IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Smart Attendance Tracking Using Raspberry Pi And Native Mobile Application

[1]Devansh Taneja, [2]Dev Bahl

[1]Software Developer, [2]Analyst

Abstract: In recent times, as technology progresses, humans have opted for process automation to facilitate their job and lives. A face is an essential element of the human anatomy as it is the unique part of the body which can classify individuals separately. Face recognition systems may be created by exploiting facial features as a biometric identifying component. Regular monitoring of attendance is the most time-consuming duty in any company or organization. The main aim of this journal is to provide a feasible
"Smart Attendance Tracker" that is operated on a 'Small form factor' hardware to get optimum efficiency out of the system.
An Open CV-based facial recognition approach has been proposed as a workable solution. A Raspberry Pi running the Python script and an external Web camera is used to concurrently monitor attendance and update the following values in a Comma Separated Values (.csv) file. The CSV file acts as a dataset for the Mobile Application to display the attendance of people and other parameters.

I. INTRODUCTION

In today's busy world, every person needs a system that can make their work simpler. In any organization, attendance and human resource tracking and management is a very monotonous and challenging task. So, if the attendance or HR tracking can be achieved by an "Automated Smart Attendance Tracker" is controlled by a system and feasible solution and easy to integrate with minimum maintenance.

Smart Attendance Tracker creates encoding of the provided portrait images and saves them in a list with the name of the people with their corresponding face encoding/s. The attendance data is stored in an excel sheet (CSV format) and also on cloud storage that is automatically updated in the system with relevant date and time. The system is deployed on a Linux based system (Raspberry Pi) and attendance details can be tracked through a mobile application.

II. RELATED WORKS

A. "Eigenfaces For Recognition," Journal of Cognitive Neuroscience, vol. 3, no. 1, 1991 [1]

As described in [1], a computer system can find and track a subject's head, then recognize them by comparing facial features to known persons. Face recognition issue as a two-dimensional (2-D) recognition challenge. "Eigenfaces" are important characteristics since they are eigenvectors (principal components) of the face set.

B. "Fast Face Recognition Using Eigen Faces," IJRITCC, vol. 2, no. 11, pp. 3615-3618, November 2014 [2]

In [2], there are eigenvectors of a group of faces, which may not be linked to basic facial characteristics like eyes, nose, and mouth. The eigenface approach recognizes pictures using PCA. The system works by comparing a pre-extracted face picture to a collection of known face images. After imitating a face in the database, it will be classified as known or unknown.

C. "Robust Real-Time Face Detection," International Journal of Computer Vision, vol. 57, no. 2, pp. 137-154, May 2004 [3]

In [3], the research offers a face detection framework that can analyze pictures quickly while detecting faces accurately. A collection of face detection studies is presented. The technology detects faces as well as the best earlier techniques (Sung and Poggio, 1998; Rowley et al., 1998; Schneiderman and Kanade, 2000; Roth et al., 2000).

DRAWBACKS

The methods in the above-mentioned literature have some shortcomings that can be ruled out by using the proposed method:

- In [1], the method used to track/detect faces called 'Eigenfaces' is somewhat accurate and very hardware intensive.

- The technique in [2] neural algorithm called CNN is used. Again, it is an effective method but is very difficult to implement and requires learning and training making it resource intensive.

- In [3], computer vision is used and this is similar to the proposed approach. But the solution offered is not very accurate because the libraries used are obsolete and newer methods are more efficient and accurate.

- None of the methods track attendance and keeps track of the records.

III. PROPOSED SYSTEM

The desired solution is designed to tackle shortcomings of the existing systems. As per the survey or research, the current available systems are not fully utilizing the face-recognition library that is available to be freely used to design systems and subsystems. The proposed system will implement a low cost/ less hardware intensive, user interactive automated smart attendance tracker. The system will work through a native android application and attendance of individuals can be tracked with ease. The offered system should have the following characteristics:

- Less resource intensive

- Better feasibility

- Robust algorithm

- Flexibility of tracking

- High targeted accuracy

IV. METHODOLOGY

In this solution, we are creating a smart autonomous attendance tracking system with appropriate analytics and statistics. The respective analysis can be tracked and recorded on an Android mobile device through a native mobile application. The system achieves the purpose in four phases:

- Dataset Creation

• The dataset is formed with the use of native mobile application.
• Add employee page in the Android mobile application allows Administrator to add an employee after entering relevant information like Name, Unique identity number, Mobile number and a face image (passport or mugshot).
• Once this process is completed from the mobile application, the data gets stored on the cloud database (Firebase).
• This No SQL cloud database stores the data in form of nodes and information can be easily retrieved from it.

- Model Training

• Face recognition system or backend is trained for the fetched images in the local storage
• Post completion of the training phase, the model is ready for feature classification/data comparison from the trained images.

- Face Detection

• The model/backend is now triggered to detect face using a webcam attached to the system.
• The model proposed has multi-mode face recognition which allows not only single face to be detected at a time but also permits multiple faces to be detected at the same time.
• Once the run cycle is completed, the backend model appends data entries in the cloud database for frontend/mobile application to fetch and display outcomes/results.

- Data Tracking

• The frontend native mobile application is used to retrieve data records from the cloud storage/csv file and display results based on attendance and parameters allowing the user to have certain useful features as well.

SYSTEM DIAGRAMS

A) Block diagram of the Smart Attendance Tracker

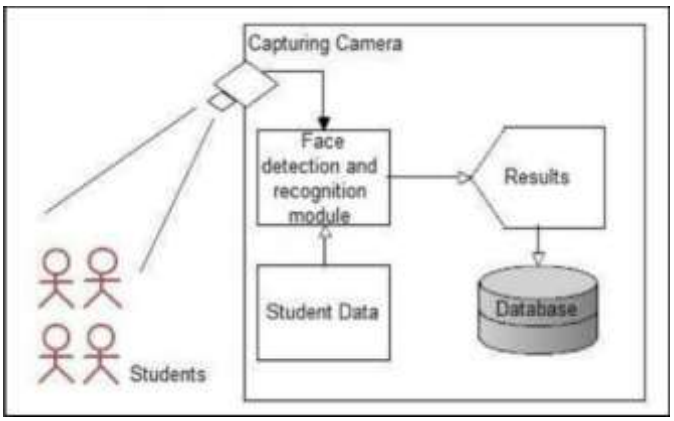

Fig 4.1: Block Diagram

The block diagram signifies and simplifies the entire system into basic blocks also called subsystems. The system is divided into students/pupils, Capturing camera, Face recognition, Student data, results and Database.

B) Flow Diagram of the Proposed System

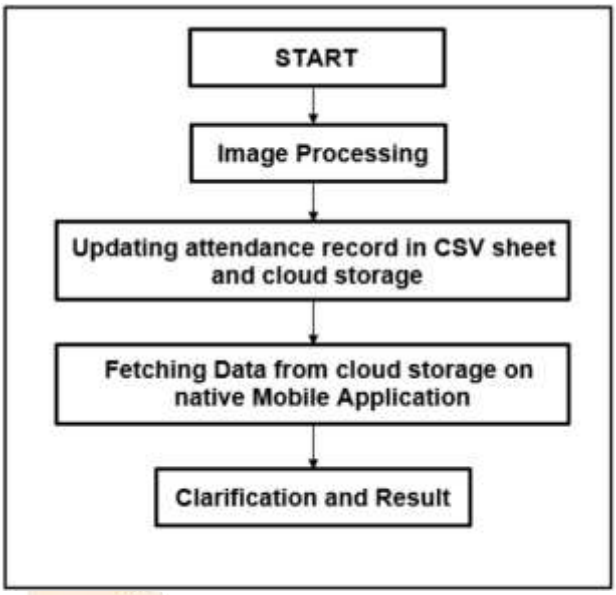

Fig 4.2: Flow Diagram

The flow diagram is a flowchart from the start to the result set. The system starts with image processing. Once, image processing is done, the records are updates in cloud storage and csv file in local storage. Once the cloud storage and csv file is updated with records, the native mobile application fetches these records. These records after getting fetched is then displayed and classified in the mobile application.

C) System Architecture Diagram with subsystems

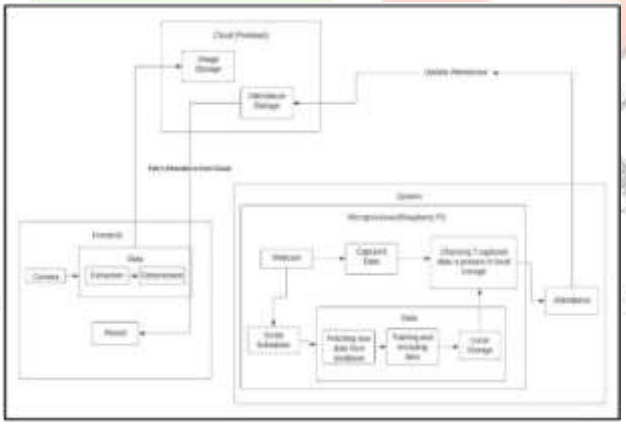

Fig 4.3: Architecture Diagram

The architecture diagram classifies the subsystems into frontend, backend and cloud storage. The frontend is controlled using native mobile application and s used to show results/attendance after data extraction. The cloud storage contains the attendance, pupil list and other details for the frontend and backend to function in sync and perform efficiently. The backend system (typically Raspberry Pi) uses webcam and fetched data from the cloud storage to train the system and detect pupil/s and update the attendance set in the cloud storage and CSV file.

V. RESULTS

- The code is currently fully working on a Windows/Mac and Linux based systems like Raspberry Pi.
- The script ‘app.py’ is run on Spyder (Anaconda3) or Thony IDE using Python 3.9.7 in the root environment with dependencies like OpenCV, Dlib, Cmake and Face-Recognition installed using ‘conda’ and ‘pip’.
- The script ‘Training.py’ is run as a script scheduler after specified time to train the system for any new data entries in the system.
- The script ‘app.py’ when run using the IDE corresponds with the training images and creates a list according to the name of the images and triggers the WebCam (Default or External) whichever connected to detect and recognize the face.
- Post the recognition in correspondence to the training images, the script is written in a way that appends the CSV file which in this case is ‘Attendance.csv’ and Firebase Cloud storage simultaneously with detected person’s name and timestamp.

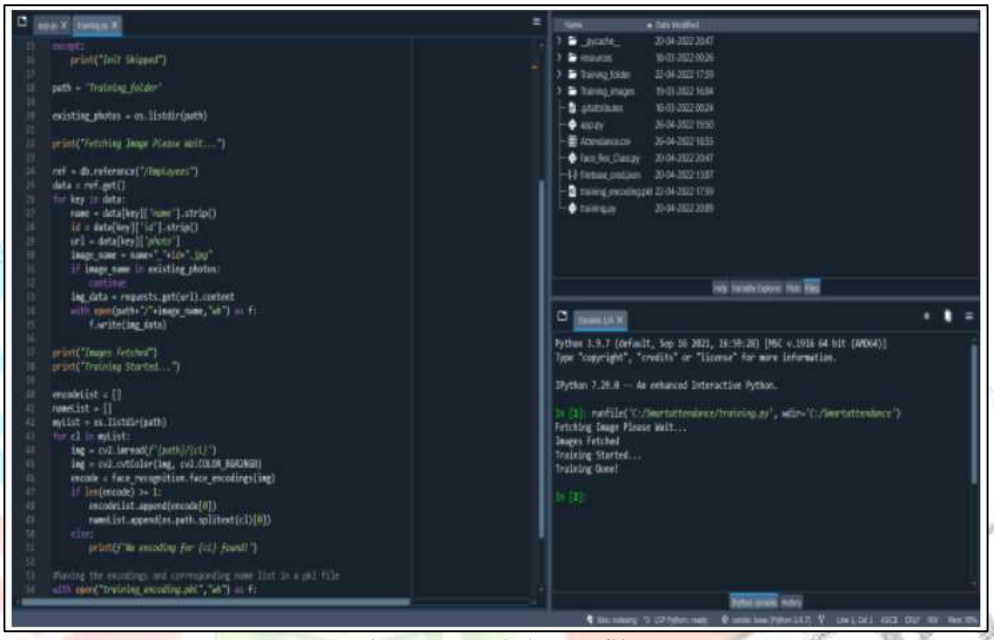

Fig 5.1: Trainig.py file

This image shows the training of the backend system getting done while fetching data from the cloud database.

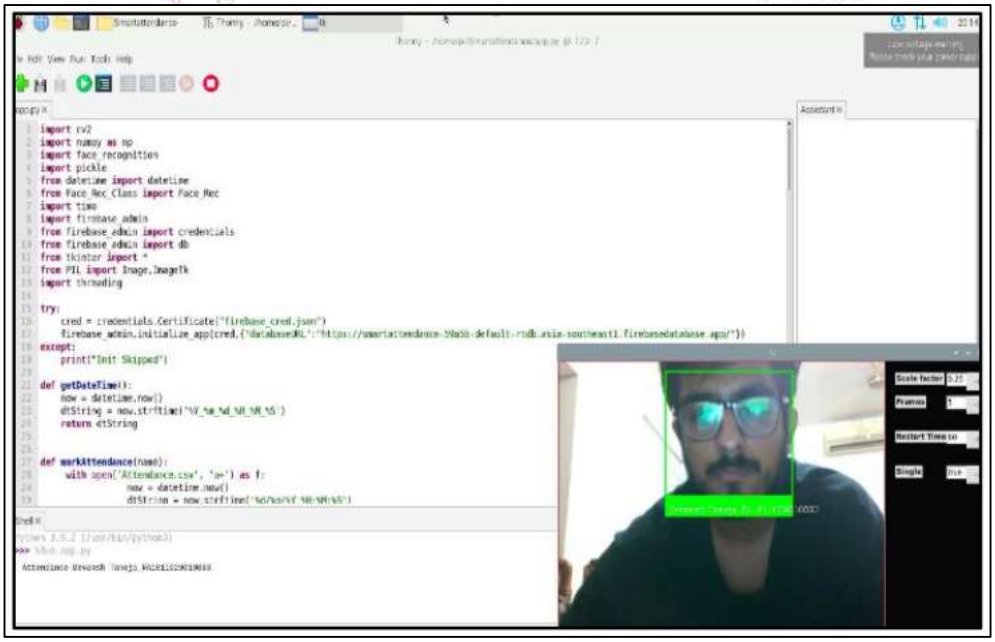

Fig 5.2: App.py file

This image shows the main app file getting executed along with webcam capturing live feed and the system detecting a person's face and updating cloud database.

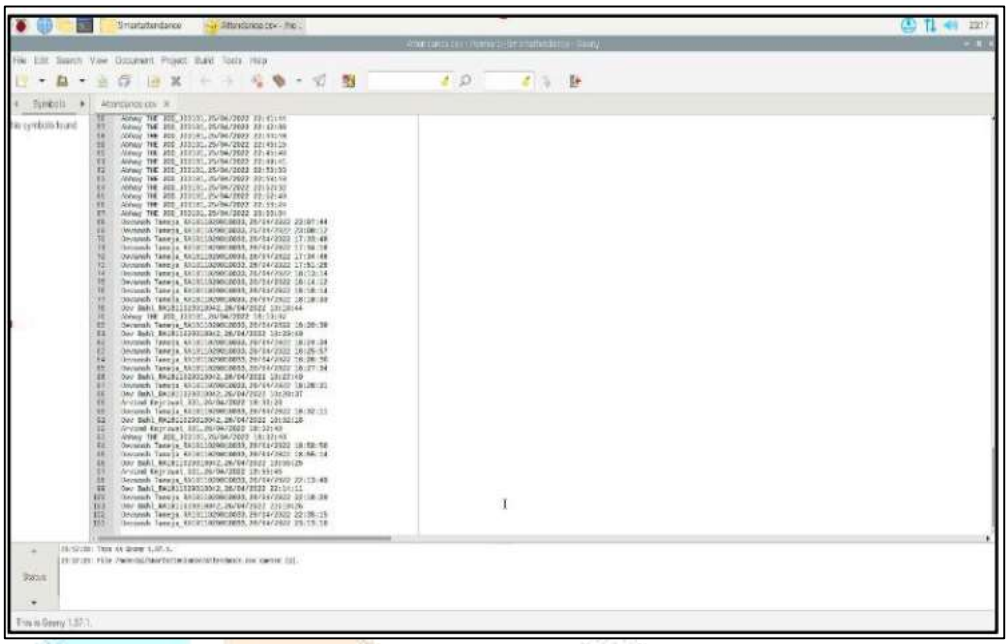

Fig 5.3: Attendance.csv file

This image shows the training of the backend system getting done while fetching data from the cloud database.

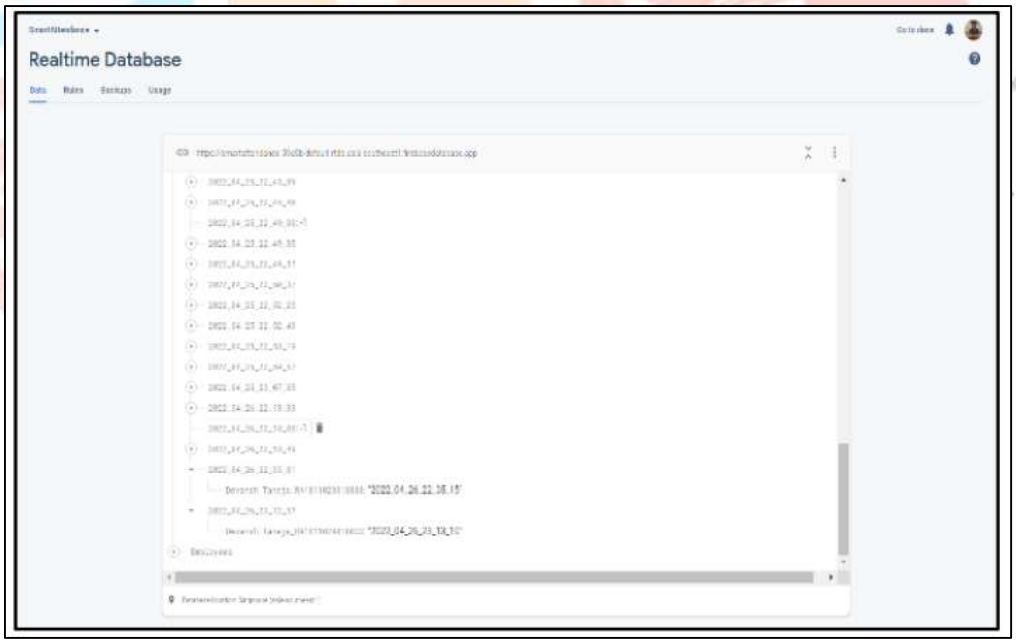

Fig 5.4: Cloud storage snapshot

This image shows the real-time firebase cloud storage along with the nodes and the data entries.

Fig 5.5: Mobile application home screen

This image is the home screen of the native mobile application which displays the overview of the attendance records in a progress bar and presentism and absenteeism in card view.

Fig 5.6: Mobile application analytics screen

This image is the analytics screen and shows the overall attendance of pupils and other relevant details of attendance for better understanding.

VI. CONCLUSION

The Smart Attendance Tracker using Raspberry Pi and Native Android Mobile Application worked successfully and efficiently. The web camera recognizes the face and the script uploads the attendance as a record to the cloud database and the comma separated values file in the local storage of backend subsystem or Raspberry Pi.

The system has the capability to record attendance of not just a single human but also multiple humans at an instance. With this, to enhance flexibility in terms of recognition speed, the system has a feature of inputting number of frames to be recognized before successfully recording the attendance. The system also has the capability to reset automatically after a designated duration of time. By default, the number of frames is set to 5 and reset time to 24 hours.

The mobile application then fetches the records from the cloud database and CSV file and displays attendance in terms of individual and overall attendance in a day. The application has the ability to send reminder SMS alerts to the people whose attendance is not recorded in due time. The overall attendance of the organization is available in the analytics screen in the application.

The entire system as a whole fulfils the desired purpose using the above technologies.

REFERENCES

[1] Matthew Turk and Alex Pentland: “Eigenfaces for Recognition," Journal of Cognitive Neuroscience, vol. 03| January 1991.

[2] Arun Vyas and Rajbola Tokas: "Fast Face Recognition Using Eigen Faces," IJRITCC, vol. 2, no. 11, pp. 3615-3618| November 2014.

[3] Paul Viola and Michael J.Jones: “Robust Real-Time Face Detection" International Journal of Computer Vision, vol. 57, no. 2, pp. 137-154| May 2004.

[4] Diaa Salama AbdELminaam ,Abdulrhman M. Almansori,Mohamed Taha,Elsayed Badr: “A deep facial recognition system using computational intelligent algorithms”| December 2020.

[5] Shahrin Azuan Nazeer, Normah Binti Omar and Marzuki Khalid: “Face Recognition System using Artificial Neural Networks Approach” , DOI:10.1109/ICSCN.2007.350774| March 2007.