Ain Shams University

Faculty of Computer & Information Sciences

Computer Science Department

Automatic Attendance System

Figure 1: Face Recognition

Ain Shams University

Faculty of Computer & Information Sciences

Computer Science Department

Automatic Attendance System

BY:

Youssef Ashraf Sobhy [CS Department]

Youssef Hany Anter [CS Department]

Mohammed Ahmed Fathy [CS Department]

Nouran Taha Abdelmoniem [CS Department]

Under Supervision of :

Prof. Zaki Taha

[CS Department]

Faculty of Computer and Information Sciences,

Ain Shams University.

Dr.Ghada Elshamy

[CS Department]

Faculty of Computer and Information Sciences,

Ain Shams University.

Acknowledgement

We would like to thank Prof.Zaki Taha for believing in us that we can make such a project and thanks for Dr.GhadaElshamy for helping us for giving us useful advices .

Abstract

Authentication is a significant issue in system control in computer based communication. Human face recognition is an important branch of biometric verification and has been widely used in many applications, such as video monitor system, human-computer interaction, and door control system and network security. This paper describes a method for Student's Attendance System which will integrate with the face recognition technology using Personal Component Analysis (PCA) algorithm. The system will record the attendance of the students in class room environment automatically and it will provide the facilities to the faculty to access the information of the students easily by maintaining a log for clock-in and clock-out time.

Contents

List of Figures

List of Abbreviations

RFID: Radio Frequency Identification

BPNN: Back Propagation Neural Network

DNN: Deep Neural Network

LMS : Learning Management System

HOG : Histogram of Oriented Gradients

SVM : Support Vector Macine

1-Introduction

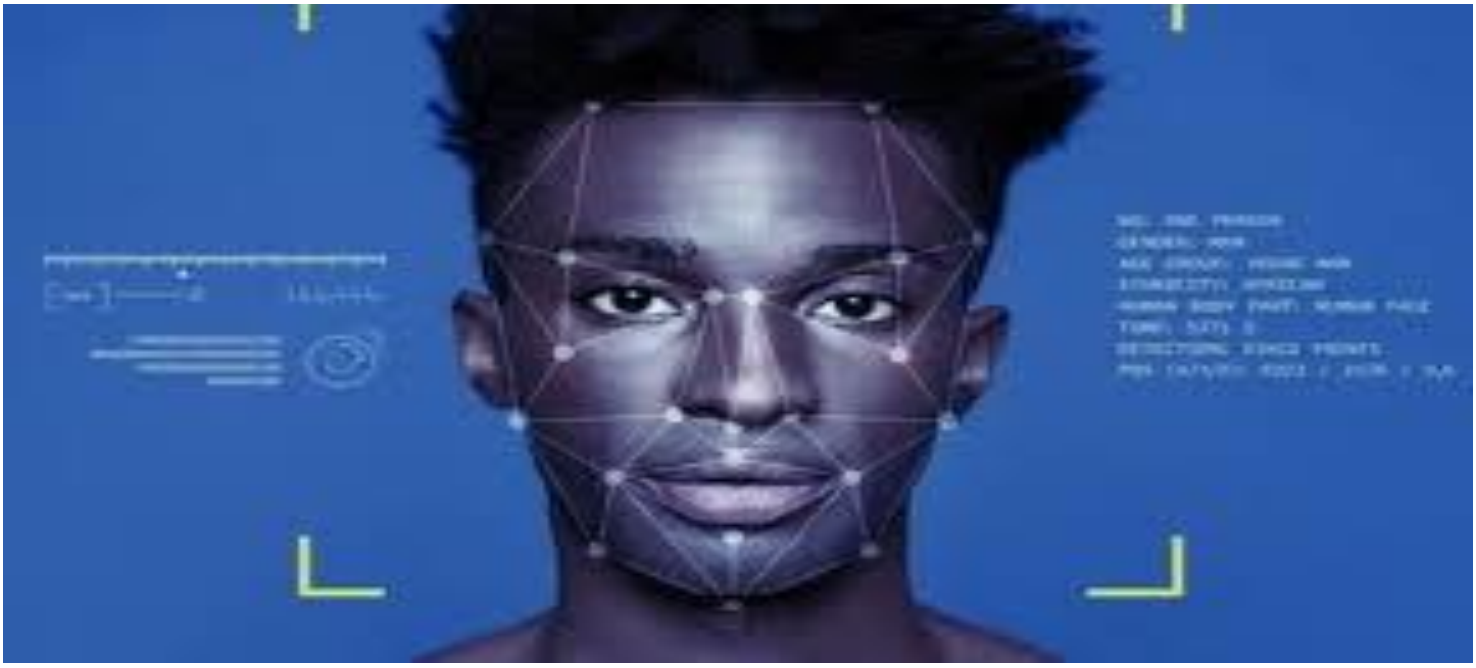
- In many institutions and organization the attendance is a very important factor for various purposes and its one of the important criteria that is to follow for students and organization employees. The previous approach in which manually taking and maintains the attendance records was very inconvenient task. After having these issues in mind we develop an automatic attendance system which automates the whole process of taking attendance and maintaining it.

- A key factor of improving the quality of education is having students attend classes regularly. Traditionally students are stimulated to attend classes using attendance points which at the end of a semester constitute a part of a student's final grade. However, traditionally this presents additional effort from the teacher, who must make sure to correctly mark attending students, which at the same time wastes a considerable amount of time from the teaching process. Furthermore it can get much more complicated if one has to deal with large groups of students.

- Facial recognition or face recognition as it is often referred to as, analyses characteristics of a person's face image input through a camera. It measures overall facial structure, distances between eyes, nose, mouth, and jaw

edges. These measurements are retained in a database and used as a comparison when a user stands before the camera. One of the strongest positive aspects of facial recognition is that it is non-intrusive. Verification or identification can be accomplished from two feet away or more, without requiring the user to wait for long periods of time or do anything more than look at the camera.

Figure 2: Feature Extraction



1.1 Motivation

- Recording attendance by signing or using fingerprint for each employee or student takes much time

- With the recent spread of corona virus , it will be dangerous to communicate between employees or students

- In general manual recording has many other issues that make a false presence, losing attendance

1.2 Problem Definition

When you have a large number of employees or student it is not easy to take the attendance for every one

- It takes too much time
- It takes too much effort
- You may lose the paper of attendance
- Make the communication between the employees or students and that is help in spreading corona virus

1.3 Objective

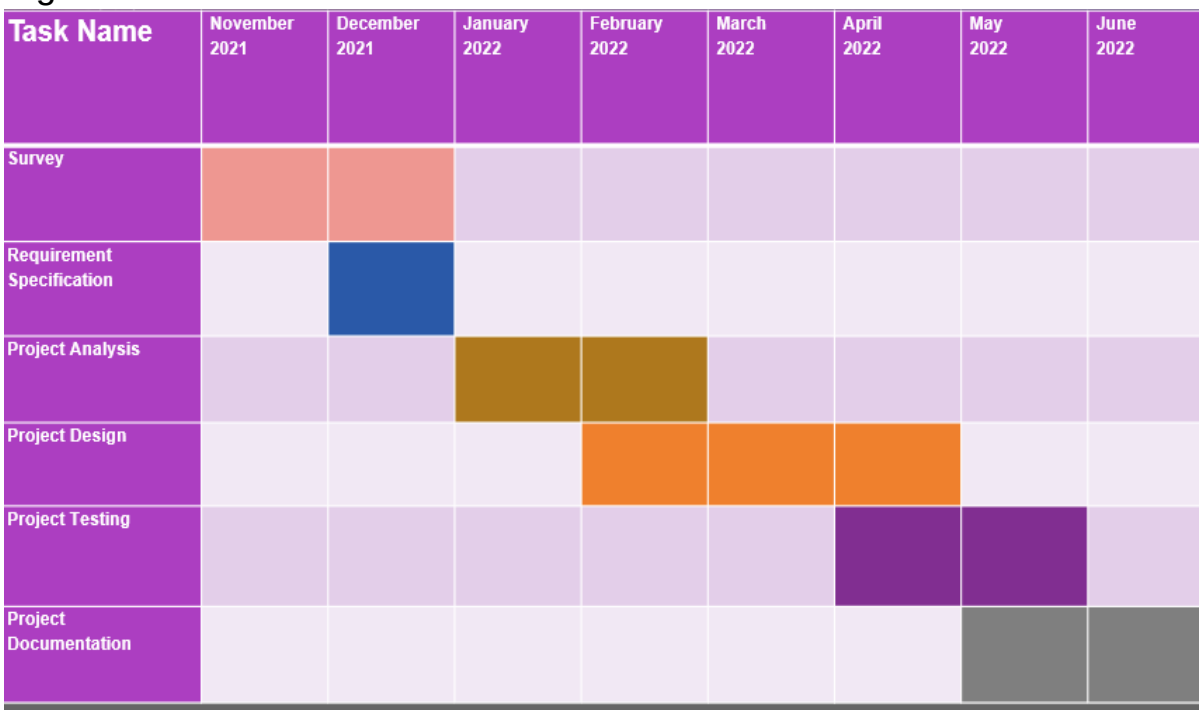
1- We aim to make attendance register process

- Easier
- Faster
- Accurate
- Safety

2- We aim to save time and effort according to humans

1.4Time Plan

Figure 3: Time Plan



2. Background

- The first attempts to use face recognition began in the 1960's with a semi-automated system. Marks were made on photographs to locate the major features; it used features such as eyes, ears, noses, and mouths. Then distances and ratios were computed from these marks to a common reference point and compared to reference data. In the early 1970's Goldstein, Harmon and Lesk created a system of 21 subjective markers such as hair colour and lip thickness. This proved even harder to automate due to the subjective nature of many of the measurements still made completely by hand. Fisher and Elschlagerb approaches to measure different pieces of the face and mapped them all onto a global template, which was found that these features do not contain enough unique data to represent an adult face. Another approach is the Connectionist approach , which seeks to classify the human face using a combination of both range of gestures and a set of identifying markers. This is usually implemented using 2-dimensional pattern recognition and neural net principles. Most of the time this approach requires a huge number of training faces to achieve decent accuracy; for that reason it has yet to be implemented on a large scale. The first fully automated system to be developed utilized very general pattern recognition. It compared faces to a generic face model of expected features and created a series of patters for an image relative to this model. This approach is mainly statistical and relies on histograms and the gray scale value.

2.1Related Work

- Today technology progress is at a faster rate and devices are shifting towards automation. As a result is rapid progress with respect to software technology . There are multiple types of monitoring and taking attendance applied in companies, industries and institutes.

- Most of the time attendance of students is taken on attendance sheet or register given to faculty members . It takes a lot of time and work. We are not sure whether authenticated student has responded or not, it also causes manual errors. Also attendance sheet can be lost or stolen by some students .

- Other biometric system consists of fingerprint identification . Utilizing this method first fingerprint of each student is collected and it is stored in database that is finger print sensor. After this the finger print is matched with the database and attendance is marked so, but its main disadvantage is students have to make a queue and wait for their turn which is a lot of work and consumes much time also if the finger print does not recognized properly by the system attendance will not be marked so it is not efficient.

- Other biometric system available is detection of Eye ball in which sensor is eyeball, blinking rate and location of iris is sensed. In this system first iris or eyeball for each person is stored in database eyeball is not same for every person. In obtained image eye ball is compared with eyeball in database and attendance is marked, but practically it is impossible to capture eyeball of each student in an image.

- Another solution is RFID based attendance system . Every student has its RFID tag and as it passes through RFID reader it mark its attendance. But tag reader is time consuming; also chance of misuse of cards is one student can mark multiple attendance of missing students too.

- To overcome above scenarios advanced system to mark attendance is using cameras to capture images of a classroom. Recently image processing is used to extract useful information for an image to mark attendance. As smart phones are becoming popular among people, to use the existing devices without making a dedicated setup to mark attendance from an image attendance is marked. Almost face of every student is unique but may have resemblance in certain features with other person.

- There are multiple systems created specifically for face detection and recognition for making attendance system each has certain drawbacks. One way is back propagation neural network (BPNN) and Viola-Jones algorithm. BPNN, uses two weighted propagation one is forward in which input is fed through network to generate output propagation activation and in backward feedback is formed by taking output as input in order to create difference among actual output and target.

- Viola–Jones algorithm uses Haar feature selection that matches common features in human faces leading to high chances of inaccuracy. Another approach is Histogram normalization, Skin classification and Noise filtering using MATLAB. Problem in such system is it is restricted to inbuilt feature of MATLAB.

- Deep neural network (DNN) used for face recognition. DNN used Deep face model that tends to be human level accurate taking multiple set of images for its training. Major problem is

its requirement of large dataset for training which is not possible to store a big number of images for each student.

2.2 Survey

- Using real time computer vision algorithms in automatic attendance management systems This paper introduces a new approach in automatic attendance management systems, extended with computer vision algorithms. The Proposed system uses real time face detection algorithms integrated on an existing Learning Management System (LMS), which automatically detects and registers students attending on a lecture. The system represents a supplemental tool for instructors, combining algorithms used in machine learning with adaptive methods used to track facial changes during a longer period of time. Face Recognition-based Lecture Attendance System proposed a system that takes the attendance of students for classroom lecture. The system takes attendance automatically using face recognition. However, it is difficult to estimate the attendance precisely using each result of face recognition independently because the face detection rate is not sufficiently high. In this paper, we propose a method for estimating the attendance precisely using all the results of face recognition obtained by continuous observation. Automatic Control of students' attendance in Classrooms Using RFID Radio frequency identification (RFID) is one of the automatic identification technologies more in vogue nowadays. There is a wide research and development in this area trying to take maximum advantage of this technology, and in coming years many new applications and research areas will continue to appear. Face Recognition based Attendance Management System using Machine Learning The most arduous task in any organization is attendance marking. In this paper we proposed an automated attendance management system which tackles the predicament of

recognition of faces in biometric systems subject to different real time scenarios such as illumination, rotation and scaling.

2.3 Methodology

- The Attendance management is the significant process that were carry out in every institute to monitor the performance of the student. Every institute does this is its own way. Some of there institute use the old paper or file-based system and some have adopted strategies of automated attendance system using some biometric technique.
- A facial recognition system is a computerized software which is suited for determining or validating a person by performing comparisons on patterns based on their facial appearances. In this system OpenCV & Face Recognition libraries were used which are one of the popular libraries for face detection by using these libraries system first capturing the student photos and storing them into the database which were further used for the training purpose after that at the time of attendance when system camera get on system will detect the faces that were present in the frame the faces were detected by using HOG i.e. (Histogram of Oriented Gradients) which were carrier out in the system. after that if image that were present in the frame is tilted then Face Landmark Estimation algorithm will be carried out and face will be transformed to be as close as possible to perfectly cantered.
- After that system will encode all the images that were present in the database as well as the face which were detected in the frame. For performing encoding Deep Conversional Neural Network algorithm will get carried out & for each face 128 measurements were generated then the measurements of the face that were detected in frame it get compared with the measurements of the faces that were present in the image which is earlier stored in the database.

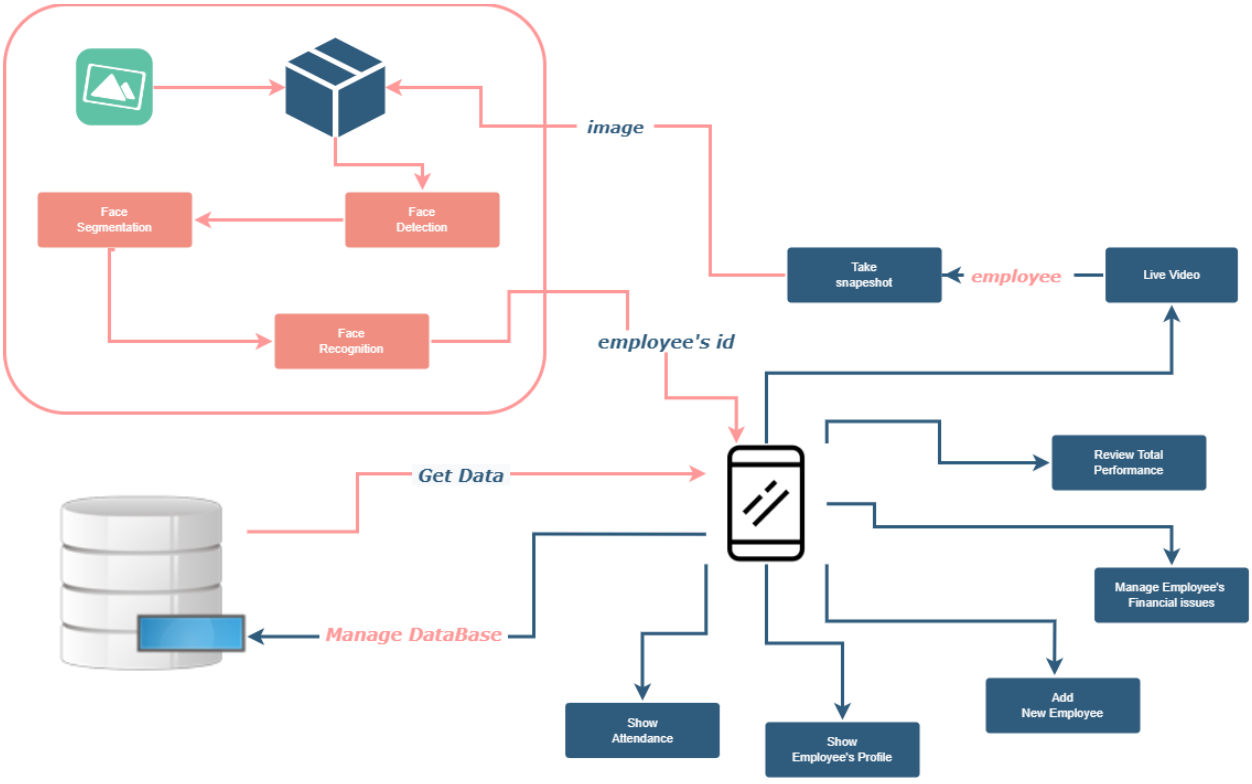
- So at last by using simple liner SVM algorithm system will find the person in database of know peoples (i.e. capture at the starting of the project) who has closest measurements to the image that were detected by camera. After finding perfect match system will generate the name and date & time & present mark and store the entry in CSV file. Which were further uploaded on the database and also user can open it with Microsoft Excel.

3- Analysis and Design

3.1 System Overview

3.1.1 System Architecture

Figure 4 : System Architecture



The system consists of two sections :

- Live video and Attendance recording
- Employees management and Data representation

First section deals with employees , it is responsible for detect employee's face and recognize him by cropping the face and send it to neural network model which extract features from the image then classify this face belongs to which person and give us back the name then record him in the attendance of that day.

Second section designed for administrator, it is responsible for showing the personal and job information for each employee and his performance during work in the form of visualized data, presence , absence and days which he arrived late, this section also makes user able to give bonus or punishment for each employee and manage his data.

3.1.2 System Users

A. ***Intended Users:****Employees , Company manager or Administrator*

Each employee needs to standing in front of the camera till the system recognize him and show him the welcome message,

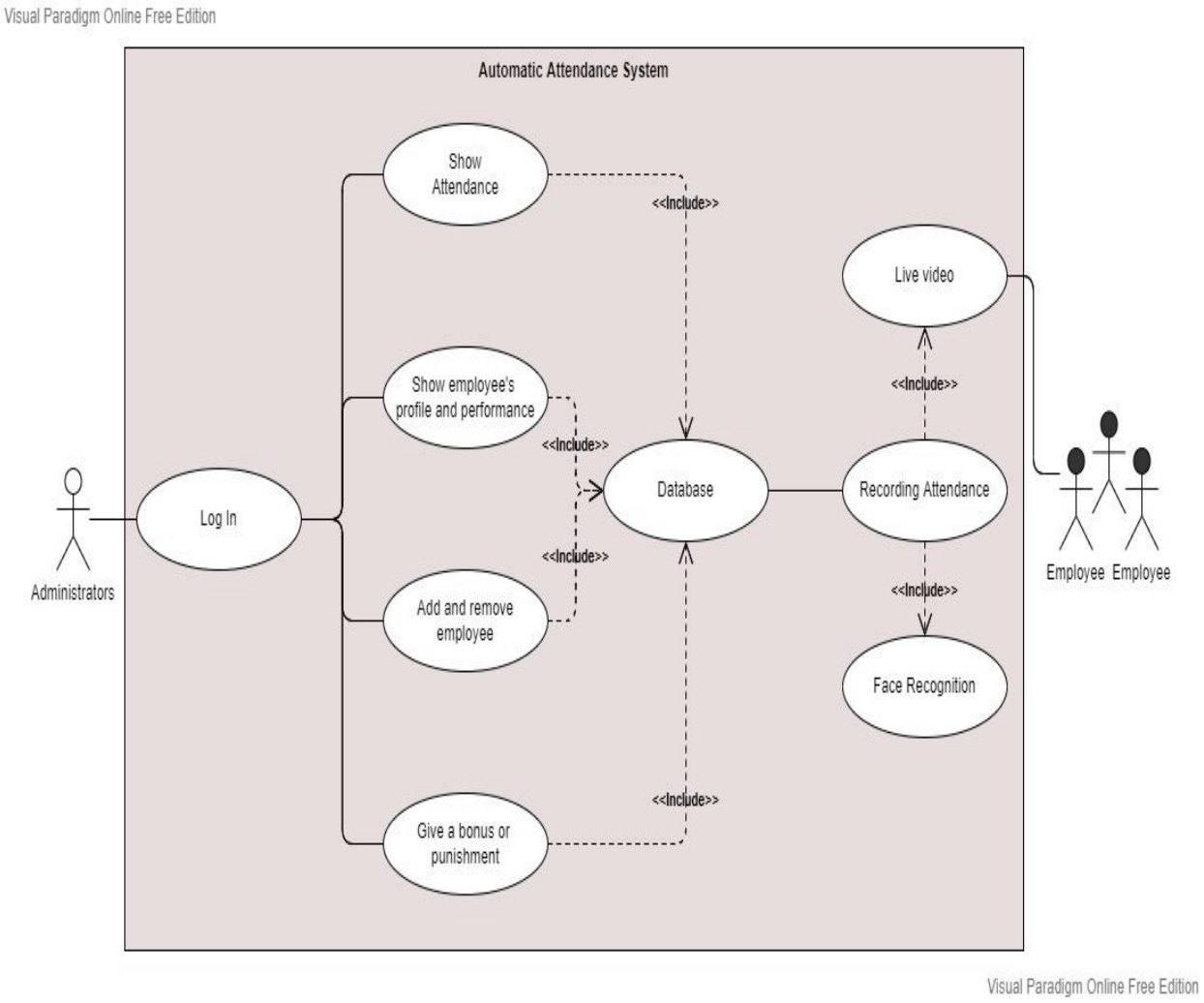
After attendance recording phase ended the administrator is able to manage the system and see the attendees.

B. ***User Characteristics:***System is not complicated it is easy to use and doesn't require skills to operate.

3.2 System Analysis & Design

3.2.1 Use Case Diagram

Figure 5 :Use Case Diagram



At the beginning Live video will detect employee's face and recognize him by neural network model then his name will be recorded in the attendance of that day,

If administrator wants to access this data he needs to login first then the application produce some functions to help manager to control employees data like showing attendance and performance for each employee , add or remove new employee and give a bonus or a punishment to encourage them to work, Also every employee have a profile which help administrator to collect enough knowledge about him.

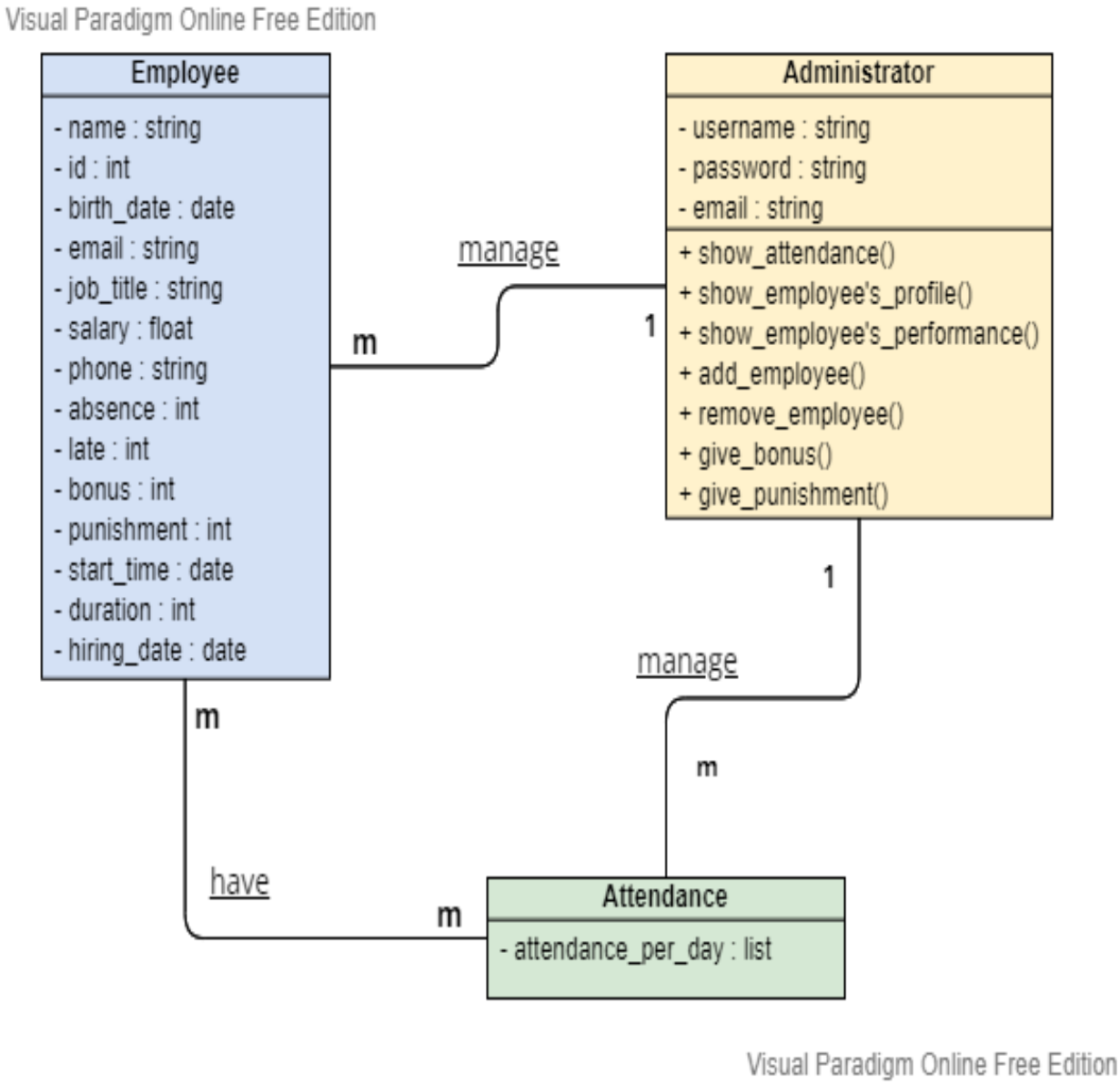
3.2.2 Class Diagram

Only administrator can use the system but he needs to login first so each administrator have username , password and email. Administrator can see attendance of any day in the current month also he can see all information about each employee from his profile so he can reward him or punish him. Adding or removing employees is administrator responsibility.

Each employee has a profile which filled with his information like his name , phone, id, email, absence, bonus, etc..

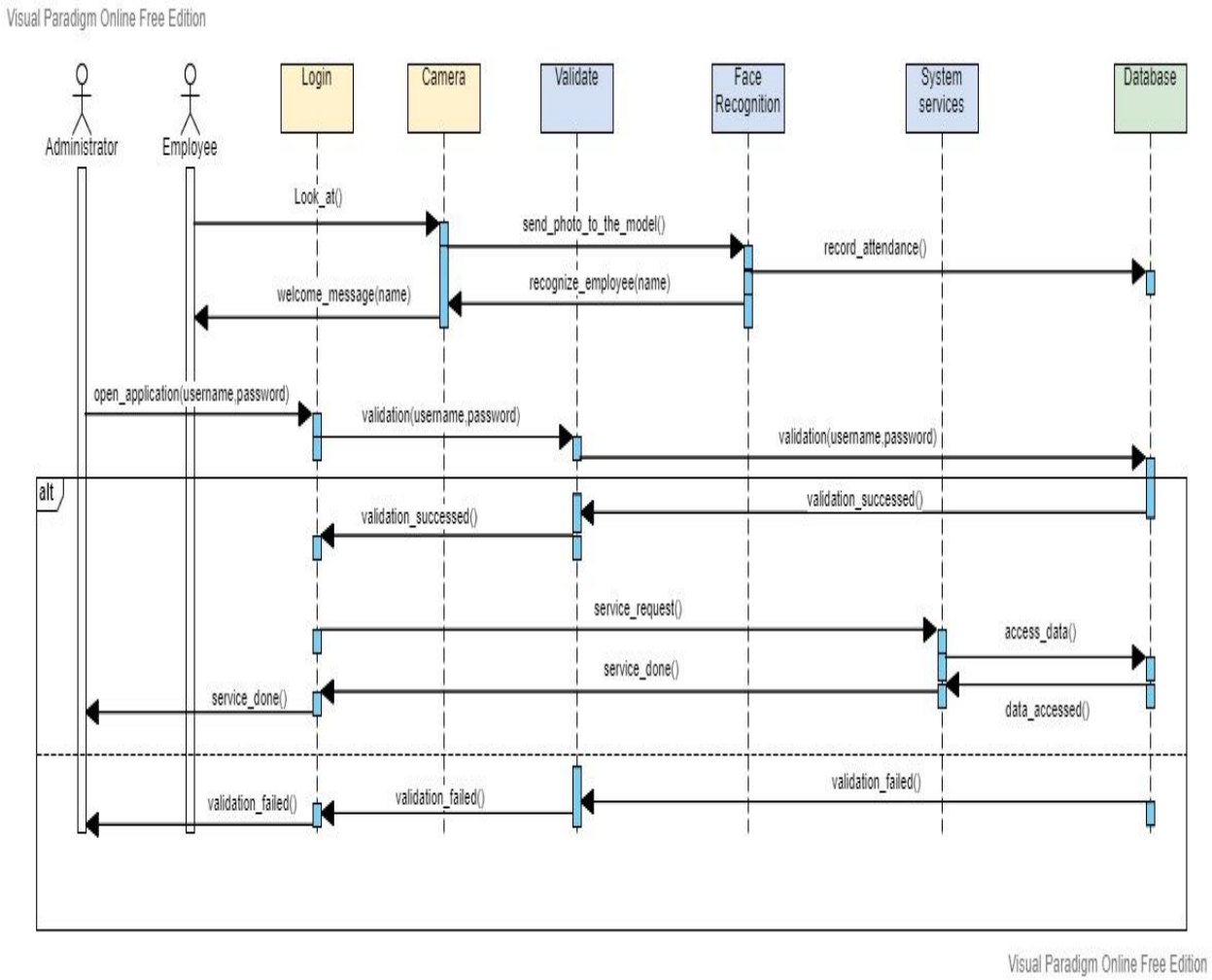
Attendance recorded per a day each day have a list of ids of employees who attended that day.

Figure 6 : Class Diagram



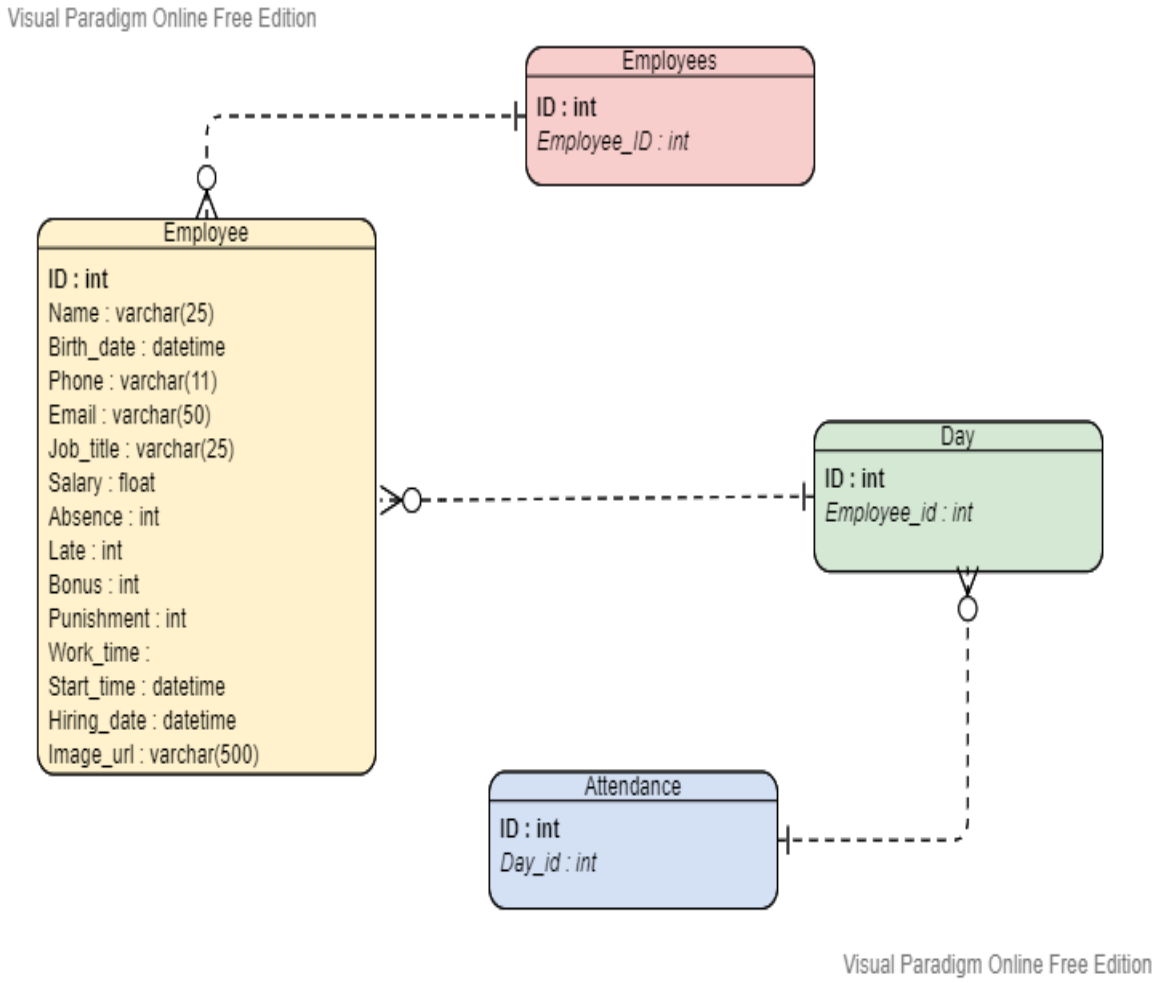
3.2.3 Sequence Diagram

Figure 7: Sequence Diagram



3.2.4 Database Diagram

Figure 8 :Database Diagram



Employees table contain ids each id refer to a unique employee.

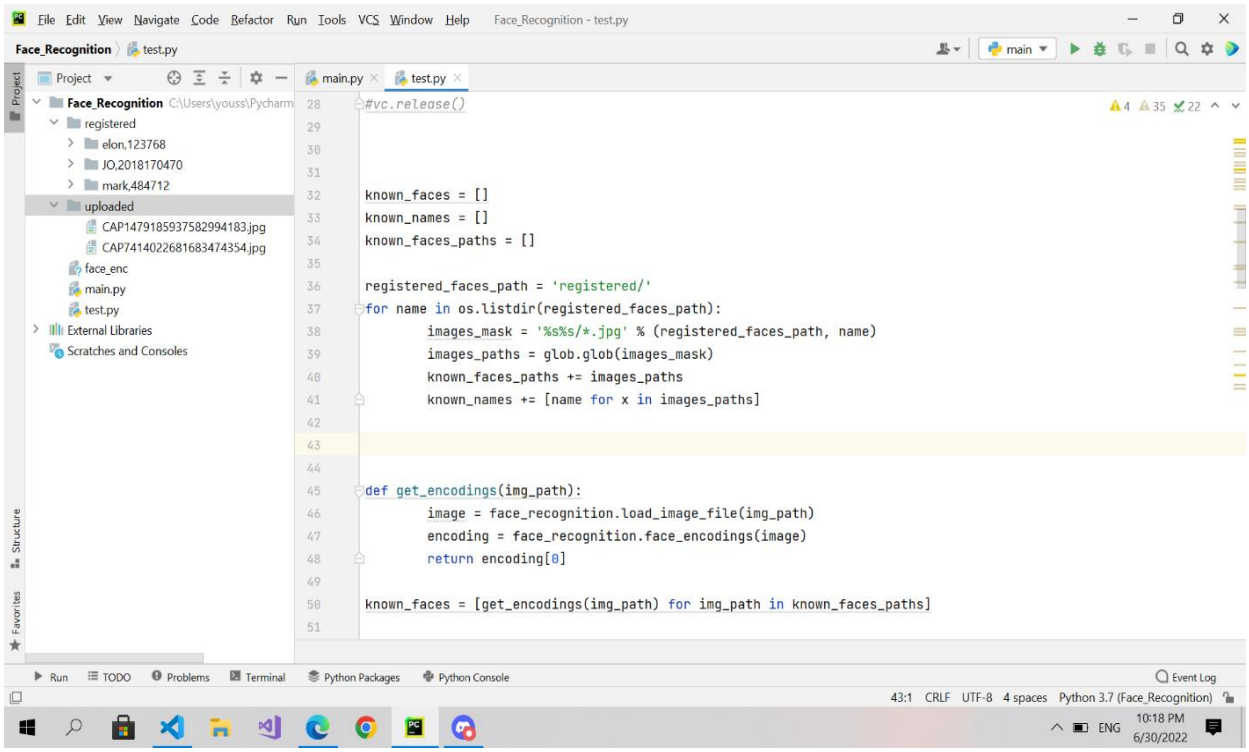
Employee table contain all necessary information about each recorded employee in the system. .

Attendance table contain ids each id refer to specific day.

Day table contain ids of all employees who attended this day.

faces. new faces will be stored to uploaded path folder that will come through flask api using flutter.First we have a folder of Known faces that the pretrained model will extract their feature to compare them with the new .

Figure9:code of adding new photos



we initialize two empty lists to add to them the Known faces and their corresponding names. we get the path of images in Known faces folder to read them and apply function called getEncoding() that uses function face_encondings() in face_recognition library to get from each image

```
f = open("face_enc", "wb")
data = {"encodings": known_faces, "names": known_names}
f.write(pickle.dumps(data))
f.close()
data = pickle.loads(open('face_enc', "rb").read())
```

Figure 10 : Saving trend model and loading

face's features. then we add those encodings to known faces list and we add folder's name to known names list .

then we save faces features to file using Pickle API to optimize recognition process and we load this file again to read features to be compared.

Figure 11: Retriving faces to encode and compare them

```
@app.route("/postImage", methods=['POST'])
def post():
        notfound=True
        imagefile = request.files['image'] # getting the response data
        filename=werkzeug.utils.secure_filename(imagefile.filename)
        global uploadedPath
        uploadedPath="./uploaded/"+filename
        imagefile.save(uploadedPath)
        data = pickle.loads(open('face_enc', "rb").read())
        frame = cv2.imread(uploadedPath)

        frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        faces = face_recognition.face_locations(frame_rgb)
        for face in faces:  # top, right, bottom, left
            face_code = face_recognition.face_encodings(frame_rgb, [face])[0]
            results = face_recognition.compare_faces(data['encodings'], face_code, tolerance=0.6)
            if any(results):
             namee = data['names'][results.index(True)]
             notfound = False
             break
```

Then we apply face encoding function again to the new picture coming from flutter to extract features and then we use compare_faces() function to compare the face with known faces that were stored in file and we find the minimum distance between them to classify person .

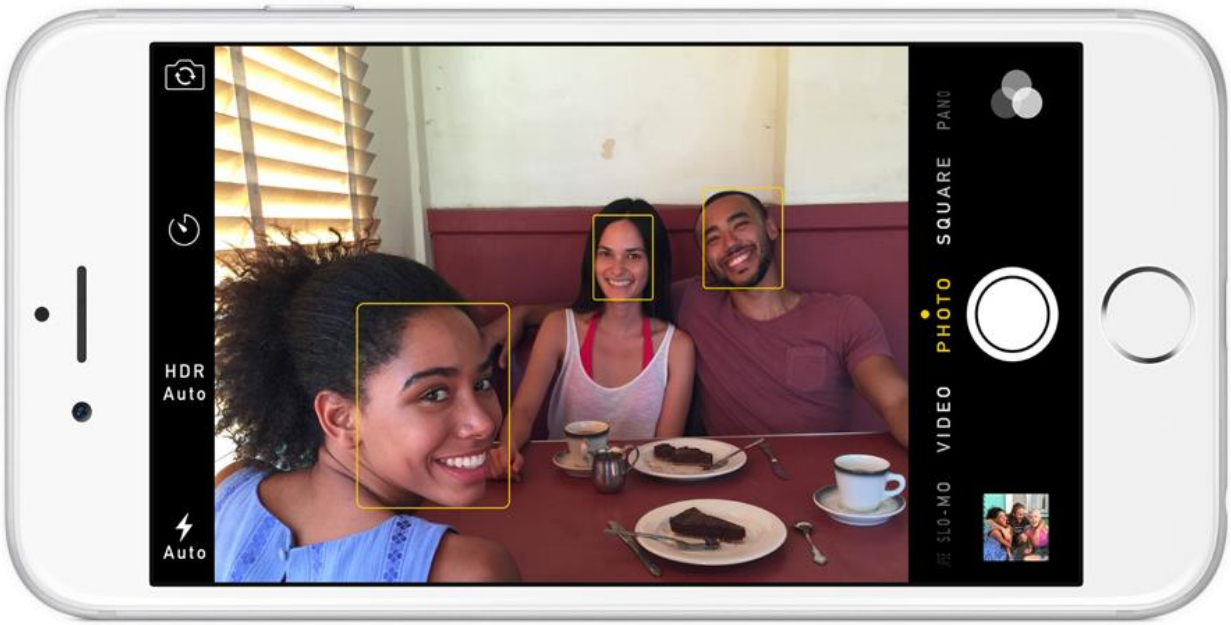
4.1 Algorithm Steps

Step 1: Finding all the Faces

The first step in our pipeline is *face detection*. Obviously we need to locate the faces in a photograph before we can try to tell them apart!

If you've used any camera in the last 10 years, you've probably seen face detection in action:

Figure 12 : Face Detection



Face detection is a great feature for cameras. When the camera can automatically pick out faces, it can make sure that all the faces are in focus before it takes the picture. But we'll use it for a different purpose — finding the areas of the image we want to pass on to the next step in our pipeline.

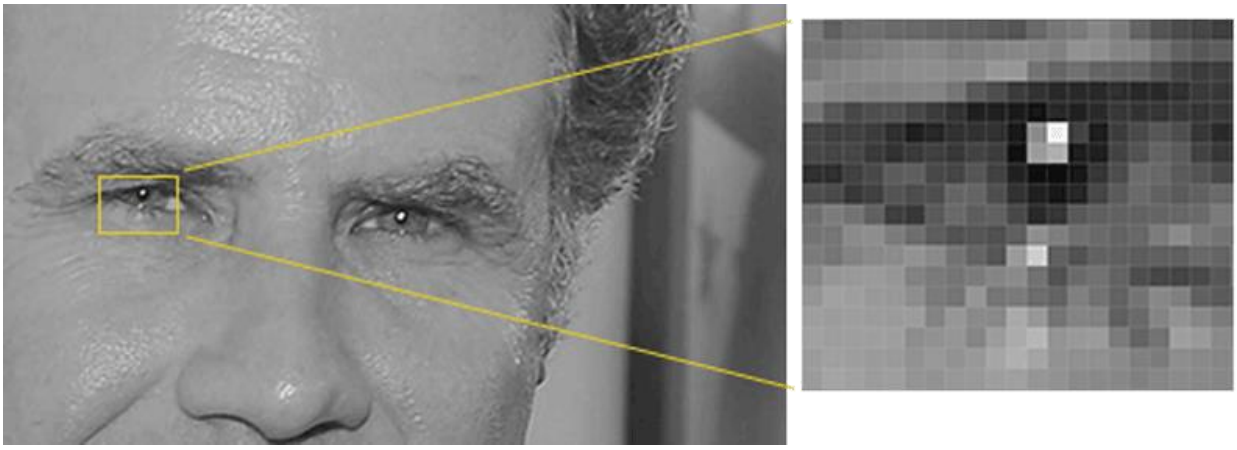
Face detection went mainstream in the early 2000's when Paul Viola and Michael Jones invented a way to detect faces that was fast enough to run on cheap cameras. However, much more reliable solutions exist now. We're going to use a method invented in 2005 called Histogram of Oriented Gradients — or just ***HOG*** for short.

To find faces in an image, we'll start by making our image black and white because we don't need color data to find faces:

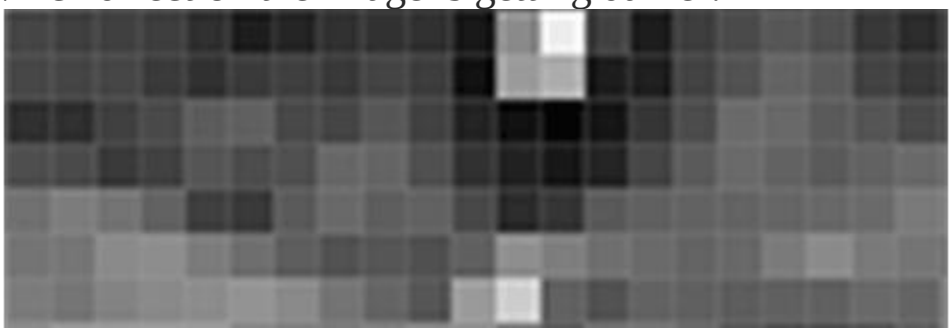
Figure 13 : Turning the image to gray scale

Then we'll look at every single pixel in our image one at a time. For every single pixel, we want to look at the pixels that directly surrounding it:

Figure 14: Convert pixels to unique arrows



Our goal is to figure out how dark the current pixel is compared to the pixels directly surrounding it. Then we want to draw an arrow showing in which direction the image is getting darker:



If you repeat that process for **every single pixel** in the image, you end up with every pixel being replaced by an arrow. These arrows are called *gradients* and they show the flow from light to dark across the entire image:

This might seem like a random thing to do, but there's a really good reason for replacing the pixels with gradients. If we analyze pixels directly, really dark images and really light images of the same person will have totally different pixel values. But by only considering the *direction* that brightness changes, both really dark images and really bright images will end up with the same exact representation. That makes the problem a lot easier to solve!

But saving the gradient for every single pixel gives us way too much detail. We end up missing the forest for the trees. It would be better if we could just see the basic flow of lightness/darkness at a higher level so we could see the basic pattern of the image.

To do this, we'll break up the image into small squares of 16x16 pixels each. In each square, we'll count up how many gradients point in each major direction (how many point up, point up-right, point right, etc…). Then we'll replace that square in the image with the arrow directions that were the strongest.

The end result is we turn the original image into a very simple representation that captures the basic structure of a face in a simple way:

To find faces in this HOG image, all we have to do is find the part of our image that looks the most similar to a known HOG pattern that was extracted from a bunch of other training faces:

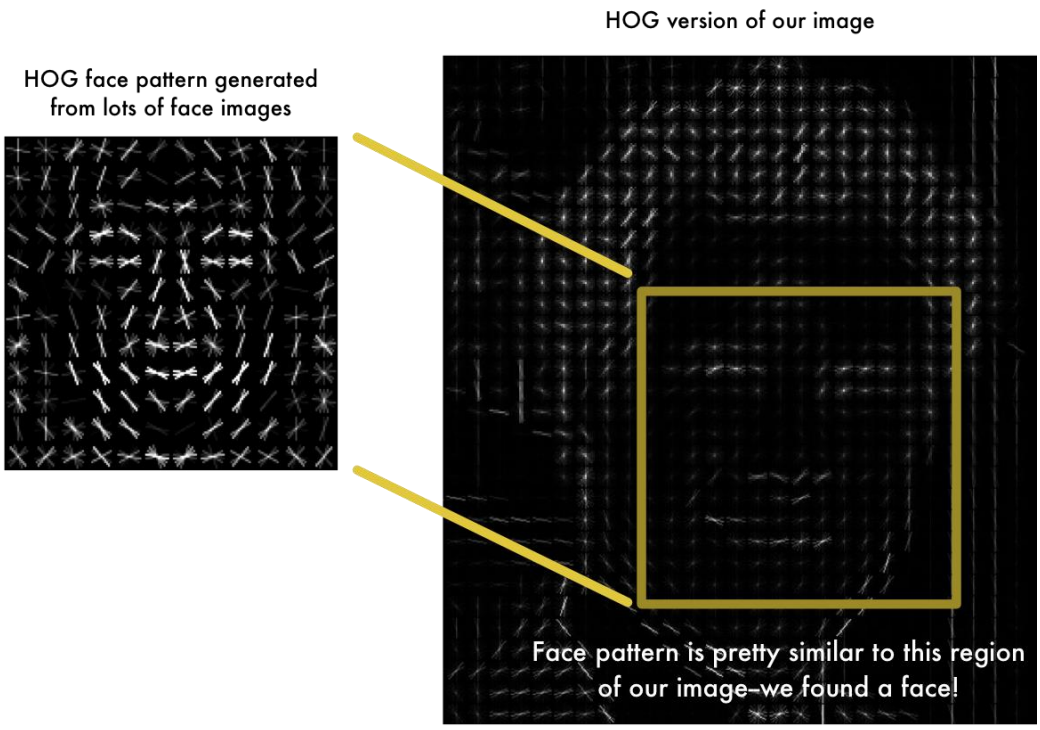
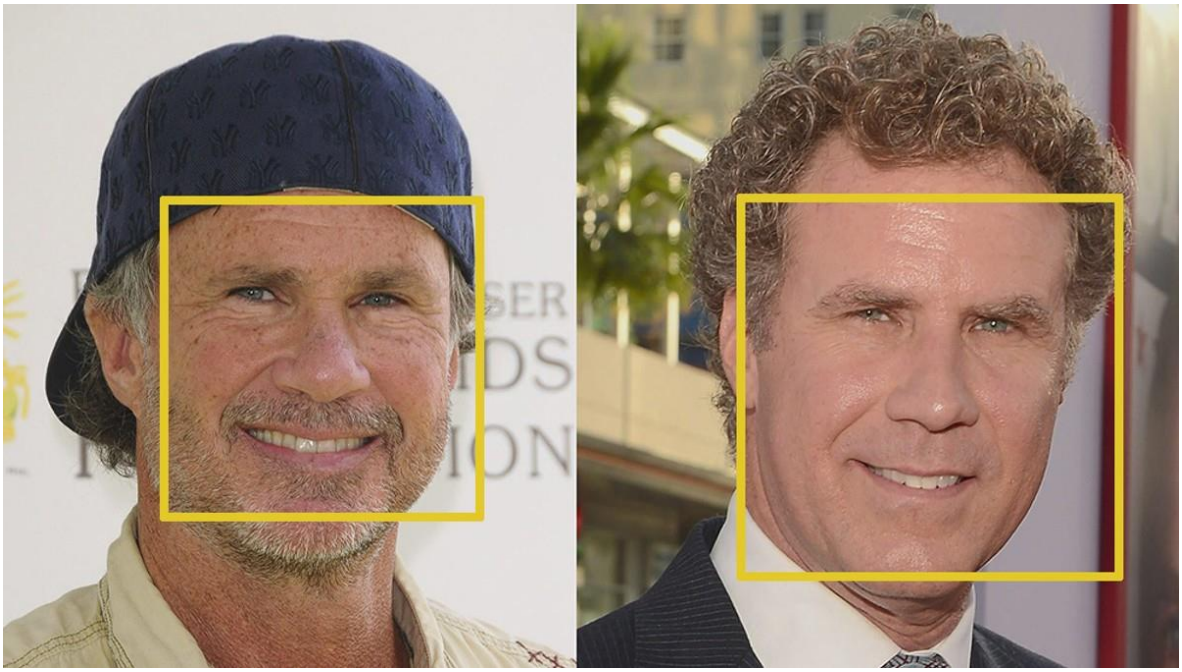


Figure 15 : Hog Transform

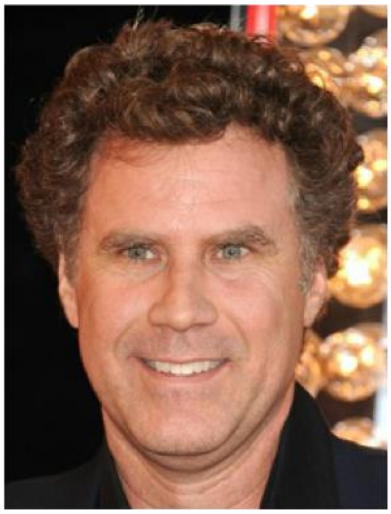
Using this technique, we can now easily find faces in any image:

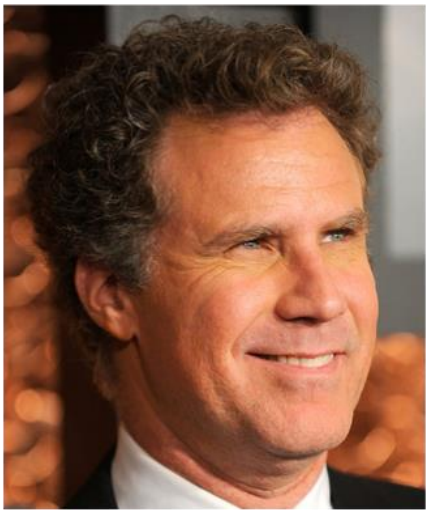


Step 2: Posing and Projecting Faces

Whew, we isolated the faces in our image. But now we have to deal with the problem that faces turned different directions look totally different to a computer:

Figure 16: Projecting faces





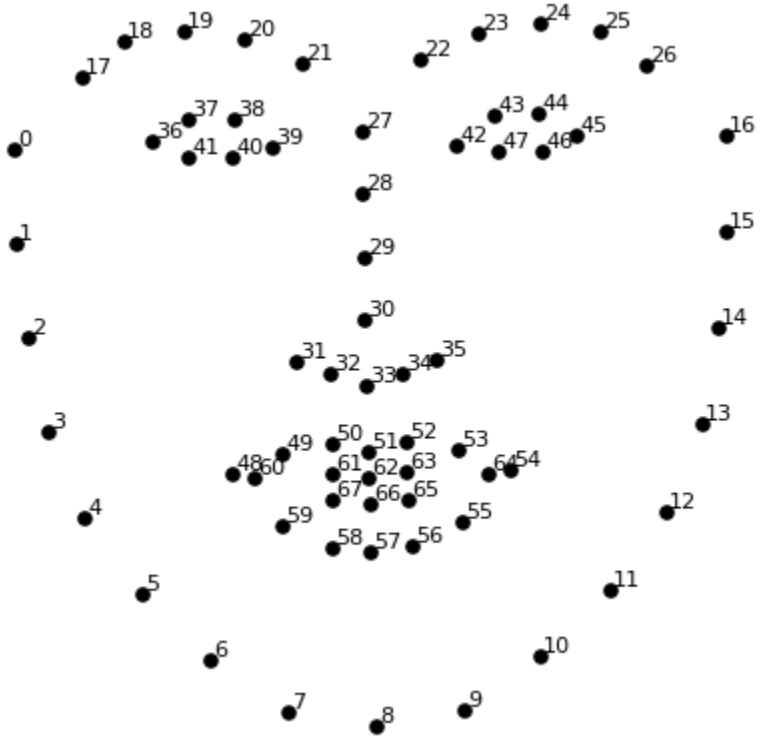
To account for this, we will try to warp each picture so that the eyes and lips are always in the sample place in the image. This will make it a lot easier for us to compare faces in the next steps.

To do this, we are going to use an algorithm called **face landmark estimation**. There are lots of ways to do this, but we are going to use the approach [invented in 2014 by Vahid Kazemi and Josephine Sullivan.]

The basic idea is we will come up with 68 specific points (called *landmarks*) that exist on every face — the top of the chin, the outside edge of each eye, the inner edge of each eyebrow, etc. Then we

will train a machine learning algorithm to be able to find these 68 specific points on any face:

Figure 17: Face Features



Here's the result of locating the 68 face landmarks on our test image:

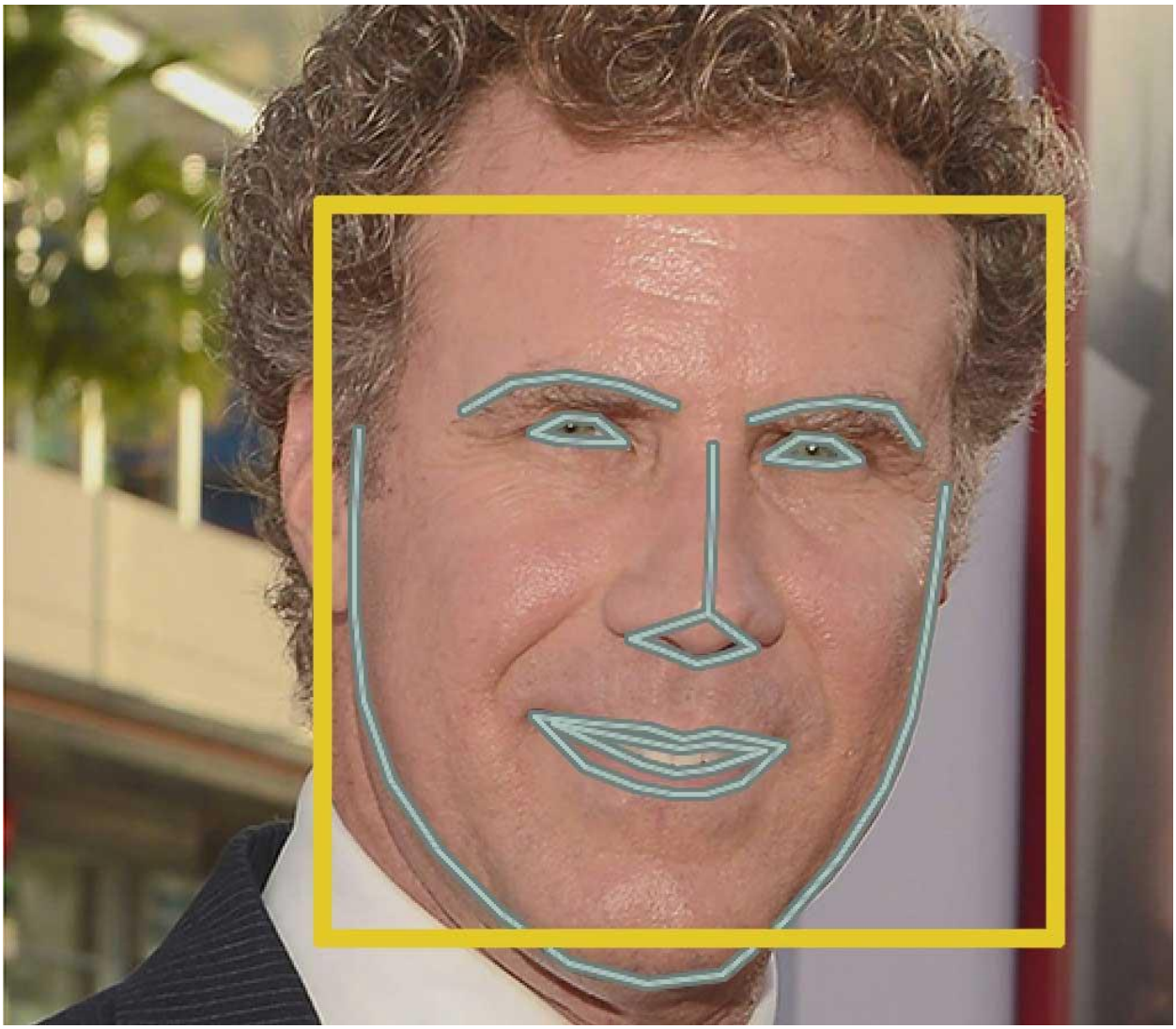


Figure 18: Face Landmarks

Now that we know were the eyes and mouth are, we'll simply rotate, scale and shear the image so that the eyes and mouth are centered as best as possible. We won't do any fancy 3d warps because that would introduce distortions into the image. We are only going to use basic image transformations like rotation and scale that preserve parallel lines (called affine transformations):

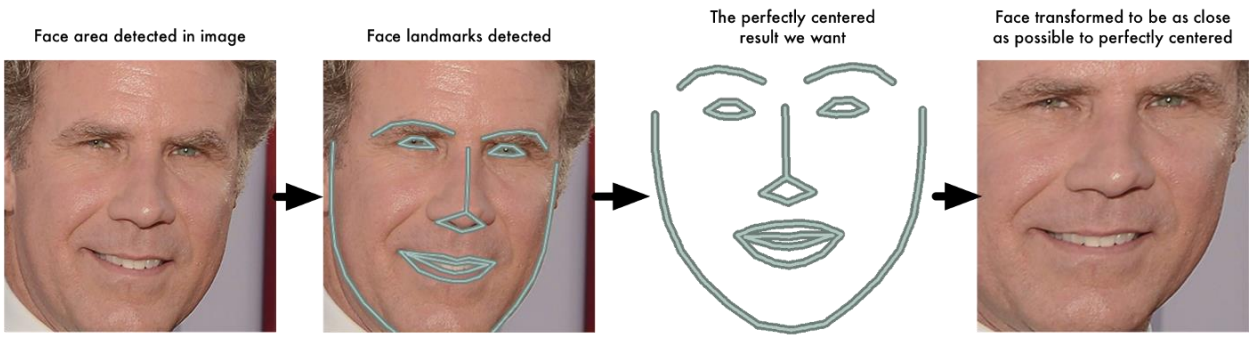


Figure 19 :Centring Face Landmarks

Now no matter how the face is turned, we are able to center the eyes and mouth are in roughly the same position in the image. This will make our next step a lot more accurate.

If you want to try this step out yourself using Python and dlib, here's the code for finding face landmarks and here's the code for transforming the image using those landmarks.

Step 3: Encoding Faces

Now we are to the meat of the problem — actually telling faces apart. This is where things get really interesting!

The simplest approach to face recognition is to directly compare the unknown face we found in Step 2 with all the pictures we have of people that have already been tagged. When we find a previously tagged face that looks very similar to our unknown face, it must be the same person. Seems like a pretty good idea, right?

There's actually a huge problem with that approach. A site like Facebook with billions of users and a trillion photos can't possibly loop through every previous-tagged face to compare it to every newly uploaded picture. That would take way too long. They need to be able to recognize faces in milliseconds, not hours.

What we need is a way to extract a few basic measurements from each face. Then we could measure our unknown face the same way and find the known face with the closest measurements. For example, we might measure the size of each ear, the spacing between the eyes, the length of the nose, etc. If you've ever watched a bad crime show like CSI, you know what I am talking about:

The most reliable way to measure a face

Ok, so which measurements should we collect from each face to build our known face database? Ear size? Nose length? Eye color? Something else?

It turns out that the measurements that seem obvious to us humans (like eye color) don't really make sense to a computer looking at individual pixels in an image. Researchers have discovered that the most accurate approach is to let the computer figure out the measurements to collect itself. Deep learning does a better job than humans at figuring out which parts of a face are important to measure.

The solution is to train a Deep Convolutional Neural Network (just like we did in Part 3). But instead of training the network to recognize pictures objects like we did last time, we are going to train it to generate 128 measurements for each face.

The training process works by looking at 3 face images at a time:

1. Load a training face image of a known person
2. Load another picture of the same known person
3. Load a picture of a totally different person

Then the algorithm looks at the measurements it is currently generating for each of those three images. It then tweaks the neural network slightly so that it makes sure the measurements it generates for #1 and #2 are slightly closer while making sure the measurements for #2 and #3 are slightly further apart:

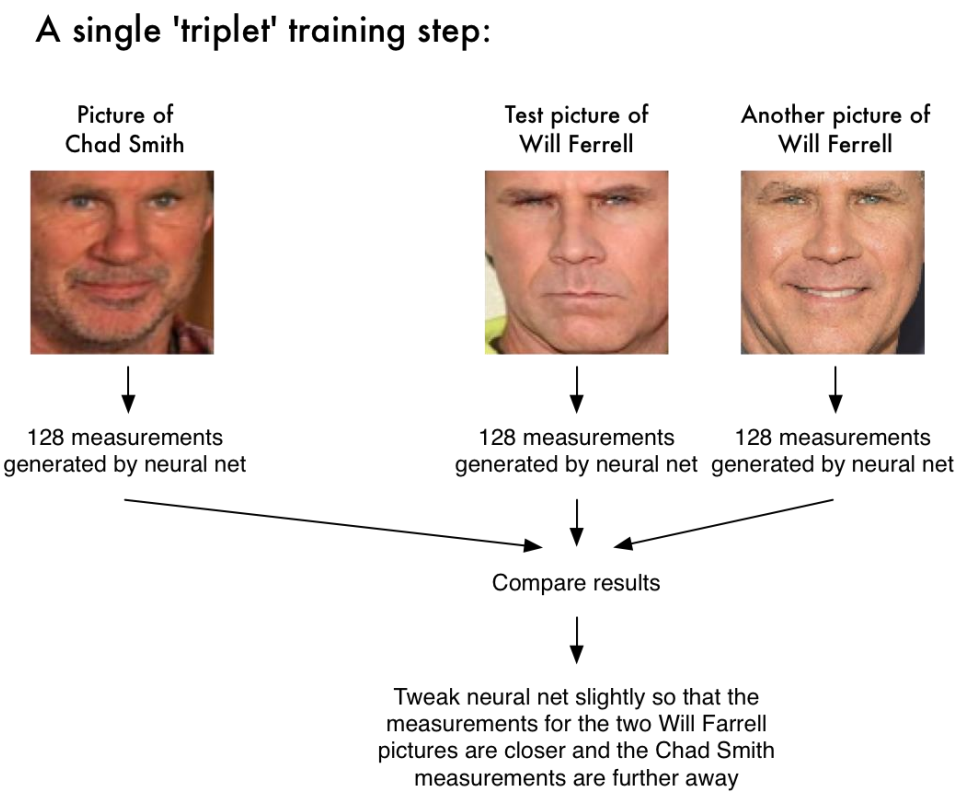


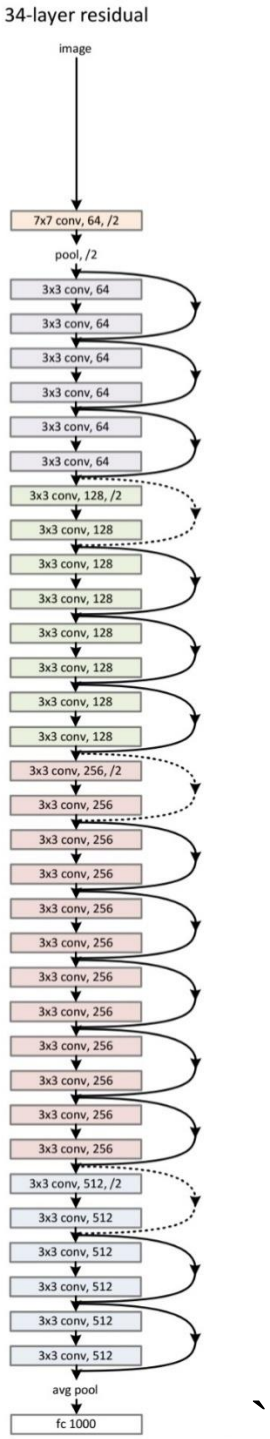
Figure 20 : Triplet Algorithm

After repeating this step millions of times for millions of images of thousands of different people, the neural network learns to reliably generate 128 measurements for each person. Any ten different pictures of the same person should give roughly the same measurements.

Machine learning people call the 128 measurements of each face an **embedding**. The idea of reducing complicated raw data like a picture into a list of computer-generated numbers comes up a lot in machine learning (especially in language translation). The exact approach for faces we are using was invented in 2015 by researchers at Google but many similar approaches exist.

4.2 Neural Network Architecture

Figure 21 :Residual Network



4.3 Technologies Used

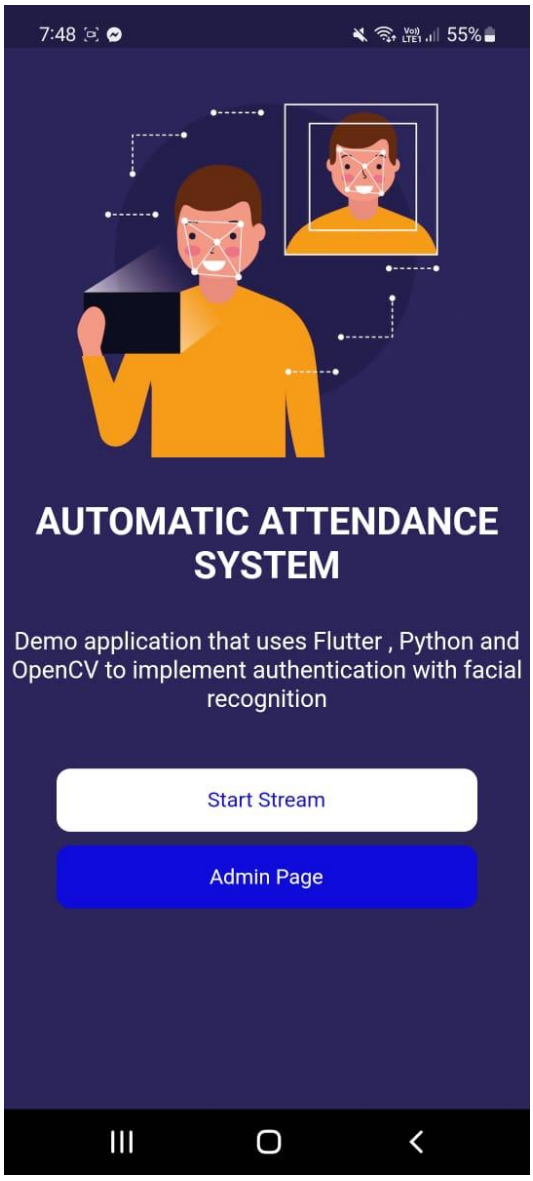
- Flask: api used to provides a local host in python so that flutter connect with it to send and retrieve images and names

- face_Recongiton :api that provides different functions to deal with face recognition with previous mentioned algorithms

-
- Flutter :Framework to create the mobile application, also a cross-platform application

5.User Manual

Figure 22 :Start Stream

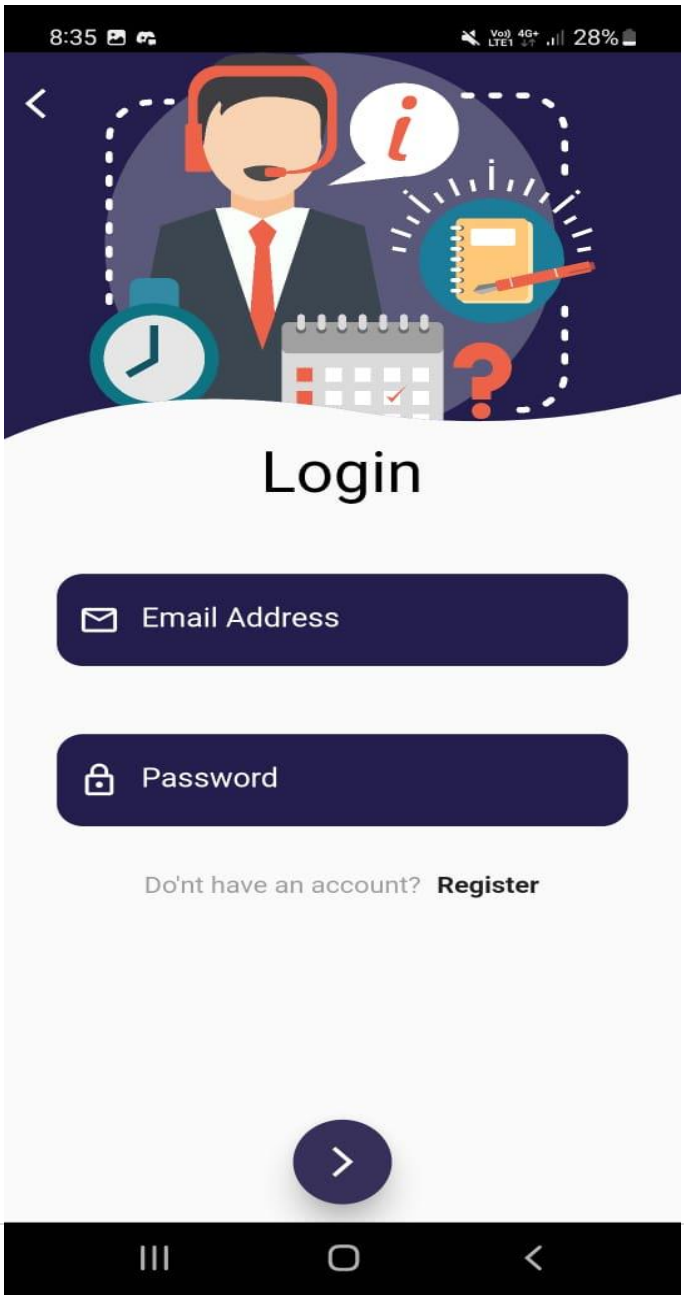


In this screen you have 2 options Start Stream and Admin Page

in Start stream option you can open camera to capture employees or students faces to exctact face features and record their attendence .

in Admin Page option the admin can login or make an account to be able to view each day's attendence , view employees or students profile and their performance , add , delete or update an employee or a student .

Figure 23 : Login Page



When you enter admin page first thing the app ask you to login

To be able to access app functions .

If you don’t have an account its simple you can press on Register

To make a new account and you only need to write your e-mail , password and id .

Like in figure 24.

in Start stream option you can open camera to capture employees or

students faces to exctact face features and record their attendence .

like in figure 22.

Figure 24: Register page

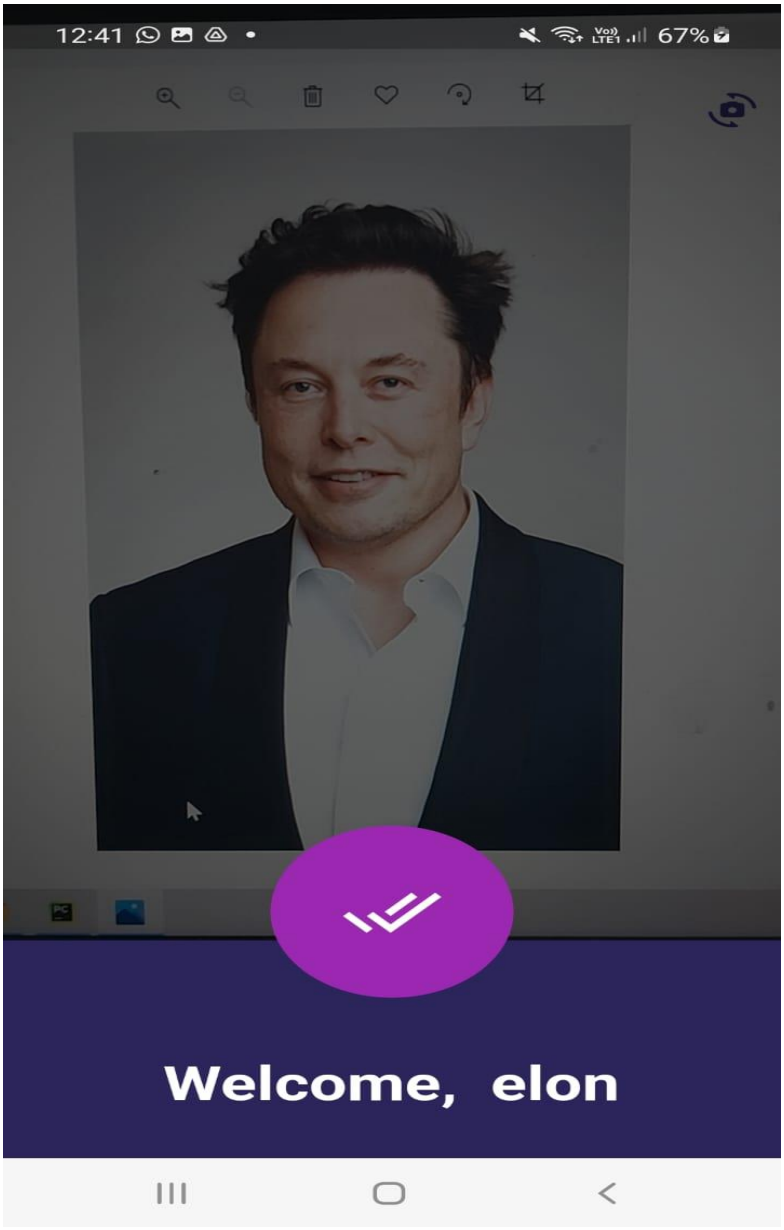


Figure 25: Start Stream

In figure 24 admin will be able to view the attendence of employees or students in each day of current month and their time arrival

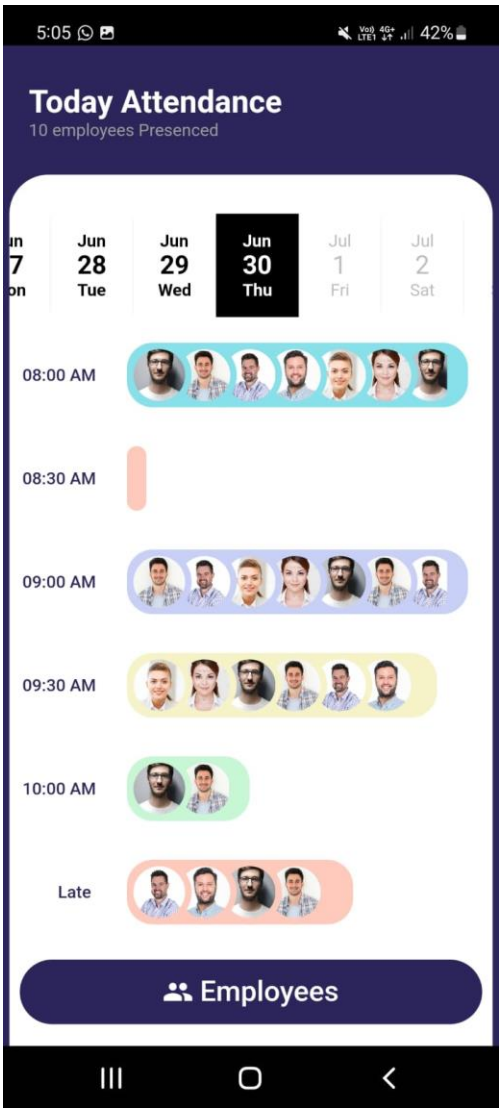


Figure 26:Taking Attendence

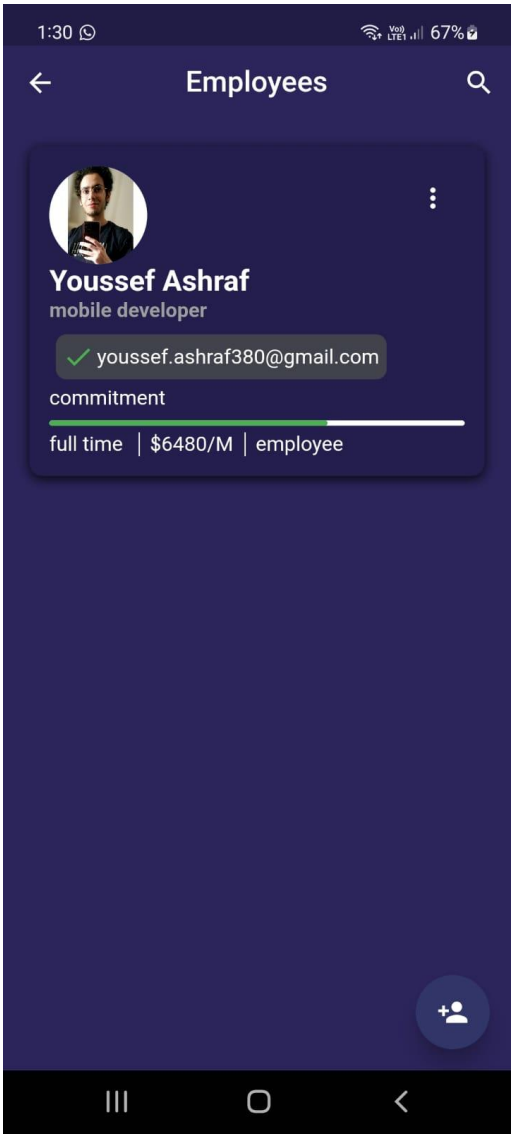


Figure 27 : Employees profile

In figure 27 admin is able to view profile of each employee or student to view or mange their data .

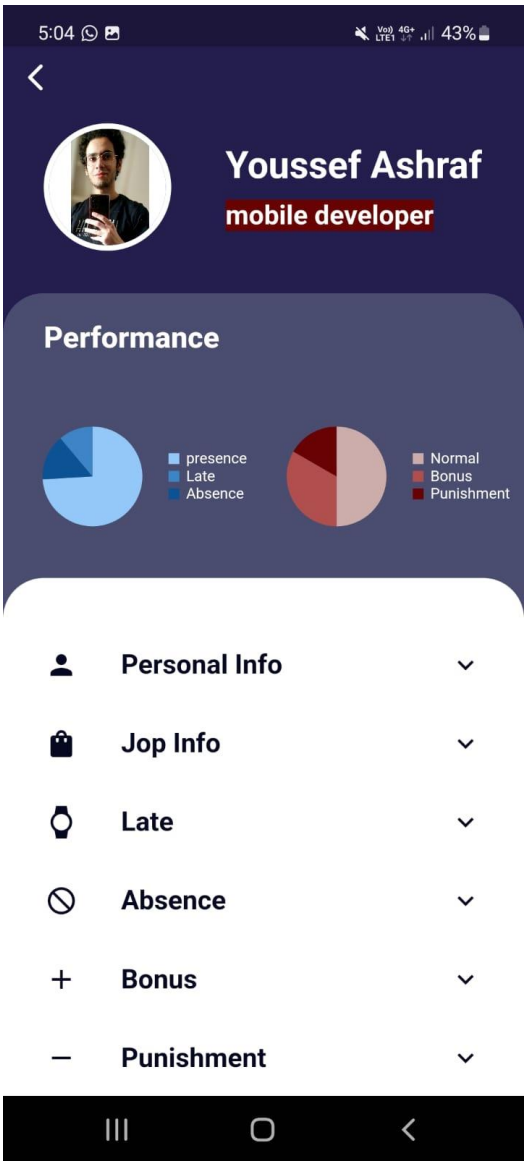


Figure 28: Employees performance and data

In figure 28admin is able to view detailed data of employees like

Personal info , job info , late ,absence , bonus and punishment .

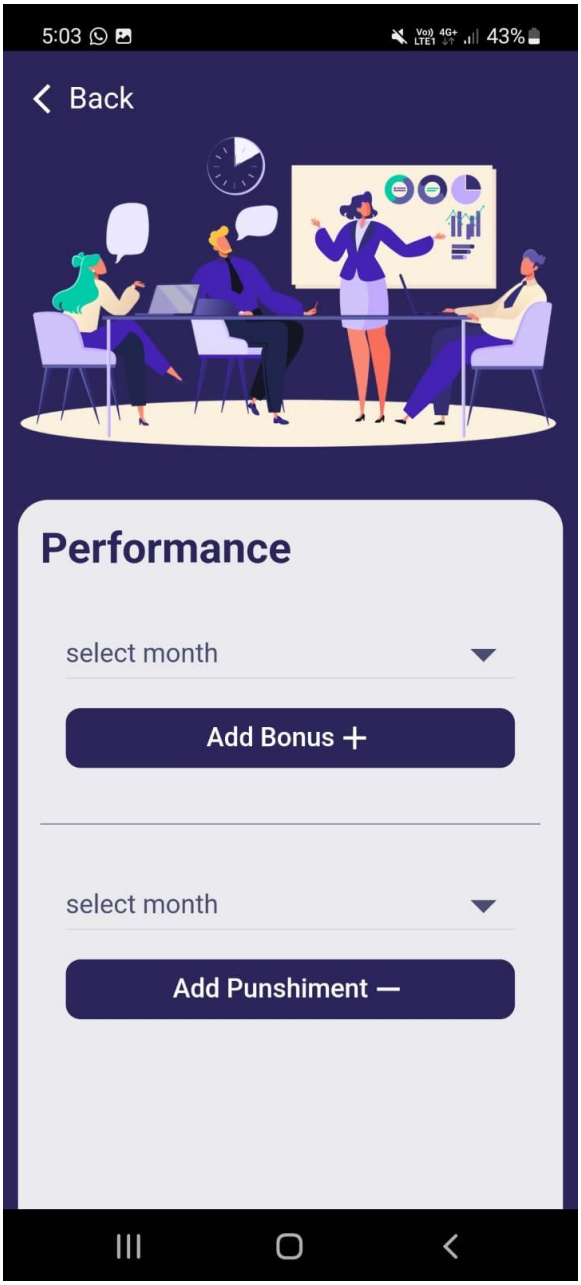


Figure 29 : Add Bonus and Add Punishment

In figure 29 admin can add bonus or punishment monthly .

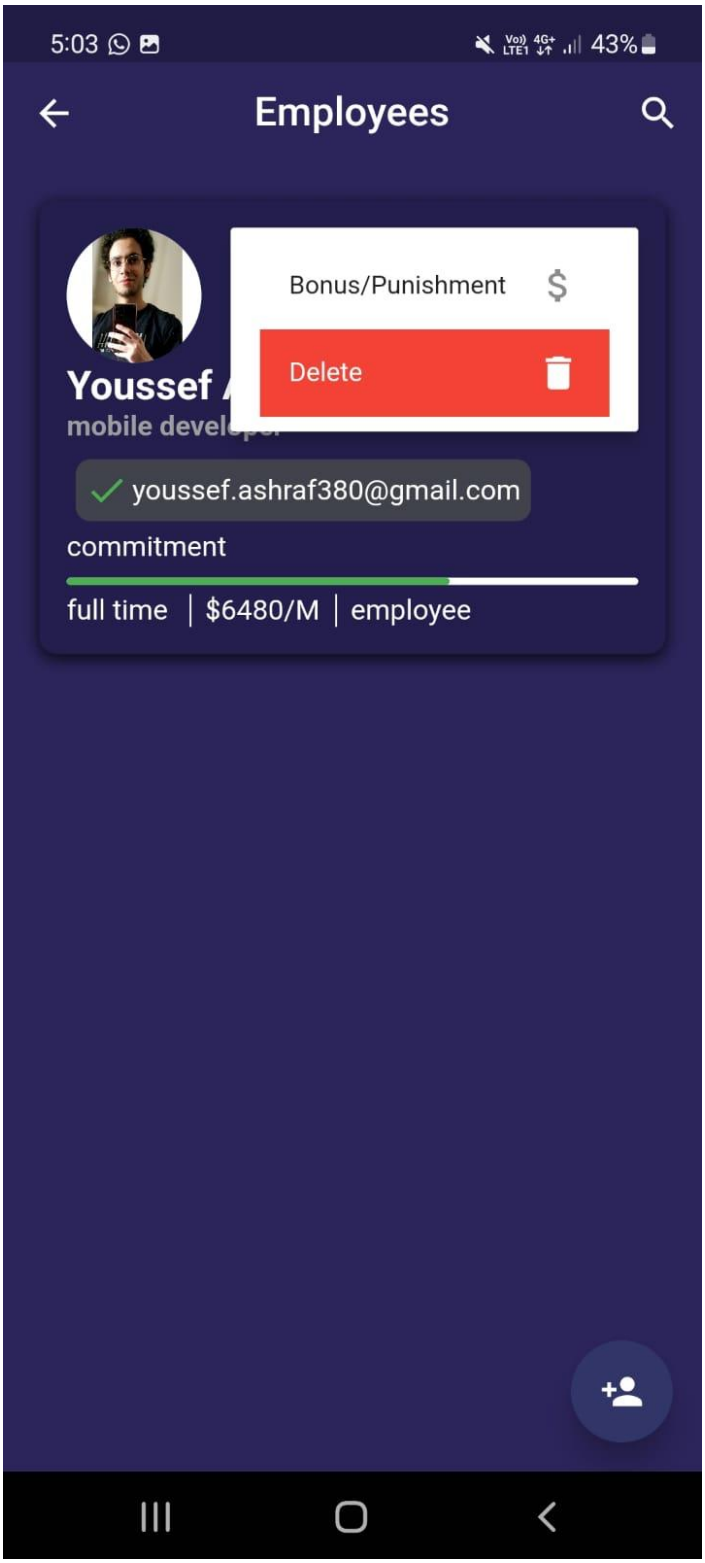


Figure 30: Edit or Delete Employees

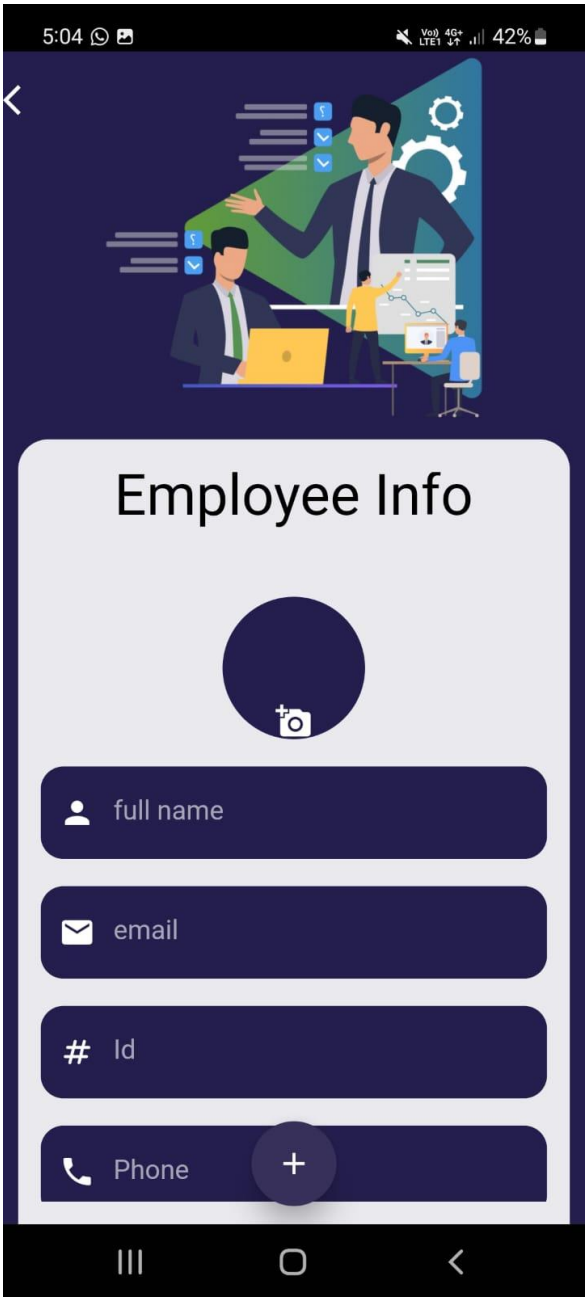


Figure 31: Add Employee

In figure 31 the admin can add a new employees.

6.Conclusion and Future Work

6.1 Conclusion

- In order to obtain the attendance of individuals and to record the entry and exit, the proposed system can be used. The system can widely be used in the institutions/organizations. The proposed system takes attendance of each student by continuous observation at the entry and exit points. The result of our preliminary experiment shows improved performance in the estimation of the attendance compared to the traditional attendance marking systems.
- A system is designed and implemented in real time environment to automate and monitor attendance system. Major goal behind taking attendance automatically is to rectify drawback in conventional method which is time consuming, that causes proxy attendance and wastage of paper. In this work we tried to eradicate all challenges and attendance will be marked using camera in a classroom. Counting of students is done and faces are recognized in image to mark attendance automatically.
- There are various types of seating arrangement, environment and lightning condition in different classroom. Most conditions are tested and system shows 100% accuracy in most cases. In a classroom student may portraying different facial expressions beard, spectacles, varying hair styles etc. all cases are tested and obtained high efficiency and accuracy. Thus concluded from above discussion is that our proposed system is cost effective, secure, reliable, fast, better and efficient module is developed to replace unreliable and manual system.

6.2 Future Work

- In the future we intend to make administrator able to manage the system at the same time the system recording attendance,

- Also we have planned for produce other versions for different operating systems like IOS, Windows, macOS and the Web form, that’s why we decided to use Flutter, It is the most popular cross platform mobile SDK

7.Refrences

[1] M. A. Turk and A. P. Pentland, “Face Recognition Using Eigenfaces,” in Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp. 586–591. 1991.

[2] A. J. Goldstein, L. D. Harmon, and A. B. Lesk, “Identification of Human Faces,” in Proc. IEEE Conference on Computer Vision and Pattern Recognition, vol. 59, pp 748 – 760, May 1971

[3] M. A. Fischler and R. A. Elschlager, “The Representation and Matching of Pictorial Structures,” IEEE Transaction on Computer, vol. C-22, pp. 67-92, 1973.

[4] S. S. R. Abibi, “Simulating evolution: connectionist metaphors for studying human cognitive behaviour,” in Proceedings TENCON 2000, vol. 1 pp 167-173, 2000.

[5] Y. Cui, J. S. Jin, S. Luo, M. Park, and S. S. L. Au, “Automated Pattern Recognition and Defect Inspection System,” in proc. 5 th International Conference on Computer Vision and Graphical Image, vol. 59, pp. 768 – 773, May 1992

[6] Y. Zhang and C. Liu, “Face recognition using kemel principal component analysis and genetic algorithms,” IEEE Workshop on Neural Networks for Signal Processing, pp. 4-6 Sept. 2002.

[7] J. Zhu and Y. L. Yu, “Face Recognition with Eigenfaces,” IEEE International Conference on Industrial Technology, pp. 434 -438, Dec. 1994.

[8] M. H. Yang, N. Ahuja, and D. Kriegmao, “Face recognition using kernel eigenfaces,” IEEE International Conference on Image Processing, vol. 1, pp. 10-13, Sept. 2000.

[9] T. D. Russ, M. W. Koch, and C. Q. Little, “3D Facial Recognition: A Quantitative Analysis,” 38th Annual 2004 International Carnahan Conference on Security Technology, 2004.

[10] P. Sinha, B. Balas, Y. Ostrovsky, and R. Russell, “Face Recognition by Humans: Nineteen Results All Computer Vision Researchers Should Know About,” in Proceedings of the IEEE, vol. 94, Issue 11, 2006.

[11] Y.-W. Kao, H.-Z. Gu, and S.-M. Yuan “Personal based authentication by face recognition,” in proc. Fourth International Conference on Networked Computing and Advanced Information Management, pp 81-85, 2008.

[12] A. T. Acharya and A. Ray, Image Processing: Principles and Applications, New York: Wiley, 2005.