Abstract - Covid-19 has generated a paradigm change in day-to-day work that has never been seen before. In today’s environment, social distance and limited physical contact are required. Despite these challenges, educational institutions must continue to operate. Most institutions’ existing techniques for documenting classroom attendance entail manually putting the student’s signature on papers and then uploading them into the system for further analysis. This technique is currently outdated and antiquated. Furthermore, it is harmful since it contradicts the contactless recommendations. This article offers a method in which a user may utilize facial recognition to register his or her attendance, and our system will automatically generate all of the data for the instructors to evaluate in a painless yet efficient way.

Key Words: Attendance System, Facial Recognition, Web-based application, Online classroom, Open source

1. INTRODUCTION

Face recognition remains the most popular among various means of identification, such as fingerprints [1], RFID [2], passcodes, and so on. Face recognition is still developing to predict [3] and verify the person, even when the others are accurate. According to the present COVID-19 scenario, it is essential to have a basic contact system and keep meticulous attendance records. The old method of collecting attendance puts users in danger of transferring diseases and viruses. In order to find a solution, this web-based application was created to offer a contactless attendance system that uses a facial recognition model to track students’ attendance [4].

Furthermore, it automates all available statistical analyses to give instructors and students hassle-free attendance monitoring and eliminates the need for any personal interaction. In part 2, the study opens with an introduction and a literature review. The suggested system is described in Section 3. In section 4, the suggested solution and approach are presented in detail. The experimental findings are discussed in Section 5. In the concluding section, part 6, the study concludes with a conclusion and future scope.

2. RELATED WORK

2.1 Literature Review

The system must be built using a real-time facial recognition model that offers correct service with the fewest potential differences. After examining the different models stated in [5][6][7][9][13][14], it was determined that the ‘python3 face-recognition program built by Adam Geitgey [4] offered the greatest accuracy. The reported accuracy on the Labeled Faces in the Wild benchmark dataset is 99.38 percent [4], and this package is constructed utilizing dlib’s state-of-the-art face recognition, a deep learning-based architecture. For facial recognition, [5] uses a 20-layered CNN model. The suggested application [5] falls short when specifying a reliable server and database architecture for real-time attendance management.

For face detection, [6] uses the Dlib CNN library. This system’s data analytics are inaccessible, and the system fails to communicate with its stakeholders through emails or alerts.

A haar-cascade classifier paired with local binary pattern histograms (LBPH) was used to create the face recognition model in [7]. It is a SQL database that’s been utilized. The database architecture of the proposed system [7] is poor, resulting in restricted data analyses.

Face recognizers Haar Cascade and LBPH are utilized in [9]. This study continues working on improving the accuracy to above 90% and making the gadget more portable. The data analytics offered to the company’s stakeholders were insufficient.

An enhanced Alexnet model is utilized in [14], which has a faster training time (109ms), fewer network parameters, and a faster feature extraction time (98ms). Although the idea is intriguing, the program as a whole falls short in many areas, including the lack of a user-friendly interface, preliminary data analysis, and lack of multiplatform compatibility.

To address the inadequacies of the previous examples, BeThere presents a comprehensive application with an exacting database management system. It provides in-depth data analytics to its consumers to present them with additional information. It also supports mobile and computer devices on a multi-device basis. Furthermore, a ‘Google Classroom’ [8]-like architecture is implemented into this system to relieve students’ reliance and instructors from manual attendance marking for a wide variety of courses taught in various batches. This structure allows each student’s enrolment to be marked independently for attendance. The classroom architecture, when combined with a face recognition system for recording attendance, provides a plethora of opportunities for the educational institution to keep a calculative and statistical record of students, teachers, attendance, absentees, mass bunks,
online coursework, and email facility that existing systems [5][6][7][9][13][14] fail to provide collectively in one single system.

Table 1

<table>
<thead>
<tr>
<th>No.</th>
<th>Features</th>
<th>Four Recognition Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>None</td>
<td>Normalize</td>
</tr>
<tr>
<td>2</td>
<td>None</td>
<td>Apply PCA</td>
</tr>
<tr>
<td>3</td>
<td>None</td>
<td>Train model on LFW</td>
</tr>
<tr>
<td>4</td>
<td>None</td>
<td>Test accuracy</td>
</tr>
</tbody>
</table>

**2.2 Mitigating bias in visual recognition and data security**

Most commercial face analysis methods favor one race, ethnicity, culture, age, or gender over another [10]. The system uses Adam Geitgey’s python facial recognition model [4]. The model is based on Labeled Faces In The Wild (LFW) and has a 99.38 percent accuracy rate. The LFW dataset has a uniform distribution of demographics [10], including a wide variety of faces belonging to diverse ethnicities, races, and genders, which helps to reduce bias in visual face identification.

Many online applications are still concerned about cross-site reference forgery (CSRF) [11]. The suggested system uses CSURF, an express.js middleware, to limit the potential of a data breach resulting in the dissemination of sensitive information, particularly user photos. Each user request produces a random CSRF token and stores it as a session cookie for that user's authentication. Furthermore, all sensitive data is safeguarded by storing sensitive data as a hash, such as passwords or personal information like a home address and phone number [12].

**3. PROPOSED SYSTEM**

**3.1 SYSTEM ARCHITECTURE**

- The training layer entails training the face recognition model using training photos for face identification, extracting the model’s features, and storing them in a database.
- Recognition layer/ML layer: This layer employs a pre-trained model to extract and recognize facial characteristics from the current real-time video frame given by the camera module and record attendance in the database.
- Web layer: Save the indicated attendance in a database so that APIs may be used for future analysis.

**3.2 SCOPE**

BeThere considers two main stakeholders: Teacher and the students. For Teachers, BeThere has following key features:

- Attendance recording: Teachers can accurately record each student’s attendance as well as the class’s average attendance. Teachers have access to a list of all enrolled pupils, as well as a list of defaulters.
- Monthly analysis: A graphical depiction of the number of lectures given, the number of people who attended, and the number of mass bunked people supplied.
- Filtering and Downloading: Teachers may sort students by Yea, Batch, and Class, as well as download the system’s color-coded attendance and absentee, excel sheets.
- Creating a one-of-a-kind code for student enrollment in the classroom: Each topic instructor may generate a one-of-a-kind classroom code for their subject and share it with their students for sharing coursework. BeThere includes the following essential characteristics for students:
  - Check average attendance for all courses: A student may view his or her average attendance for all courses in which he or she is enrolled.
  - Defaulters list: Students may see the attendance of courses highlighted in red because their attendance is less than the required level of 75.
  - Joining a classroom: Students may join a classroom by entering a unique code given to them by their subject instructor.
4. PROPOSED SYSTEM AND METHODOLOGY

4.1 TECHNOLOGY STACK

The technologies used to create BeThere are shown in the diagram below (Fig2). A javascript server (nodejs and expressjs), a python flask server, and a NoSQL MongoDB database are the main components of the technology. On the V8 engine, Node.js is a backend JavaScript runtime environment that is cross-platform and open-source. Handlebar is a templating engine that allows frontend and backend APIs to communicate with dynamic data. Express.js is a Node.js backend framework that is used to create APIs. A python flask server hosts the facial recognition model [4]. Opencv is also incorporated into the Python server to record real-time camera feeds. All of the data is kept in the MongoDB database, which allows several users to view and write the same data simultaneously.

4.2 System Modules

- User Registration/Authentication: When a user accesses the portal for the first time, he or she must register. All input fields, such as the email field, the roll number field, and the password field, are verified during registration. Students and instructors register individually to keep track of crucial information for future reference. Users must use the same credentials every time they log in.

- Face Recognition[4]: After a student has been enrolled, this module is in charge of collecting his or her attendance. It opens a camera and takes the face of the student in that frame, then saves the student’s attendance in the database based on those recorded faces.

- Classrooms: Teachers may establish topic classrooms using a unique class code and then manually or, using the code, add students to each class.

- Attendance module: Once a class is formed, this module calculates the average attendance of each student in each enrolled class, the average attendance of all courses, and separates the defaulters’ list.

- Email notification: Notifies students when they are added to a new class by a teacher or when they are placed in the defaulters’ category (attendance below 75)

- Statistics and graphs: The monthly graph analysis of both total lectures performed and total mass bunks by students are shown on both the student and instructor portals.

4.3 DATABASE DESIGN

The diagram below (Fig3) represents the database Entity Relationship for this system. The Database is a NoSQL based database i.e. Mongodb (Fig2).

The main entities in the database are users, classes, images and attendance. The internal relationship of these entities can be detailed as,
provided information on data analysis and features they would want to see in this system. Table III shows the system’s major performance metrics, which were computed after extensive testing with a class of 20 pupils. Because of COVID 19 rules, the experiment was done using an online meeting program. The meeting program allowed for a total of nine students to be seen in one view (Fig. 4), and the system correctly recognized the students with an average accuracy of 98 percent. A 2.3GHz two-core CPU with 8GB RAM was used to test the system. Figures 4, 5, 6, 7, 8, 9, 10 and 11 show the outcomes of a teacher presenting a lecture through an online meeting application using a web browser.

Table II

<table>
<thead>
<tr>
<th>Feature Analysis of CaptureIt!</th>
<th>Yes</th>
<th>python3 faceRecognition package(by Adam Geitgey)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Face recognition model used for CaptureIt!</td>
<td>Yes</td>
<td>faceRecognition package(by Adam Geitgey)</td>
</tr>
<tr>
<td>Mass Bank Indicators</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Statistical analysis of attendance and absenteeism</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Export to detailed color coded excel sheet</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Multi-platform</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Spoof detection</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Notification alerts</td>
<td>Yes</td>
<td></td>
</tr>
</tbody>
</table>

Table III

<table>
<thead>
<tr>
<th>Performance Analysis of CaptureIt!</th>
<th>15.245 seconds</th>
<th>30 fps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average camera start Latency</td>
<td>15.245 seconds</td>
<td>30 fps</td>
</tr>
<tr>
<td>Frame Rate of video stream</td>
<td>500</td>
<td></td>
</tr>
<tr>
<td>Maximum concurrent database connections</td>
<td>500</td>
<td></td>
</tr>
<tr>
<td>Maximum concurrent database operations</td>
<td>100 per second</td>
<td></td>
</tr>
<tr>
<td>Database read latency</td>
<td>1 millisecond</td>
<td></td>
</tr>
<tr>
<td>Response time for successful facial identification</td>
<td>0.428 seconds</td>
<td></td>
</tr>
<tr>
<td>Average notification sound delay</td>
<td>0.6 seconds</td>
<td></td>
</tr>
<tr>
<td>Accuracy of CaptureIt! in detecting faces in a class correctly</td>
<td>98%</td>
<td></td>
</tr>
</tbody>
</table>

6. CONCLUSION AND FUTURE SCOPE

This web-based application (BeThere!) offers a variety of extensive features that have yet to be explored in systems like this[5][6][7][9][13][14]. A comprehensive automated experience is provided by the smooth integration of face recognition and custom APIs and the interaction between students and instructors through the web application. It is also a goal to continue improving this application in the future. Because this is an open-source project, the open-source community will give more development iterations to improve the application’s usability and usefulness. Various experiments, such as UI changes, modular advancements for preventing spoofing when registering attendance, and
performance increases, have been deferred for the time being. Future work will use the classroom schema to include additional instructor and student interactions elements, such as quizzes and file sharing. Experimenting and testing to lower the latency of different tasks listed before may increase the system's performance. Several models for detecting liveliness will be evaluated to make the system spoof-proof. The application was created in the hopes of providing a safe, educational atmosphere in this coronavirus epidemic, according to the regulatory authorities' safety standards and requirements. Project URL: The technology has been implemented open source by Team BeThere! Anyone may download and install the system, contribute to developing new features by filing a pull request, and report defects.

Fig. 9. Student Dashboard

Fig. 10. Individual subject classroom with statistics

Fig. 11. Defaulter students list page for teachers

Fig. 12. Exported attendance xlsx sheet for teachers; color coding schema used: (Bad<less than 50%): Red, Average(50-60%): Orange, Good(60-75%): Yellow

Fig. 13. Absentee list for a particular day

REFERENCES


[12] Friedrich Wiemer, Ralf Zimmermann, “High-speed implementation of bcrypt password search using special-purpose hardware”, 2014 International Conference on ReConFigurable Computing and FPGAs (ReConFig14), doi: 10.1109/ReConFig.2014.7032529.
