AUTOMATED ATTENDANCE SYSTEM

PROJECT REPORT

submitted by

SARAN PRASAD (TVE16CS052)

SONU SADANANDAN (TVE16CS056)

VIMAL SEBASTIAN (TVE16CS061)

VIVEK V (TVE16CS064)

to

the APJ Abdul Kalam Technological University

in partial fulfillment of the requirements for the award of the Degree

of

Bachelor of Technology

in

Computer Science and Engineering

Department of Computer Science and Engineering

College of Engineering, Trivandrum

Kerala

June 30, 2020

DECLARATION

We undersigned hereby declare that the project report **Automated Attendance System**, submitted for partial fulfillment of the requirements for the award of degree of Bachelor of Technology of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by us under supervision of **Dr Salim A**. This submission represents our ideas in our own words and where ideas or words of others have been included, we have adequately and accurately cited and referenced the original sources. We also declare that we have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in our submission. We understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place: Trivandrum

Date: June 30, 2020

Saran Prasad

Sonu Sadanandan

Vimal Sebastian

Vivek V

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
COLLEGE OF ENGINEERING, TRIVANDRUM

CERTIFICATE

This is to certify that the report entitled **"AUTOMATED ATTENDANCE SYSTEM",** submitted by **Saran Prasad**, **Sonu Sadanandan**, **VimalSebastian**, **Vivek V** to the **APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY** in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering is a bonafide record of the project presented by them under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Dr. Saritha R	**Dr. Salim A**
Assistant Professor	Professor & HOD
Department of CSE	Department of CSE
(Coordinator)	(Guide)

ACKNOWLEDGEMENT

It is with extreme respect that we remember the names of all who had been a great help and guidance throughout our sessions. Firstly, we would like to thank the Almighty for giving us the wisdom and grace for presenting our project. With a profound sense of gratitude, we would like to express our heartfelt thanks to our guide, **Dr. Salim A**, Professor, Department of Computer Science and Engineering for his expert guidance, co-operation and immense encouragement in pursuing this seminar. We are extending our gratitude to our project coordinator, **Dr. Saritha R**, Assistant Professor, Department of Computer Science and Engineering for her co-operation and support. We are very much thankful to **Dr. Salim A**, Head of the Department of Computer Science and Engineering, for providing necessary facilities and his sincere co-operation. Our sincere thanks is extended to all the teachers of the Department of CSE and to all our friends for their help and support.

Saran Prasad

Sonu Sadanandan

Vimal Sebastian

Vivek V

ABSTRACT

A lot of attendance management systems have been introduced in the past 3-5years. Many of them can be labelled nearly successful while the efficiency of most the models have to be questioned. Attendance of a batch of people whether in office or in a student classroom can be processed via individual recognition techniques like finger print or face recognition. Although the former stood as a reasonable solution it will result in the cost of such a project to be high as it is not feasible to have a finger print sensor in every classroom. The proposed project is an automated attendance system that helps teachers by simplifying the current attendance system. So a face recognition system with most efficient model would serve the need. Identification of the most efficient pre-trained model with one shot learning facility "FACENET" stood as the backbone of this project. The project also provided an mobile application to every tutor to ease the system. The whole system is controlled through a Command Line Interface which seems to be the heart and brain of the project. The registered attendance of a subject can be viewed through the subject specific google sheet titled with subject name and subject code. The proposed model seem to be successful in easing individual teacher efforts by providing an AUTOMATED ATTENDANCE SYSTEM.

This project is dedicated to ease the attendance process for all teachers who has been able to provide a meaningful insight to student minds.

Contents

List of Figures

List of Tables

ABBREVIATIONS

(List in the alphabetical order)

NOTATION

(List in the alphabetical order)

Chapter 1

Introduction

The proposed project is an automated attendance system that helps teachers by simplifying the current attendance system. The current attendance system consumes a lot of time and effort of the tutor. The motive of the project is to simplify it. After a lot of discussion we finalized to do it via Image Recognition. The system is aided by a Face detection algorithm which provides one shot learning namely "FACENET" which is the backbone of this project. The algorithm is based on a google research paper.The attendance updation involves teacher taking group photo of the class via a provided mobile app. The taken photo is sent to the server where it is compared with trained face cropped image data set of students of the specific class. The attendance stats is sent back to the teacher's application where he/she can cross verify it. The data after confirmation will be automatically updated in the resultant attendance google sheet of the particular subject. The mobile application provided contains an account specific to each tutor while the student as well as data is maintained in the database. The creation and maintenance of teacher,subject and google sheet is handled by an admin via a Command Line Interface,which controls the whole attendance model in a systematic approach. The google sheet is created and maintained via OAuth 2.0 protocol which access Google sheet API in javascript. The proposed model seem to be successful in easing individual teacher efforts by providing an AUTOMATED ATTENDANCE SYSTEM.

Chapter 2

Automated Attendance System

The proposed solution needed an accurate face-recognition model to correctly map present students in the class with their trained photo.Photo taken via a camera would have served the need.Since almost all of the teachers have smartphone present with them taking photo wont be much of a task.Three or more photos for cross confirmation make the system foolproof. The image is taken via an mobile application with separate teacher accounts through which images are sent to the server.At the server recognition of faces and appropriate attendance marking takes place.

The attendance stats are sent back to the teacher for cross verification within seconds.When the data is confirmed automatic updation of attendance takes place in the google sheet.For controlling the whole system a proper Command Line Interface system has been provided.It deals with the database containing student data,teacher data,Subject data as well as provides commands for the automatic creation of subject-wise attendance maintaining google sheet.

Chapter 3

Background and Literature Review

The proposed project is an automated attendance system that helps teachers by simplifying the current attendance system. The current system consumes a lot of time and effort of the tutor which can be reduced using this systematic project. By saving the significant time from the class hours will help teachers to provide meaningful insight to the students and help them. The current scenario and the proposed solution are discussed below.

3.1 Current Attendance System

The traditional attendance system is lengthy and takes away a lot of productive time from the teachers.The current tiresome system is difficult for teachers to maintain.
The process starts with teachers keeping a manual register of student names and their roll numbers in a register book or google sheet. This is followed by the much time consuming process of taking attendance stats of students in every teaching hour. This is followed by a cross verification to check whether noted attendance is correct. This usually consumes 5-10 minutes from every period and takes away quality time of the lecture.
Teacher has to mark the confirmed attendance in the register or google sheet. Although the introduction of google classroom has made this procedure much easier by linking google sheet with classroom so that students can always verify it,this system is not used by all the tutors.
Hence the idea was to provide an automatic system which would aid the teachers in this process.

3.2 PROPOSED SOLUTION

The proposed solution needed an accurate face-recognition model to correctly map the present students in the class with their trained photo. Photo taken via a camera would have served the need. Since almost all of the teachers have smartphone present with them taking photo wont be much of a task. Three or more photos for cross confirmation make the system foolproof.

The image is taken via an mobile application with separate teacher accounts through which images are sent to the server. At the server recognition of faces and appropriate attendance marking takes place.

The attendance stats are sent back to the teacher for cross verification within seconds. When the data is confirmed automatic updation of attendance takes place in the google sheet. For controlling the whole system a proper Command Line Interface system has been provided. It deals with the database containing student data,teacher data,Subject data as well as provides commands for the automatic creation of subject-wise attendance maintaining google sheet. The google sheets are created and maintained using OAuth 2.0 protocol.The google sheets API functions on the command of the Command line Interface.This is the proposed solution for the system.

3.3 SDLC

3.3.1 Planning and requirement analysis

The initial planning phase took around a week. This included bringing up the idea, and a quick overview of implementation. Later on, within the next two weeks a more structured plan was made. We planned about which parts to do first and then the respective order. The requirements included which software's to use and the data sets. The initial plan was to mount a camera at the front of each class, so that it captures the entire class. This photo is sent to the backend server where it is processed and the final results are updated in a google sheet. Later, instead of the mounted camera, we switched to mobile phone cameras which makes it easy for teachers. The dataset includes a single photo of each student at

the beginning. Identification of the most efficient pre-trained model with one shot learning facility “FACENET" stood as the backbone of this project. The software’s we planned to use were

- Node
- Angular
- MongoDb
- OpenCV
- Keras
- Express

3.3.2 Defining Requirements

- The front end or the user interface is built using Angular JS. AngularJS extends HTML with new attributes. It is perfect for Single Page Applications. AngularJS is a JavaScript framework. It can be added to an HTML page with a <script> tag. AngularJS extends HTML attributes with Directives, and binds data to HTML with Expressions.

- The backend consists of two parts. The face recognition part and the admin dashboard. The face recognition is done using the FACENET model. This model directly learns a mapping from face images to a compact Euclidean space where distances directly correspond to a measure of face similarity. Once this space has been produced, tasks such as face recognition, verification and clustering can be easily implemented using standard techniques with FACENET embeddings as feature vectors.
  The Admin dashboard or the command Line Interface is mainly focused for a single admin user to control the whole application. Addition of a user, subjects and students are done through this dashboard. The admin has access to every user’s attendance sheets. All the necessary edits and updates are done through the dashboard. The application is built in node js platform using javascript. The data are stored in the

backend using MongoDB. MongoDB is an open-source document database and leading NoSQL database. We use mongoose to interact with mongo. Mongoose is an Object Data Modeling (ODM) library for MongoDB and Node.js. It manages relationships between data, provides schema validation, and is used to translate between objects in code and the representation of those objects in MongoDB.Other modules used are readline, fs and crypto.

3.3.3 Designing the Product Architecture

The architecture includes a mobile camera through which the photos are taken and are send to the image processing server to match the students present. The later information is sent to rest api, and to a google sheet produced to represent the attendance. The admin dashboard controls everything. It has connection to whole system.

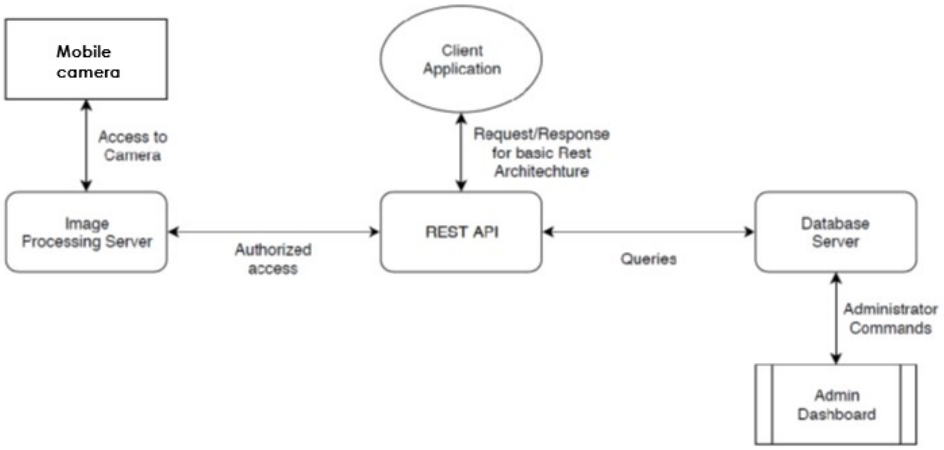


Figure 3.1: Flow/architecture diagram

3.3.4 Building or Developing the Product

The product took about one month for completion after all the requirements were acquired. Production started from first developing the face recognition system. The system requires only one image of a person to study his/her features. This was tested with a group photo consisting of 60+ students and faculties and it resulted in cent

percent accuracy. Later on, we started developing the admin dashboard and front-end user interface. The admin dashboard is built using node js. One of the major tasks raised was to edit the google sheets as required. It took about two weeks to complete the whole process.

Chapter 4

Methodology

The project introduces a simple mobile application for automated attendance system based on the FaceNet face embedding technique, which uses a one-shot training method to learn the collected data. As the initial step, a single image of each student is collected in the order of the corresponding semesters and the details of the teachers collected along with their current lecturing subjects. These tasks are consolidated by the admin along with handling the database using the provided command line interface(CLI). Admin hereby also generates distinct user id and password for each users for handling their corresponding accounts. By using the provided login credentials the user logs in to the system. Using the developed application user selects the subject to which he/she has to take the attendance followed by uploading the photos of the class.

Implementing the mobile application based on FaceNet for the automated attendance helps to reduce the tiring process of taking attendance to a great extend. Once the user gets logged in to their account, they can select the subject to which the attendance to be taken. Uploading the photos of the class using the application itself recognises the students who are currently present in the classroom and automatically marks the attendance to the google sheet of corresponding subject when the user confirms.

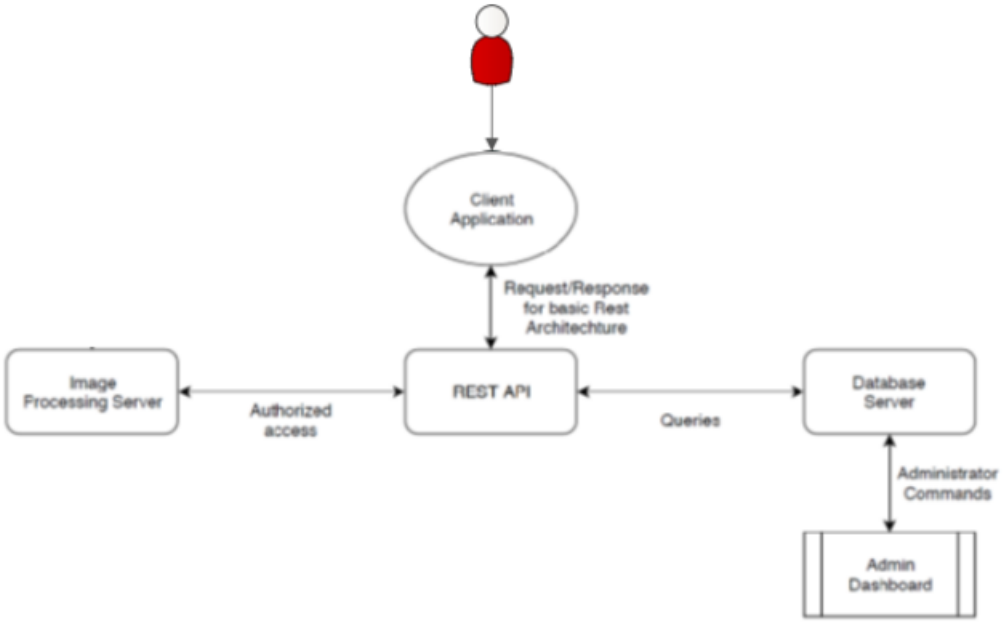


Figure 4.1: Workflow diagram

- The client interface for the Automated Attendance System is provided with a mobile application. Initially the users are provided with user credentials generated by the admin to login to their account. The Client application uses a session based authentication where the session is created and session data gets stored in the server memory when the user logins and session id is stored in a cookie on the browser. Once the User logs in, the application requests the Rest API for the details of the corresponding user and the required response is returned to the application.

- The Rest API is the bridge between the three main components Image processing server, Database server and the Client application. The requests and responses from the Rest API to all three components are OAuth2 authorization protected.

- The image processing server of the system is based on the FaceNet, which learns a mapping from face images to a compact Euclidean space where distances directly correspond to a measure of face similarity. When the space has been produced, learning can be implemented based on the FaceNet embeddings as the feature vectors. Once the photos of the class gets uploaded by the Client application, the Rest API will forward the

snaps to the image processing server where it will detect all the faces and compare the similarity between the learned training images for recognition. Followed by sending the result containing the list of present students to the rest API.

- The database of the application is based on MongoDB, which is an open-source and NoSql database. Schemas used in the application are User schema, Subject schema and Student schema. The rest API uses queries in order to access the data in the database based on the requests from the client application and the image processing server.

- An Admin dashboard is provided in order to control the whole application. It is a command line interface (CLI) to which only the admin has the complete access. The admin is entitled to add users, subjects, and students to the corresponding schemas in the database along with the necessary updation and deletion. The changes made by the admin will get reflected in the application.

Chapter 5

Design and Implementation

5.1 FACENET MODEL

Our strategy depends on learning an Euclidean installing for every picture utilizing a profound convolutional arrange. The arrange is prepared with the end goal that the squared L2 separates in the installing space directly correspond to face similarity. When this embedding has been delivered, at that point the previously mentioned undertakings become straight-forward: face check basically includes thresholding the separation between the two embeddings; acknowledgment turns into a k-NN order issue; and grouping can be accomplished utilizing off-theshelf methods, for example, k-implies or agglomerative clustering.

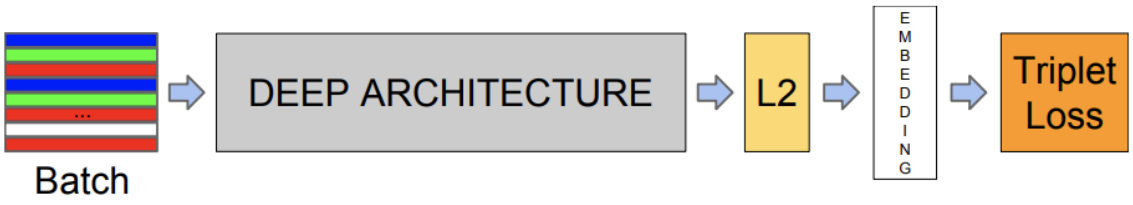


Figure 5.1: Facenet Model

FaceNet uses a deep CNN. FaceNet approaches in two different architectures, the 1x1 convolution networks and the recent Inception networks. To this end FaceNet utilizes the triplet misfortune that straightforwardly reflects facial embeddings to accomplish in face check, acknowledgment and grouping. Specifically, we make progress toward an installing f(x), from a picture x into a component space Rd , with the end goal that the squared sepa-

ration between all faces, free of imaging conditions, of a similar character is little, while the squared separation between a couple of face pictures from various personalities is huge. In spite of the fact that FaceNet doesn't straightforwardly contrast with different misfortunes, for example the one utilizing sets of positives and negatives, the triplet misfortune is increasingly appropriate for face confirmation. The inspiration is that the misfortune energizes all appearances of one personality to be anticipated onto a solitary point in the implanting space. The triplet misfortune, be that as it may, attempts to uphold an edge between each pair of appearances from one individual to every other face. This enables the appearances for one personality to live on a complex, while as yet upholding the separation and in this way discriminability to other personalities. The accompanying segment portrays this triplet misfortune and how it tends to be adapted effectively at scale.

5.1.1 Triplet Loss

Even though multiple architectures were demonstrated by the paper, the main stand-out point that the system stressed on was the use of its unique training metric known as the Triplet Loss. Unlike conventional models, instead of focusing on the classification task, FaceNet directly trains the embeddings in the Euclidean plane such that distances between these vectors are reduced in case of face similarities. The Triplet Loss metric is as follows,

$$\|f(x_i^a) - f(x_i^p)\|_2^2 + \alpha < \|f(x_i^a) - f(x_i^n)\|_2^2$$

Figure 5.2: Triplet Loss

What this function aims to learn:

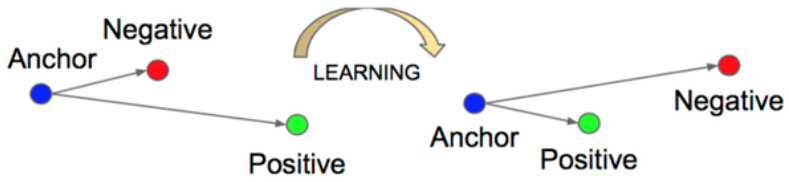


Figure 5.3: Triplet Loss Intuition

5.1.2 Triplet Selection

So as to guarantee quick intermingling it is significant to choose triplets that disregard the triplet imperative in. This implies that, given xai , we want to select an xpi (hard positive) such that argmax(f(xai) - f(xpi)) and similarly xni (hard negative) such that argim(f(xai) - f(xni)).

- Produce triplets disconnected each n steps, utilizing the most late system checkpoint and figuring the argmin what's more, argmax on a subset of the information.
- Create triplets on the web. This should be possible by choosing the hard positive/negative models from inside a smaller than normal group.

Here, we center around the online age and utilize huge smaller than expected clumps in the request for a couple thousand models and just register the argmin and argmax inside a smaller than usual bunch. To have a significant portrayal of the anchorpositive separations, it should be guaranteed that an insignificant number of models of any one personality is available in each smaller than usual cluster. In our examinations we test the preparation information to such an extent that around 40 faces are chosen for every character per minibatch. Also, haphazardly examined negative faces are added to every less clump. Rather than picking the hardest positive, we utilize every anchorpositive pair in a smaller than expected bunch while as yet choosing the hard negatives. We don't have a next to each other correlation of hard stay positive sets versus all grapple positive matches inside a little bunch, yet we found by and by that the all anchor positive strategy was progressively steady and combined somewhat quicker toward the start of preparing. We additionally investigated the disconnected age of triplets related to the online age and it might permit the utilization of littler group sizes, yet the tests were uncertain. Choosing the hardest negatives can by and by lead to awful nearby minima from the get-go in preparing, explicitly it can result in a crumbled model (for example f(x) = 0). So as to moderate this, it chooses xni with the end goal that

5.2 JUST CLICK:MOBILE APP

The project mainly focuses on reducing the tiring process of taking class attendance. A simple mobile application is developed for effectively handling the system by the teachers.

The front-end of the client application is developed using the AngularJS framework. It is a structural framework for creating dynamic web applications.

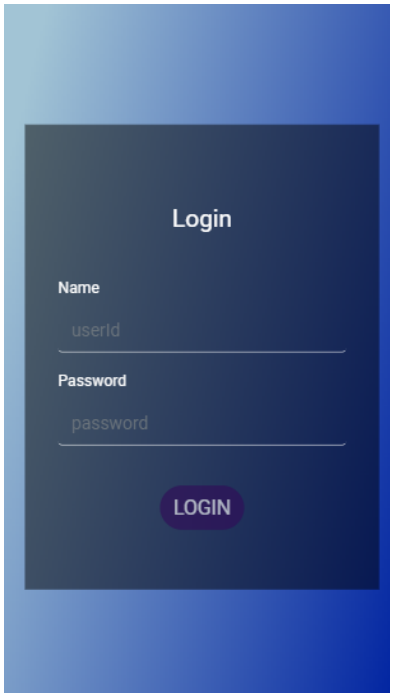


Figure 5.4: login page

Figure 5.5: home page

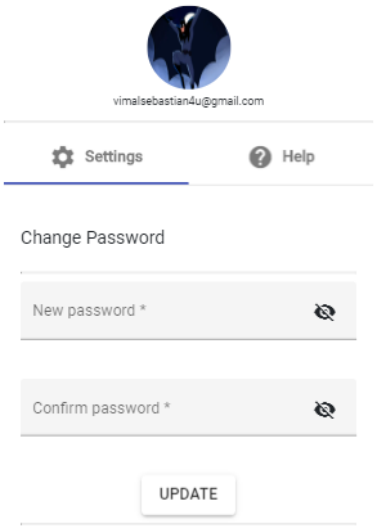


Figure 5.6: profile page

- Initially, all the users gets provided with credentials generated by the admin. A session based authentication is used in the application where the sessions gets created and stored in memory. Passport module is used for mapping of the session id when the corresponding user logs in. The credentials provided by the admin can be changed later by the user in the profile page.

- Once the user gets logged in, the home page will display all the subjects which the corresponding teacher is teaching in that current academic period.

- After selecting a particular subject, it takes into the corresponding subject page as shown in figure 5.4, where it provides three option. There will be an option that displays the google sheet containing the attendance list of the subject till that date. A manual option is also provided as shown in figure 5.5.

Figure 5.7: subject page

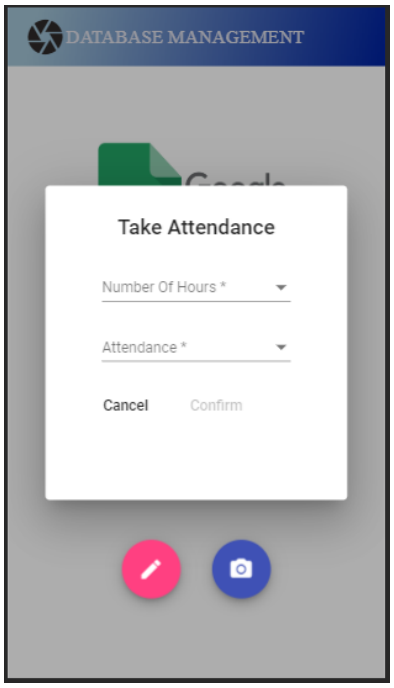


Figure 5.8: manual option

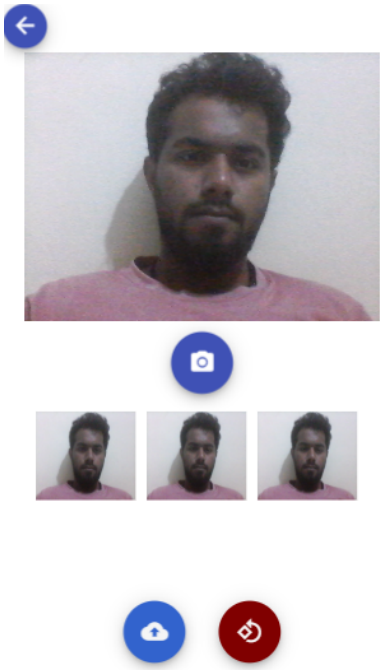


Figure 5.9: camera page

- On clicking the camera option the rear camera of the mobile gets activated followed by taking photos of the class. There must be atleast three photos of the class which should cover the whole students in the class. After taking the photos Upload button can be used to upload the photos to the server and within seconds the results will get popped up on the screen giving the complete list of absent students and the number of present students. A retake option is also provided inorder to cancel the taken photos and to take new set of photos.

5.3 COMMAND LINE INTERFACE CONTROLLING THE ENTIRE SYSTEM

The Admin dashboard or the command Line Interface is mainly focused for a single admin user to control the whole application. Addition of a user, subjects and students are done through this dashboard. The admin has access to every user's attendance sheets. All the necessary edits and updates are done through the dashboard.

5.3.1 Components

The application is built in node js platform. Node.js is an open source server environment. Node.js allows you to run JavaScript on the server. The data are stored in the backend using MongoDB. MongoDB is an open-source document database and leading NoSQL database. We use mongoose to interact with mongo. Mongoose is an Object Data Modeling (ODM) library for MongoDB and Node.js. It manages relationships between data, provides schema validation, and is used to translate between objects in code and the representation of those objects in MongoDB. Other modules used are readline, fs and crypto.

The readline module is used to read lines or commands from the command prompt. fs module has readFile() function to read .CSV file which contains the student details for StudentSchema.

The application starts by typing “attendance” at the command prompt. It then takes the user to our application command prompt. Exiting from the interface is done by typing the “exit” command.

Various Commands available are:

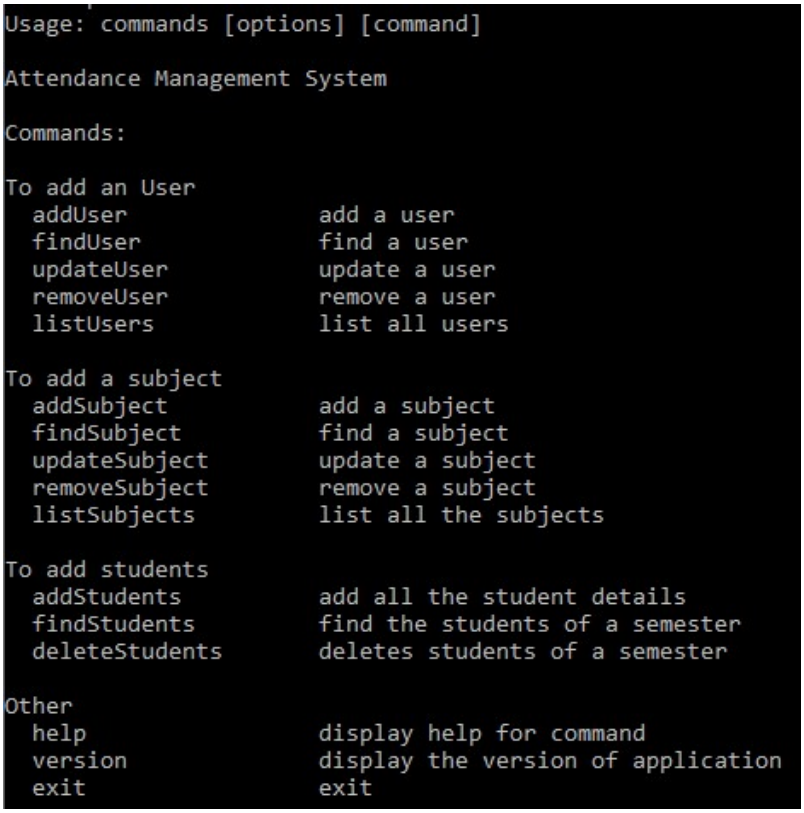


Figure 5.10: Commands

The description of each commands is shown right next to each command.

5.3.2 Methodology

When a User is added username, phone number, and email id are asked. We automatically generate a 10-digit random password for the first time. User can login using the randomly generated password for the first time and can then change it later from the user dashboard. Corresponding changes are made to the database.

Three schemas used in the application are UserSchema, SubjectSchema and StudentsSchema. UserSchema is used to store the details of user. We store hashed password plus salt in the database for security issues. SubjectSchema contains information about the subject and semester. It also contains the user id, such that each teacher is associated with a subject and a google sheet id were the attendance of each subject is updated. The StudentSchema

contains the name and roll number of all the students of a particular semester. There are update commands available for each schema.

Column Name	Data Type		Memory
user id(10)	string	1 byte x 10	10 bytes
name(20)	string	1 byte x 20	20 bytes
password()	string(hex)	1 byte x 8	8 bytes
Email(30)	string	1 byte x 30	30 bytes
PhnNo(12)	string	1 byte x 12	12 bytes

Column Name	Data Type		Memory
SubjectId(10)	String	1 byte x 10	10 bytes
SubjectName(30)	String	1 byte x 30	30 bytes
Semester(3)	String	1 byte x 3	3 bytes
Department(5)	String	1 byte x 5	5 bytes

Column Name	Data Type		Memory
Semester	String	1 byte x 3	3 bytes
Roll No	Number	1 byte x 4	4 bytes
Name	String	1 byte x 20	20 bytes

Figure 5.11: schemas

After the users are added, the admin can assign subjects for each user using the addsubject command. When adding a subject, a googlesheet is generated with the admins Gmail which is used by the corresponding user to mark his/her class attendance.

5.3.3 Salt hashing

Salt hashing is a technique in which we take the user entered password and a random string of characters called as salt, hash the combined string with a suitable hashing algorithm and store the result in the database.
Since hashes of the same password are the same, they are much easier to crack using lookup tables and rainbow tables, here someone pre-calculates the hashes of common passwords and stores them for others to use. Having same password hash for two or more users also makes it easier for the attacker to predict the password. So as a rule of thumb, no two users should have the same password hash. Adding salt to a password and then hashing the result reduces the possibility of having duplicate hashes and if your salt length is long enough, chances are minimal.

5.4 GOOGLE SHEET API

Google APIs use the OAuth 2.0 protocol for authentication and authorization. Google supports common OAuth 2.0 scenarios such as those for web server, client-side, installed, and limited-input device applications.
To begin, obtain OAuth 2.0 client credentials from the Google API Console. Then your client application requests an access token from the Google Authorization Server, extracts a token from the response, and sends the token to the Google API that you want to access. This can be done using OAuth 2.0 with Google (including the option to use your own client credentials),with the OAuth 2.0 Playground.
Basic steps:
All applications follow a basic pattern when accessing a Google API using OAuth 2.0. At a high level, you follow five steps:
1. Obtain OAuth 2.0 credentials from the Google API Console: Visit the Google API Console to obtain OAuth 2.0 credentials such as a client ID and client secret that are known to both Google and your application. The set of values varies based on what type of application you are building.

2. Obtain an access token from the Google Authorization Server: Before your application can access private data using a Google API, it must obtain an access token that grants access to that API. A single access token can grant varying degrees of access to multiple APIs. A variable parameter called scope controls the set of resources and operations that an access token permits. During the access-token request, your application sends one or more values in the scope parameter.

3. Examine scopes of access granted by the user: Here the scope that we should grant to access is access google sheet API for creation as well as edit.

4. Send the access token to an API: After an application obtains an access token, it sends the token to a Google API in an HTTP Authorization request header. It is possible to send tokens as URI query-string parameters, but we don't recommend it, because URI parameters can end up in log files that are not completely secure.

5. Refresh the access token, if necessary: Access tokens have limited lifetimes. If your application needs access to a Google API beyond the lifetime of a single access token, it can obtain a refresh token. A refresh token allows your application to obtain new access tokens.

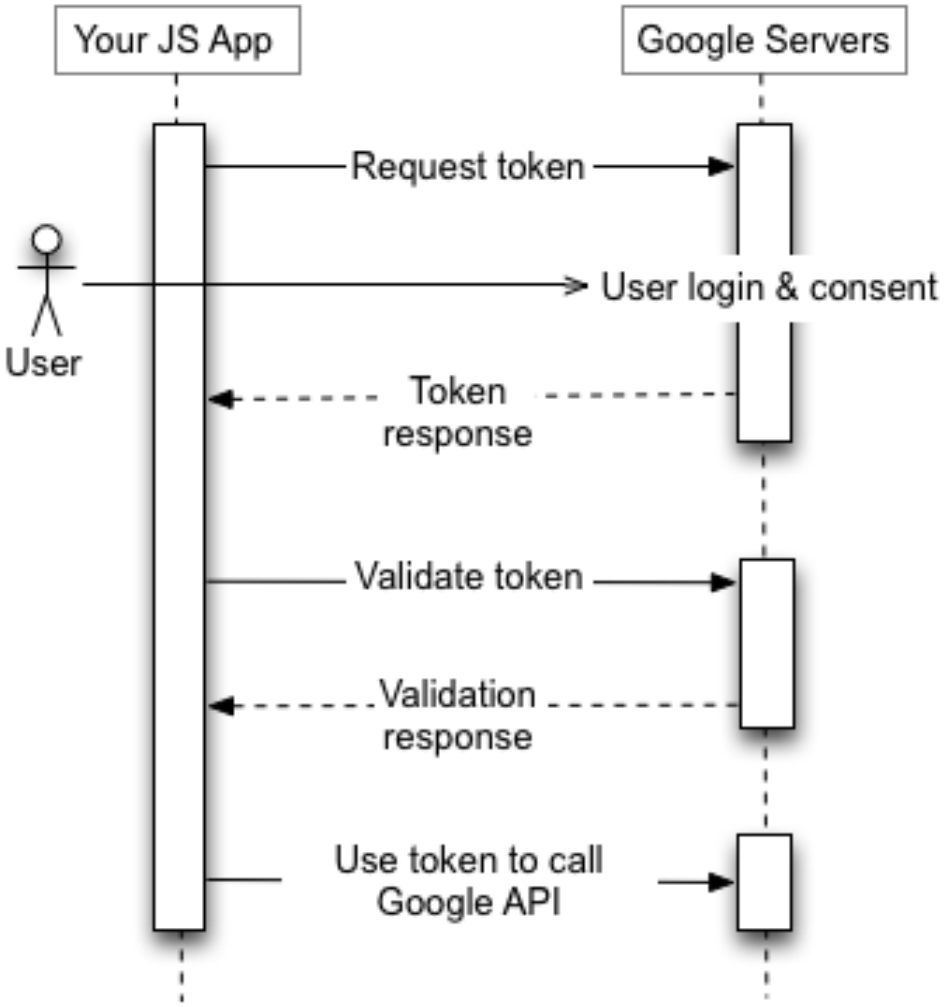


Figure 5.12: Javascript intrepretation of Google sheet API

5.4.1 Properties

The google sheet API functions are provided via admin commands given through the command line interface.

The google Sheet API provides the following functions:

1.Creation of google sheet and entry of static data into the sheet:

Whenever a new subject is added to the database a function is automatically called which creates a google sheet with a newly designed template .As the parameter an object is passed

to the function which contains an array of student names and roll number of students in the class,teacher name,semester are also sent.Subject name with subject code is set as the title of the newly created spreadsheet.Formula wise stats of the attendance sheet is provided. Spreadsheetid of the newly created sheet is stored in the database.

2.Appending attendance taken to the spreadsheet:

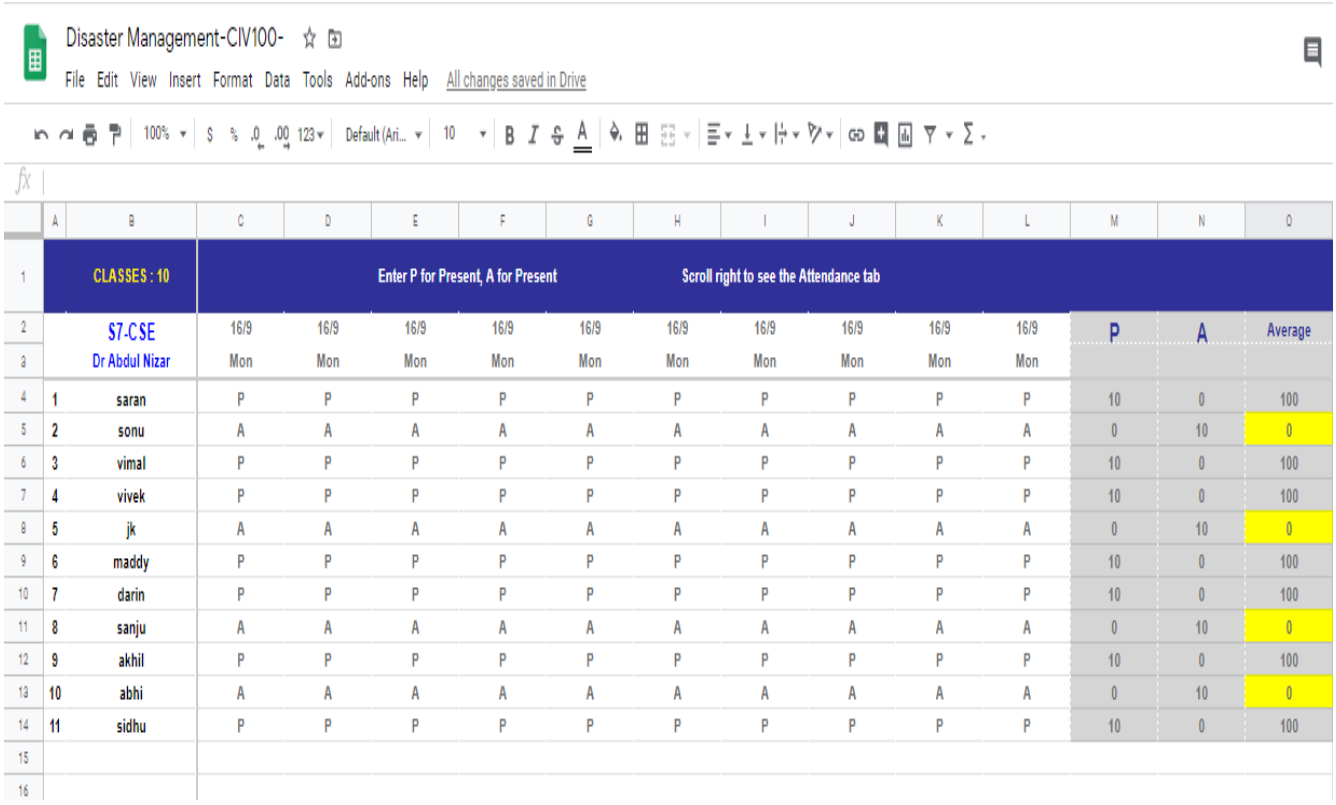
Data from the deep network model is processed and sent to the sheet which is updated as a particular period's attendance based on the number of hours taken by the tutor.Further stats of the attendance are also shown in the right end of every spreadsheet highlighting students with below par attendance.

3.Manual attendance addition:

Attendance can be added manually by tutors in any case and further edit permission to the sheet will be only given to specific teaching staff.

4.Static data Updation:

Further if admin makes any mistakes during database data insertion or a teacher teaching a particular subject changes then admin should be able to update this static data.Necessary objects should be passed to the function by admin from the database.



1	CLASSES : 10	Enter P for Present, A for Present					Scroll right to see the Attendance tab							
2	S7-CSE	16/9	16/9	16/9	16/9	16/9	16/9	16/9	16/9	16/9	16/9	P	A	Average
3	Dr Abdul Nizar	Mon	Mon	Mon	Mon	Mon	Mon	Mon	Mon	Mon	Mon			
4	1 saran	P	P	P	P	P	P	P	P	P	P	10	0	100
5	2 sonu	A	A	A	A	A	A	A	A	A	A	0	10	0
6	3 vimal	P	P	P	P	P	P	P	P	P	P	10	0	100
7	4 vivek	P	P	P	P	P	P	P	P	P	P	10	0	100
8	5 jk	A	A	A	A	A	A	A	A	A	A	0	10	0
9	6 maddy	P	P	P	P	P	P	P	P	P	P	10	0	100
10	7 darin	P	P	P	P	P	P	P	P	P	P	10	0	100
11	8 sanju	A	A	A	A	A	A	A	A	A	A	0	10	0
12	9 akhil	P	P	P	P	P	P	P	P	P	P	10	0	100
13	10 abhi	A	A	A	A	A	A	A	A	A	A	0	10	0
14	11 sidhu	P	P	P	P	P	P	P	P	P	P	10	0	100
15														
16														

Figure 5.13: An updated spreadsheet would look like this.

These are the different sections into which the overall project was divided for easier development of the project as the Agile methodology.These sections were integrated so as to provide the successful completion of the project.They represent the appropriate integration of this well processed Automatic Attendance System via an mobile app and server.

Chapter 6

Experiment Report

6.1 FaceNet Results

FaceNet system was experimented on many deep convolutional network architectures. One of the most successful architectures were, the NN1 CNN architecture. The architecture is as follows

layer	size-in	size-out	kernel	param	FLPS
conv1	220×220×3	110×110×64	7×7×3, 2	9K	115M
pool1	110×110×64	55×55×64	3×3×64, 2	0	
rnorm1	55×55×64	55×55×64		0	
conv2a	55×55×64	55×55×64	1×1×64, 1	4K	13M
conv2	55×55×64	55×55×192	3×3×64, 1	111K	335M
rnorm2	55×55×192	55×55×192		0	
pool2	55×55×192	28×28×192	3×3×192, 2	0	
conv3a	28×28×192	28×28×192	1×1×192, 1	37K	29M
conv3	28×28×192	28×28×384	3×3×192, 1	664K	521M
pool3	28×28×384	14×14×384	3×3×384, 2	0	
conv4a	14×14×384	14×14×384	1×1×384, 1	148K	29M
conv4	14×14×384	14×14×256	3×3×384, 1	885K	173M
conv5a	14×14×256	14×14×256	1×1×256, 1	66K	13M
conv5	14×14×256	14×14×256	3×3×256, 1	590K	116M
conv6a	14×14×256	14×14×256	1×1×256, 1	66K	13M
conv6	14×14×256	14×14×256	3×3×256, 1	590K	116M
pool4	14×14×256	7×7×256	3×3×256, 2	0	
concat	7×7×256	7×7×256		0	
fc1	7×7×256	1×32×128	maxout p=2	103M	103M
fc2	1×32×128	1×32×128	maxout p=2	34M	34M
fc7128	1×32×128	1×1×128		524K	0.5M
L2	1×1×128	1×1×128		0	
total				140M	1.6B

Figure 6.1: NN1 Architecture

The paper was assessed their technique on four datasets and except for Labeled Faces in the Wild and YouTube Faces framework was assessed on the face confirmation task. For example given a couple of two face pictures a squared L2 separation limit D(xi, xj) is utilized to decide the order of same and extraordinary. All faces sets (xi, xj) of a similar personality are signified with Psame, while all sets of various characters are indicated with Pdiff

True Accepts:

$$\mathrm{TA}(d) = \{(i,j) \in \mathcal{P}_{\text{same}}, \text{with } D(x_i, x_j) \leq d\}$$

Figure 6.2: True Accepts

These are the face sets (xi, xj) that were effectively characterized as same at edge d.

Correspondingly **False Accepts:**

$$\mathrm{FA}(d) = \{(i,j) \in \mathcal{P}_{\text{diff}}, \text{with } D(x_i, x_j) \leq d\}$$

Figure 6.3: False Accepts

is the arrangement of all matches that was erroneously delegated same (false accept). The approval rate VAL(d) and the false accept rate FAR(d) for a given face separation d are then characterized as **Validation Rate and False Acceptance Rate**

$$\mathrm{VAL}(d) = \frac{|\mathrm{TA}(d)|}{|\mathcal{P}_{\text{same}}|}, \quad \mathrm{FAR}(d) = \frac{|\mathrm{FA}(d)|}{|\mathcal{P}_{\text{diff}}|}.$$

Figure 6.4: Equations for VAR and FAR

6.2 VAL Results

In view of these measurements the FaceNet framework distributed its outcome when explored different avenues regarding distinctive CNN models. The outcome is as per the following, This table looks at the exhibition of our model designs on the hold out test set. Announced is the mean approval rate VAL at 10E-3 false accept rate. Additionally indicated is the standard mistake of the mean over the five test parts.

architecture	VAL
NN1 (Zeiler&Fergus 220×220)	87.9% ± 1.9
NN2 (Inception 224×224)	89.4% ± 1.6
NN3 (Inception 160×160)	88.3% ± 1.7
NN4 (Inception 96×96)	82.0% ± 2.3
NNS1 (mini Inception 165×165)	82.4% ± 2.4
NNS2 (tiny Inception 140×116)	51.9% ± 2.9

Figure 6.5: Table of architecture - VAL results

The convergence graph of the above claimed result is given below,

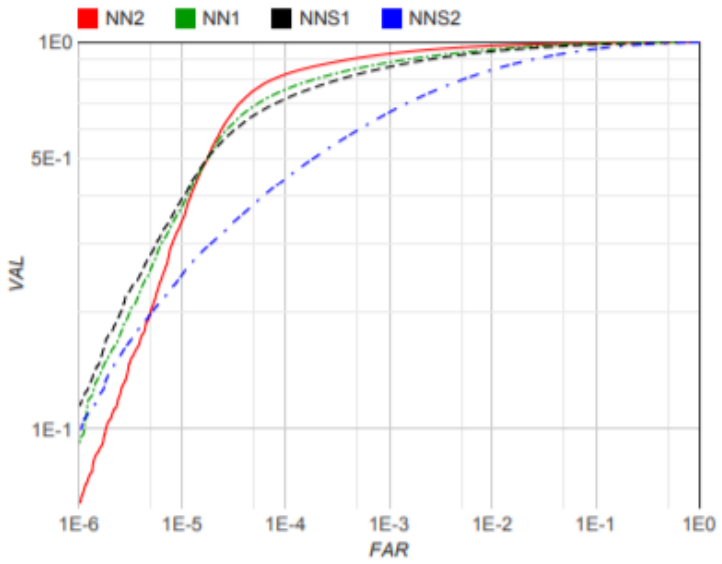


Figure 6.6: Convergence graph of different models

This plot shows the total ROC for the four unique models on our own photographs test set. The sharp drop at 10E-4 FAR can be clarified by commotion in the groundtruth marks. The models arranged by execution are: NN2: 224×224 info Inception based model; NN1: ZeilerFergus based system with 1×1 convolutions; NNS1: little Inception style model with just 220M

Failures; NNS2: little Inception model with just 20M FLOPS.

6.3 Image Quality

Results based on image qualities. It is obvious that the val-rate turns more accurate as we increase image quality.

jpeg q	val-rate
10	67.3%
20	81.4%
30	83.9%
50	85.5%
70	86.1%
90	86.5%

#pixels	val-rate
1,600	37.8%
6,400	79.5%
14,400	84.5%
25,600	85.7%
65,536	86.4%

Figure 6.7: VAL Rate based on jpeg q and no. of pixels

The table on the left shows the impact on the approval rate at 10E-3 accuracy with changing JPEG quality. The one on the correct shows how the picture size in pixels impacts the approval rate at 10E-3 exactness. This examination was finished with NN1 on the primary split of our test hold-out dataset.

6.4 OUR MODEL

The pre-trained FaceNet model was fed the following cropped images to produce their corresponding vectors and stored to locally compare new queries. Example of trained faces, Due to unavailability of real classroom photos, we had to test our model on multiple photos from different online medias. The model was tested on multiple photos of groups of students like the following,

	Total Students	Detected Correctly	Percentage %
Test-1	45	45	100
Test-2	50	50	100

Table 6.1: Accuracy on example photos

Figure 6.8: Trained Faces

Figure 6.9: Test photo 1

Figure 6.10: Test photo 2

Chapter 7

Conclusion and Future Scope

Our model suggests an easy automated method to compute attendance. Earlier models were much hard to implement and was also time consuming. This model brings the attendance of whole class within a click. Practical implementation almost took maximum of 2 to 3 seconds to generate the attendance. Our model provides almost all the necessary functionalities required for an institution to take attendance. Further developments and updates can be made later if needed. The strategy proposed shows critical enhancements over condition of -the-workmanship techniques, particularly for quick transformation from pictures to google sheets. The model is almost 98% accurate. We tested with a data-set containing the images from our classroom which was medium quality images and it ended in cent percent accurate results.

References

[1] FaceNet Paper – *FaceNet: A Unified Embedding for Face Recognition and Clustering*

[2] Salt Hash passwords using NodeJS crypto– *https://ciphertrick.com/salt-hash-passwords-using-nodejs-crypto/*

[3] Google Sheet API : Node.js Quickstart – *https://developers.google.com/sheets/api/quickstart/nodejs*

[4] Introduction to Node.js – *https://nodejs.org/en/docs/guides/.*

[5] Angular – *https://angular.io/docs/*