2018

Attendance Tracking by Facial Recognition

Gurismran Kaur

Grand Valley State University

Attendance Tracking By Facial Recognition

By

Gursimran Kaur

April, 2018

Attendance Tracking By Facial Recognition

By

Gursimran Kaur

A project submitted in partial fulfillment of the requirements for the degree of

Master of Science in

Computer Information Systems

at

Grand Valley State University

April, 2018

Yonglei Tao **Date**

Table of Contents

Abstract

Current systems that are generally used for tracking attendance for online exams/courses is either manual or marked automatically by successful logins. The proposed system uses facial detection and recognition to mark the attendance. This can be further expanded to track employees, replace traditional paper attendance and so on. Facial recognition system also increases security apart from frauds, it ensures no accidental data is leaked to unauthorized persons and no human intervention is needed to monitor the attendance or registration [1].

The proposed system allows pre-registration of users so that their details can be stored in the database, it also stores several sample images of the user's face. The program compares the user's face accessed whose attendance is sought to be marked in the real time through the laptop/computer's in-built camera. This is compared with the samples stored in the database and if a match is found, the attendance is marked automatically.

This technology is currently in use for high end security organizations, government offices, immigration services, etc. However, it is still not expanded to be used for general purposes as the skills to develop facial recognition systems were limited and expensive until recent developments in machine learning took place. The advancement in research for artificial neural networks (ANN) has provided us with the concept of deep learning which can be used interchangeably with ANN now-a-days [1]. It dissects facial images into pixels and patterns of pixels. It then generates a general pattern for a particular image known as the histogram of Gradients and then runs the calculations for basic measurements of the features [4]. Based on these calculations, algorithms like eigenfaces and Local Binary Patterns Histograms are used for further smarter processing.

Introduction

Facial recognition has many applications, one of them is explored in the project, which is authentication [2]. This project is an attempt to implement authentication through facial recognition in order to track attendance of the users.

This system can be used for verifying attendees for an online test. Authentication in the online tests is necessary to identify if the same person is taking the online test. Currently, online tests have secured login but do not have a fool proof system for identifying if the same person is taking the test for which the credentials have been submitted. If online tests can verify this, it would be much easier to prevent frauds for e-exams.

Also, it can be used to mark the attendance of students/teachers to prevent masquerade [3]. It can be used for tracking the entries of employees in the software systems/ databases with confidential data. Using such a system will prevent security breach and theft of data. It will also help in implementing data abstraction at organization level – information could be use only by authorized users.

Program Requirements and Design

The system implements basic algorithms of artificial neural networks to carry out facial detection and recognition. For security and authorization at system level, a super user i.e. an admin account has to be created during run time with the help of a username and password. The system can be accessed by only the users which have an authorized username and password.

The homepage is the main dashboard of Django web interface. It displays the tabs available for navigation and also allows admin to manage users and view the photos in the current database instance. The screenshots of all the tabs are provided in the results section of this project report.

A user, for example a new student/test taker/employee has to be initially added in the system database via the add user tab. An existing picture and other details have to be provided so that the system can compare facial features during the time of recognition. It can be considered as a type of registration which would help in further validation.

After that various positions of the face need to be captured and stored in the database so that when the user has to be identified, the system can compare the features with these pictures taken. The no. of captures can be decided and adjusted by the admin.

Then the system has to be trained for comparing the picture and associating it with the user details added. This is done at the backend by the code completely, but in order to invoke the code, a train button and a tab is provided for the admin. This can be triggered after adding a whole new batch of users or even after adding a new individual user, depending on the need of the program usage.

Once, the system is trained, it is all set to recognize the users in the database. A recognize tab is provided which captures three clicks in order to compare the features with the images in the database and returns the name of the users. Also, once the user is recognized, his/her attendance is automatically marked with the current time stamp. This can be verified in the attendance tab.

Implementation

It is a python and Django based system. Django is used for the front end and web interface and the Django database is used for storing images and their details while python is used for implementing artificial neural networks. The facial detection and recognition is implemented through eigenfaces, fisherfaces algorithm and Local binary patterns histograms (LBPH) algorithm.

These platforms and algorithms were chosen as I had done a course of datamining (CIS-635) in which I learnt R and some basics of Artificial neural networks (ANN). Therefore, I wanted go deeper into those algorithms and I learnt these new algorithms. Python was chosen as I already knew some basics of R, so, wanted to learn a new language and moreover, python is easier to learn and tutor oneself in comparison with R.

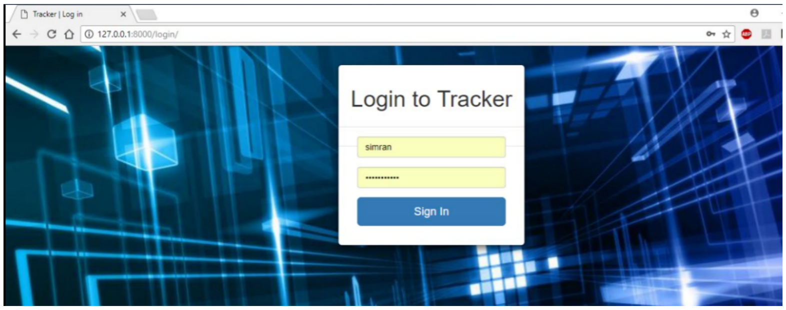
Now, lets take a sneak peak of how the algorithms are implemented.

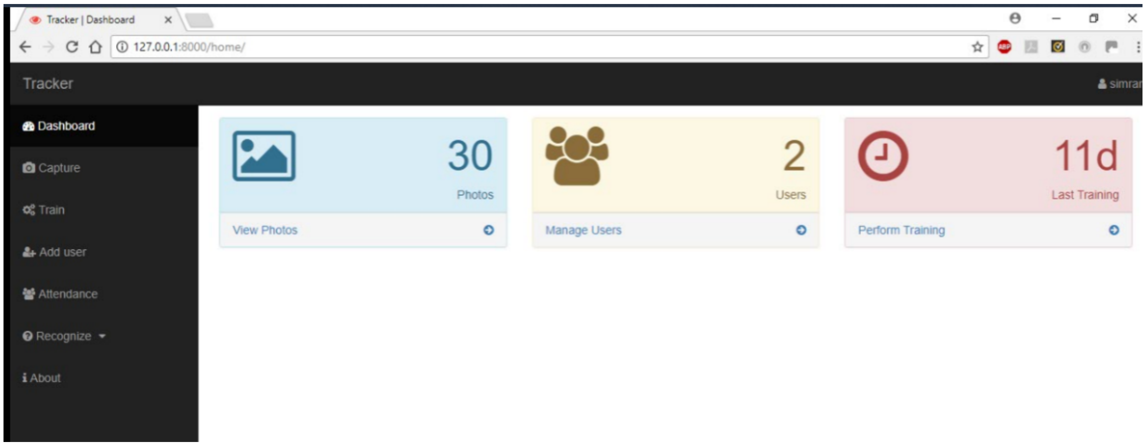
A widely used efficient algorithm in the world of ANN is Principal component analysis (PCA). This is commonly referred to as the “eigenface” method. It computes a reduced set of orthogonal basis vectors or eigenfaces of the training face images. A new face image can be approximated by a weighted sum of these Eigenfaces. PCA provides an optimal linear transformation from the original image space to an orthogonal eigenspace with reduced dimensionality in the sense of least mean squared reconstruction error [3]. Furthermore, it seeks to find a linear transformation by maximizing the between-class variance and minimizing the within-class variance [3].

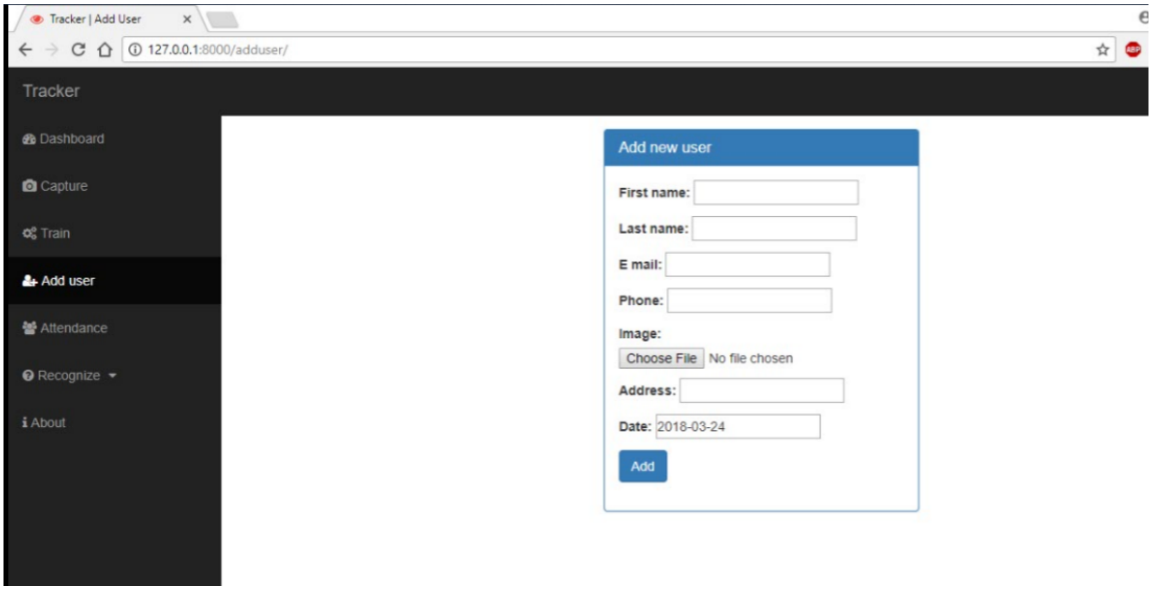
Next component used is the LBPH algorithm. The face image is first divided into small regions from which the Local Binary Pattern (LBP) features are extracted and concatenated into a single feature histogram efficiently representing the face image. The textures of the facial regions are locally encoded by the LBP patterns while the whole shape of the face is recovered by the construction of the face feature histogram [4]. The idea behind using the LBP features is that the face images can be seen as composition of micro-patterns which are invariant with respect to monotonic grey scale transformations. Combining these micro-patterns, a global description of the face image is obtained [2].

Results and Reflection

Here are the screenshots of the projects working which serve as the result of the code processing. (The backend coding is done by using python, references used for that are mentioned in the reference section - [5][6][7]).







Add new user

First name:

Last name:

E mail:

Phone:

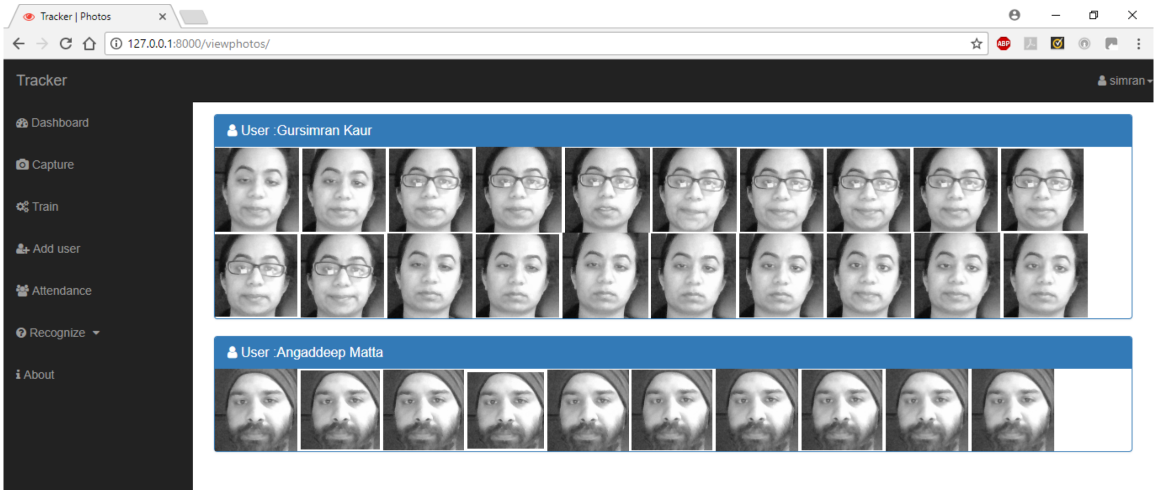
Image:

Choose File No file chosen

Address:

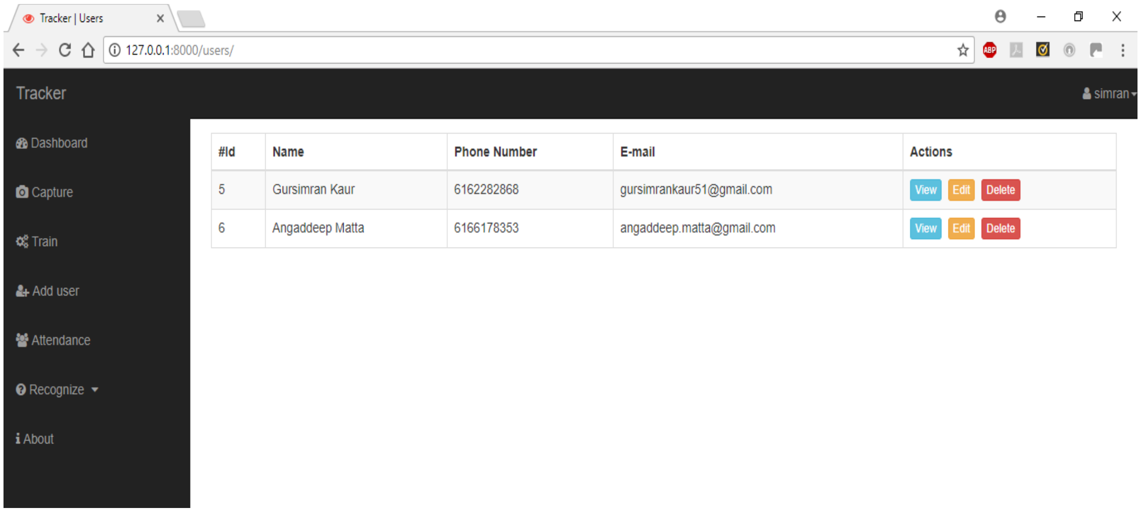
Date: 2018-03-24

Add

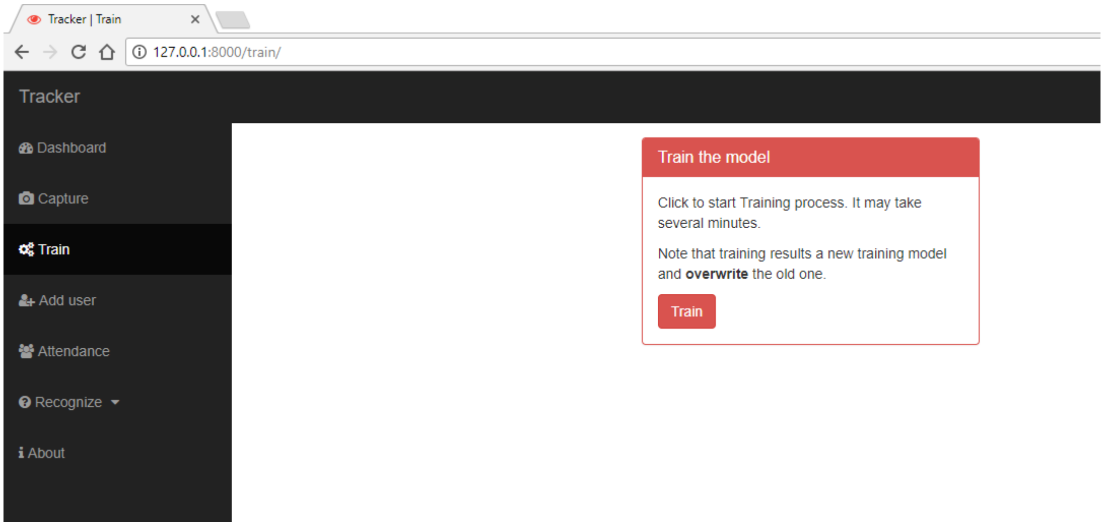


User :Gursimran Kaur

User :Angaddeep Matta



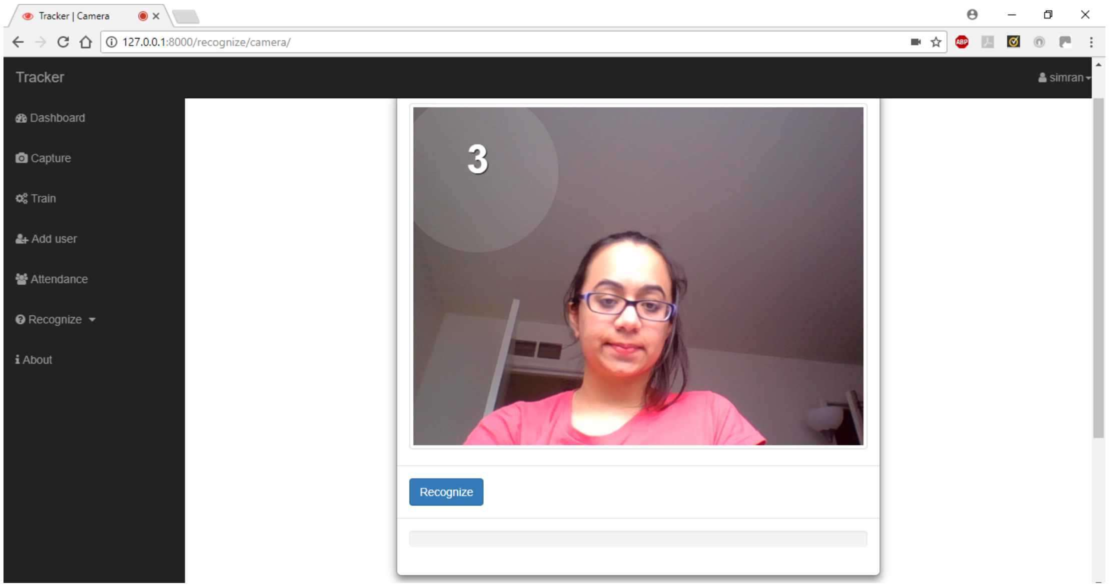
#Id	Name	Phone Number	E-mail	Actions
5	Gursimran Kaur	6162282868	gursimrankaur51@gmail.com	View Edit Delete
6	Angaddeep Matta	6166178353	angaddeep.matta@gmail.com	View Edit Delete

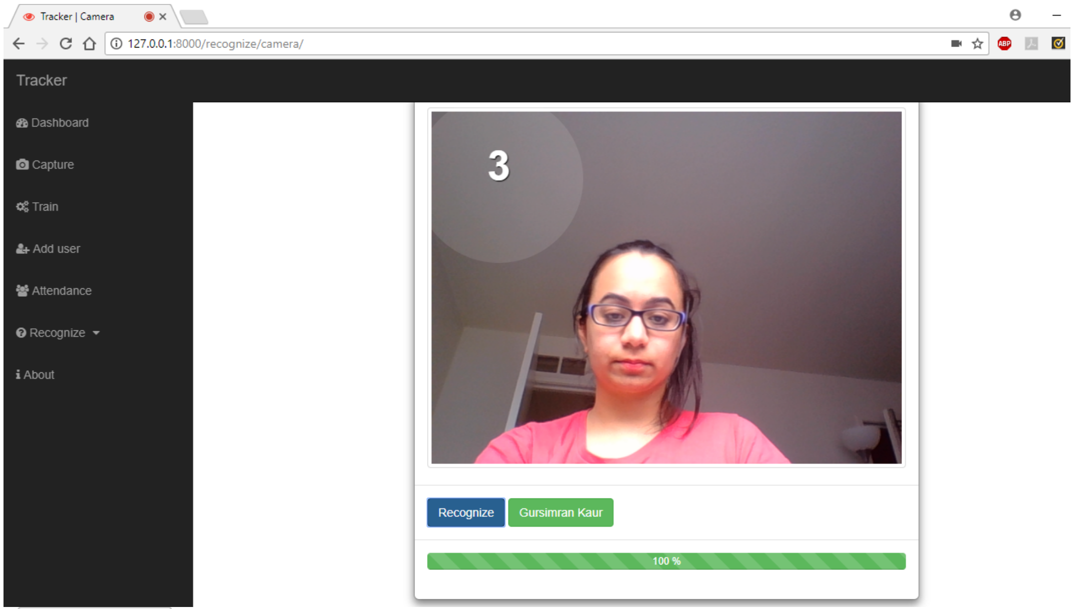


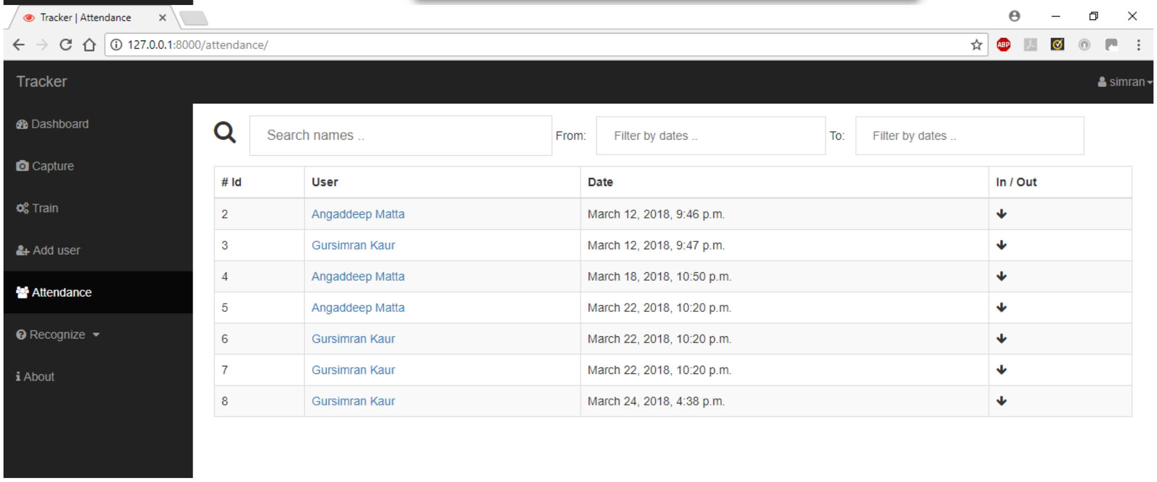
Train the model

Click to start Training process. It may take several minutes.

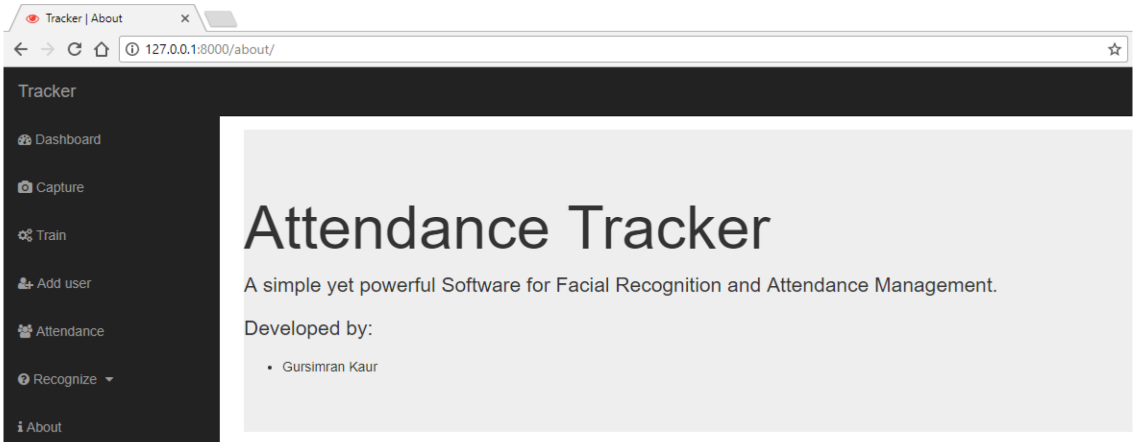
Note that training results a new training model and **overwrite** the old one.







# Id	User	Date	In / Out
2	Angaddeep Matta	March 12, 2018, 9:46 p.m.	↓
3	Gursimran Kaur	March 12, 2018, 9:47 p.m.	↓
4	Angaddeep Matta	March 18, 2018, 10:50 p.m.	↓
5	Angaddeep Matta	March 22, 2018, 10:20 p.m.	↓
6	Gursimran Kaur	March 22, 2018, 10:20 p.m.	↓
7	Gursimran Kaur	March 22, 2018, 10:20 p.m.	↓
8	Gursimran Kaur	March 24, 2018, 4:38 p.m.	↓



Attendance Tracker

A simple yet powerful Software for Facial Recognition and Attendance Management.

Developed by:

- Gursimran Kaur

Conclusions and Future Work

To summarize, this system is effective for tracking attendance by facial recognition and reduce the risks of frauds. The front end is a web-based portal that is uses Django web interface and implements Django database to store and retrieve information. The logic and facial recognition systems are implemented in python which provides in built libraries and hence, the ease of programming and coding these AI based systems. The simultaneous use of Eigenfaces and LBPH (Local Binary patterns Histogram) makes it a robust system and can recognize faces even with a little blurred/ change in facial expression. These are self-learning algorithms and there is an initial need of registration that prompts the user to provide a picture and stores the picture along with the details in a database. Once the face is matched with an already stored image in the database, the related details are pulled and the attendance is marked for that particular person along with the current time stamp. This helps in maintaining a time based tracking.

The currently implemented system takes about 20-25 seconds approximately to recognize the faces and pull out the details from the database, this performance can be reduced in the future. This system can further be extended to include modules that can be used for assigning and tracking tasks. Moreover, functionality can be added to run reports automatically instead of outsourcing to another tool.

References

1. Principles of Artificial Neural Networks – Daniel Graupe
2. An introduction to neural networks – Kevin Gurney
3. Toward a Practical Face Recognition System: Robust Alignment and Illumination by Sparse Representation – Andrew wagner, Arviind Ganesh – University of Illionois.
4. Learning Discriminative LBP-Histogram Bins for Facial Expression Recognition Caifeng Shan and Tommaso Gritti
5. Python Programming for the absolute beginner, third edition – Michael Dawson
6. Python.org
7. Pythonforbeginners.com