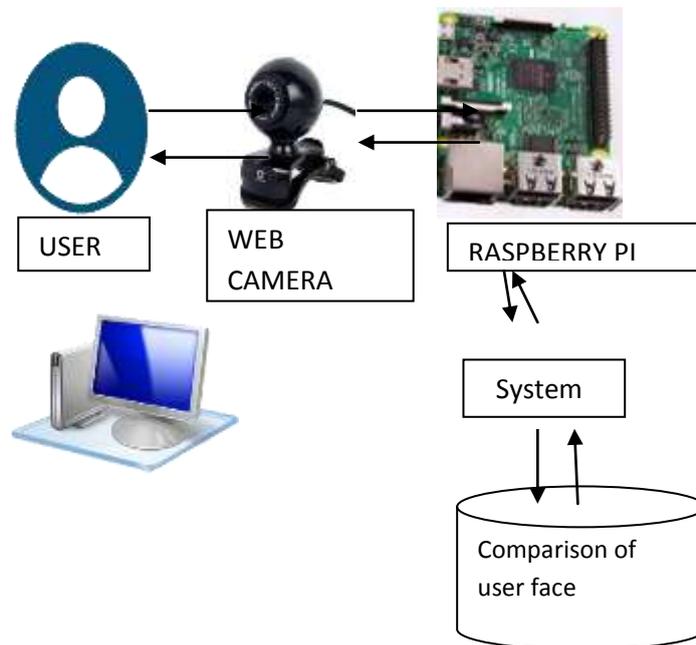
- Interfacing of camera module to capture live Face image.
- Create a database of authorized person.
- Capture current face, save it and compare with data base image.
- Interface relay as an output module.

**Advantages:**

- It's cheap, fast, highly reliable and provides enough flexibility to suit the requirements of different systems.
- It is Hygienic because there is no physical contact, no diseases can be spread.
- Avoid buddy punch.
- It is accurate to use.

**IV. SYSTEM ARCHITECTURE:**

In the shown figure1 there is a system architecture which devices are get connected. First web camera is connected with the raspberry pi. Then raspberry pi is connected with that of the system. The system contains database which contains user data. When the user will give the image it will check with the database and compare and send response to the user either positive or negative. The below figure explains the architecture.



**Fig 1: System Architectur**

The steps involved in the architecture is as follows:

- 1) User will give the image to the web camera.
- 2) After giving the image the web camera will capture the image and it will send to the raspberry pi.
- 3) The raspberry pi will accepts the image and it will send to the system.
- 4) The System will check in the database if the given image is matched or not with a threshold value.
- 5) If the image is matched "attendance is done".
- 6) If the image is not matched the mail alert will be sent to the authorized person.

**V. MODULES:**

- Creating the database
- Capturing the image
- Face detection and segmentation using Haar casc-ades
- Face recognition using LBP
- Attendance marker

**Creating the database:**

A database of all the students will be created using python and open cv. It is a onetime process so that we will have a real time database to train our system and to match the captured faces. For creating a person's database, the person has to sit in front of the camera around 80cms away from the camera with light on the opposite side of the face. The camera must be at level of the face of the person. When the code runs, the person has to give 8 poses with different expressions so that a database of different types of photo gets made. The poses can be looking sideways, up down or any direction in which face is visible. The expressions to be recorded can be happy, sad, bored, yawning etc. Also the faces detected from the captured images will be added to the database so that the database is updated continuously. Once the database is done we are ready with our implementation part.

**Capturing the image:**

A high definition camera will be installed in the classroom above the board so that it could capture all the students present in the class. The camera can be manually controlled or programmed as per the choice of the user. After capturing the image, it will be sent to the system for further processing.

**Face detection and segmentation using Haar cas-cades:**

Once the system gets the input image, it will be processed and all the faces present will be detected using the haar cascade feature of open cv. The image then will be segmented to all the faces present and will be stored in a file for that particular date.

**Face recognition using LBP:**

Once all the faces are segmented into different faces, we will run the face recognition code. Each of the faces from the particular date folder will be checked with the database using the local binary pattern algorithm and if similar face is found the photo will be added to the database for better efficiency in future.

**Attendance Marker:**

If a face from the particular date folder is matched with the database, then the particular student will be marked present. Following the same procedure, we will have list of all students who were present in the class. Rest of the class students will be marked absent.

**VI. PROPOSED ALGORITHM****Haar -like features:**

There is different type of haar feature which are applied on the image to see whether the feature exists in the image. The 24X24 window is applied all over the image, in which for each operation, we are subtracting the sum of pixels in white region with, sum of pixels in black region which output an integer value, that determines the validation of the corresponding feature.

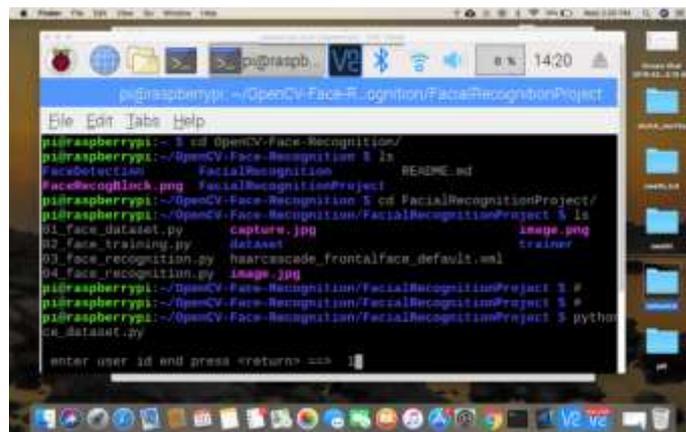


Fig 2: Haar cascade features

The steps are:

1. Convert the image into grayscale image.
2. For each pixel in the image, take neighborhood of size = r(say 8) around the pixel
3. Calculation of the LBP value goes like this. Any value greater than the Centre pixel(threshold) is assigned a value 1 whereas others lesser than the threshold are assigned 0.
4. They are then written in an array of length 8 in clockwise or anti-clockwise direction from the first pixel.
5. Finally, the LBP value for the pixel is calculated by multiplying the values of the array with values equal to 2 raise to the power the index of the array.
6. This LBP value for each pixel stored in a 2D array of size same as the size of the input image
7. A histogram of the frequency of each number over the cell (the number greater and the number smaller) is plotted.
8. The histogram is then normalized.
9. Thus, we get 256 bin histograms.
10. Finally, the histograms corresponding to all the cells are concatenated to get the final feature vector.

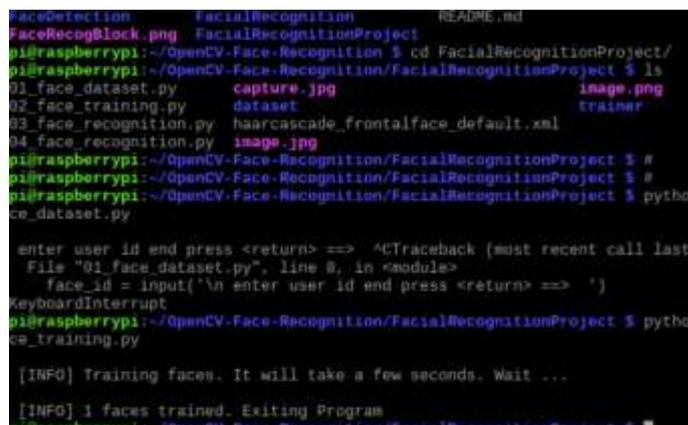
## VII. RESULT SCREENS:



```
pi@raspberrypi:~/OpenCV-Face-Recognition/FacialRecognitionProject
File Edit Tabs Help
pi@raspberrypi:~/OpenCV-Face-Recognition/
pi@raspberrypi:~/OpenCV-Face-Recognition $ ls
faceDetection FacialRecognitionProject README.md
faceRecogBlock.png FacialRecognitionProject
pi@raspberrypi:~/OpenCV-Face-Recognition/FacialRecognitionProject/
pi@raspberrypi:~/OpenCV-Face-Recognition/FacialRecognitionProject $ ls
01_face_dataset.py capture.jpg image.png
02_face_training.py dataset trainer
03_face_recognition.py haarcascade_frontalface_default.xml
04_face_recognition.py image.jpg
pi@raspberrypi:~/OpenCV-Face-Recognition/FacialRecognitionProject $ #
pi@raspberrypi:~/OpenCV-Face-Recognition/FacialRecognitionProject $ #
pi@raspberrypi:~/OpenCV-Face-Recognition/FacialRecognitionProject $ python
ce_dataset.py
enter user id end press <return> ==> 1
```

Fig 3: User will give Id

As shown the below figure the user will enroll with his id and name. After enrolling the image will be taken. For each user it will take 30 poses.



```
faceDetection FacialRecognitionProject README.md
faceRecogBlock.png FacialRecognitionProject
pi@raspberrypi:~/OpenCV-Face-Recognition $ cd FacialRecognitionProject/
pi@raspberrypi:~/OpenCV-Face-Recognition/FacialRecognitionProject $ ls
01_face_dataset.py capture.jpg image.png
02_face_training.py dataset trainer
03_face_recognition.py haarcascade_frontalface_default.xml
04_face_recognition.py image.jpg
pi@raspberrypi:~/OpenCV-Face-Recognition/FacialRecognitionProject $ #
pi@raspberrypi:~/OpenCV-Face-Recognition/FacialRecognitionProject $ #
pi@raspberrypi:~/OpenCV-Face-Recognition/FacialRecognitionProject $ python
ce_dataset.py
enter user id end press <return> ==> ^CTraceback (most recent call last)
  File "01_face_dataset.py", line 8, in <module>
    face_id = input('\n enter user id end press <return> ==> ')
KeyboardInterrupt
pi@raspberrypi:~/OpenCV-Face-Recognition/FacialRecognitionProject $ python
ce_training.py
[INFO] Training faces. It will take a few seconds. Wait ...
[INFO] 1 faces trained. Exiting Program
pi@raspberrypi:~/OpenCV-Face-Recognition/FacialRecognitionProject $
```

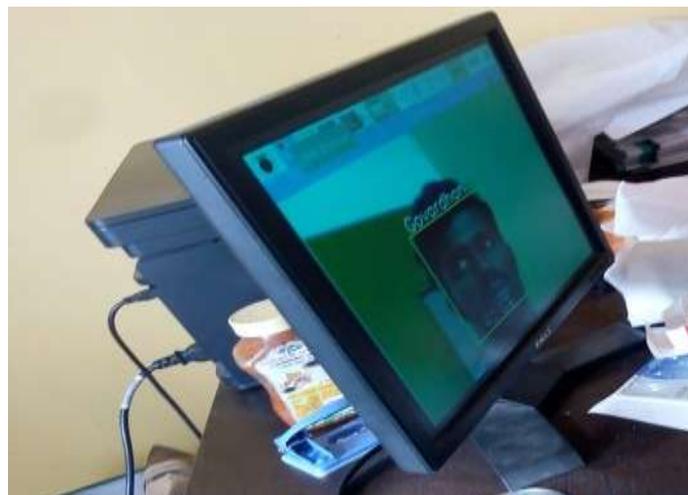
Fig 4: Images are loading

In the fig 4 all the trained images will be get loaded and user should give the image through web camera.



**Fig 5:** User Sitting infront of web camera

In the above figure the user will give image through web camera which is connected with that of raspberry pi and to the system.



**Fig 6:** Image captured

In the above figure it will shows the image with the name and the threshold value that we set with in a range.

```
[INFO] 1 faces trained. Exiting Program
pi@raspberrypi:~/OpenCV-Face-Recognition/FacialRecognitionProject $ python
ce_recognition.py
^CTraceback (most recent call last):
  File "03_face_recognition.py", line 45, in <module>
    cv2.imshow('camera',img)
KeyboardInterrupt
pi@raspberrypi:~/OpenCV-Face-Recognition/FacialRecognitionProject $ python
ce_recognition.py
^CTraceback (most recent call last):
  File "04_face_recognition.py", line 116, in <module>
    minSize = (int(minW), int(minH)),
KeyboardInterrupt
pi@raspberrypi:~/OpenCV-Face-Recognition/FacialRecognitionProject $ python
on 04_face_recognition.py
Name: Nirmal
Status: Present
Time: 2019-03-20 14:24:50.651259
```

Fig 7: Showing attendance as present

In the above figure it is showing attendance as present with three attributes name, status and time.

```
[INFO] 1 faces trained. Exiting Program
pi@raspberrypi:~/OpenCV-Face-Recognition/FacialRecognitionProject $ python
ce_recognition.py
^CTraceback (most recent call last):
  File "03_face_recognition.py", line 45, in <module>
    cv2.imshow('camera',img)
KeyboardInterrupt
pi@raspberrypi:~/OpenCV-Face-Recognition/FacialRecognitionProject $ python
ce_recognition.py
^CTraceback (most recent call last):
  File "04_face_recognition.py", line 116, in <module>
    minSize = (int(minW), int(minW)),
KeyboardInterrupt
pi@raspberrypi:~/OpenCV-Face-Recognition/FacialRecognitionProject $ python
on 04_face_recognition.py
Name: Nirmal
Status: Present
Time: 2019-03-20 14:24:50.651259
pi@raspberrypi:~/OpenCV-Face-Recognition/FacialRecognitionProject $ python
on 04_face_recognition.py
Unauthorized Person - Email Sent
pi@raspberrypi:~/OpenCV-Face-Recognition/FacialRecognitionProject $
```

Fig 8: Unauthorized person detected

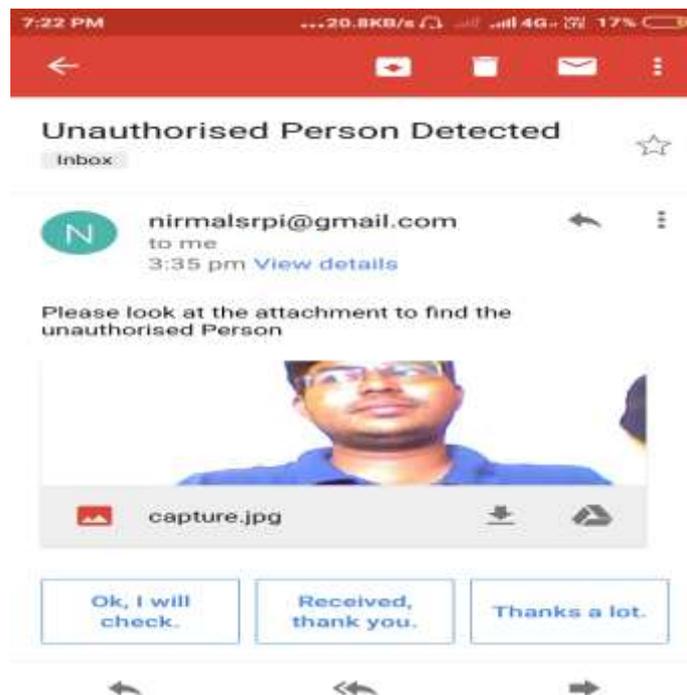


Fig 9: Sent mail to the authorized person

In the above figure it is showing that unauthorized person is detected and mail alert will be sent to authorized person.

## VII. CONCLUSION:

The purpose of our project is to mainly avoid the diseases like cold which is spreading through finger print scanner, that is hidden and not known to the user. In the proposed project we use facial readings to check and keep the attendance and if unauthorized person is detected the mail alert will be send.

## VIII. REFERENCE

- [1] B.K.P. Horn and M. Brooks, Seeing Shape from Shading. Cambridge, Mass.: MIT Press, 1989
- [2] A.F. Abate, M. Nappi, D. Riccio, and G. Sabatino, "2D and 3D face recognition: A survey", Pattern Recognition Letters, vol.28, issue 15, pp.1885-1906, Oct 2007.
- [3] Yael Adini, Yael Moses, and Shimon Ullman, "Face Recognition: The Problem of Compensating for Changes in Illumination Direction"
- [4] Kanan C, Cottrell GW (2012) Color-to-Grayscale: Does the Method Matter in Image Recognition? <https://doi.org/10.1371/journal.pone.0029740>
- [5] Grundland M, Dodgson N (2007) Decolorize: Fast, contrast enhancing, color to grayscale conversion. Pattern Recognition

## IX. BIOGRAPHIES



G. Divya zion,  
Assistant Professor,  
Dept. Of CSE, MTIET, Palamaner.



D. Govardhan, 15HR1A0522,  
Dept. Of CSE, MTIET,  
Palamaner.



B. Sravana Kumari, 15HR1A0510,  
Dept. Of CSE, MTIET,  
Palamaner.



K. Vani, 15HR1A0548,  
Dept. Of CSE, MTIET,  
Palamaner.



K.M. Karthik, 15HR1A0543,  
Dept. Of CSE, MTIET,  
Palamaner.



D. Madhu Sudhan Reddy, 15HR1A0521,  
Dept. Of CSE, MTIET,  
Palamaner.