Biometric Attendance Management System using Raspberry Pi

Varun Panditpautra[1], Adrija Goswami[2], Aishwarya Khavare[3], Prof. Sarita Ambadekar[4]

Department of Computer Engineering, K.J. Somaiya Institute of Engineering & I.T, Sion, Mumbai, Maharashtra, India-400022

[1]v.panditpautra@somaiya.edu, [2]adrija.g@somaiya.edu

[3]a.khavare@somaiya.edu, [4]sarita.ambadekar@somaiya.edu

***Abstract*- In every organization or company, attendance is an aspect that is of major importance and is recorded daily. Currently, in many institutions, attendance is recorded by using a sheet of paper on which students sign in front of their name. The major drawback of this system is that a student may forge his/her friends' signature even when they are absent. Keeping a track of the number and names of students present in the class is a time-consuming job and costs a lot of working hours of the faculty. In this paper, we describe ways to curb the existing problem through our biometric attendance management system with customized report generation. Raspberry Pi is used to transmit data and our hardware modules are connected to it. The system provides two ways-Fingerprint Module and Facial Recognition Module in order to record attendance uniquely by checking pre-registered data. The system, after recording attendance, sends data to Real-time database using Firebase and this data can be retrieved on the Web Application. The system is also able to generate customized report of attendance. Our system stands out from other existing biometric attendance management systems as, our system provides a fully-functional backup method to record attendance in case our primary method fails or takes inappropriately long time.**

Keywords-**Facial Recognition, Fingerprint Module, LBPH Face Recognizer, Haar Cascade, Firebase, Customized report**

I. INTRODUCTION

Till today, lecturers in, most universities are still found using the conventional methods of taking students' attendance either by calling out the student names or by passing around an attendance sheet for students to sign confirming their presence. In addition to the time-consuming issue, such a method is also at higher risk of having students cheating about their attendance, especially in a large classroom. This method can be replaced by a new method of recording attendance by using Biometrics and Facial Recognition. These two technologies can be implemented on a computer. But, using these on a computer will not allow the lecturer to record attendance of students. It can be recorded but it has many drawbacks, mainly, it is time consuming. To avoid this disadvantage, Biometrics (Fingerprint & Facial Recognition) can be used in Raspberry Pi, A credit card sized computer, to which various modules performing facial recognition and biometric can be attached. This will allow the system to be portable and secure at the same time. Thus, our system is compact as all the main functional unit get equipped within a portable handheld box.

Attendance Management is a very crucial part of an Institute as it ensures discipline and required performance measures by students. Thus, an accurate and efficient Attendance Management system is highly required and popular.

II. LITERATURE REVIEW

We have reviewed several IEEE papers relevant to our system which are as follows:

Initially Jennifer C. Dela Cruz, Arnold C. Paglinawan, Miguel Isiah R. Bonifacio, Allan Jake D. Flores, Earl Vic B. Hurna introduced Biometrics Based Attendance Checking using Principal Component Analysis [1] in the year 2010, where they proposed fingerprint identification through Arduino, Face detection and recognition using Viola-Jones Face Detection Method and Principal Component Analysis (PCA) through MATLAB respectively. We have used Raspberry Pi instead of Arduino because it connects easily to the Internet and we can use SD cards as the flash memory in the total system, allowing you to quickly swap out different versions of the operating system or software updates to debug. Because of this device is independent network connectivity, one can also set it up to access via SSH, or transfer files to it using FTP. Next, we reviewed the paper by Yash Mittal, Aishwary Varshney, Prachi Aggarwal, Kapil Matani and V. K. Mittal proposed in 2015 on Fingerprint Biometric based Access Control and Classroom Attendance Management System [2], in which the proposed Access Control System presents a scalable solution for identification and authentication purposes. The system is capable of replacing the existing RFID card-based Systems. The generic nature of the ACS makes it possible to extend the system to multiple locations and employ the technology in various situations. We did not use ACS in our project as our requirement includes student attending the lecture till it completes and not just access to lecture hall. Hence, we will circulate the hardware device to every bench. Next paper reviewed was End-to-end Encryption based Biometric SaaS [3] by Dr. Vinayak A. Bharadi, V. J. Kaul, Sameer Amrutia in the year 2015. In this paper a low-cost computer Raspberry Pi is used as a remote enrolment and authentication node. The enrolment part is done, and the captured data is sent to the cloud. The Fingerprint & face capturing sensors along with Wi-Fi adapter is interfaced to Raspberry Pi. We will make use of the fact that only specific characteristics, which are unique to every fingerprint, are filtered and saved as an encrypted biometric key or mathematical representation. No image of a fingerprint is ever saved, only a series of numbers (a binary code), is used for verification. The algorithm cannot be reconverted to an image, so no one can duplicate your fingerprints. The next paper was on Attendance Management System [4] by Shailendra, Manjot Singh, Md. Alam Khan, Vikram Singh, Avinash Patil, Sushma Wadar in the year 2015. This paper presents a design and framework for taking attendance and thereby making troublesome process of taking and compiling of attendance simple and efficient. This website uses MySQL database which is open source relational database management system (RDBMS). We have used Real-time Firebase database because here, clients from various platforms share one Realtime Database instance and automatically receive updates with the newest data. We also reviewed paper on Just IoT Internet of

Things based on the Firebase Real-time Database [5] by Wu-Jeng Li, Chiaming Yen, You-Sheng Lin, Shu-Chu Tung, and ShihMiao Huang in the year 2018. This paper designs an Internet of Things system called JustIoT. It can be used for educational purposes and help students to design IoT applications. Firebase Cloud had recently introduced the Cloud Functions feature which allows users to write programs in JavaScript language and put them on the Firebase cloud platform. We have used Real-time database for our project as Cloud Firestore is still in its beta testing stage and real-time database stores data as one large JSON tree which makes it very easy to store data. Lastly, we have reviewed and used the method described in A Real-time Face Recognition System Based on the Improved LBPH Algorithm[6] by XueMei Zhao, ChengBing Wei in year 2017 where they describe why LBPH face recognition algorithm is better than Eigenfaces and Fisherfaces face recognition algorithms as it can not only recognize the front face, but also recognize the side face, and can recognize faces under any light which is more flexible. Most of our facial recognition module is similar to what is proposed in this paper.

III. METHODOLOGY

Fig 1 shows the main flow of the system in order to record attendance and send it to database.

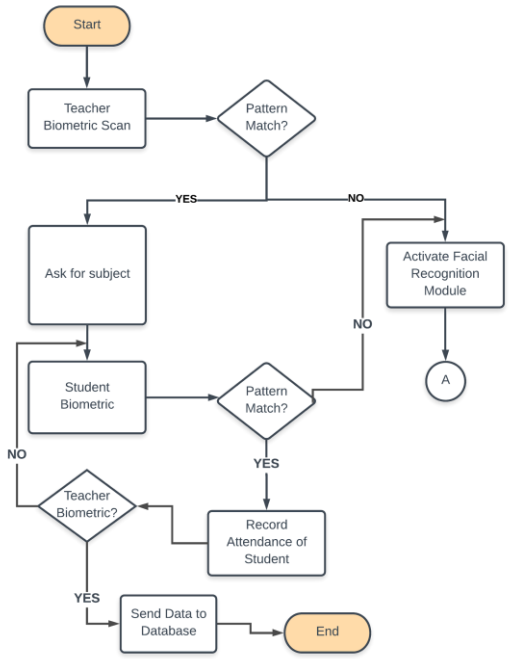

Fig 1. Block diagram of attendance recording process

A. Hardware and Software Modules

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. We have used Raspberry Pi 3 B+ to transmit data. This is the main hardware component of our system through which data will be transferred to online database. It has inbuilt Wi-Fi which helps to transmit data wirelessly, Optical Fingerprint Reader 307, as shown in Fig 2, to scan fingerprint and PiCamera module, as shown in Fig 3, to mark attendance by recording and validating the facial features of the user. We have used python as our programming language to interact with firebase. The attendance can we retrieved on a Web Application directly from Firebase. Fingerprint detection and Facial Recognition algorithms are executed to register as well as validate the respective biometric data.

B. Fingerprint Module

In our fingerprint module, the sensor is able to enroll new finger as well as detect already existing finger. Fingerprint data gets stored in the Optical Fingerprint Reader R307 Module itself. R307 can store up to 1000 fingerprints. R307 module is capable of independent fingerprint collection, fingerprint registration, fingerprint comparison (1:1) and fingerprint search (1: N) function. R307 can be accessed by using PyFingerprint Library.

Fig 2. Optical Fingerprint Reader R307

In our system, R307 module performs Fingerprint Collection, Fingerprint Comparison and Fingerprint Search. These features are performed at different stages of the project. Fingerprint Collection: A data set is created of new fingerprint samples, to be used further for recognition.

Fingerprint Comparison: The user needs to enter finger twice to validate the new finger. Once user enters finger second time, it gets stored at a new position. It gives an error saying “Fingers Don’t Match” upon comparison, in case user enters different finger second time during validation.

Fingerprint Search: Now, when an already enrolled finger is entered, the module checks whether it is already there in the data set by comparing and displays its position number.

C. Facial Recognition

In our face detection module, the camera is able to register a new face as well as detect already existing image displaying the name and confidence percentage. The recorded images will be stored in the dataset folder. The camera takes a predefined number of images before registering it in the dataset and training the classifier. Once the image is recorded, it is then used for training the classifier. In our case we have used Haar Cascade Classifier. For training the classifier we need to convert the detected face into a NumPy array which needs to be passed to the trainer.

Fig 3. PiCamera

There are 3 steps to train the classifier for facial recognition which are as follows: Data Gathering: Gather face data (face images in this case) of the people we want to identify. Train the Recognizer: Feed that face data and respective names of each face to the recognizer so that it can learn. Recognition: Feed new faces of that people and see if the face recognizer we just trained recognizes them. The idea with LBPH is not to look at the image as a whole, but instead, try to find its local structure by comparing each pixel to the neighbouring pixels. The names of those face recognizers and their OpenCV calls are: EigenFaces – cv2.face.createEigenFaceRecognizer() FisherFaces – cv2.face.createFisherFaceRecognizer() Local Binary Patterns Histograms (LBPH) – cv2.face.createLBPHFaceRecogniz Eigenfaces and Fisherfaces are both affected by light and, in real life, we cannot guarantee perfect light conditions. LBPH face recognizer is an improvement to overcome this drawback.

The LBPH Face Recognizer Process: In this method we take a 3×3 window and move it across one image. At each move (each local part of the picture), compare the pixel at the centre, with its surrounding pixels. Denote the neighbours with intensity value less than or equal to the centre pixel by 1 and the rest by 0.

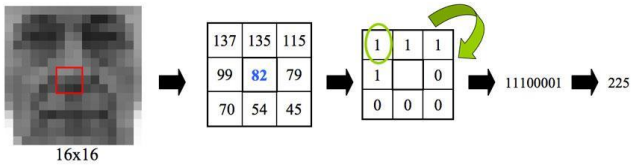

Fig 4. LBPH decimal value generation[7]

After we read these 0/1 values under the 3×3 window in a clockwise order, you will have a binary pattern like 11100011 that is local to a particular area of the picture. When we finish doing this on the whole image, we will have a list of local binary patterns. Now, after we get a list of local binary patterns, you convert each one into a decimal number using binary to decimal conversion. These steps are shown in Fig 4.[7] Then you make a histogram of all of those decimal values. A sample histogram is shown in Fig 5[7] where Y-axis represents frequency and X-axis represents decimal values.

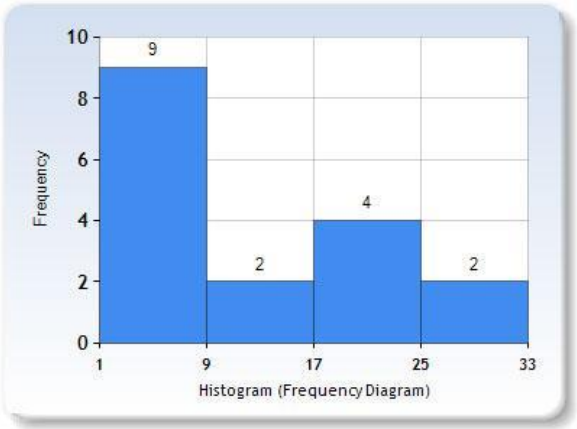

Fig 5. Sample Histogram[7]

At the end of this process we will have one histogram for each face in the training data set which will be stored for later recognition. The algorithm also keeps track of which histogram belongs to which person.

Later during recognition, the process is as follows: Feed a new image to the recognizer for face recognition. The recognizer generates a histogram for that new picture. It then compares that histogram with the histograms it already has. Finally, it finds the best match and returns the person label associated with that best match.

In the existing system, as mentioned in [7], using Fisherfaces for Facial Recognition we get an accuracy of 36.4% and while using Eigenfaces we get an accuracy of 15.09%. Whereas in our system we get an accuracy of 86.47%. This makes our system more accurate and reliable. Accuracy of our algorithm in comparison with other algorithms is as shown in Table 1. Fig 6 shows graphical representation of the comparison between various algorithms.

Algorithms	Detected faces	Correct	Accuracy %
Eigenfaces	1020	154	15.09%
Fisherfaces		372	36.4%
LBPH		882	86.47%

Table 1: Comparison of Exisitng and Proposed System

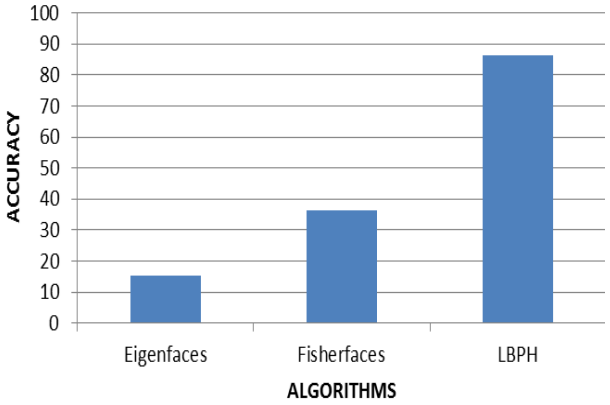

Fig 6: A comparison of accuracy

D. Firebase Module

Firebase is a mobile and web application development platform developed by Firebase, Inc. Firebase provides a real-time database and backend as a service. The service provides application developers an API that allows application data to be synchronized across clients and stored on Firebase's cloud. The company provides client libraries that enable integration with Android, iOS, JavaScript, Java, Objective-C, Swift and Node.js applications.

We have successfully created a database and stored data sent through raspberry pi in it. We have used Python to interact with Firebase. Fig 6 shows sample data displayed on Real-time Database on Firebase as received from Raspberry Pi.

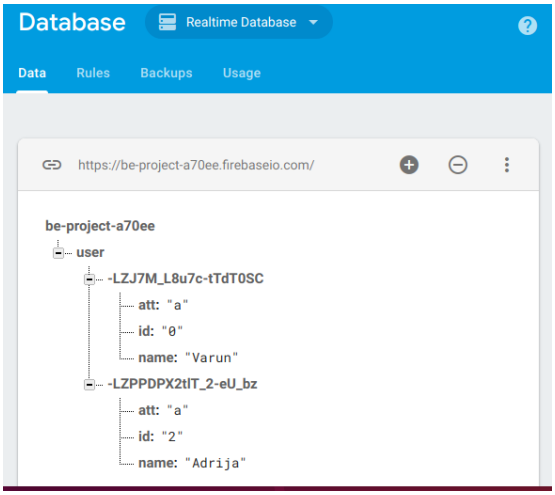

Fig 7. Sample Firebase Database

IV. INPUT AND OUTPUT

For the proposed system having two modules – Biometric Recognition and Facial Recognition, we have provided different inputs to the system.

A. Biometric Recognition

For the biometric recognition module, we provide fingerprint as the input. The input is a finger to the R307 Biometric Module which is then converted into a greyscale image as shown in Fig 8.

Fig 8. Input Fingerprint Image

Biometric Recognition is required in 2 parts of the system. First, while registering the fingerprint for the first time. This process takes places as shown in Fig. 9.

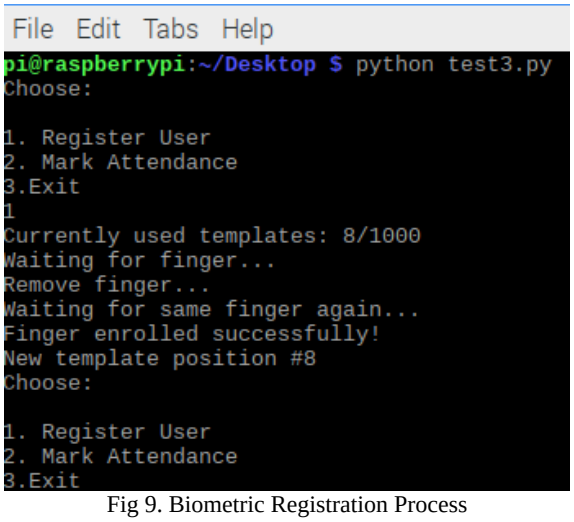

Fig 9. Biometric Registration Process

Second, Biometric Recognition is required while Marking the attendance. The procedure of marking the attendance is as shown in Fig. 9.

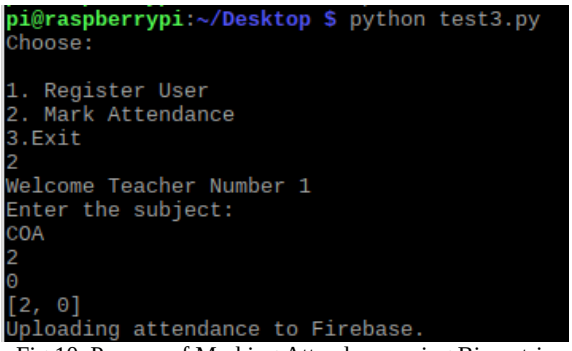

Fig 10. Process of Marking Attendance using Biometric

After the data is uploaded to Firebase it looks as shown in Fig 6. This data is then fetched by the Web Application which is used to generate reports.

B. Facial Recognition

While using the facial recognition module. The input given to the system is recorded initially by recording a dataset of each user. This dataset may contain images of the user's face ranging from 30-150 in number. More number of images ensures better accuracy. Recording the dataset shows the window as shown in Fig 11.

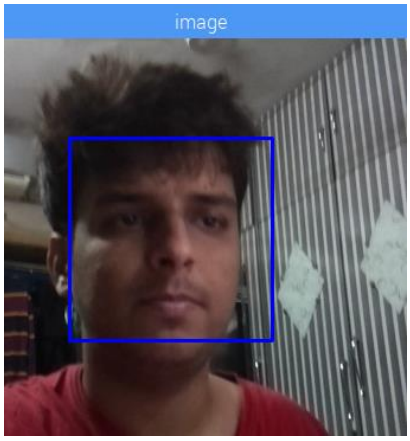

Fig 11. Recording Dataset

Once recorded, these images are then converted into greyscale for better results. Conversion of these images is done while storing the image in a folder. Image after converting to greyscale looks as shown in Fig 12.

Fig 12. Greyscale Images of Dataset.

After training the system using the dataset generated. Facial Recognition is performed. While performing Facial Recognition, the output is as shown in Fig 13.

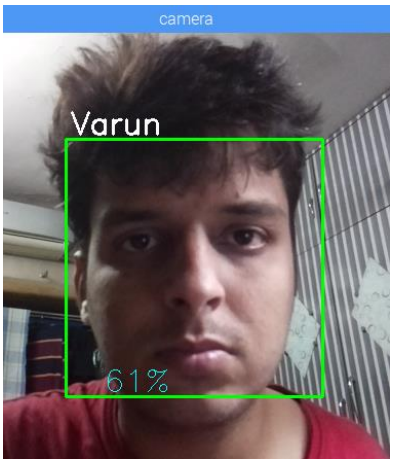

Fig 13. Output of Facial Recognition

V. CONCLUSION

In the present generation systems, there is a lot of time, data, space when it comes to keeping track of attendance. We have studied the overall flow of the system i.e. how the system would work, what would be the phases of the system. We have also studied the various algorithms through which the data can be transferred to the firebase module and retrieved on the web application. The code for the respective web application has been written in html language with the help of java-script for the purpose of connecting it to the real-time database on the firebase console. The web page will retrieve the data from the database and display the required data on the click of a button on the web page developed. The report generation is done on the basis of data retrieved from Web Application and can be custom according to the user. The outputs for all the trial runs were recorded. On observation, the designed system gives satisfactory results. Our system stands out from other existing biometric attendance management systems as, our system provides a fully-functional backup method to record attendance in case our primary method fails or takes inappropriately long time. Once after all the students have recorded attendance, the data gets sent to the Firebase.

REFERENCES

[1] Jennifer C. Dela Cruz, Arnold C. Paglinawan, Miguel Isiah R. Bonifacio, Allan Jake D. Flores, Earl Vic B. Hurna, "Biometrics Based Attendance Checking using Principal Component Analysis", School of EECE, Mapúa Institute of Technology Intramuros, Manila, Philippines, 2015

[2] Yash Mittal, Aishwary V arshney, Prachi Aggarwal, KapilMatani and V. K. Mittal, "Fingerprint Biometric based Access Control and Classroom Attendance Management System", Indian Institute of Information Technology Chittoor, Sri City, 2015

[3] Dr. Vinayak A. Bharadi, V. J. Kaul, SameerAmrutia, "End-to-end Encryption based Biometric SaaS", Thakur College of Engineering and Technology Mumbai, India, 2015

[4] Shailendra, Manjot Singh, Md. Alam Khan, Vikram Singh, AvinashPatil, SushmaWadar, "Attendance Management System", Army Institute of Technology, Pune, India,

[5] 2010Wu-Jeng Li, Chiaming Yen, You-Sheng Lin, Shu-Chu Tung, and ShihMiao Huang, "JustIoT Internet of Things based on the Firebase Real-time Database", 2018

[6] XueMei Zhao, ChengBing Wei, "A Real-time Face Recognition System Based on the Improved LBPH Algorithm", School of Electronic InformationQingdao UniversityQingdao, China, 2017

[7] Xiao-Ming Bai, Bao-Cai Yin, Qin shi, Yan-Feng Sun, "Face recognition using extended fisherface with 3D morphable model"

[8] https://www.superdatascience.com/blogs/ opencv-face-recognition