International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 05 | May 2019 www.irjet.net p-ISSN: 2395-0072

Face-track : Smart Attendance System using Face Recognition

G.S SanthoshKumar[1], Martin Siby[2], Philson K Philip[3], Rajat Jaic Mendus[4], Shine Abraham[5]

[1]G.S SanthoshKumar: Professor, Dept. of Computer Science, Amal Jyothi College of Engineering, Kerala, India
[2]Martin Siby: Student, Dept. of Computer Science, Amal Jyothi College of Engineering, Kerala, India
[3]Philson K Philip: Student, Dept. of Computer Science, Amal Jyothi College of Engineering, Kerala, India
[4]Rajat Jaic Mendus: Student, Dept. of Computer Science, Amal Jyothi College of Engineering, Kerala, India
[5]Shine Abraham: Student, Dept. of Computer Science, Amal Jyothi College of Engineering, Kerala, India

---

Abstract - *Regular student attendance in the current academic system plays an important role in performance evaluation and quality monitoring. Calling names or signing on papers are the conventional methods practiced in most institutions, which is highly time consuming and insecure. This article presents for convenience or data reliability the automatic attendance management system. Using Face Recognition technology, the system is developed by integrating ubiquitous components to create a portable device for managing student attendance.*

Key Words: **Facetrack, Smart Attendance, AWS Rekognition, School Attendance, Machine Learning.**

1. INTRODUCTION

Student attendance administration is a critical content of the administration of school/college/university students. The traditional scenario of tracking student attendance is by calling out roll numbers of every student in every other period. This traditional method is inefficient and insecure (prone to proxies or impersonations) at the same time.

The main focus of this paper is to introduce the application of Smartphone in attendance system through the facetrack app. Nowadays Smartphone is very common to all the teachers so that they can easily take attendance and process the attendance where it needs.

To resolve this problem of attendance, many attendance management systems have been introduced in recent years. Jain developed a desktop based application in which students are given attendance by clicking a checkbox next to their name and then by clicking the register button to mark their presence. In 2013, Bhalla have proposed blue-tooth based attendance system. Application software installed in mobile phone enables to register the attendance via blue-tooth connection and transfer the notification to the instructor.

However, most of these systems have respective limitations in portability, accessibility, authenticity or cost. So an endeavor to overcome the shortcomings of the respective systems leads to the development of Facetrack based on face recognition. Unlike other biometric and non-biometric means of attendance system, face recognition technology stands tall with its unique advantages. Every student has a separate facial identity and it can not be faked by mere proxies. Moreover, the class teachers feel more acquainted with the student. The main benefits of this system is that the teacher can save time required in taking attendance for a class.

2. METHODOLOGY

2.1 Capturing frames from the app

From the live camera in the mobile app on clicking the screen, the mobile captures a frame and sends it to the AWS Lambda function through the AWS API Gateway. From the android application, the API call is made using Volley. This step happens only after the frame is base64 encoded. For detecting faces from the Frame, we have used a functionality from AWS (Amazon Web Services) called as Rekognition. The Lambda function invokes the detection and recognition functions which are used to identify and recognize faces in the image.

2.2 AWS Rekognition

The frame sent in to the AWS server indexes and matches faces already present in the user-made group called collection (Currently selected class).

Some of the functionalities we use from AWS Rekognition to enable these are:

- ***Search faces***: Searches for matching faces in the collection to which the face belongs for a given input face identification. Using the IndexFaces operation, you get a face identification when you add a face to the collection. The operation compares input face features with faces in the collection specified.
  The response to the operation returns a matching array of faces, ordered first by a similarity score with the highest similarity. More specifically, for each face match found, it is an array of metadata. The response includes a confidence value for each face match in addition to the metadata, indicating the confidence that the specific face matches the face of the input.

- ***Index faces***: The actual faces that are detected are not saved by Amazon Rekognition. Instead, the detection algorithm underlying the input image first detects the faces. The algorithm extracts facial characteristics into a feature vector for each face and stores it in the database of the backend.

2.3 Dashboard

The dashboard provides a mechanism to manage and manipulate the AWS Rekognition collections which are presently emulated as classes in this case. The dashboard presents a user-friendly interface to handle different functions.

The functionalities of the dashboard are:

1. Adding, deleting and managing classes :
   We can use this feature to add collections easily AWS Rekognition and check it's status, the faces present in it and an option to delete the class.

2. Adding face groups into individual classes:
   The dashboard provides a platform to add faces or face groups of students in a particular collection(class).

3. Viewing attendance taken:
   The dashboard lists the attendance marked using the app for a particular class at a particular time period.

3. ISSUES FACED IN DEVELOPMENT

Some of the major issues faced during development was sending the images taken from the phone to the server. As the size of the image increased, the response time from the server also increased. The difficulty faced was to compress the image to the right size without losing the quality of the image.

Another major issue faced was when video was taken as input for face detection and identification. Videos consisted of many frames and sending all of them to the server took a long time. Video based detection proved to be an impractical solution due to the high server response time. Videos can be processed but not immediately. It could be stored and processed later for information which is not ideal in this case.

4. TECHNOLOGIES AND HARDWARE USED

4.1 Hardware

To simplify its application we have designed it in such a way that it works even with low end mobile phones that are easily available. The application is developed in Android so android phones are preferred.

4.2 Technology used

Some of the technologies we have used to implement this mobile application are:

1. **Android Studio**

   Android Studio is the official integrated development environment (IDE) for the Android operating system of Google, built on the IntelliJ IDEA software of JetBrains and specifically designed for Android development. It can be downloaded from operating systems based on Windows, macOS and Linux. It is a replacement for the Eclipse Android Development Tools (ADT) as the primary IDE for native Android application development.

2. **AWS (Amazon Web Services)**

- *Rekognition*: It is a developer tool with multiple functions, including facial recognition, "pathing "— which involves tracking an object through a video frame, like a soccer ball — and finding and reading text in images and on video that is hard to see with the naked eye.
- *DynamoDB*: Amazon DynamoDB is a key value and document database that delivers performance at any scale of a single-digit millisecond. It is a fully managed, multi-regional, multimedia database with built-in security, backup and restore, and internet-scale applications in-memory caching. DynamoDB can handle more than 10 trillion requests per day and support peaks of more than 20 million requests per second.
- *Lambda:* AWS Lambda allows you to execute code without running servers. You only pay for the calculation time you consume-when your code is not running, there is no charge. For virtually any type of application or backend service, you can run code with Lambda-all with zero administration. Just upload your code and Lambda takes care of everything required to run and scale your code with high availability. You can set up your code to automatically trigger from other AWS services or call it directly from any web or mobile app.
- *API Gateway*: Amazon API Gateway is a fully managed service that makes it easy to create, publish, maintain, monitor, and secure APIs on any scale for developers. With a few clicks in the AWS Management Console, you can create REST and WebSocket APIs that act as a "front door" for applications to access data, business logic, or functionality from your backend services, such as workloads running on Amazon Elastic Compute Cloud (Amazon EC2), code running on AWS Lambda,

any web application, or real-time communication applications.

3. **Python Flask**

   Flask is a Python-written micro-web framework. As it does not require specific tools or libraries, it is classified as a microframework. It has no database abstraction layer, form validation, or any other components where common functions are provided by pre-existing third-party libraries.

4. **HTML and CSS**

   The standard markup language for creating web pages and web applications is Hypertext Markup Language (HTML). It forms a cornerstone technology trial for the World Wide Web with Cascading Style Sheets (CSS) and JavaScript.

5. PERFORMANCE OF THE SYSTEM

Right now, when we open the mobile app, we have the option to choose the class and once a class is chosen we are directed to a camera view on which when we capture a frame of a group of students, a few seconds later the name of the students detected will be displayed on the side and the count will be displayed below.

We have a dashboard portal parallel to the mobile app in order to create and manipulate classes. The attendance taken will be shown live on the online dashboard for each class respectively.

6. CONCLUSION

With the help of the mobile app we developed, we were successful in creating multiple classes, adding multiple face groups in each classes and then taking attendance via FaceTrack mobile app and verified it with our PC console dashboard.

REFERENCES

[1] Smart Attendance Monitoring System (SAMS): A Face Recognition based Attendance System for Classroom Environment by Shubhobrata Bhattacharya, Gowtham Sandeep Nainala, Prosenjit Das and Aurobinda Routray.

[2] Development of Smartphone-based Student Attendance System by Md. Milon Islam, Md. Kamrul Hasan, Md Masum Billah, and Md. Manik Uddin.