FACE RECOGNITION BASED ATTENDANCE SYSTEM

FACE RECOGNITION BASED ATTENDANCE SYSTEM

A PROJECT REPORT IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR IN TECHNOLOGY

IN

COMPUTER SCIENCE ENGINEERING

BY

NAME	HALLTICKET NO.
P. SAI NAGENDRA	17017T0919
K. HRIDAYA SAIVALLI	17017T0910
G. SAI KIRAN	17017T0909
V. MAHESH	17017T0929
A. SAI KRISHNA	180170936L

UNDER THE GUIDANCE OF

DR. NAGAVELLI RAMANA,

ASSISTANT PROFESSOR, DEPARTMENT OF COMPUTER SCIENCE ENGINEERING,

UNIVERSITY COLLEGE OF ENGINEERING, KAKATIYA UNIVERSITY, KOTHAGUDEM

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING,
UNIVERSITY COLLEGE OF ENGINEERING, KAKATIYA UNIVERSITY,
KOTHAGUDEM

DECLARATION

We hereby declare that the project work with the title "Face Recognition based Attendance System" is our original work done under Dr. Nagavelli Ramana Sir, Assistant Professor, Department of CSE, University College of Engineering, Kakatiya University, Kothagudem. We have learned and followed all the rules and regulations provided by our guide while writing this thesis. This project work is being submitted in the fulfillment of the requirements for the degree of Bachelor of Technology in Computer Science Engineering at University College of Engineering, Kakatiya University, Kothagudem for the academic session 2017– 2021.

NAME	HALLTICKET NO.
P. SAI NAGENDRA	17017T0919
K. HRIDAYA SAIVALLI	17017T0910
G. SAI KIRAN	17017T0909
V. MAHESH	17017T0929
A. SAI KRISHNA	180170936L

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING,

UNIVERSITY COLLEGE OF ENGINEERING, KAKATIYA UNIVERSITY,

KOTHAGUDEM

CERTIFICATE

This is to certify that the major project report titled "Face - recognition based Attendance System" is a record of Bonafide work submitted by P.Sai Nagendra (17017T0919), K.Hridaya Saivalli (17017T0910), G.Sai Kiran (17017T0909), V.Mahesh (17017T0929), A.Sai Krishna (180170936L) in partial fulfillment of the requirements for the award of Bachelor of Technology Degree in Computer Science Engineering at University College of Engineering, Kakatiya University, Kothagudem is an authentic work carried out by them under my supervision and guidance.

Supervisor	Head of the Department
Dr. N. Ramana	Mr. K. Kishor Kumar

External Examiner

ACKNOWLEDGEMENT

On the submission of our thesis report on "Face Recognition based Attendance System", we would like to express our indebted gratitude and special thanks to our guide Dr. Nagavelli Ramana Sir, Principal, Assistant Professor, Department of CSE who in spite of being extraordinarily busy, spare time for guidance and keep us on the correct path and allowing us to carry out our work in the previous one year. We truthfully appreciate and value his admired supervision and support from the start to the end of this project. We are obliged to him for having helped us shape the trouble and providing insights towards the way out.

We would like to offer our special thanks to Mr. Katha. Kishor Kumar Sir, Assistant Professor, Head Of Department for CSE, for providing a solid backdrop for out studies and explore afterward. They have been great sources of motivation to me and we thank them from the core of our hearts. Last but not the least we would like to thank each and every person who is involved directly or indirectly to make this project successful.

NAME	HALLTICKET NO.
P. SAI NAGENDRA	17017T0919
K. HRIDAYA SAIVALLI	17017T0910
G. SAI KIRAN	17017T0909
V. MAHESH	17017T0929
A. SAI KRISHNA	180170936L

ABSTRACT

Automatic face recognition (AFR) technologies have seen dramatic improvements in performance over the past years, and such systems are now widely used for security and commercial applications. An automated system can be used for human face recognition in a real time background for a college to mark the attendance of their students and staff. So Smart Attendance using Real Time Face Recognition is a real-world solution which comes with day-to-day activities of handling students and staff. The task is exceedingly difficult as the real time background subtraction in an image is still a challenge. To detect real time human faces, face recognition technology is used. The recognized face is used to mark attendance of the college students and staff. Our system maintains the attendance records of college automatically. Manual entering of attendance in logbooks becomes a challenging task and it also wastes the time. So, we designed an efficient module that comprises of face recognition to manage the attendance records of college. Our module enrolls the students' and staff's face. During enrolling, their face will be stored in the database. We require a system since enrolling is a onetime process. The presence of each student and staff will be updated in a database. The results showed improved performance over manual attendance management system. Attendance is marked after students and staff identification. This product gives much more solutions with accurate results in user interactive manner rather than existing attendance and leave management systems. If there is any unauthorized person other than the students and staff in database is detected then this model will give the alert to the admin (Principal).

LIST OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

The main objective of this project is to develop face recognition based automated student attendance system. In order to achieve better performance, the test images and training images of this proposed approach are limited to frontal and upright facial images that consist of a single face only. In addition, the students have to register in the database to be recognized. The enrolment can be done on the spot through the user-friendly interface.

1.1 Background

Face recognition is crucial in daily life in order to identify family, friends or someone we are familiar with. We might not perceive that several steps have actually taken in order to identify human faces. Human intelligence allows us to receive information and interpret the information in the recognition process. We receive information through the image projected into our eyes, by specifically retina in the form of light. Light is a form of electromagnetic waves which are radiated from a source onto an object and projected to human vision. Robinson-Riegler, G., & Robinson-Riegler, B. mentioned that after visual processing done by the human visual system, we actually classify shape, size, contour and the texture of the object in order to analyze the information. The analyzed information will be compared to other representations of objects or face that exist in our memory to recognize. In fact, it is a hard challenge to build an automated system to have the same capability as a human to recognize faces. However, we need large memory to recognize different faces, for example, in the Universities, there are a lot of students with different race and gender, it is impossible to remember every face of the individual without making mistakes. In order to overcome human limitations, computers with almost limitless memory, high processing speed and power are used in face recognition systems.

The human face is a unique representation of individual identity. Thus, face recognition is defined as a biometric method in which identification of an individual is performed by comparing real-time capture image with stored images in the database of that person. Nowadays, face recognition system is prevalent due to its simplicity and awesome performance. For instance, airport protection systems and FBI use face recognition for criminal investigations by tracking suspects, missing children and drug activities. Apart from that, Facebook which is a popular social networking website implement face recognition to allow the users to tag their friends in the photo for entertainment purposes. Furthermore, Intel Company allows the users to use face recognition to get access to their online account. Apple allows the users to unlock their mobile phone, iPhone X by using face recognition.

The work on face recognition began in 1960. Woody Bledsoe, Helen Chan Wolf and Charles Bisson had introduced a system which required the administrator to locate eyes, ears, nose and mouth from images. The distance and ratios between the located features and the common reference points are then calculated and compared. The studies are further enhanced by Goldstein, Harmon, and Lesk in 1970 by using other features such as hair color and lip thickness to automate the recognition. In 1988, Kirby and Sirovich first suggested principal component analysis (PCA) to solve face recognition problem. Many studies on face recognition were then conducted continuously until today.

1.2 Problem Statement

Traditional student attendance marking technique is often facing a lot of trouble. The face recognition student attendance system emphasizes its simplicity by eliminating classical student attendance marking technique such as calling student names or checking respective identification cards. There are not only disturbing the teaching process but also causes distraction for students during exam sessions. Apart from calling names, attendance sheet is passed around the classroom during the lecture sessions. The lecture class especially the class with a large number of students might find it difficult to have the attendance sheet being passed around the class. Thus, face recognition student attendance system is proposed in order to replace the manual signing of the presence of students which are burdensome and causes students get distracted in order to sign for their attendance. Furthermore, the face recognition based automated student attendance system able to overcome the problem of fraudulent approach and lecturers does not have to count the number of students several times to ensure the presence of the students.

The paper proposed by Zhao, W et al. (2003) has listed the difficulties of facial identification. One of the difficulties of facial identification is the identification between known and unknown images. In addition, paper proposed by Pooja G.R et al. (2010) found out that the training process for face recognition student attendance system is slow and time-consuming. In addition, the paper proposed by Priyanka Wagh et al. (2015) mentioned that different lighting and head poses are often the problems that could degrade the performance of face - recognition based student attendance system.

Hence, there is a need to develop a real time operating student attendance system which means the identification process must be done within defined time constraints to prevent omission. The extracted features from facial images which represent the identity of the students have to be consistent towards a change in background, illumination, pose and expression. High accuracy and fast computation time will be the evaluation points of the performance.

1.3 Aims and Objectives

The objective of this project is to develop face recognition based automated student attendance system. Expected achievements in order to fulfill the objectives are:

- To capture the image from the external video device.
- To detect the face from the image.
- To extract the useful features from the face detected.
- To classify the features in order to recognize the face detected.
- To record the attendance of the identified student.

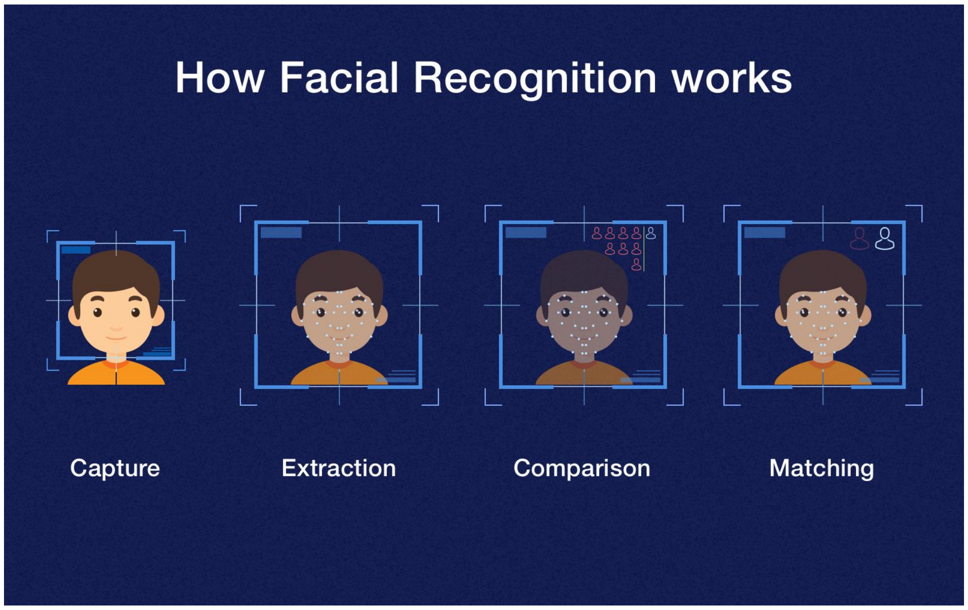


Fig. 1.1 Block Diagram of the General Framework

1.4 Thesis Organization

This section gives an overview of the whole documentation. Next, we will discuss chapter 2 about Literature review. Chapter 3 discusses about various technologies we used in various stages of this project. Chapter 4 discusses about the methodology of our project in detail. Chapter 5 explains about the ER diagram. Chapter 6 shows the results and discussions of our whole project and finally Chapter 7 gives various conclusions and future enhancements we want implement further in the future. Chapter 8 gives the references of our project in terms of concepts we used for finishing this project documentation.

CHAPTER 2

LITERATURE REVIEW

2.1 Student Attendance System

There are a lot disadvantages for RFID (Radio Frequency Identification) card system, fingerprint system and iris recognition system. RFID card system is implemented due to its simplicity. However, the user tends to help their friends to check in as long as they have their friend's ID card. The fingerprint system is indeed effective but not efficient because it takes time for the verification process so the user has to line up and perform the verification one by one. However for face recognition, the human face is always exposed and contain less information compared to iris. Iris recognition system which contains more detail might invade the privacy of the user. Voice recognition is available, but it is less accurate compared to other methods. Hence, face recognition system is suggested to be implemented in the student attendance system.

System type	Advantages	Disadvantages
RFID card system	Simple	Fraudulent usage
Fingerprint system	Accurate	Time-consuming
Voice recognition system	_	Less accurate compared to others
Iris recognition system	Accurate	Privacy Invasion

Table 2.1 Advantages & Disadvantages of Different Biometric System

2.2 Face Detection

Difference between face detection and face recognition are often misunderstood. Face detection is to determine only the face segment or face region from image, whereas face recognition is to identify the owner of the facial image. Some factors can be presented which cause face detection and face recognition to encounter difficulties. These factors consist of background, illumination, pose, expression, occlusion, rotation, scaling and translation. The definition of each factor is tabulated in Table 2.2.

Background	Variation of background and environment around people in the image which affect the efficiency of face recognition.
Illumination	Illumination is the variation caused by various lighting environments which degrade the facial feature detection.
Pose	Pose variation means different angle of the acquired the facial image which cause distortion to recognition process, especially for Eigen face and Fisher face recognition method.
Expression	Different facial expressions are used to express feelings and emotions. The expression variation causes spatial relation change and the facial-feature shape change.
Occlusion	Occlusion means part of the human face is unobserved. This will diminish the performance of face recognition algorithms due to deficiency information.
Rotation, scaling and translation	Transformation of images which might cause distortion of the original information about the images.

Table 2.2 Factors causing Face Detection Difficulties

There are a few face detection methods that the previous researchers have worked on. However, most of them used frontal upright facial images which consist of only one face. The face region is fully exposed without obstacles and free from the spectacles.

We often see face detection feature in our cameras in mobile phones. 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. We can use this feature for finding the areas of the image we captured.

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:

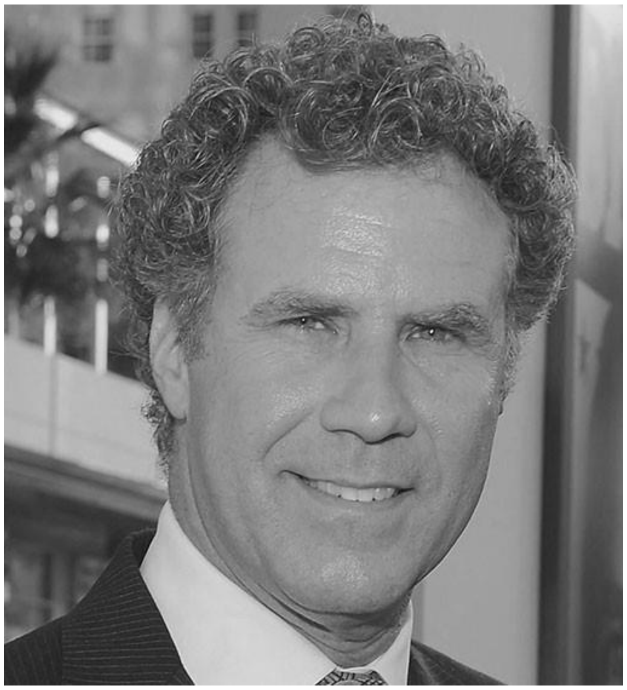


Fig 2.1 Test Image

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.

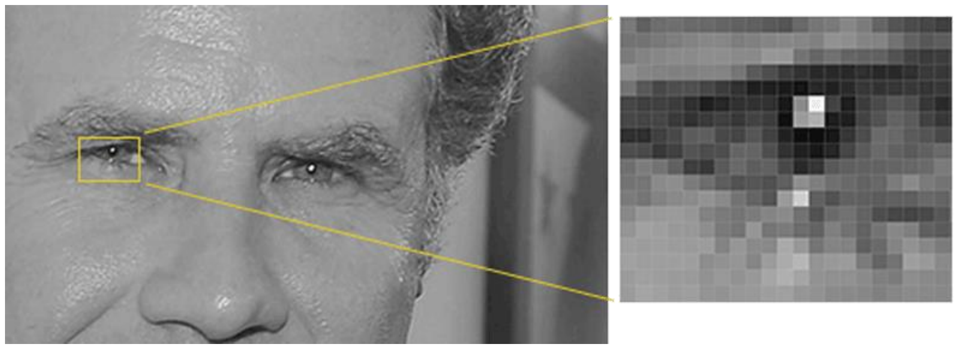


Fig 2.2 Looking into every pixel of our test image

We need 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.

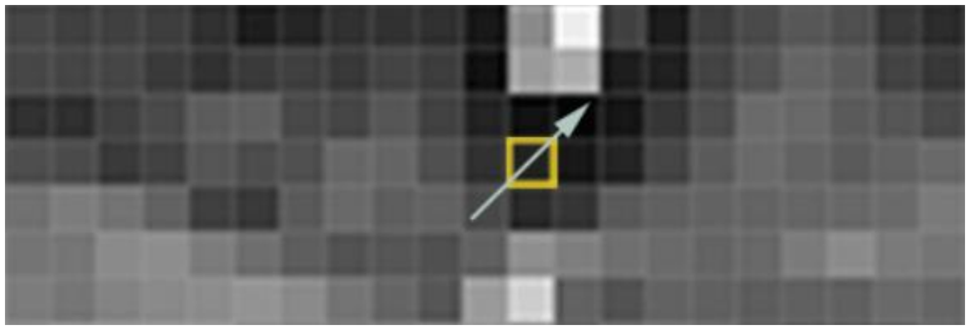


Fig 2.3 Looking at which direction the image is getting darker

Note: The image is getting darker towards upper right in case of above picture.

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.

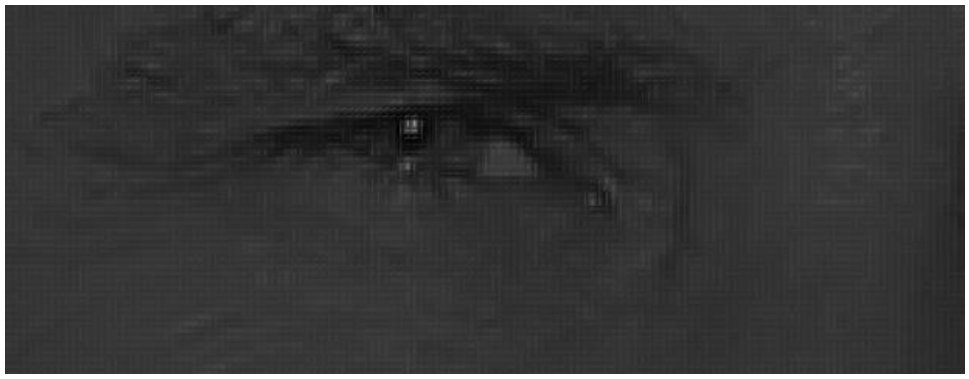
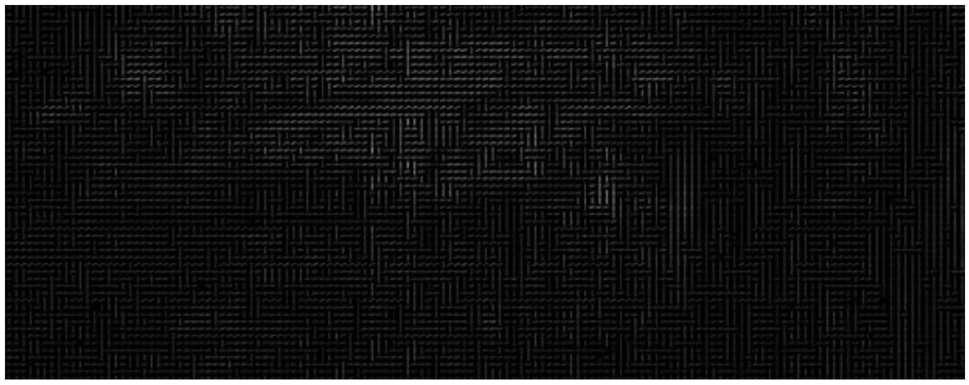



Fig 2.4 Series of images extracting the gradient image

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.

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.

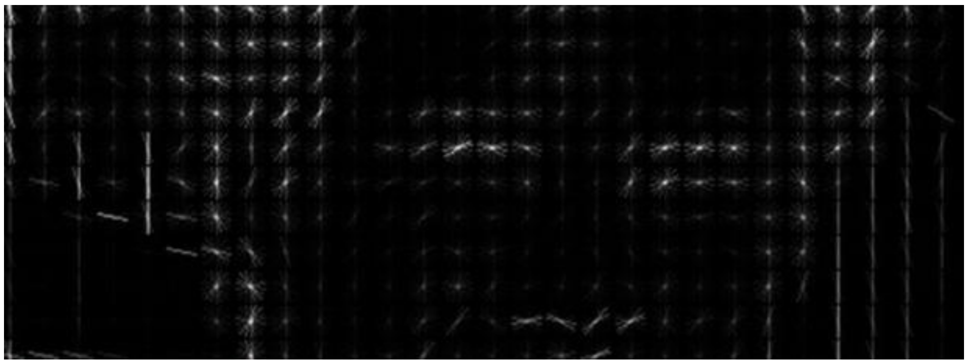


Fig 2.5 Optimized gradient picture

The original image is turned into a HOG representation that captures the major features of the image regardless of image brightness.

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.

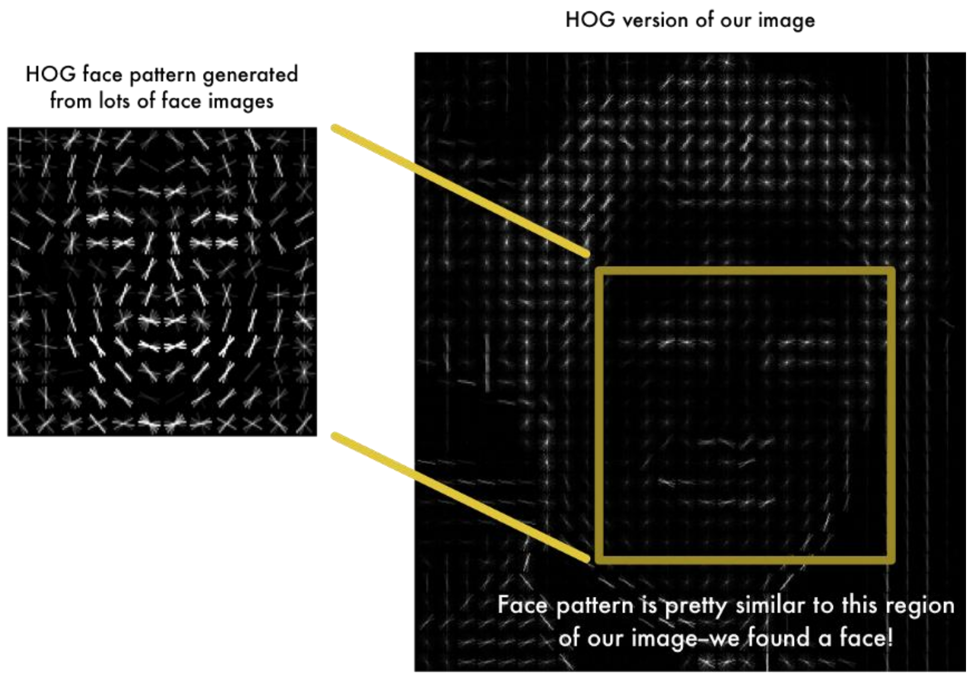

HOG version of our image
HOG face pattern generated from lots of face images
Face pattern is pretty similar to this region of our image–we found a face!

Fig 2.6 HOG Image

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

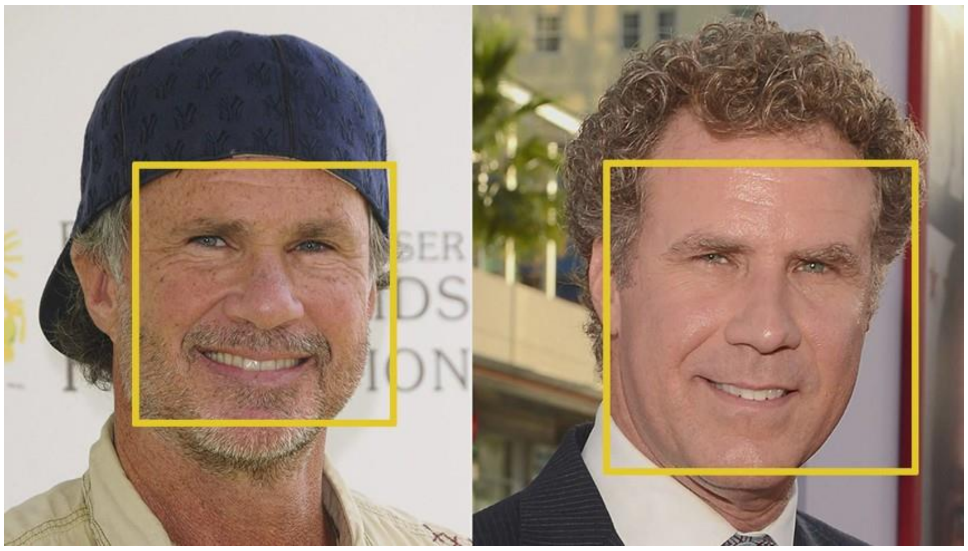


Fig 2.7 Face detection Images

We have discussed several face detection difficulties in before paragraphs. These includes tilted images also. A human can find to whom a tilted face belongs but for computer it is still two different images.

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.

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.

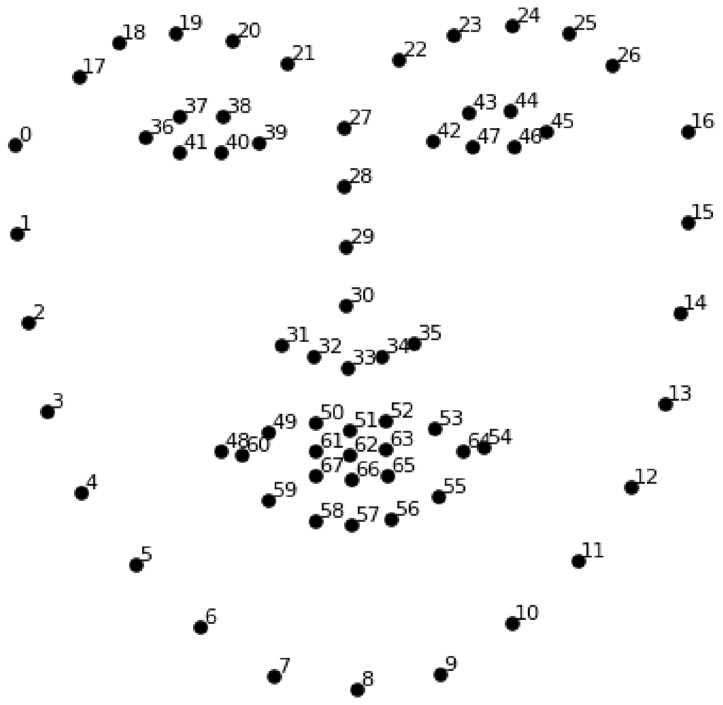


Fig 2.8 68 landmarks of detected face

The result of locating the 68 face landmarks on our above image will look like this.

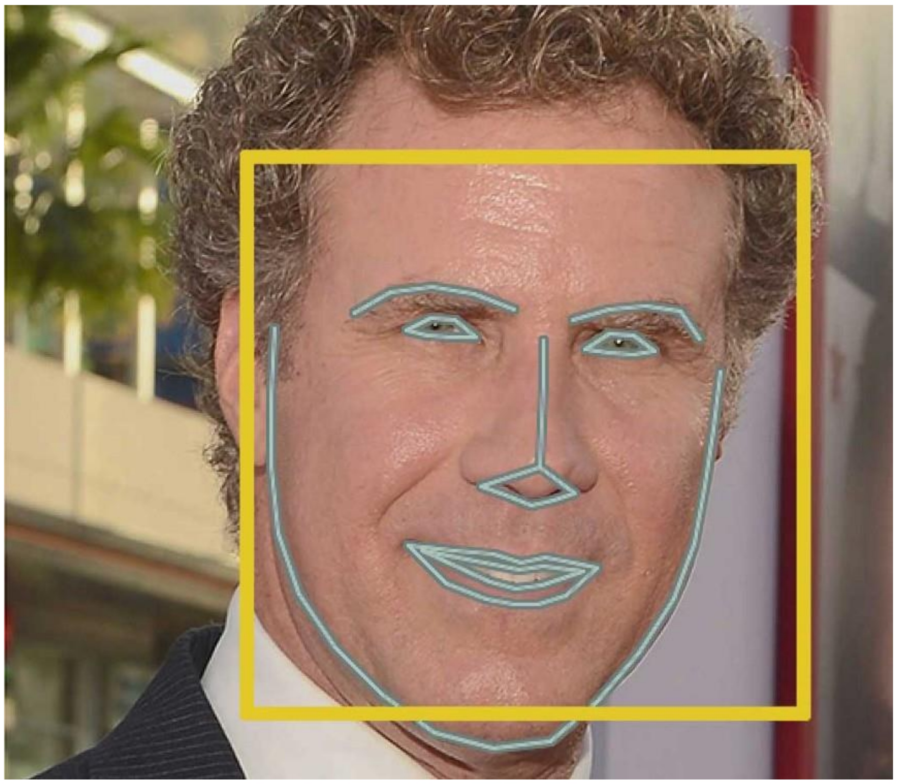


Fig 2.9 Result of 68 landmarks of detected face

Now that we know where 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).

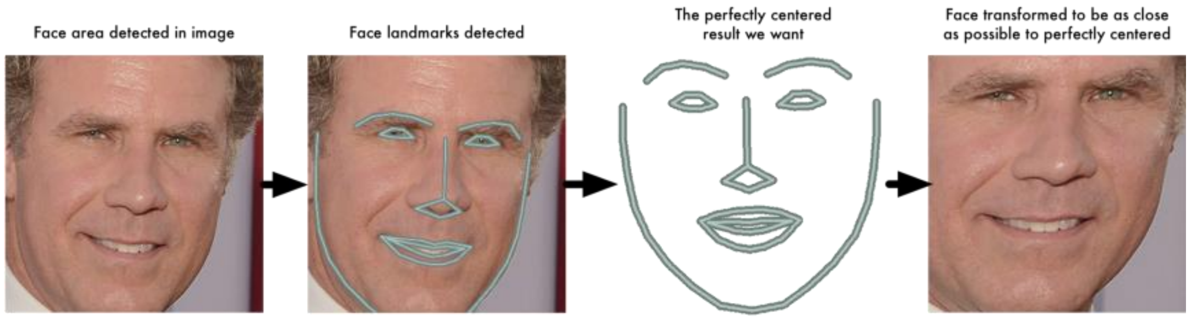

Face area detected in image
Face landmarks detected
The perfectly centered result we want
Face transformed to be as close as possible to perfectly centered

Fig 2.10 Perfectly centered face image

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.

2.3 Face Recognition

The recognition of face of human is challenging in computer-human interaction. The face is our essential center of consideration in societal life playing a critical part in assigning identification and emotion of the person. We can perceive various appearances adapted all through our lifespan and distinguish faces initially even following quite a while of detachment. This expertise is very vigorous notwithstanding of substantial varieties in visual boost because of evolving condition, maturing and diversions, for example, facial hair, glasses or changes in haircut.

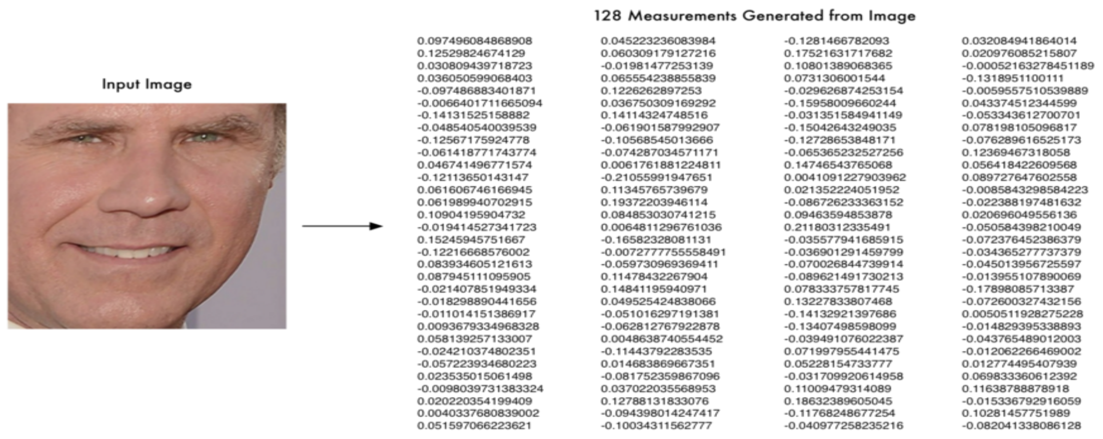
For the application of face recognition, detection of face is very important and the first step. After detecting face, the face recognition algorithm can only be functional. Face detection itself involves some complexities for example surroundings, postures, enlightenment etc.

The simplest approach to face recognition is to directly compare the unknown face 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.

But with this approach, we cannot perform the same operation of bulk number of images. To overcome that, we let the computer figure out the measurements to collect the measurements 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. But instead of training the network to recognize pictures objects, we are going to train it to generate 128 measurements for each face.

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. But for us it is impossible as we require a lot of computational power in order to achieve this. What we can we do ourselves is to run our face images through a pre-trained network to get the 128 measurements for each face. Here’s the measurements for our example image.


Input Image

128 Measurements Generated from Image

0.097496084868908	0.045223236083984	-0.1281466782093	0.032084941864014
0.12529824674129	0.060309179127216	0.17521631717682	0.020976085215807
0.030809439718723	-0.01981477253139	0.10801389068365	-0.00052163278451189
0.036050599068403	0.065554238855839	0.0731306001544	-0.1318951100111
-0.097486883401871	0.1226262897253	-0.029626874253154	-0.0059557510539889
-0.0066401711665094	0.036750309169292	-0.15958009660244	0.043374512344599
-0.14131525158882	0.14114324748516	-0.031351584941149	-0.053343612700701
-0.048540540039539	-0.061901587992907	-0.15042643249035	0.078198105096817
-0.12567175924778	-0.10568545013666	-0.12728653848171	-0.076289616525173
-0.061418771743774	-0.074287034571171	-0.065365232527256	0.12369467318058
0.046741496771574	0.0061761881224811	0.14746543765068	0.056418422609568
-0.12113650143147	-0.21055991947651	0.0041091227903962	0.089727647602558
0.061606746166945	0.11345765739679	0.021352224051952	-0.0085843298584223
0.061989940702915	0.19372203946114	-0.086726233363152	-0.022388197481632
0.10904195904732	0.084853030741215	0.09463594853878	0.020696049556136
-0.019414527341723	0.0064811296761036	0.21180312335491	-0.050584398210049
0.15245945751667	-0.16582328081131	-0.035577941685915	-0.072376452386379
-0.12216668576002	-0.0072777755558491	-0.036901291459799	-0.034365277737379
0.083934605121613	-0.059730969369411	-0.070026844739914	-0.045013956725597
0.087945111095905	0.11478432267904	-0.089621491730213	-0.013955107890069
-0.021407851949334	0.14841195940971	0.078333757817745	-0.17898085713387
-0.018298890441656	0.049525424838066	0.13227833807468	-0.072600327432156
-0.011014151386917	-0.051016297191381	-0.14132921397686	0.0050511928275228
0.0093679334968328	-0.062812767922878	-0.13407498598099	-0.014829395338893
0.058139257133007	0.0048638740554452	-0.039491076022387	-0.043765489012003
-0.024210374802351	-0.11443792283535	0.071997955441475	-0.012062266469002
-0.057223934680223	0.014683869667351	0.05228154733777	0.012774495407939
0.023535015061498	-0.081752359867096	-0.031709920614958	0.069833360612392
-0.0098039731383324	0.037022035568953	0.11009479314089	0.11638788878918
0.020220354199409	0.12788131833076	0.18632389605045	-0.015336792916059
0.0040337680839002	-0.094398014247417	-0.11768248677254	0.10281457751989
0.051597066223621	-0.10034311562777	-0.040977258235216	-0.082041338086128

Fig 2.11 128 measurements from detected face image

We need to find the person in our database of known people who has the closest measurements to our example image.

CHAPTER 3

USED TECHNOLOGIES

3.1 What is Computer Vision?

Computer vision is a process by which we can understand the images and videos how they are stored and how we can manipulate and retrieve data from them. Computer Vision is the base or mostly used for Artificial Intelligence. Computer-Vision is playing a major role in self-driving cars, robotics as well as in photo correction apps.

3.2 What is OpenCV?

In the field of Artificial Intelligence, Computer Vision is one of the most interesting and Challenging tasks. Computer Vision acts like a bridge between Computer Software and visualizations around us. It allows computer software to understand and learn about the visualizations in the surroundings. For Example: Based on the color, shape and size determining the fruit. This task can be very easy for the human brain however in the Computer Vision pipeline, first we gather the data, then we perform the data processing activities and then we train and teach the model to understand how to distinguish between the fruits based on size, shape and color of fruit.

Currently, various packages are present to perform machine learning, deep learning and computer vision tasks. By far, computer vision is the best module for such complex activities. OpenCV is an open-source library. It is supported by various programming languages such as R, Python. It runs on most of the platforms such as Windows, Linux and MacOS.

3.3 Advantages of OpenCV

- OpenCV is an open-source library and is free of cost.
- As compared to other libraries, it is fast since it is written in C/C++.
- It works better on System with lesser RAM
- It supports most of the Operating Systems such as Windows, Linux and MacOS.

3.4 Implementation of OpenCV

We shall implement face recognition using OpenCV and Python. For this, we need some libraries and have to install them in our IDE. The libraries are:

- OpenCV
- dlib
- Face_recognition

OpenCV is an image and video processing library and is used for image and video analysis, like facial detection, license plate reading, photo editing, advanced robotic vision, optical character recognition, and a whole lot more.

The dlib library, maintained by Davis King, contains our implementation of "deep metric learning" which is used to construct our face embeddings used for the actual recognition process.

The face_recognition library, created by Adam Geitgey, wraps around dlib's facial recognition functionality, and this library is super easy to work with and we will be using this in our code. Remember to install dlib library first before you install face_recognition.

To install OpenCV, type in command prompt

```
pip install opencv-python
```

We can install dlib using anaconda or we can either download and install a dlib file. We downloaded and installed the dlib file.

To install dlib using anaconda, we use the following command in our command prompt.

```
conda install –c conda-forge dlib
```

Next to install face_recognition, we type the following command in command prompt.

```
pip install face_recognition
```

3.5 Python Functions

As we implemented the backend of our project in Python, we used several basic python functions and methods in our backend development. These are very basic and can be understand by anyone due to its simplicity.

They are:

1. append()
2. split()
3. upper()
4. readlines()
5. writelines()

1.append() :

The append() method appends an element to the end of the list.

```
list.append(elmnt)
```

- elmnt is required data of any type (string, int, number, object etc.)

2. split() :

The split() method splits a string into a list.

You can specify the separator, default separator is any whitespace.

```
string.split(separator, maxsplit)
```

- separator is used to separate the words derived from splitting and it is optional,
- maxsplit is also the optional argument which specifies how many splits to do. It's default value is –1.

3. upper() :

The upper() method returns a string where all characters are in upper case.

Symbols and Numbers are ignored.

```
string.upper()
```

4. readlines() :

The readlines() method returns a list containing each line in the file as a list item.

Use the hint parameter to limit the number of lines returned. If the total number of bytes returned exceeds the specified number, no more lines are returned.

```
file.readlines(hint)
```

- hint is an optional argument. If the number of bytes returned exceed the hint number, no more lines will be returned. Default value is -1, which means all lines will be returned.

5.writelines() :

The writelines() method writes the items of a list to the file.

Where the texts will be inserted depends on the file mode and stream position.

"a": The texts will be inserted at the current file stream position, default at the end of the file.

"w": The file will be emptied before the texts will be inserted at the current file stream position, default 0.

```
file.writelines(list)
```

- list is the list of texts or byte objects that will be inserted.

3.6 OpenCV Library and Functions

We import OpenCV into our application by using the following statement.

```
import cv2
```

After executing the above statement, we can use all functions and methods belongs to OpenCV. For our project, we used some of the functions and methods of OpenCV for different type of implementations.

They are:

1. imread()
2. imshow()
3. cvtColor()
4. VideoCapture()
5. VideoCapture.read()
6. resize()
7. Rectangle()
8. putText()
9. waitKey()

1.imread() :

cv2.imread() method loads an image from the specified file. If the image cannot be read (because of missing file, improper permissions, unsupported or invalid format) then this method returns an empty matrix.

```
cv2.imread(path, flag)
```

- path is a string representing the path of the image to be read,
- flag specifies the way in which image should be read. It's default value is cv2.IMREAD_COLOR. We use cv2.COLOR_BGR2RGB which converts the image in BGR format into RGB format.
- This method returns an image that is loaded from the specified value.

2. imshow() :

cv2.imshow() method is used to display an image in a window. The window automatically fits to the image size.

```
cv2.imshow(window_name, image)
```

- window_name is a string representing the name of the window in which image to be displayed,
- image is the image that is to be displayed.
- It doesn't return anything.

3. cvtColor() :

cv2.cvtColor() method is used to convert an image from one color space to another. There are more than 150 color-space conversion methods available in OpenCV.

```
cv2.cvtColor(src, code[, dst[, dstCn]])
```

- src is the image whose color space is to be changed,
- code is the color space conversion code,
- dst is the output image of the same size and depth as src image. It is an optional parameter,
- dstCn is the number of channels in the destination image. If the parameter is 0 then the number of the channels is derived automatically from src and code. It is an optional parameter,

- It returns an image.

4. Video Capture() :

With OpenCV, we can capture a video from the camera. It lets us create a video capture object which is helpful to capture videos through webcam and then we may perform desired operations on that video.

Steps to capture a video:

- Use cv2.VideoCapture() to get a video capture object for the camera.
- Set up an infinite while loop and use the read() method to read the frames using the above created object.
- Use cv2.imshow() method to show the frames in the video.
- Breaks the loop when the user clicks a specific key.

Simple implementation is given below.

```
# import the opencv library

import cv2

# define a video capture object

vid = cv2.VideoCapture(0)

while(True):

    # Capture the video frame

    # by frame

    ret, frame = vid.read()

    # Display the resulting frame

    cv2.imshow('frame', frame)
    # the 'q' button is set as the
    # quitting button you may use any
    # desired button of your choice
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
# After the loop release the cap object
vid.release()
# Destroy all the windows
cv2.destroyAllWindows()
```

5.VideoCapture.read() :

If we observe above example table, we find vid.read() in which this method is reading the images from the live video capture.

We can use these derived images for finding faces, facial measurements and etc.

6.resize() :

Resizing an image means changing the dimensions of it, be it width alone, height alone or changing both of them. Also, the aspect ratio of the original image could be preserved in the resized image. To resize an image, OpenCV provides cv2.resize() function.

```
cv2.resize(src, dsize[, dst[, fx[, fy]]])
```

- src is source or input image,
- dsize is desired size for output image,
- fx is scale factor along horizontal axis,
- fy is scale factor along the vertical axis.

7.rectangle() :

cv2.rectangle() method is used to draw a rectangle on any image.

We can use this for drawing a rectangle on the face which is detected.

```
cv2.rectangle(image, start_point, end_point, color, thickness)
```

- image is the image on which rectangle is to be drawn,
- start_point is the starting coordinates of rectangle,
- end_point is the ending coordinates of rectangle,
- color is the color of border line of rectangle to be drawn,
- thickness is the thickness of the rectangle border line in px.
- It returns an image.

8. putText() :

cv2.putText() method is used to draw a text string on any image.

```
cv2.putText(image, text, org, font, fontScale, color[, thickness[, lineType[, bottomLeftOrigin]]])
```

- image is the image on which text is to be drawn,
- text is the Text string to be drawn,
- org is the coordinates of the bottom-left corner of the text string in the image,
- font denotes the font type,
- fontScale denotes the font scale factor that is multiplied by the font-specific base size,
- color is the color of text string to be drawn,
- thickness is the thickness of the line in px,
- lineType is an optional parameter.It gives the type of the line to be used,
- bottomLeftOrigin is an optional parameter. When it is true, the image data origin is at the bottom-left corner. Otherwise, it is at the top-left corner.
- It returns an image.

9.waitKey() :

This function is very important, without this function cv2.imshow() won't work properly.

```
cv2.waitkey(wait_time_in_milliseconds)
```

- wait_time_in_milliseconds is the time in milliseconds form I.e., if we enter the wait time as 6000 then the picture will be dispalyed fo 6s and then get closed. If we use 0 as the value then the image will be displayed for an infinite time until we stop the process.

3.7 Face_recognition tool and its methods

It is the world's simplest face recognition library which helps in recognizing and manipulating faces from Python or from the command line.

This model is built using deep learning with an accuracy of 99.38% on the labelled faces in the wild benchmark.

This model also provides a simple face_recognition command line tool that lets us do face recognition on a folder of images from the command line.

Features :

It can find faces that appear in a picture. For this purpose, we can use a method called face_locations which takes an image as input.

```
face_recognition.face_locations(img, number_of_times_to_upsample=1, model='hog')
```

- Where img is an image (as a numpy array),
- Number_of_times_to_upsample is to show how many times to upsample the image looking for faces. Higher numbers find smaller faces,
- Model tells which face detection model to use. Default model is hog.
- It returns a list of tuples of found face locations in css (top,right,bottom,left) order.

It finds and manipulates facial features in pictures. We can get the locations of each person's eyes, nose, mouth and chin.

We use face_landmarks for getting the facial locations of a face in the image.

```
face_recognition.face_landmarks(face_image, face_locations=None, model='large')
```

- Where face_image is an image to search,
- Face_locations is providing a list of face locations to check. This is optional,
- Model is optional which tells which model to use. The default model is hog.
- It returns a list of dicts of face feature locations (eyes, nose, etc).

We can recognize the faces in pictures.

For that, we can use face_encodings method which belongs to face_recognition.

```
face_recognition.face_encodings(face_image, known_face_locations=None, num_jitters=1,
model='small')
```

- Where face_image is the image that contains one or more faces,
- known_face_locations is the bounding boxes of each face if we already know them. This is optional,
- num_jitters tell us how many times to re-sample the face when calculating encoding (i.e., 100 is 100x slower),
- model is optional which tells us which model to use. 'large' or 'small' which only returns 5 points but is faster.
- It returns a list of 128-dimensional face encodings for each face detected.

We can get the Euclidean distance for each comparison face. For that, we use face_distance method.

```
face_recognition.face_distance(face_encodings, face_to_compare)
```

- Where face_encodings is a list of face encodings to compare,
- Face_to_compare is a face encoding to compare against.

- It returns a numpy ndarray with the distance for each face in the same order as the 'faces' array.

Compare a list of face encodings against a candidate encoding to see if they match.

We can use compare_faces method to compare two faces with a list of face encodings. If known face encodings match with face encodings of other face, it returns True or else returns false.

```
face_recognition.compare_faces(known_face_encodings,face_encoding_to_check,tolerance=0.6)
```

- Where known_face_encodings is a list of face encodings,
- face_encoding_to_check is a single face encoding to compare against the list,
- tolerance tells us how much distance between faces to consider it a match. Lower is stricter. 0.6 is typical best performance.
- It returns a list of True/False values indicating which known_face_encodings match with the face_encoding to_check.

3.8 Numpy module

NumPy is a Python library used for working with arrays.

It also has functions for working in domain of linear algebra, fourier transform, and matrices.

NumPy stands for Numerical Python.

The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy.

Arrays are very frequently used in data science, where speed and resources are very important.

In our project, we used numpy argmin() to store the minimum face distance of a known face and an unknown detected face.

```
numpy.argmin(array, axis = None, out = None)
```

- Array is an input array to work on,

- Axis is an int parameter which is used to look at the rows and columns individually. This is optional,
- Out provides a feature to insert output to the out array and it should be of appropriate shape and data type. This is an optional argument.
- It returns the indices of the minimum values along an axis. In case of multiple occurrences of the minimum values, the indices corresponding to the first occurrence are returned.

3.9 OS Module

OS Module :

The OS module in Python provides functions for interacting with the operating system. OS comes under Python's standard utility modules. This module provides a portable way of using operating system dependent functionality. The os and os.path modules include many functions to interact with the file system.

os.listdir() method in python is used to get the list of all files and directories in the specified directory. If we don't specify any directory, then list of files and directories in the current working directory will be returned.

In our project, we are using this method for getting all the images from a specified path and to store them into a list of images.

```
os.listdir(path)
```

- path is an optional argument which defines the path of the directory.
- It returns the list of all files and directories in the specified path. The return type of this method is list.

3.10 DateTime Module

Datetime module :

In Python, date and time are not a data type of its own, but a module named datetime can be imported to work with the date as well as time. Datetime module comes built into Python, so there is no need to install it externally.
Datetime module supplies classes to work with date and time. These classes provide a number of functions to deal with dates, times and time intervals. Date and datetime are an object in Python, so when you manipulate them, you are actually manipulating objects and not string or timestamps.

datetime class is a combination of date and time along with the attributes year, month, day, hour, minute, second, microsecond, and tzinfo.

We use the below methods now() for current local time and today() for current date at which the student entered the college premises.

datetime.today() : This returns the current local datetime, with tzinfo None.

```
datetime.today()
```

datetime.now() : This returns the current local date and time.

```
datetime.now()
```

3.11 Sessions

There are four types of sessions that we can use for user authentication.

Database-backed sessions - This is a type where the session details are stored in database.
File-based sessions - This is a type where the session details are stored in file.
Cached sessions - This is a type where the session details are stored in cache.
Cookie-based sessions - This is a type where the session details are stored in cookies.

For our project, we used cookie-based session for authentication. Using this, we let the user to stay logged in until he logs out.

3.12 FullCalendar API

We have used FullCalendar API to show the calendar in attendance page.
An application programming interface is a connection between computers or between computer programs. It is a type of software interface, offering a service to other pieces of software. A document or standard that describes how to build such a connection or interface is called an API specification.

We used FullCalendar API in our project, to show the full month calendar with present, absent, week holiday and any public holidays are given.

The output for the same can be seen in Fig. 5.5, 5.7, 5.10, 5.13 and 5.15.

3.13 Select2

While retrieving the student and staff details to view their attendance we have used Select2.
Select2 gives you a customizable select box with support for searching, tagging, remote data sets, infinite scrolling, and many other highly used options.

We can see in the figures 5.8, 5.9 and 5.14.

3.14 AJAX

The attendance in the calendar changes on changing the month. This has obtained by AJAX in javascript.
The same can be seen in figures 5.5, 5.7, 5.10, 5.13 and 5.15.

3.15 Views

A view is nothing more than a SQL statement that is stored in the database with an associated name. A view is actually a composition of a table in the form of a predefined SQL query.

A view can contain all rows of a table or select rows from a table. A view can be created from one or many tables which depends on the written SQL query to create a view.

3.16 Stored Procedures

A stored procedure is a set of Structured Query Language (SQL) statements with an assigned name, which are stored in a relational database management system (RDBMS) as a group, so it can be reused and shared by multiple programs.

3.17 HTML

HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets and scripting languages such as JavaScript.
For our project, we used HTML to create our HTML pages in the front end.

3.18 CSS

Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.
We used CSS to present our webpages elegantly.

3.19 JavaScript

JavaScript, often abbreviated as JS, is a programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time compiled, and multi-paradigm. It has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions.

For our project, we use Json Responses. JsonResponse is an HTTP Response subclass that helps to create a JSON-encoded response. Its default Content-Type header is set to application/json.

If the safe parameter is set to False, any object can be passed for serialization; otherwise only dict instances are allowed.

CHAPTER 4

METHODOLOGY

4.1 Methodology Flow

The approach performs face - recognition based student attendance system. The methodology flow begins with the capture of image i.e., by performing enrollment using simple and handy interface, followed by pre-processing of the captured facial images. All the data regarding every person who got enrolled during enrollment process is stored in our database.

The proposed attendance system chiefly consists of 3 phases. They are: Image acquisition, Face Detection, Face Recognition.

The operating of the system is represented as follows:

1. Image Acquisition - The system consists of a camera that captures the pictures of the surroundings of the college and sends it to the image pre-processing. Then that image will be sent for face detection.

2. Face Detection - This method separates the facial space from the remainder of the background image. The faces area unit will be kept within the information and is detected when that face extraction is additionally administered.

3. Face Recognition - The face image is then compared with the stored image in the database. If the face image is matched with the stored image, then the face is recognized. Then for that individual the attendance is recorded.

The block diagram and the flow chart for the proposed system are provided below. Figure 3.1 shows the block diagram of our proposed system and figure 3.2 shows the flow chart of the proposed system.

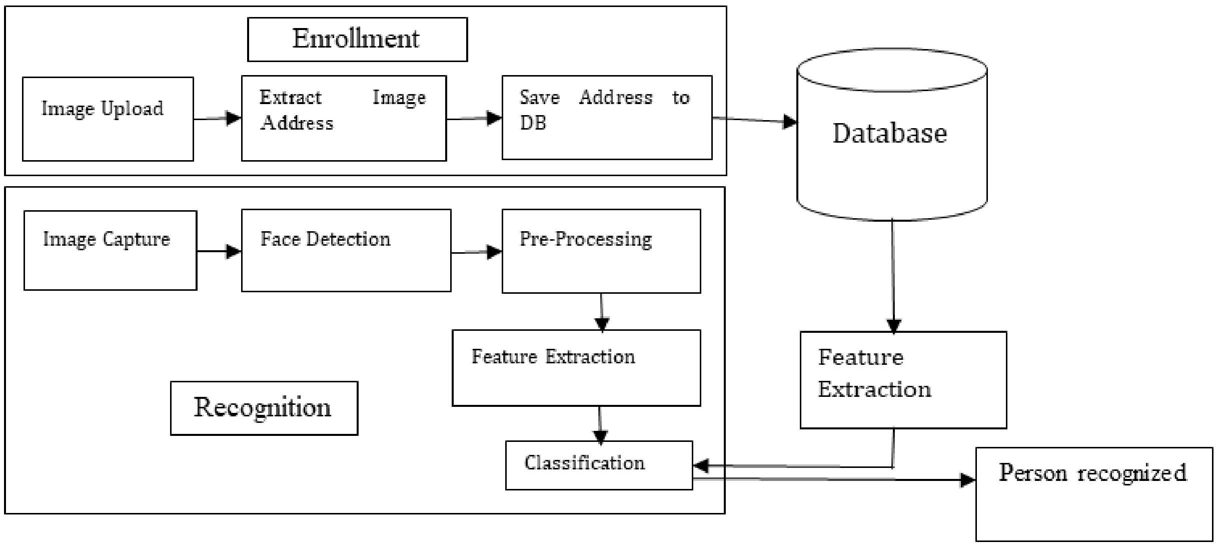


Fig 3.1 Block diagram of the proposed system

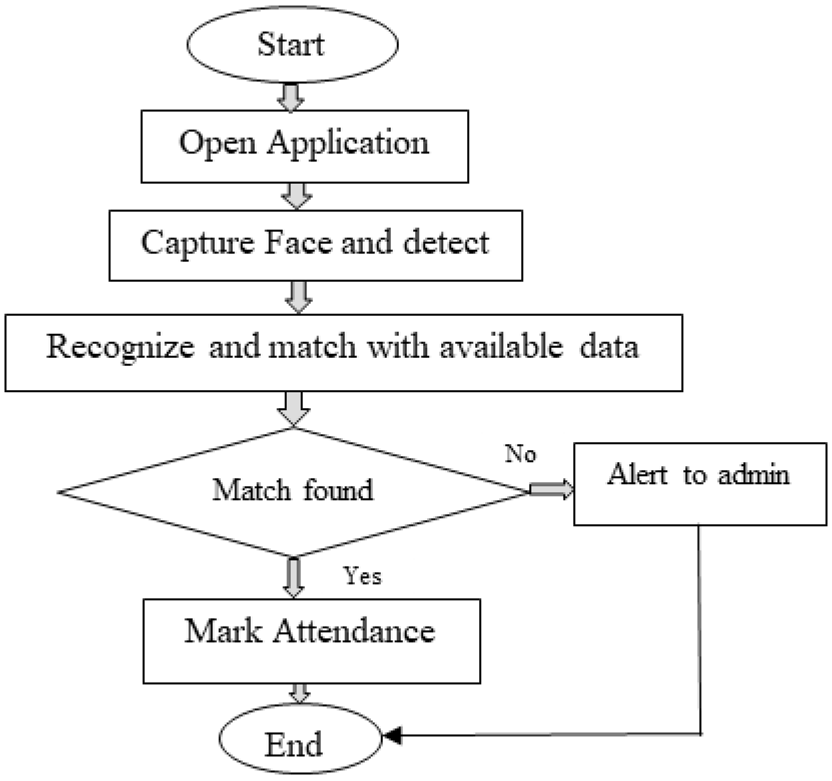


Fig 3.2 Flow chart of the proposed system

4.2 Enrollment

We provide the students and lecturers with a user registration form to enroll themselves. Hence, we can have enough data required to detect and recognize the faces. This enrollment is a one-time process. All the data given during enrollment process will be stored in database. When the application started then the face recognition model will take the facial measurements of the recognized person which are about to 128. These measurements will be stored in the array. These measurements will be compared with the facial measurements of the images in the database. If the face distance of these two measurements is near to 0 then the person is recognized by the model. Else the model will check for the remaining images in the database. Then the attendance gets recorded for the respective person. If the face is detected, but not recognized in our database, we will provide the admin with an alert that an unauthorized person entered the college premises.

4.3 Databases

Our project is about marking attendance of students and staff so we need to store all the required information in order to mark the attendance every day. For that, we created and used several tables. These tables include :

1. student – stores id, rollnumber, academicYear, branch_id and person_id of the student. id is the primary key.

Column Name	Data Type
id	INT
rollnumber	VARCHAR(10)
academicYear	VARCHAR(10)
branch_id	INT
person_id	INT

Table 3.1Student Table

2. staff – stores id, designation_id and person_id of the staff. id is the primary key.

Column Name	Data Type
id	INT
designation_id	INT
person_id	INT

Table 3.2 Staff Table

3. studentattendance – id, isPresent, date, student_id and stdattdate of the student where isPresent is a flag to denote whether the student is present or not and stdattdate is the date on which the student was attended. id is the primary key.

Column Name	Data Type
id	INT
isPresent	TINYINT(1)
date	DATETIME(6)
student_id	INT
stdattdate	DATE

Table 3.3 Studentattendance Table

4. staffattendance – stores id, isPresent, date, staff_id and stfattdate of the staff where stfattdate is the date on which the staff was attended. id is the primary key.

Column Name	Data Type
id	INT
isPresent	TINYINT(1)
date	DATETIME(6)
staff_id	INT
stfattdate	DATE

Table 3.4 staffattendance Table

5. auth_user – This is used for the authentication of the user. It stores the id, password, last_login, is_superuser, username, first_name, last_name, email, is_staff, is_active, date_joined. Here the user is either student or staff. id is the primary key.

Column Name		Data Type	
id		INT	
password		VARCHAR(128)	
last_login	date_joined	DATETIME(6)	DATETIME(6)
is_superuser		TINYINT(1)	
username		VARCHAR(150)	
first_name	last_name	VARCHAR(150)	VARCHAR(150)
email		VARCHAR(254)	
is_staff	is_active	TINYINT(1)	TINYINT(1)

Table 3.5 auth_user Table

6. branch – stores id and stream of the student. id is the primary key.

Column Name	Data Type
id	INT
stream	VARCHAR(5)

Table 3.6 branch Table

7. person – stores id, firstname, lastname, mobile, picture, address_id and user_id of the person. id is the primary key.

Column Name	Data Type
id	INT
firstname	VARCHAR(30)
lastname	VARCHAR(30)
mobile	VARCHAR(254)

picture	VARCHAR(100)
address_id	INT
user_id	INT

Table 3.7 person Table

8. designation – stores id and role of the staff. id is the primary key.

Column Name	Data Type
id	INT
role	VARCHAR(20)

Table 3.8 designation Table

9. address – stores the id, addressLine1, addressLine2, city, state and pincode of the person.

Column Name	Data Type
id	INT
addressLine1	VARCHAR(50)
addressLine2	VARCHAR(50)
city	VARCHAR(30)
state	VARCHAR(30)
pincode	INT

Table 3.9 address Table

10. calendardates – stores id, Date, Day for the calender api.

Column Name	Data Type
id	BIGINT
Date	DATE
Day	VARCHAR(45)

Table 3.10 calendardates Table

11. holidays – stores id, Date and Type where Type is the type of holiday which can be seen in calendar.

Column Name	Data Type
id	BIGINT
Date	DATE
Type	VARCHAR(45)

Table 3.11 holidays Table

CHAPTER 5

ER DIAGRAM

An object-relational mapper (ORM) is a code library that automates the transfer of data stored in relational database tables into objects that are more commonly used in application code.

ORMs provide a high-level abstraction upon a relational database that allows a developer to write Python code instead of SQL to create, read, update and delete data and schemas in their database. Developers can use the programming language they are comfortable with to work with a database instead of writing SQL statements or stored procedures.

We used this model in our project. The below model shows ER diagram of our project.

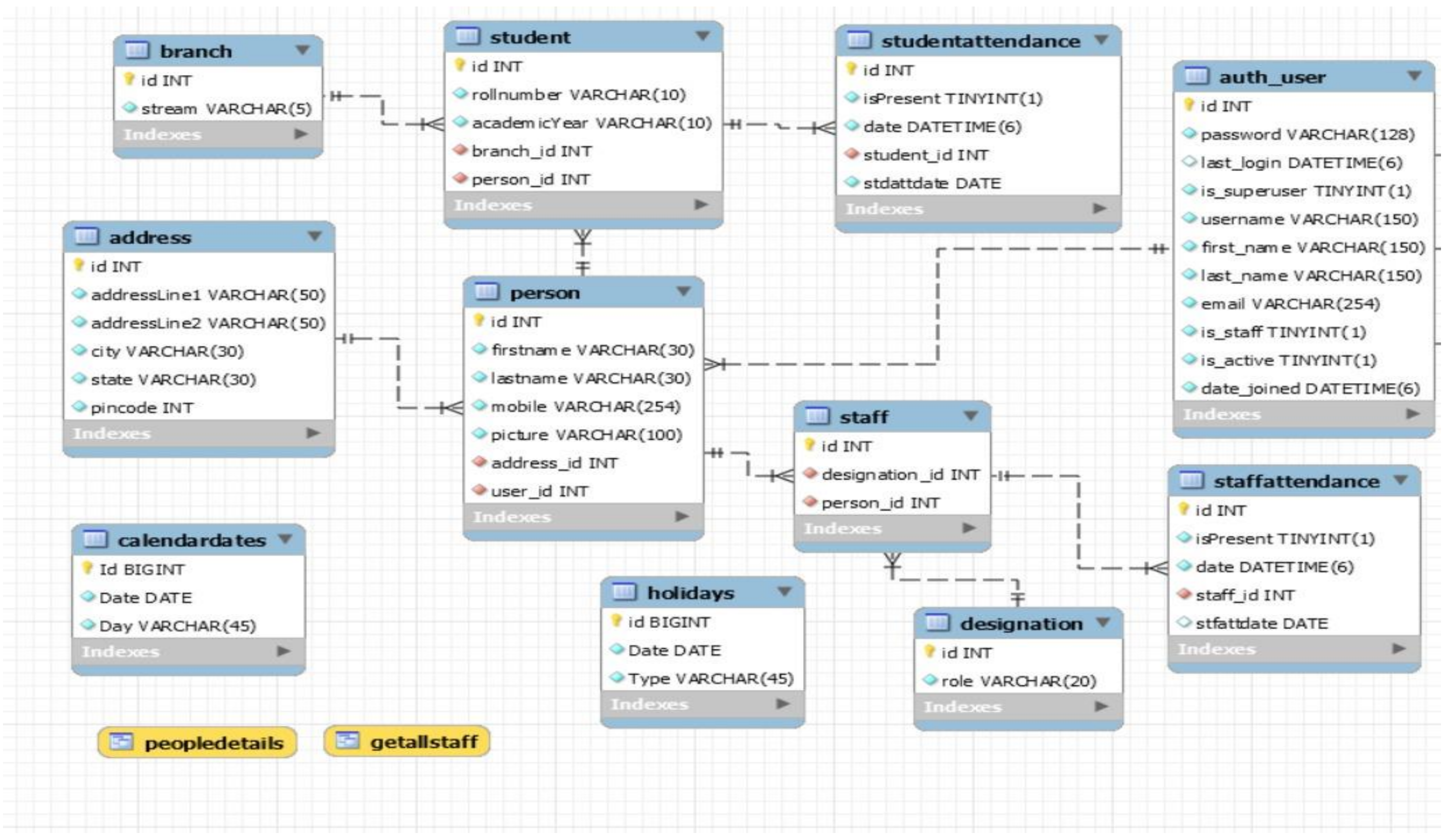


Fig 4.1 ER Diagram

CHAPTER 6

RESULTS & DISCUSSIONS

As we discussed in above sections, we enroll the students and the staff by providing a registration page.

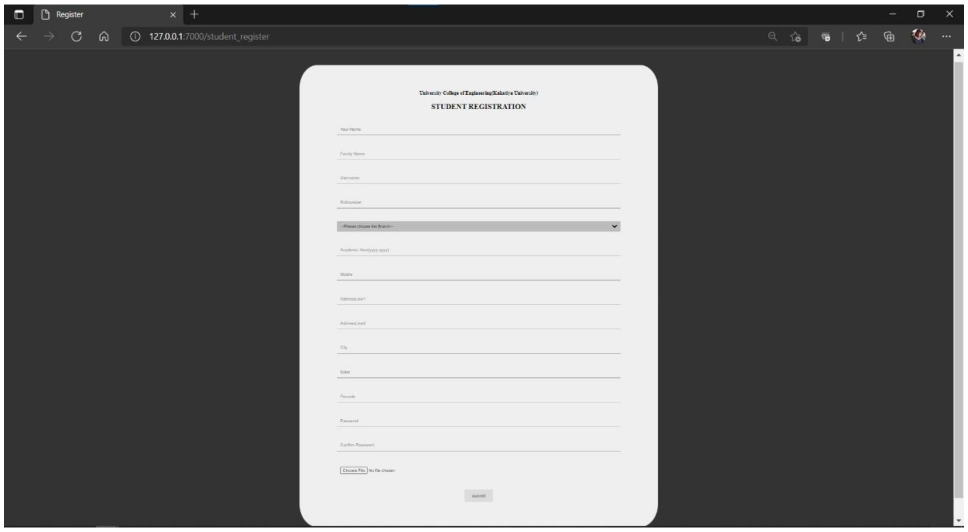


Fig 5.1 Student Registration Page

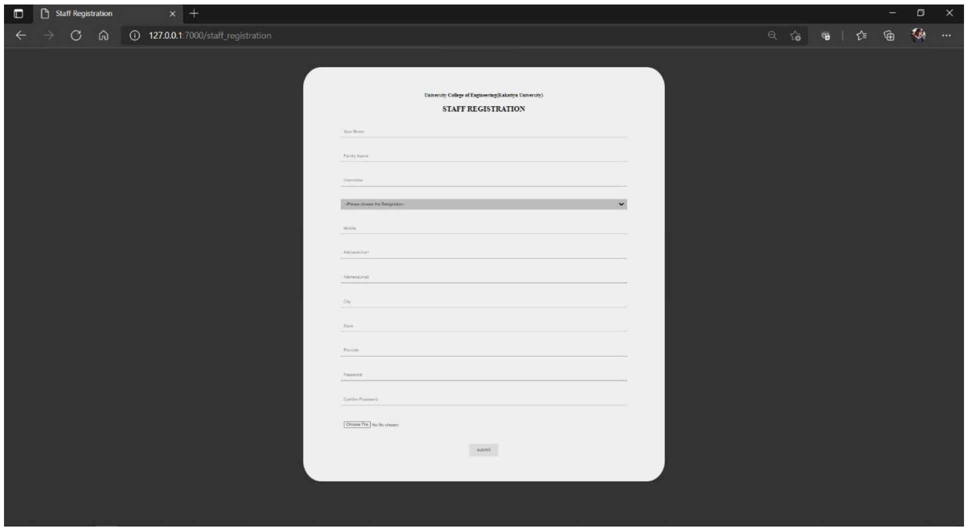

Staff Registration
127.0.0.1:7000/staff_registration
University College of Engineering(Kakatiya University)
STAFF REGISTRATION
Your Name
Family Name
Username
--Please choose the Designation--
Mobile
AddressLine1
AddressLine2
City
State
Pincode
Password
Confirm Password
Choose File No file chosen
submit

Fig 5.2 Staff Registration Page

After enrolling, the users (student, staff or admin) can login to view their details and attendance.

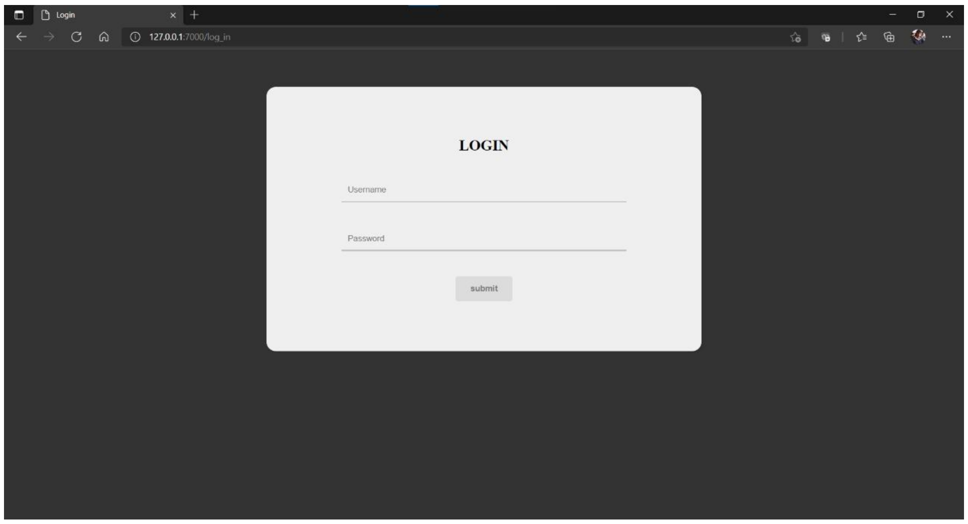
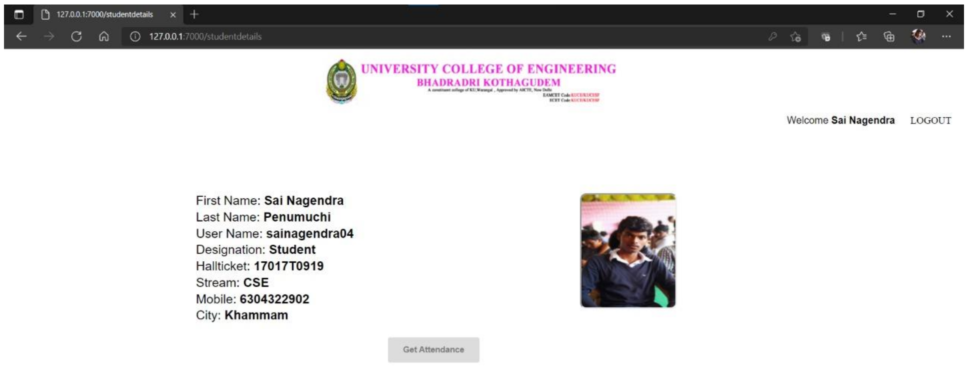


Fig 5.3 Login Page

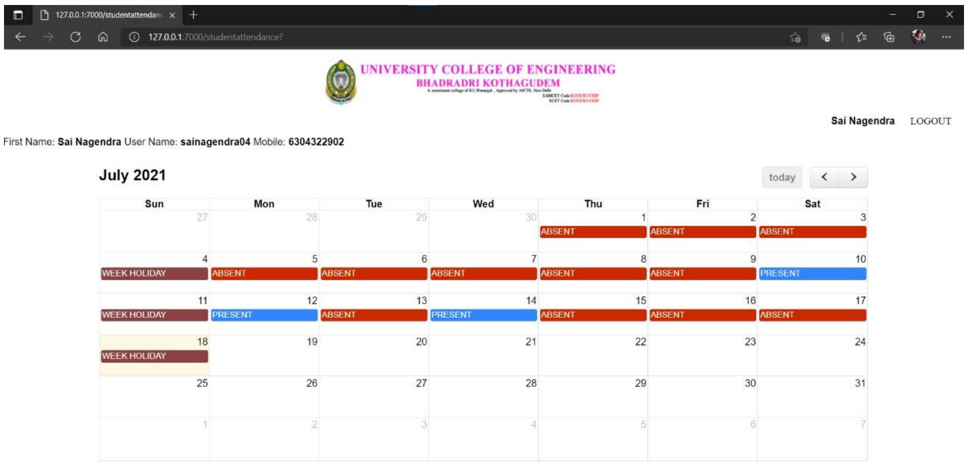
Upon login, the students will see their details page.


127.0.0.1:7000/studentdetails
UNIVERSITY COLLEGE OF ENGINEERING
BHADRADRI KOTHAGUDEM
Welcome Sai Nagendra
LOGOUT
Get Attendance

First Name: **Sai Nagendra**
Last Name: **Penumuchi**
User Name: **sainagendra04**
Designation: **Student**
Hallticket: **17017T0919**
Stream: **CSE**
Mobile: **6304322902**
City: **Khammam**

Fig 5.4 Student Details Page

On clicking 'Get Attendance' button, the students can see their attendance page.



127.0.0.1:7000/studentattendance?

UNIVERSITY COLLEGE OF ENGINEERING
BHADRADRI KOTHAGUDEM

Sai Nagendra LOGOUT

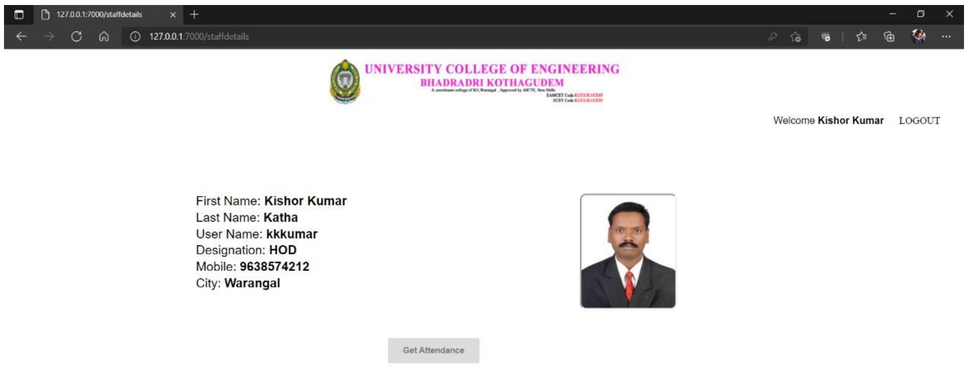
First Name: **Sai Nagendra** User Name: **sainagendra04** Mobile: **6304322902**

July 2021

Sun	Mon	Tue	Wed	Thu	Fri	Sat
27	28	29	30	1 ABSENT	2 ABSENT	3 ABSENT
4 WEEK HOLIDAY	5 ABSENT	6 ABSENT	7 ABSENT	8 ABSENT	9 ABSENT	10 PRESENT
11 WEEK HOLIDAY	12 PRESENT	13 ABSENT	14 PRESENT	15 ABSENT	16 ABSENT	17 ABSENT
18 WEEK HOLIDAY	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

Fig 5.5 Student Attendance Page

The process is the same for staff too. They can login and can see their details page.



127.0.0.1:7000/staffdetails

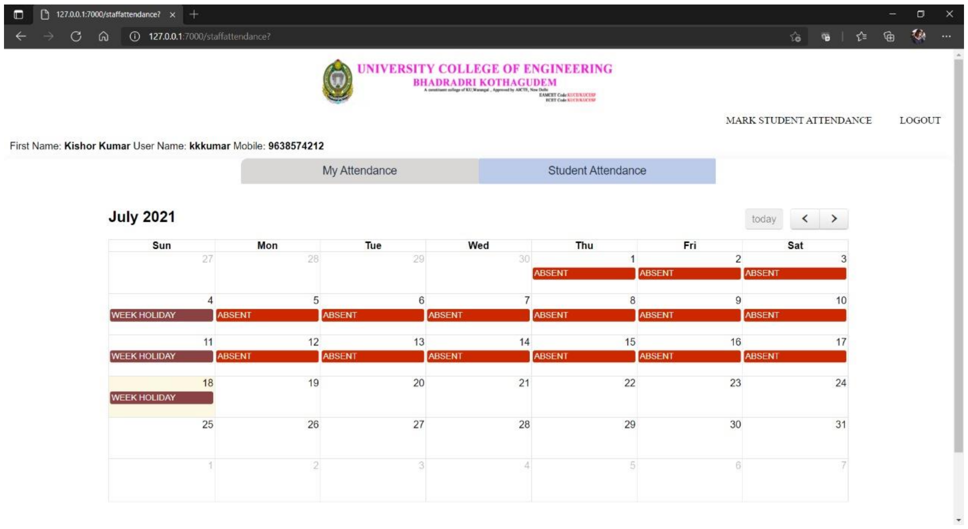
UNIVERSITY COLLEGE OF ENGINEERING
BHADRADRI KOTHAGUDEM

Welcome **Kishor Kumar** LOGOUT

First Name: **Kishor Kumar**
Last Name: **Katha**
User Name: **kkkumar**
Designation: **HOD**
Mobile: **9638574212**
City: **Warangal**

Fig 5.6 Staff Details Page

On clicking ‘Get Attendance’, the staff can see their attendance page.



127.0.0.1:7000/staffattendance?

UNIVERSITY COLLEGE OF ENGINEERING
BHADRADRI KOTHAGUDEM

MARK STUDENT ATTENDANCE LOGOUT

First Name: **Kishor Kumar** User Name: **kkkumar** Mobile: **9638574212**

My Attendance | Student Attendance

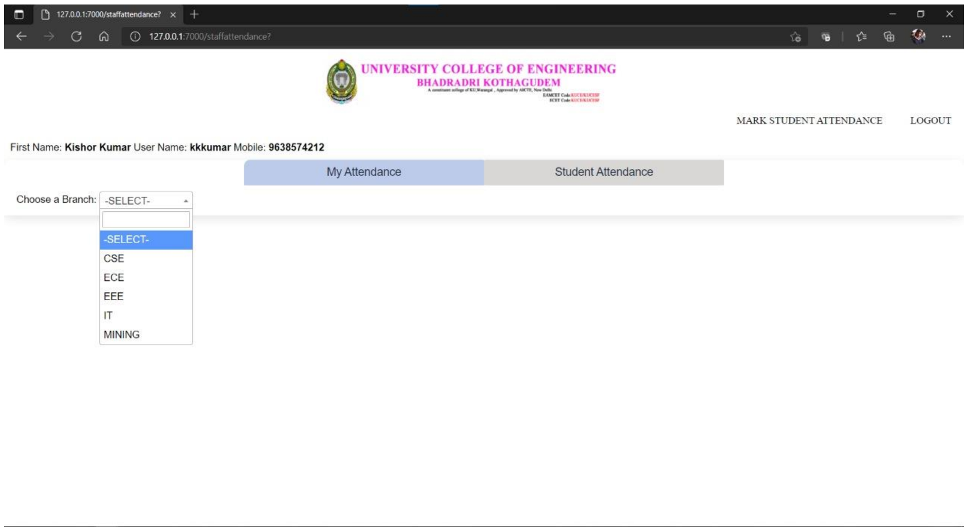
July 2021

today

Sun	Mon	Tue	Wed	Thu	Fri	Sat
27	28	29	30	1 ABSENT	2 ABSENT	3 ABSENT
4 WEEK HOLIDAY	5 ABSENT	6 ABSENT	7 ABSENT	8 ABSENT	9 ABSENT	10 ABSENT
11 WEEK HOLIDAY	12 ABSENT	13 ABSENT	14 ABSENT	15 ABSENT	16 ABSENT	17 ABSENT
18 WEEK HOLIDAY	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

Fig 5.7 Staff Attendance Page

Staff can view the attendance of the student by selecting branch and the student they want to see.



127.0.0.1:7000/staffattendance?

UNIVERSITY COLLEGE OF ENGINEERING
BHADRADRI KOTHAGUDEM

MARK STUDENT ATTENDANCE LOGOUT

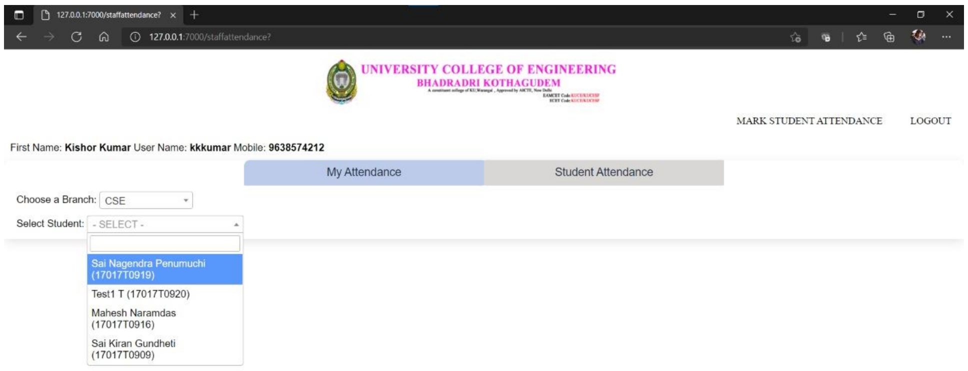
First Name: **Kishor Kumar** User Name: **kkkumar** Mobile: **9638574212**

My Attendance | Student Attendance

Choose a Branch: -SELECT-

- -SELECT-
- CSE
- ECE
- EEE
- IT
- MINING

Fig 5.8 Choosing branch of the required student from staff account



127.0.0.1:7000/staffattendance?

UNIVERSITY COLLEGE OF ENGINEERING
BHADRADRI KOTHAGUDEM

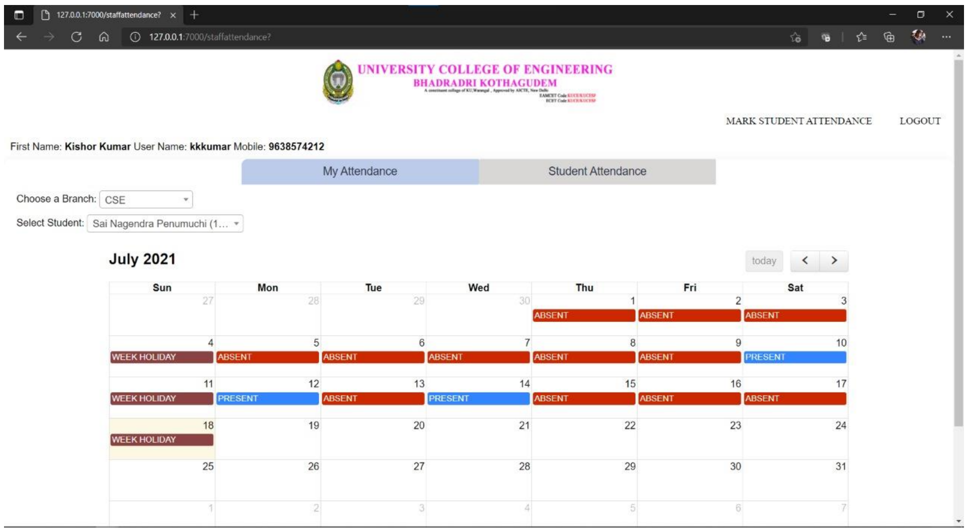
First Name: **Kishor Kumar** User Name: **kkkumar** Mobile: **9638574212**

Choose a Branch: CSE

Select Student: - SELECT -

- Sai Nagendra Penumuchi (17017T0919)
- Test1 T (17017T0920)
- Mahesh Naramdas (17017T0916)
- Sai Kiran Gundheti (17017T0909)

Fig 5.9 Selecting the student in the specified branch from staff account



127.0.0.1:7000/staffattendance?

UNIVERSITY COLLEGE OF ENGINEERING
BHADRADRI KOTHAGUDEM

First Name: **Kishor Kumar** User Name: **kkkumar** Mobile: **9638574212**

Choose a Branch: CSE

Select Student: Sai Nagendra Penumuchi (1…

July 2021

Sun	Mon	Tue	Wed	Thu	Fri	Sat
27	28	29	30	1 ABSENT	2 ABSENT	3 ABSENT
4 WEEK HOLIDAY	5 ABSENT	6 ABSENT	7 ABSENT	8 ABSENT	9 ABSENT	10 PRESENT
11 WEEK HOLIDAY	12 PRESENT	13 ABSENT	14 PRESENT	15 ABSENT	16 ABSENT	17 ABSENT
18 WEEK HOLIDAY	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

Fig 5.10 Viewing attendance of the required student from staff account

Staff can mark attendance to students for those who didn't get detected due to technical or any other issues. This is done by staff manually by giving Roll number.

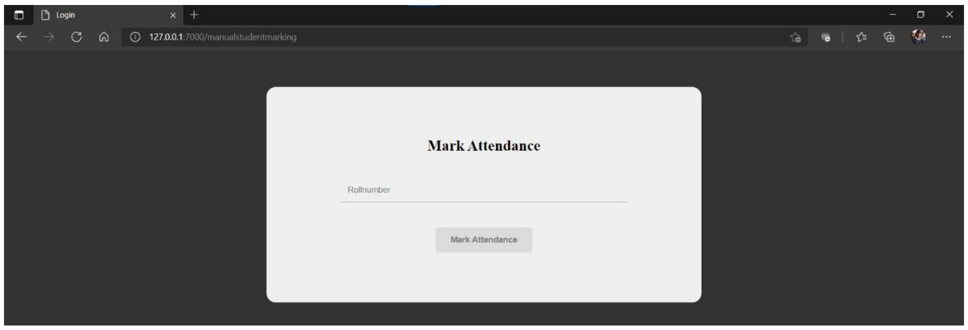
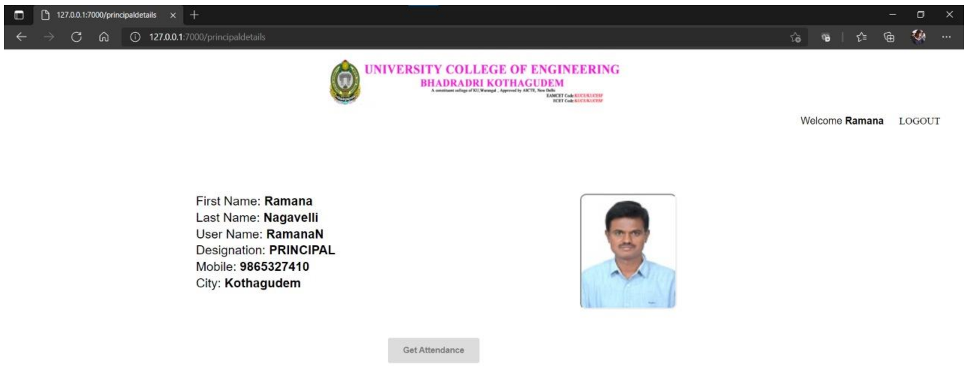


Fig 5.11 Marking Attendance to the student

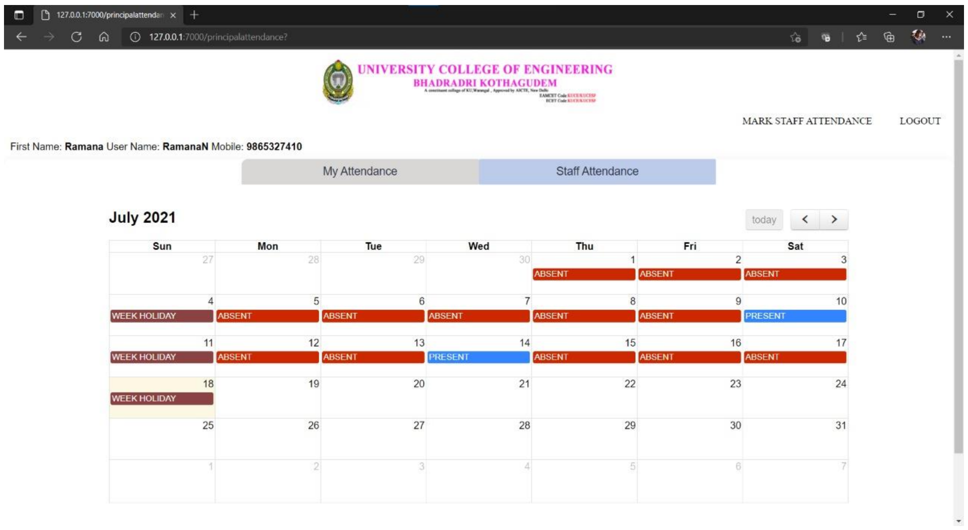
Upon successful login, the admin can see his details page.


127.0.0.1:7000/principaldetails
UNIVERSITY COLLEGE OF ENGINEERING
BHADRADRI KOTHAGUDEM
Welcome Ramana
LOGOUT
Get Attendance

First Name: **Ramana**
Last Name: **Nagavelli**
User Name: **RamanaN**
Designation: **PRINCIPAL**
Mobile: **9865327410**
City: **Kothagudem**

Fig 5.12 Admin Details Page

On clicking 'Get Attendance' page, admin can see his attendance.



127.0.0.1:7000/principalattendance?

UNIVERSITY COLLEGE OF ENGINEERING
BHADRADRI KOTHAGUDEM

MARK STAFF ATTENDANCE LOGOUT

First Name: **Ramana** User Name: **RamanaN** Mobile: **9865327410**

My Attendance | Staff Attendance

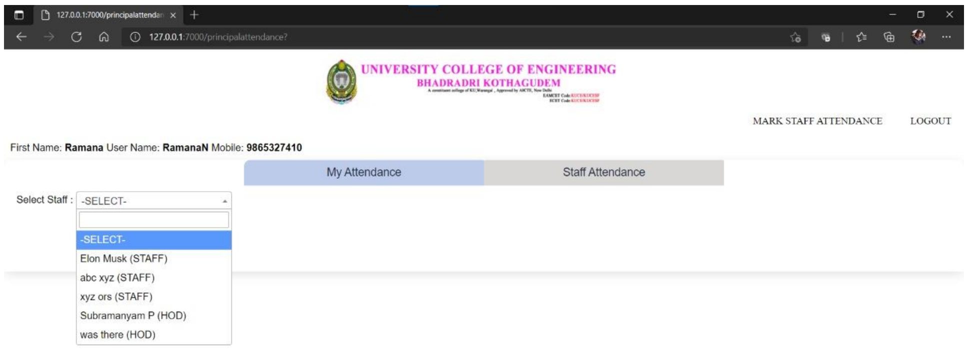
July 2021

today < >

Sun	Mon	Tue	Wed	Thu	Fri	Sat
27	28	29	30	1 ABSENT	2 ABSENT	3 ABSENT
4 WEEK HOLIDAY	5 ABSENT	6 ABSENT	7 ABSENT	8 ABSENT	9 ABSENT	10 PRESENT
11 WEEK HOLIDAY	12 ABSENT	13 ABSENT	14 PRESENT	15 ABSENT	16 ABSENT	17 ABSENT
18 WEEK HOLIDAY	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

Fig 5.13 Admin Attendance Page

Admin can see the staff attendance by selecting the particular staff he wants to see.



127.0.0.1:7000/principalattendance?

UNIVERSITY COLLEGE OF ENGINEERING
BHADRADRI KOTHAGUDEM

MARK STAFF ATTENDANCE LOGOUT

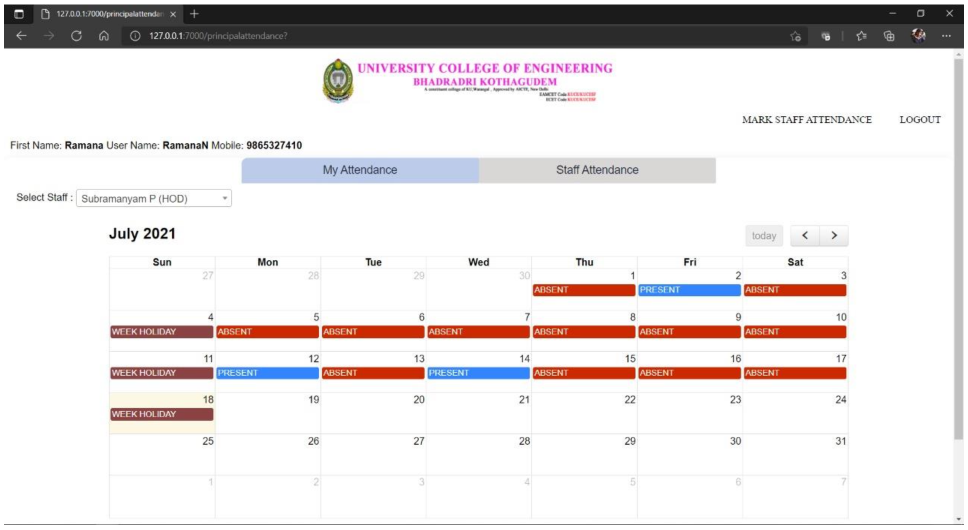
First Name: **Ramana** User Name: **RamanaN** Mobile: **9865327410**

My Attendance | Staff Attendance

Select Staff : -SELECT-

- -SELECT-
- Elon Musk (STAFF)
- abc xyz (STAFF)
- xyz ors (STAFF)
- Subramanyam P (HOD)
- was there (HOD)

Fig 5.14 Selecting the required staff to view their attendance



First Name: **Ramana** User Name: **RamanaN** Mobile: **9865327410**

Select Staff : Subramanyam P (HOD)

July 2021

Sun	Mon	Tue	Wed	Thu	Fri	Sat
27	28	29	30	1 ABSENT	2 PRESENT	3 ABSENT
4 WEEK HOLIDAY	5 ABSENT	6 ABSENT	7 ABSENT	8 ABSENT	9 ABSENT	10 ABSENT
11 WEEK HOLIDAY	12 PRESENT	13 ABSENT	14 PRESENT	15 ABSENT	16 ABSENT	17 ABSENT
18 WEEK HOLIDAY	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

Fig 5.15 Viewing the required staff attendance from Admin account

Admin can mark attendance for the staff who didn't get attendance due to technical or any other issues. Admin can mark the attendance for the staff manually by giving mobile number.

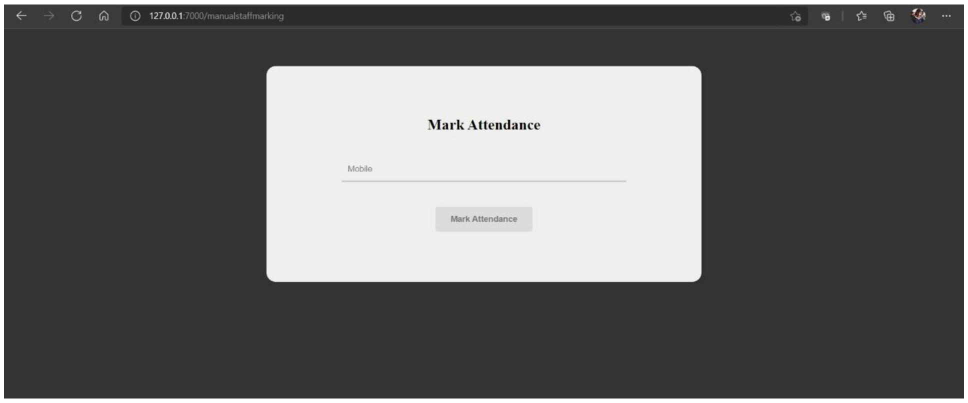


Fig 5.16 Marking attendance to the staff

Logout page will look like this.

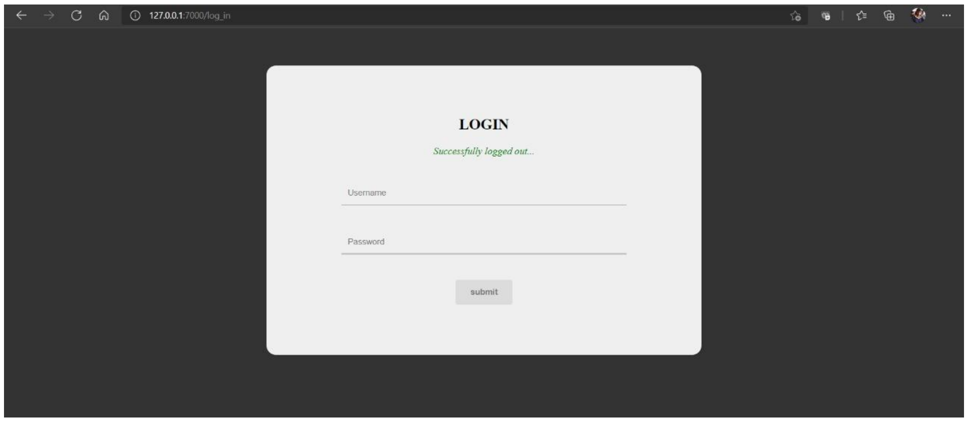


Fig 5.17 Logout Page

A person who enters the college premises will get detected by our external video capturing device like this. This is a sample picture. Accuracy and quality of the image is based on the camera device we are using.

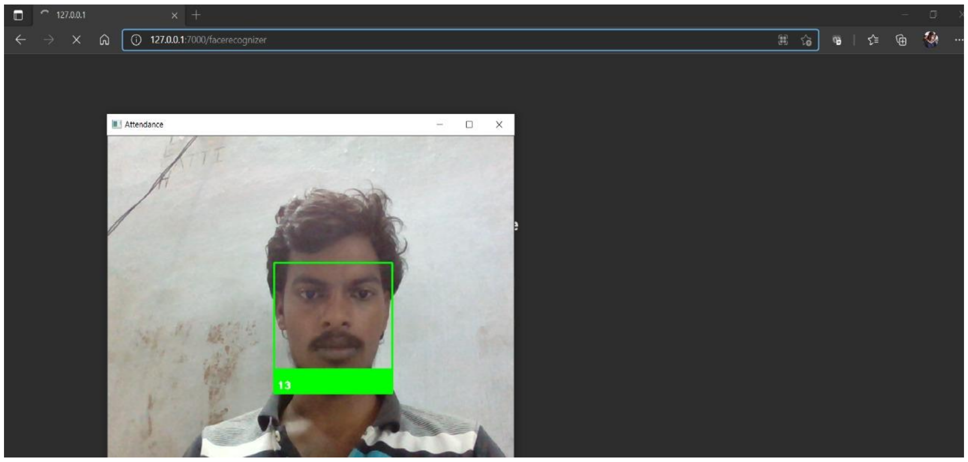


Fig 5.18 Face recognition of the people who entered the college premises

CHAPTER 7

CONCLUSION

In the system, we have implemented recording of attendance for both students and the college staff. It saves time and effort, especially if the attendance to be recorded is for a large number of people.

And also, we succeeded in identifying the strangers who entered the college premises without any authentication. The background details like the face of the stranger and also the date and time that he/she entered the college will be sent to admin.

In terms of implementation, where biometric system can verify and identify people reliably over time, a facial scan can be difficult, but not impossible. And is also very effective and faster than other existing systems.

Thus, this Attendance System helps in increasing the accuracy and speed ultimately achieving the high-precision real-time attendance to meet the need for automatic classroom evaluation.

7.1 Limitation of our Work

The working of this project would become a tedious task for the working on identical twins or we can say that the proposed idea will not work for two identical twins.

In the system, scanning of genuine person is done using camera, so sometimes it may take large amount of time for configuring the genuine identity due to the lack of server issue or the failure of the database.

Unable to recognize face with different angles, image quality, size and light intensity.

The feasibility of the model can be increased if a cloud can be hired to store details.

7.2 Future Enhancements

The system we have developed has successfully able to accomplish the task of marking attendance of the college students as well as staff automatically and the output will be obtained in a document as desired in real-time.

In further work, we intend to extend our application to the fact that it can take attendance of students with subject name in a classroom where it is conducted physical or virtual.

On the other hand, our system can be improved by integrating video-streaming service and lecture archiving system, to provide more profound applications in the field of Distance Education and Course Management System (CMS).

We can also use cloud integration for storing our databases if the application is used for extensively large number of people.

CHAPTER

REFERENCES

[1] GRD Journals- Global Research and Development Journal for Engineering | Volume 6 | Issue 4 | March 2021.

[2] International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-9, Issue-3, February 2020.

[3] International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 Volume: 07 Issue: 10 | Oct 2020.

[4] International Journal for Research in Applied Science & Engineering Technology (IJRASET) ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429 Volume 8 Issue VI June 2020.

[5] Quest Journals Journal of Software Engineering and Simulation Volume 5 ~ Issue 2 (2019) pp: 18-29 ISSN(Online) :2321-3795 ISSN (Print):2321-3809.

[6] International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-8, Issue-3S, February, 2019.

[7] Face Recognition with Python and OpenCV

[8] What is Facial Recognition? – Applications & How it Works

[9] Face Recognition with Python

[10] Real-Time Face Recognition: An End-To-End Project