Face Recognition Based Attendance System Report.

DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

Date: 19/03/2024

This is to certify that, the mini project work embodied in this report entitled, "**Face Recognition Based Attendance System**"

submitted by "**Jimit Andarpa"** Application id: 171504,

"**Priyanka Upadhyay"** Application id: ,

"**Sakshi Sen"** Application id: ,

"for the award of **VCET Engineering College** degree in the subject of **Information Technology**, is a work carried out by them under my guidance and supervision within the institute. The work described in this mini project report is carried out by the concerned students and has not been submitted for the award of any other degree of the University of Mumbai.

Further, it is certify that the students were regular during the academic year 2021-22 and have worked under the guidance of concerned faculty until the submission of this mini project work at **IDOL University OF Mumbai.**

Prof.
Mini Project Guide

Dr.
Head of Department

Dr.
Principal

ABSTRACT

We are living in a world where everything is automated and linked online. The internet of things, image processing, and machine learning are evolving day by day. Many systems have been completely changed due to this evolve to achieve more accurate results. The attendance system is a typical example of this transition, starting from the traditional signature on a paper sheet to face recognition. This Project proposes a method of developing a comprehensive embedded class attendance system using facial recognition with showing whether the face of the person is the student for that specified class or not. The system is based on the machine learning algorithm which is to be implemented on python language and using computer/laptop camera for the input image of the students or a normal outer camera can also be used which has to be connected to the system which is programmed to handle the face recognition by implementing the Local Binary Patterns algorithm LBPs.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

To maintain the attendance record with day-to-day activities is a challenging task. The conventional method of calling name of each student is time consuming and there is always a chance of proxy attendance. The following system is based on face recognition to maintain the attendance record of students. The daily attendance of students is recorded subject wise which is stored already by the administrator. As the time for corresponding subject arrives the system automatically starts taking snaps and then apply face detection and recognition technique to the given image and the recognize students are marked as present and their attendance update with corresponding time and subject id. We have used deep learning techniques to develop this system, histogram of oriented gradient method is used to detect faces in images and deep learning method is used to compute and compare feature facial of students to recognize them. Our system is capable to identify multiple faces in real time. 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. The test images and training images have to be captured by using the same device to ensure no quality difference. 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.

2. LITERATURE REVIEW

Study of the major research has been checked and studied thoroughly and got some key points to refer Camera-based object Detection, identification and distance estimation technical paper to build our project. So basically, we focus image identification part in this paper and studied the paper to deal with image identification.

2.1.1 Survey of existing system

As we analyzed these research papers and got the major idea that most of them used CNN as their technology. Some of them had limitation and some of them had proper approach. Authors in this proposed a method to automate the attendance system by integrating the face recognition technology using Eigen face database and Principal Component Analysis (PCA) algorithm with MATLAB GUI. The architecture of the system first, captures the student image, pre- process it, applied Eigenface generated database then test the captured face image with Eigenface image. When the similarity distance test scored more than the threshold value of 0.3 then the face was not recognized finally attendance marking, was stored in a Microsoft Excel sheet integrated with the MATLAB GUI. The original face database consists of images for 15 persons each has 10 images with different position and direction.

In this paper [3], the authors developed and implemented a classroom attendance system using radio frequency identification (RFID) and face verification techniques. The system recognizes students by using the RFID card and for more confirmation of the student's identity, face recognition technique has been added using Fast Adaptive Neural Network Classifier (FANNC). The classifier was trained and tested to identify human face images. Every student needs to take seven dissimilar head poses images in order for the classifier to identify students' images. The facial system tested on six distinct images of students and it achieved accuracy up to 98% for the front face.

[4] the authors focused on changing the traditional manual attendance to a digitized system using facial recognition. For face recognition module, the system used MATLAB software to implement the Principal Component Analysis (PCA) algorithm. The code was loaded on an embedded hardware system using Microcontroller PIC which also connected to a servo motor to open the door once the facial authentication was successful. From the experiment, the result was found that the system was so sensitive when there was a change in the background and with different head orientation.

2.1.2 Limitation of existing system or Research gap

It is developed for a particular system so we implement this library OpenCV does not provide the same ease of use when compared to MATLAB. OpenCV has a flann library of its own. This causes conflict issues when you try to use OpenCV library with the PCL library.

Sr. no.	Existing System	Features	Benefits	Limitations
1.	Automated attendance management system using face recognition.	Use Eigen faces for Recognition	High accuracy	Multiple faces were not recognized.
2.	Face recognition attendance system by nevon	Stores the faces that are detected and automatically marks attendance	Used for security purposes in organizations	Doesn't recognize properly in poor light.
3.	Smart Attendance Management System Using Face Recognition	Student Registration Face Recognition Addition of subject with their corresponding time. Attendance sheet gets generated and imported to Excel (xlsx) format.	In this the data is stored in sorted manner so that it can easily accessible	Requires high-definition camera
4.	Face Recognition - A Tool for Automated Attendance	Face detection, Pre-processing, Feature extraction, and Classification stages	High accuracy	Camera should be attached at a specific position
5.	Smart Application for AMS Using Face Recognition	Uses CCTV and Android mobile	3D face recognition algorithm is used	Android phone is expensive and detect one face at time
6.	Student Attendance System in Classroom Using Face Recognition Technique	Use of Discrete Wavelet Transform and Discrete Cosine Transform.	Multiple face detection was possible	Success rate is only 82%
7.	Attendance System based on Face Recognition using Eigen face and PCA Algorithms	In this Illumination invariant algorithm is used	The problem of light intensity problem and head pose was overcoming.	Masked faces were not recognized.
8.	Algorithm for Efficient Attendance Management: Face Recognition based approach	Median filter and skin classification is used	Multiple faces can be detected at a time and no special hardware is needed	Accuracy is low only 50% faces were recognized

Table 2.1 Literature Review

3. PROBLEM STATEMENT

To develop an automated attendance system using face recognition. Concept In a classroom with large number of students, it is a very tedious and time-consuming task to take the attendance manually. Therefore, we can implement an effective system which will mark the attendance of students automatically by recognizing their faces. The process of this face recognition system is divided into various steps, but the important steps are detection of face and recognition of face. Firstly, to mark the attendance of students, the image of students' faces will be required. This image can be snapped from the camera device, which will be placed in the classroom at a suitable location from where the whole classroom can be covered. This image will act as input to the system. For the effective face detection, the image needs to be enhanced by using some image processing techniques like grayscale conversion of image and histogram equalization. To identify the students sitting on the last rows neatly, the histogram equalization of image needs to be done. 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.

4. OBJECTIVES

To identify the student faces accurately. To mark the attendance automatically. To reduce the time and the efforts required for manual attendance to provide a valuable attentive system for both teacher and students. It provides flexibility and reduces the time loss. There will be no chance for a proxy.

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 detect the face segment from the video frame.
- 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.

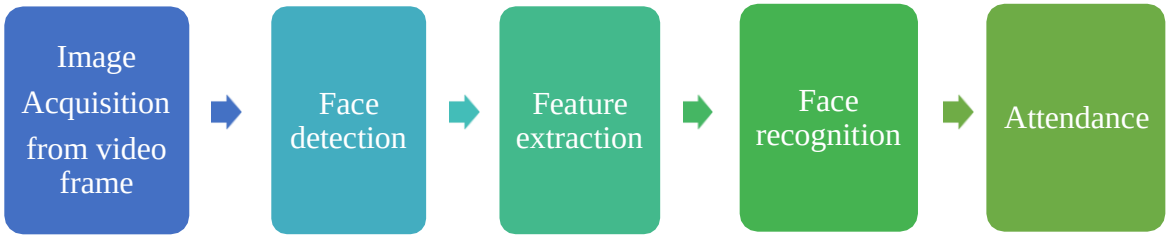


Fig 4.1 (Block Diagram of the General Framework)

5. PROPOSED SYSTEM

All the students of the class must register themselves by entering the required details and then their images will be captured and stored in the dataset. During each session, faces will be detected from live streaming video of classroom. The faces detected will be compared with images present in the dataset. If match found, attendance will be marked for the respective student. The task of the proposed system is to capture the face of each student and to store it in the database for their attendance. The face of the student needs to be captured in such a manner that all the feature of the students' face needs to be detected, even the seating and the posture of the student need to be recognized. There is no need for the teacher to manually take attendance in the class because the system records a video and through further processing steps the face is being recognized and the attendance database is updated.

5.1 Details of hardware and software

5.1.1 Hardware requirements

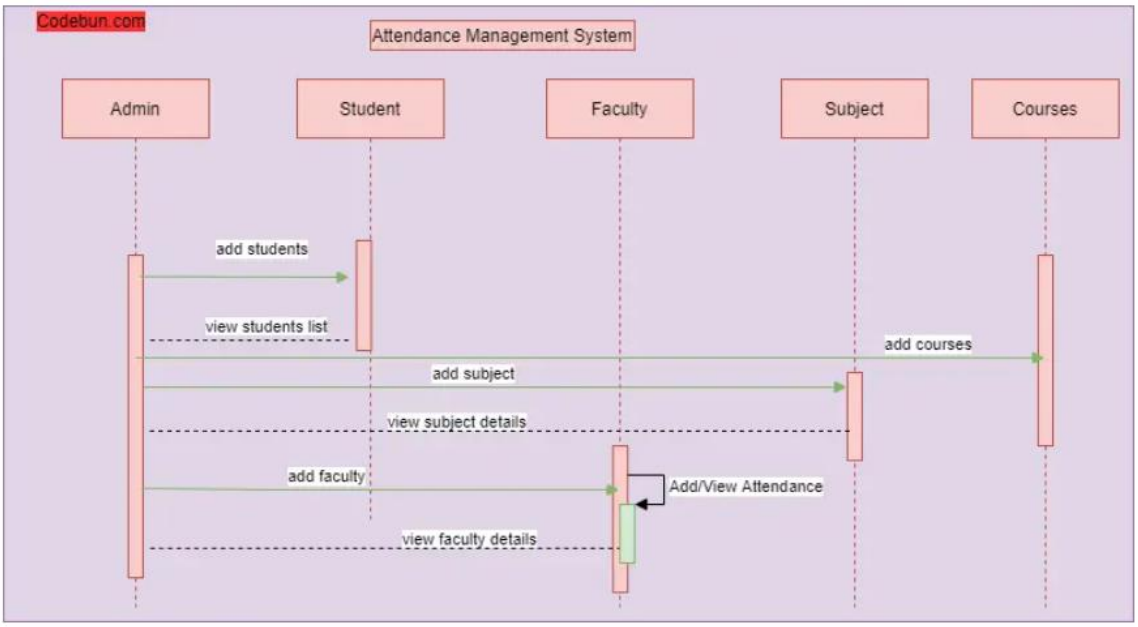
- Laptop with 8 GB RAM or above
- Camera 720p or above
- Nvidia GEFORCE RTX 1650 and above
- Windows 10 and above

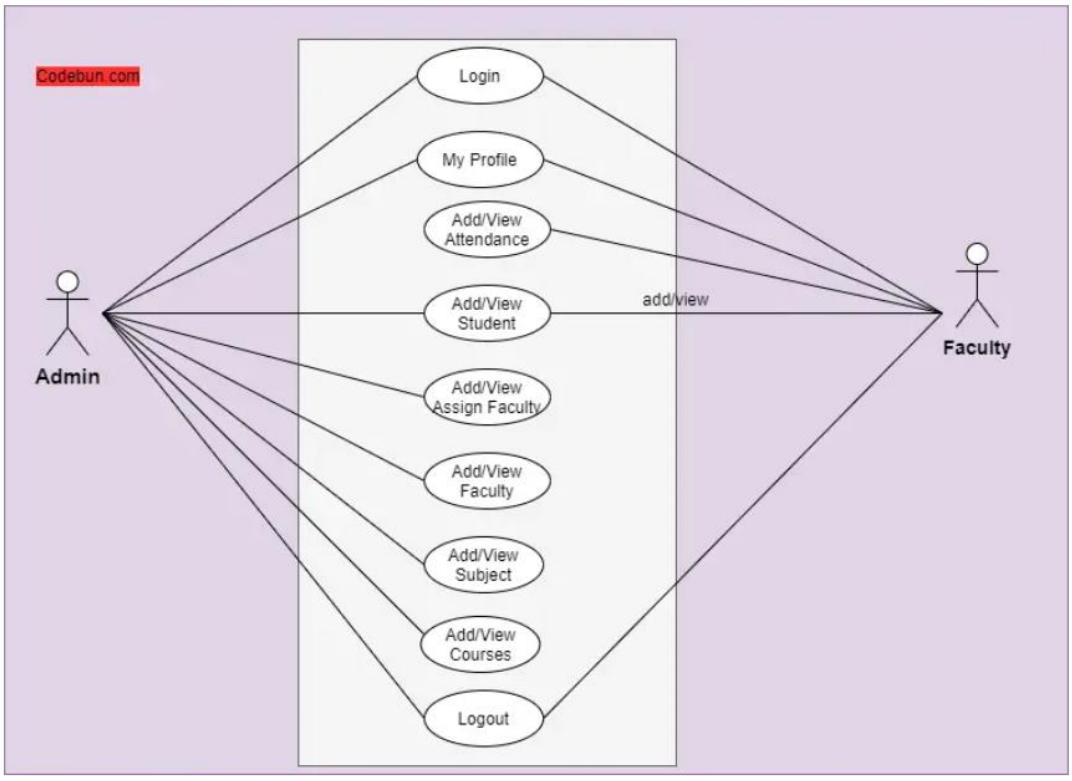
5.1.2 Software requirements

- Visual Studio Code
- Microsoft Office
- Tkinter
- Python Anaconda IDE

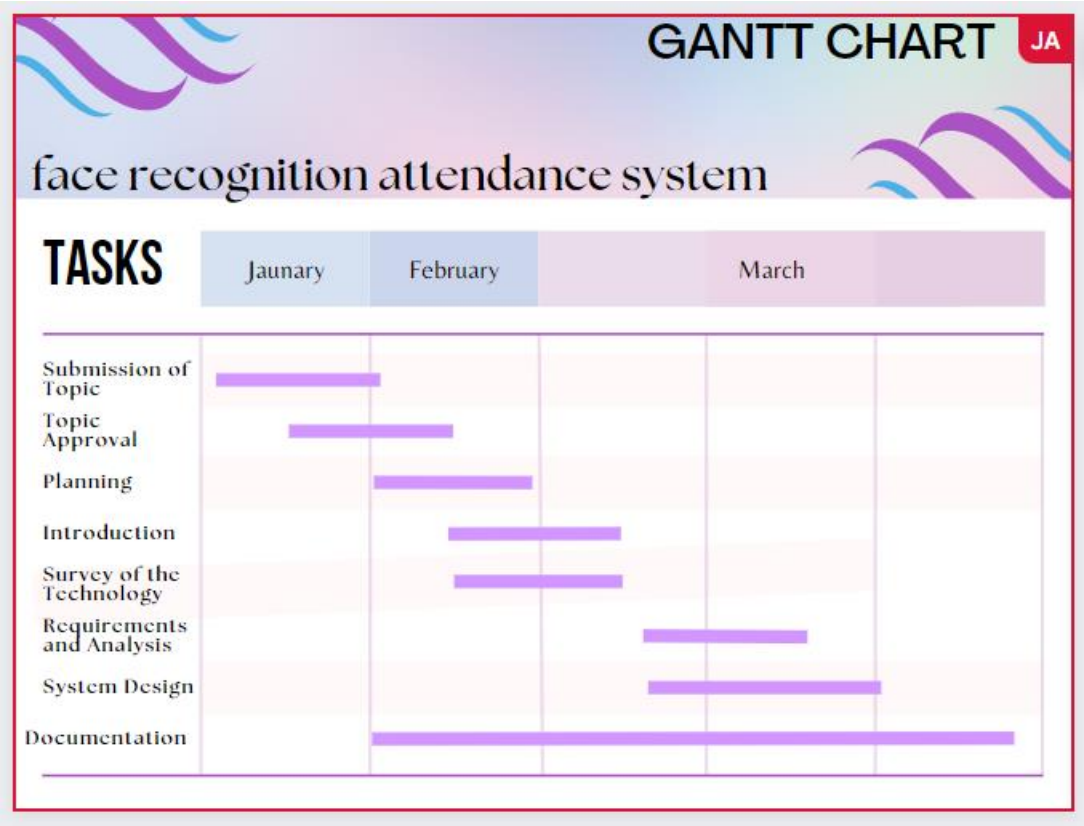
5.2 Design Details

5.2.1 Use Case Diagram





5.2.2 GANTT Chart



5.3 Design Details

5.3.1 System Flow / System Architecture

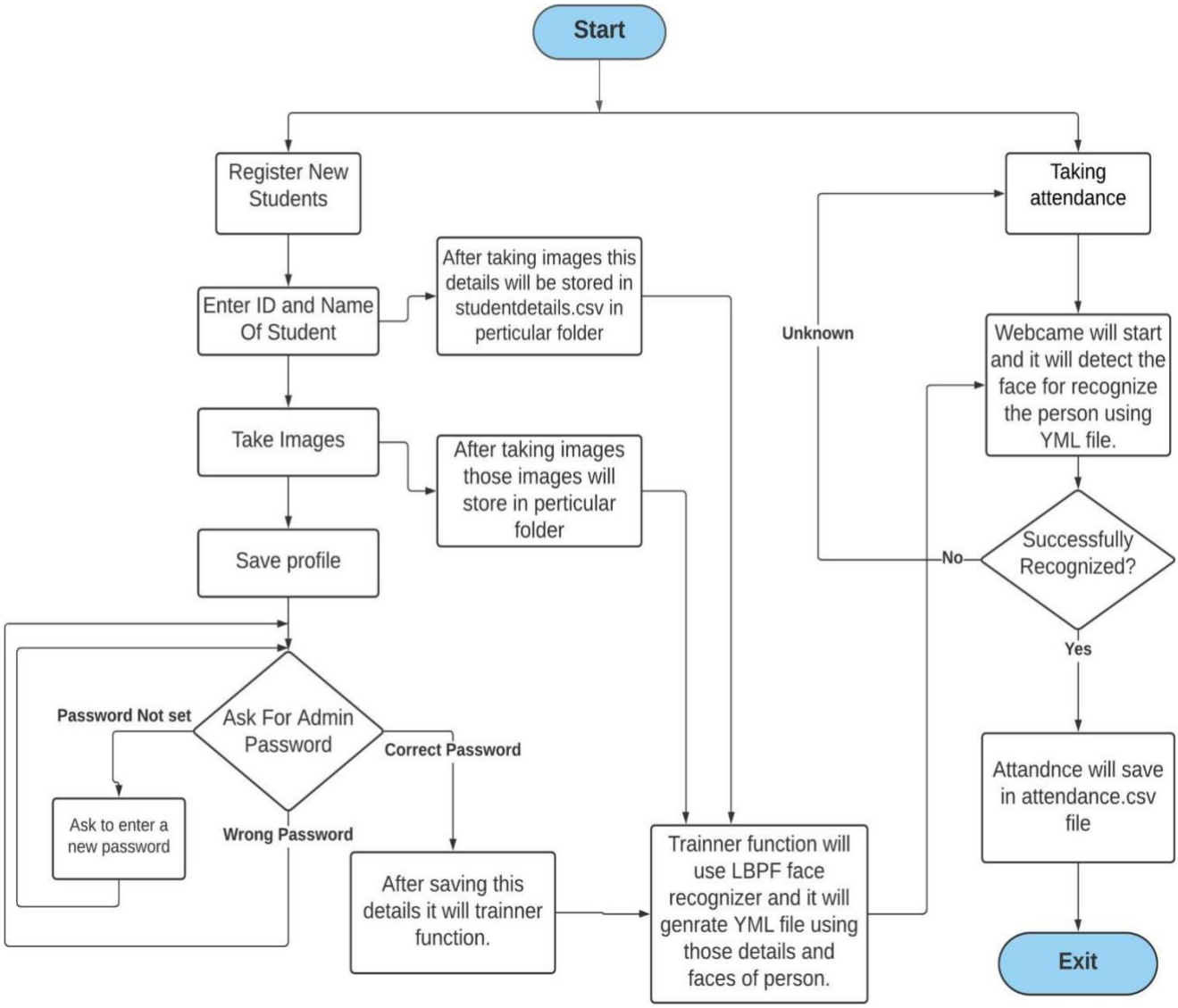
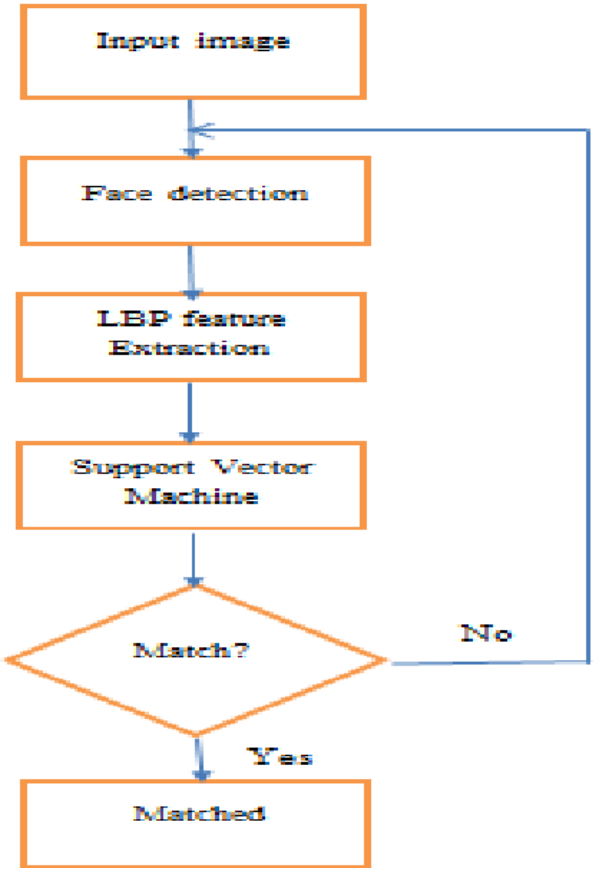
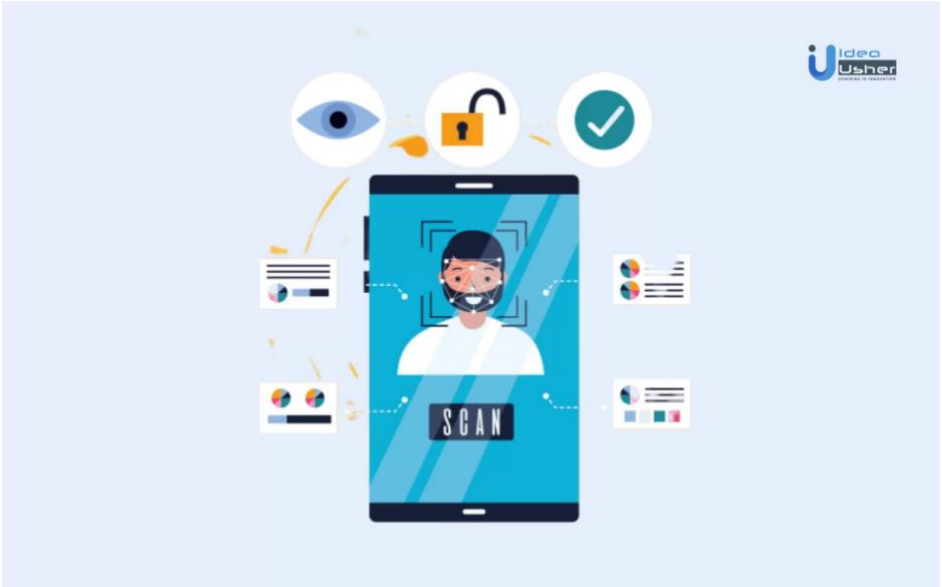


Fig 5.3.1 (System Flow)

5.3.2 Module Design and Organization

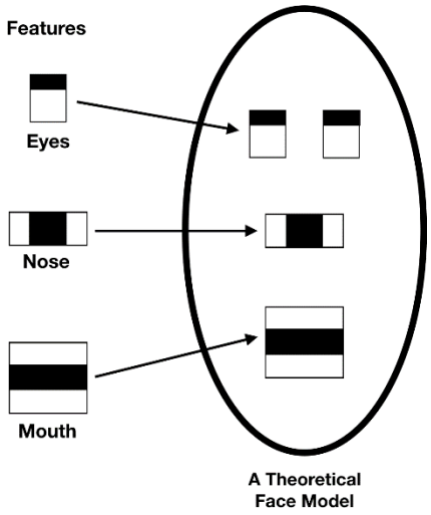




5.3.3 Analysis / Framework / Algorithm

Haar cascade Algorithm : -

It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images (where positive images are those where the object to be detected is present, negative are those where it is not). It is then used to detect objects in other images. Luckily, OpenCV offers pre-trained Haar cascade algorithms, organized into categories (faces, eyes and so forth), depending on the images they have been trained on.



5.3.3 (Haar Features)

LBPH Algorithm : -

Local Binary Pattern (LBP) is a simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number. It has further been determined that when LBP is combined with histograms of oriented gradients (HOG) descriptor, it improves the detection performance considerably on some datasets.

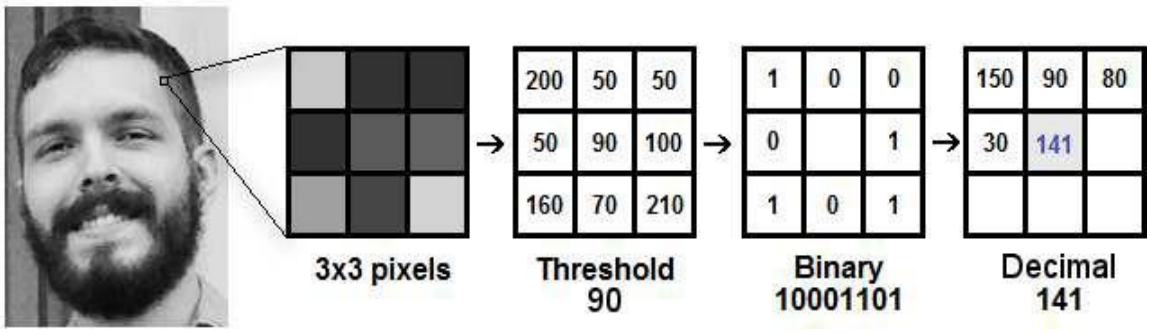


Fig 5.3.3 (LBPH Algorithm)

OpenCV Library :-

OpenCV (Open-Source Computer Vision Library) is an open-source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

NumPy package :-

NumPy is a Python package which stands for 'Numerical Python'. It is the core library for scientific computing, which contains a powerful n-dimensional array object, provide tools for integrating C, C++ etc. It is also useful in linear algebra, random number capability etc.

Pandas Library :-

Pandas is a high-level data manipulation tool developed by Wes McKinney. It is built on the NumPy package and its key data structure is called the Data Frame. Data Frames allow you to store and manipulate tabular data in rows of observations and columns of variables.

Tkinter Module :-

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit

Time Module :-

Python has a module named time to handle time related task. To use functions defined in the module, we need to import the module first.

Date Time Module :-

A date in python is not a date type of its own, but we can import a module named date time work with dates as a date objects.

6. Testing and Validation

Test process

The main test process of a facial recognition model. Firstly, we need to prepare a certain amount of test data of two parts: one is a set of test materials, such as a pair of face photos, the other one is to tag label for the materials according to the model to be tested, that is, the matchup of face photos. After the data preparation is done, we can start testing.

A message requesting the service of the model is constructed for each test material and sent to the “face recognition service”, and compares the returned result with the label. After all the test data are been tested, calculate the accuracy rate, recall rate, misrecognition rate, and other indicators based on the result set to evaluate the tested model.

Considerations for facial recognition data

Light and dark

Hairstyle

Whether to wear glasses?

Whether to wear a hat?

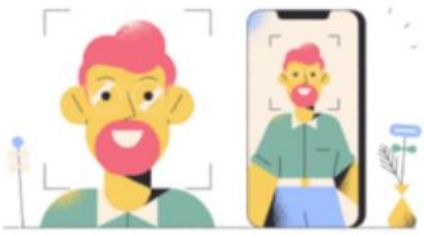
Whether the face is blocked?

Is the face facing the camera?

Background-color

Whether there are other faces in the background?

Face synthesis



Connectivity	Interface connectivity
Interface format	Missing parameters
	Empty parameter
	Invalid parameter format (length, type)
	Error code verification
model	Accuracy

7.User Manual

Install requirements package pip In cmd

tk-tools

opencv-contrib-python

datetime

pytest-shutil~

python-csv

numpy

pillow

pandas

times

Source Code

```
################################################ IMPORTING
################################################
import tkinter as tk
from tkinter import ttk
from tkinter import messagebox as mess
import tkinter.simpledialog as tsd
import cv2,os
import csv
import numpy as np
from PIL import Image
import pandas as pd
import datetime
import time

################################################ FUNCTIONS
################################################

def assure_path_exists(path):
    dir = os.path.dirname(path)
    if not os.path.exists(dir):
        os.makedirs(dir)

#####################################################################################
```

```
def tick():
time_string = time.strftime('%H:%M:%S')
clock.config(text=time_string)
clock.after(200,tick)

####################################################################################

def contact():
mess._show(title='Contact us', message="Please contact us on : 'styash54@gmail.com' ")

####################################################################################

def check_haarcascadefile():
exists = os.path.isfile("haarcascade_frontalface_default.xml")
if exists:
pass
else:
mess._show(title='Some file missing', message='Please contact us for help')
window.destroy()

####################################################################################

def save_pass():
assure_path_exists("TrainingImageLabel/")
exists1 = os.path.isfile("TrainingImageLabel\psd.txt")
if exists1:
tf = open("TrainingImageLabel\psd.txt", "r")
key = tf.read()
else:
master.destroy()
new_pas = tsd.askstring('Old Password not found', 'Please enter a new password below', show='*')
if new_pas == None:
mess._show(title='No Password Entered', message='Password not set!! Please try again')
else:
tf = open("TrainingImageLabel\psd.txt", "w")
tf.write(new_pas)
mess._show(title='Password Registered', message='New password was registered successfully!!')
return
op = (old.get())
newp= (new.get())
nnewp = (nnew.get())
if (op == key):
if(newp == nnewp):
txf = open("TrainingImageLabel\psd.txt", "w")
txf.write(newp)
else:
mess._show(title='Error', message='Confirm new password again!!!')
return
else:
mess._show(title='Wrong Password', message='Please enter correct old password.')
return
mess._show(title='Password Changed', message='Password changed successfully!!')
master.destroy()

####################################################################################

def change_pass():
global master
master = tk.Tk()
master.geometry("400x160")
master.resizable(False,False)
```

```
master.title("Change Password")
master.configure(background="white")
lbl4 = tk.Label(master,text='    Enter Old Password',bg='white',font=('times', 12, ' bold '))
lbl4.place(x=10,y=10)
global old
old=tk.Entry(master,width=25 ,fg="black",relief='solid',font=('times', 12, ' bold '),show='*')
old.place(x=180,y=10)
lbl5 = tk.Label(master, text='   Enter New Password', bg='white', font=('times', 12, ' bold '))
lbl5.place(x=10, y=45)
global new
new = tk.Entry(master, width=25, fg="black",relief='solid', font=('times', 12, ' bold '),show='*')
new.place(x=180, y=45)
lbl6 = tk.Label(master, text='Confirm New Password', bg='white', font=('times', 12, ' bold '))
lbl6.place(x=10, y=80)
global nnew
nnew = tk.Entry(master, width=25, fg="black", relief='solid',font=('times', 12, ' bold '),show='*')
nnew.place(x=180, y=80)
cancel=tk.Button(master,text="Cancel", command=master.destroy ,fg="black"  ,bg="red" ,height=1,width=25 ,
activebackground = "white" ,font=('times', 10, ' bold '))
cancel.place(x=200, y=120)
save1 = tk.Button(master, text="Save", command=save_pass, fg="black", bg="#3ece48", height = 1,width=25,
activebackground="white", font=('times', 10, ' bold '))
save1.place(x=10, y=120)
master.mainloop()

####################################################################################

def psw():
    assure_path_exists("TrainingImageLabel/")
    exists1 = os.path.isfile("TrainingImageLabel\psd.txt")
    if exists1:
        tf = open("TrainingImageLabel\psd.txt", "r")
        key = tf.read()
    else:
        new_pas = tsd.askstring('Old Password not found', 'Please enter a new password below', show='*')
        if new_pas == None:
            mess._show(title='No Password Entered', message='Password not set!! Please try again')
        else:
            tf = open("TrainingImageLabel\psd.txt", "w")
            tf.write(new_pas)
            mess._show(title='Password Registered', message='New password was registered successfully!!')
            return
    password = tsd.askstring('Password', 'Enter Password', show='*')
    if (password == key):
        TrainImages()
    elif (password == None):
        pass
    else:
        mess._show(title='Wrong Password', message='You have entered wrong password')

####################################################################################

def clear():
    txt.delete(0, 'end')
    res = "1)Take Images  >>>  2)Save Profile"
    message1.configure(text=res)


def clear2():
    txt2.delete(0, 'end')
    res = "1)Take Images  >>>  2)Save Profile"
```

```
message1.configure(text=res)

##########################################################################################

def TakeImages():
check_haarcascadefile()
columns = ['SERIAL NO.', '', 'ID', '', 'NAME']
assure_path_exists("StudentDetails/")
assure_path_exists("TrainingImage/")
serial = 0
exists = os.path.isfile("StudentDetails\StudentDetails.csv")
if exists:
with open("StudentDetails\StudentDetails.csv", 'r') as csvFile1:
reader1 = csv.reader(csvFile1)
for l in reader1:
serial = serial + 1
serial = (serial // 2)
csvFile1.close()
else:
with open("StudentDetails\StudentDetails.csv", 'a+') as csvFile1:
writer = csv.writer(csvFile1)
writer.writerow(columns)
serial = 1
csvFile1.close()
Id = (txt.get())
name = (txt2.get())
if ((name.isalpha()) or (' ' in name)):
cam = cv2.VideoCapture(0)
harcascadePath = "haarcascade_frontalface_default.xml"
detector = cv2.CascadeClassifier(harcascadePath)
sampleNum = 0
while (True):
ret, img = cam.read()
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
faces = detector.detectMultiScale(gray, 1.3, 5)
for (x, y, w, h) in faces:
cv2.rectangle(img, (x, y), (x + w, y + h), (255, 0, 0), 2)
# incrementing sample number
sampleNum = sampleNum + 1
# saving the captured face in the dataset folder TrainingImage
cv2.imwrite("TrainingImage\ " + name + "." + str(serial) + "." + Id + '.' + str(sampleNum) + ".jpg",
gray[y:y + h, x:x + w])
# display the frame
cv2.imshow('Taking Images', img)
# wait for 100 miliseconds
if cv2.waitKey(100) & 0xFF == ord('q'):
break
# break if the sample number is morethan 100
elif sampleNum > 25:
break
cam.release()
cv2.destroyAllWindows()
res = "Images Taken for ID : " + Id
row = [serial, '', Id, '', name]
with open('StudentDetails\StudentDetails.csv', 'a+') as csvFile:
writer = csv.writer(csvFile)
writer.writerow(row)
csvFile.close()
message1.configure(text=res)
else:
if (name.isalpha() == False):
```

```
res = "Enter Correct name"
message.configure(text=res)

##################################################################################################

def TrainImages():
check_haarcascadefile()
assure_path_exists("TrainingImageLabel/")
recognizer = cv2.face_LBPHFaceRecognizer.create()
harcascadePath = "haarcascade_frontalface_default.xml"
detector = cv2.CascadeClassifier(harcascadePath)
faces, ID = getImagesAndLabels("TrainingImage")
try:
recognizer.train(faces, np.array(ID))
except:
mess._show(title='No Registrations', message='Please Register someone first!!!')
return
recognizer.save("TrainingImageLabel\Trainner.yml")
res = "Profile Saved Successfully"
message1.configure(text=res)
message.configure(text='Total Registrations till now  : ' + str(ID[0]))

##################################################################################################3

def getImagesAndLabels(path):
# get the path of all the files in the folder
imagePaths = [os.path.join(path, f) for f in os.listdir(path)]
# create empth face list
faces = []
# create empty ID list
Ids = []
# now looping through all the image paths and loading the Ids and the images
for imagePath in imagePaths:
# loading the image and converting it to gray scale
pilImage = Image.open(imagePath).convert('L')
# Now we are converting the PIL image into numpy array
imageNp = np.array(pilImage, 'uint8')
# getting the Id from the image
ID = int(os.path.split(imagePath)[-1].split(".")[1])
# extract the face from the training image sample
faces.append(imageNp)
Ids.append(ID)
return faces, Ids

##################################################################################################

def TrackImages():
check_haarcascadefile()
assure_path_exists("Attendance/")
assure_path_exists("StudentDetails/")
for k in tv.get_children():
tv.delete(k)
msg = ''
i = 0
j = 0
recognizer = cv2.face.LBPHFaceRecognizer_create()  # cv2.createLBPHFaceRecognizer()
exists3 = os.path.isfile("TrainingImageLabel\Trainner.yml")
if exists3:
recognizer.read("TrainingImageLabel\Trainner.yml")
else:
mess._show(title='Data Missing', message='Please click on Save Profile to reset data!!')
```

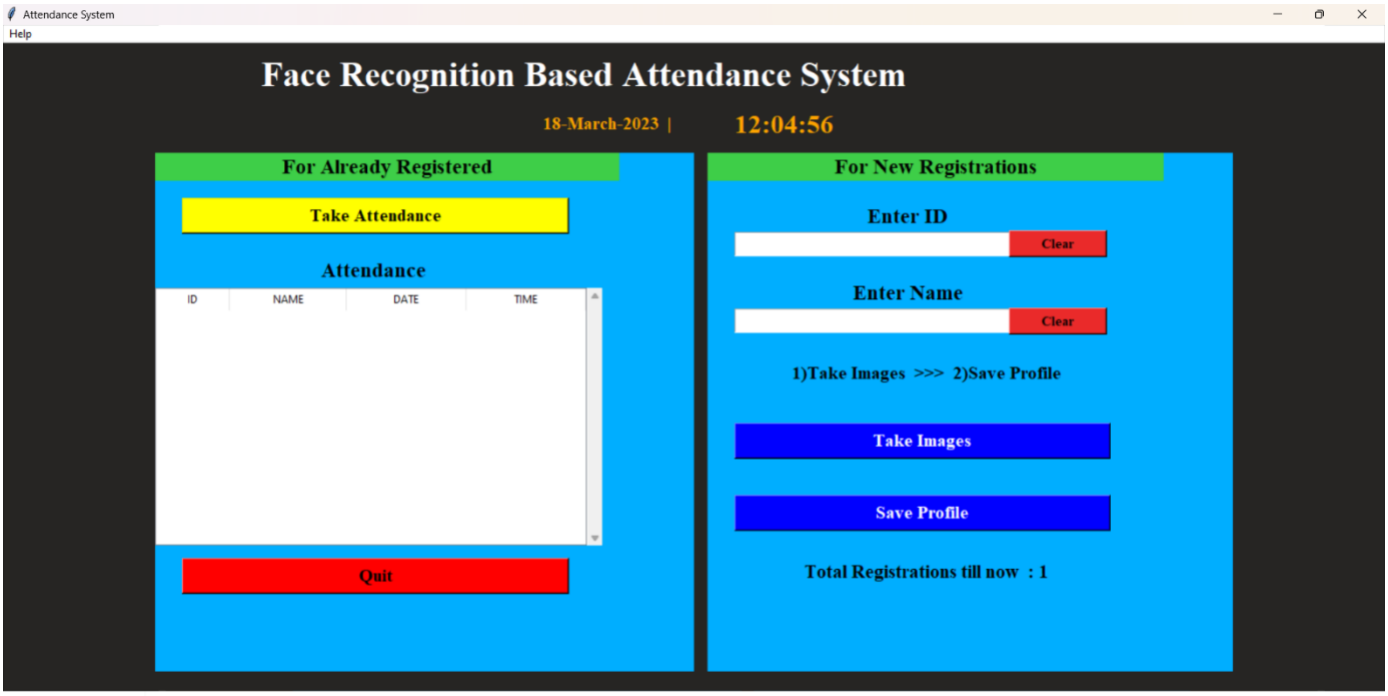
```
return
harcascadePath = "haarcascade_frontalface_default.xml"
faceCascade = cv2.CascadeClassifier(harcascadePath);

cam = cv2.VideoCapture(0)
font = cv2.FONT_HERSHEY_SIMPLEX
col_names = ['Id', '', 'Name', '', 'Date', '', 'Time']
exists1 = os.path.isfile("StudentDetails\StudentDetails.csv")
if exists1:
df = pd.read_csv("StudentDetails\StudentDetails.csv")
else:
mess._show(title='Details Missing', message='Students details are missing, please check!')
cam.release()
cv2.destroyAllWindows()
window.destroy()
while True:
ret, im = cam.read()
gray = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)
faces = faceCascade.detectMultiScale(gray, 1.2, 5)
for (x, y, w, h) in faces:
cv2.rectangle(im, (x, y), (x + w, y + h), (225, 0, 0), 2)
serial, conf = recognizer.predict(gray[y:y + h, x:x + w])
if (conf > 60):
ts = time.time()
date = datetime.datetime.fromtimestamp(ts).strftime('%d-%m-%y')
timeStamp = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
aa = df.loc[df['SERIAL NO.'] == serial]['NAME'].values
ID = df.loc[df['SERIAL NO.'] == serial]['ID'].values
ID = str(ID)
ID = ID[1:-1]
bb = str(aa)
bb = bb[2:-2]
attendance = [str(ID), '', bb, '', str(date), '', str(timeStamp)]

else:
Id = 'Unknown'
bb = str(Id)
cv2.putText(im, str(bb), (x, y + h), font, 1, (255, 255, 255), 2)
cv2.imshow('Taking Attendance', im)
if (cv2.waitKey(1) == ord('q')):
break
ts = time.time()
date = datetime.datetime.fromtimestamp(ts).strftime('%d-%m-%Y')
exists = os.path.isfile("Attendance\Attendance_" + date + ".csv")
if exists:
with open("Attendance\Attendance_" + date + ".csv", 'a+') as csvFile1:
writer = csv.writer(csvFile1)
writer.writerow(attendance)
csvFile1.close()
else:
with open("Attendance\Attendance_" + date + ".csv", 'a+') as csvFile1:
writer = csv.writer(csvFile1)
writer.writerow(col_names)
writer.writerow(attendance)
csvFile1.close()
with open("Attendance\Attendance_" + date + ".csv", 'r') as csvFile1:
reader1 = csv.reader(csvFile1)
for lines in reader1:
i = i + 1
if (i > 1):
if (i % 2 != 0):
```

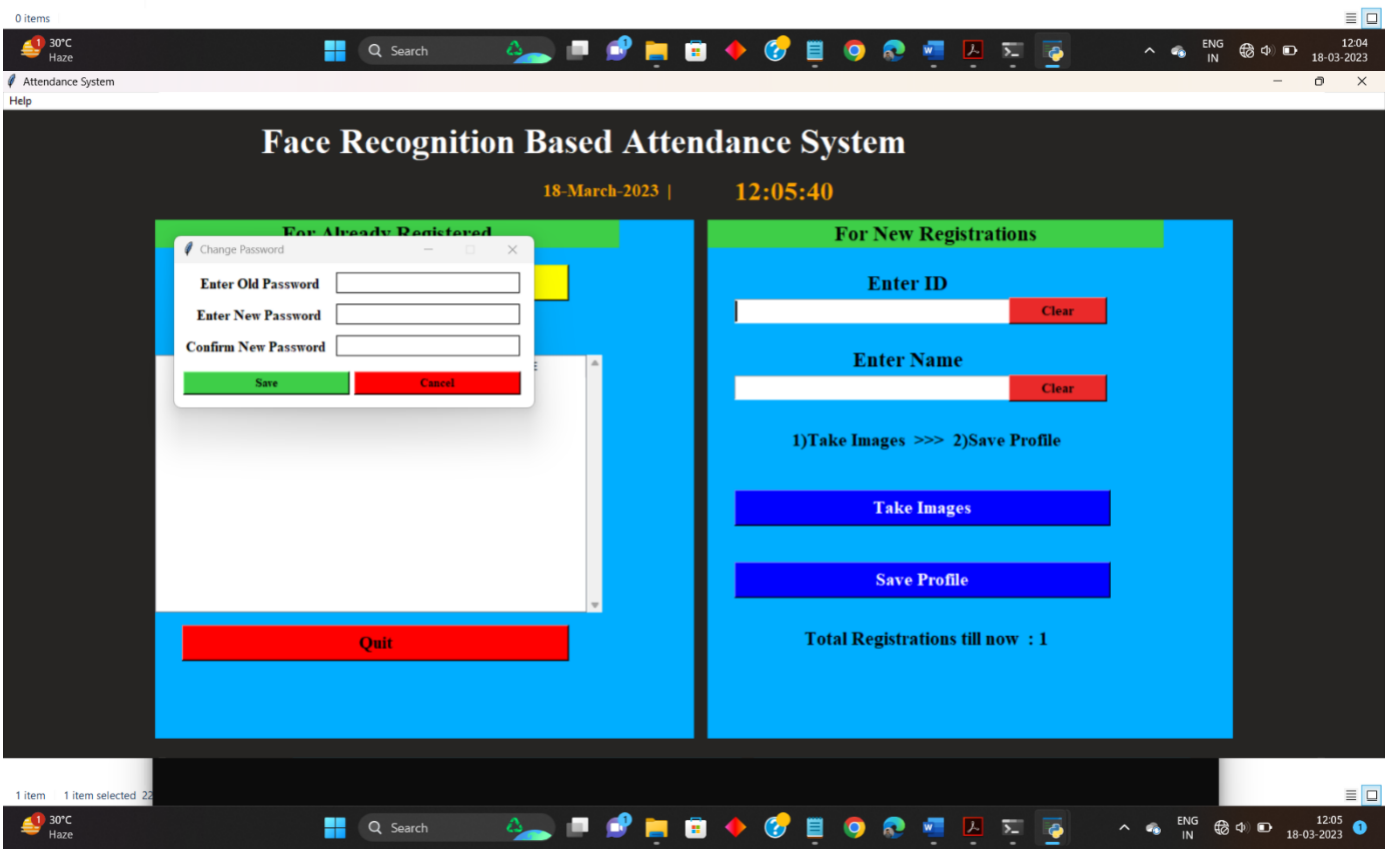
```
                iidd = str(lines[0]) + '   '
            tv.insert('', 0, text=iidd, values=(str(lines[2]), str(lines[4]), str(lines[6])))
            csvFile1.close()
            cam.release()
            cv2.destroyAllWindows()

        ######################################## USED STUFFS
        ############################################

            global key
            key = ''

            ts = time.time()
        date = datetime.datetime.fromtimestamp(ts).strftime('%d-%m-%Y')
            day,month,year=date.split("-")

            mont={'01':'January',
                '02':'February',
                '03':'March',
                '04':'April',
                '05':'May',
                '06':'June',
                '07':'July',
                '08':'August',
                '09':'September',
                '10':'October',
                '11':'November',
                '12':'December'
                }

        ######################################## GUI FRONT-END
        ###########################################

            window = tk.Tk()
            window.geometry("1280x720")
            window.resizable(True,False)
            window.title("Attendance System")
            window.configure(background='#262523')

            frame1 = tk.Frame(window, bg="#00aeff")
            frame1.place(relx=0.11, rely=0.17, relwidth=0.39, relheight=0.80)

            frame2 = tk.Frame(window, bg="#00aeff")
            frame2.place(relx=0.51, rely=0.17, relwidth=0.38, relheight=0.80)

        message3 = tk.Label(window, text="Face Recognition Based Attendance System" ,fg="white",bg="#262523"
            ,width=55 ,height=1,font=('times', 29, ' bold '))
            message3.place(x=10, y=10)

            frame3 = tk.Frame(window, bg="#c4c6ce")
            frame3.place(relx=0.52, rely=0.09, relwidth=0.09, relheight=0.07)

            frame4 = tk.Frame(window, bg="#c4c6ce")
            frame4.place(relx=0.36, rely=0.09, relwidth=0.16, relheight=0.07)

        datef = tk.Label(frame4, text = day+"-"+mont[month]+"-"+year+"  |  ", fg="orange",bg="#262523" ,width=55
            ,height=1,font=('times', 15, ' bold '))
            datef.pack(fill='both',expand=1)

        clock = tk.Label(frame3,fg="orange",bg="#262523" ,width=55 ,height=1,font=('times', 22, ' bold '))
            clock.pack(fill='both',expand=1)
```

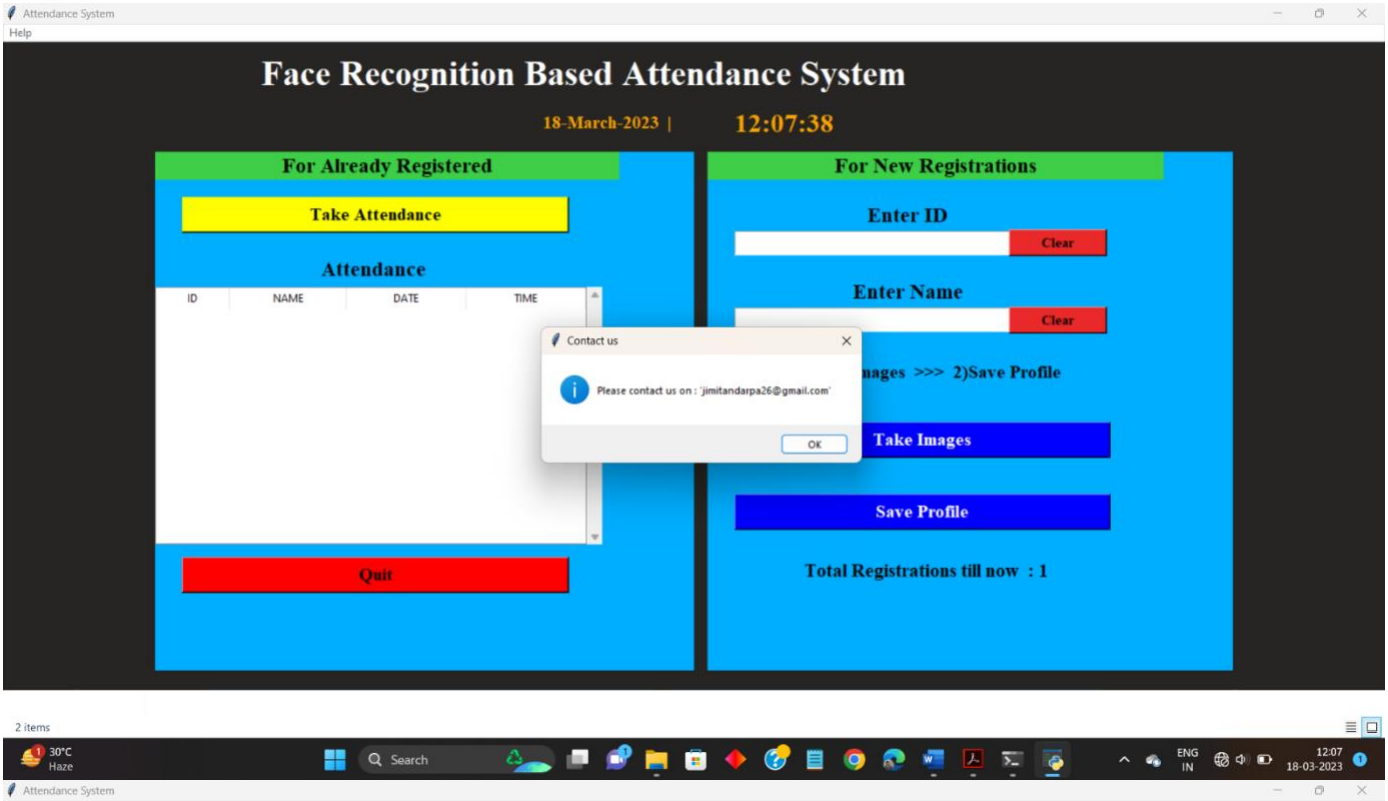
```
tick()

head2 = tk.Label(frame2, text="                For New Registrations                ", fg="black",bg="#3ece48"
,font=('times', 17, ' bold ') )
head2.grid(row=0,column=0)

head1 = tk.Label(frame1, text="                For Already Registered                ", fg="black",bg="#3ece48"
,font=('times', 17, ' bold ') )
head1.place(x=0,y=0)

lbl = tk.Label(frame2, text="Enter ID",width=20  ,height=1  ,fg="black"  ,bg="#00aeff" ,font=('times', 17, ' bold ') )
lbl.place(x=80, y=55)

txt = tk.Entry(frame2,width=32 ,fg="black",font=('times', 15, ' bold '))
txt.place(x=30, y=88)

lbl2 = tk.Label(frame2, text="Enter Name",width=20  ,fg="black"  ,bg="#00aeff" ,font=('times', 17, ' bold '))
lbl2.place(x=80, y=140)

txt2 = tk.Entry(frame2,width=32 ,fg="black",font=('times', 15, ' bold ')  )
txt2.place(x=30, y=173)

message1 = tk.Label(frame2, text="1)Take Images  >>>  2)Save Profile" ,bg="#00aeff" ,fg="black"  ,width=39
,height=1, activebackground = "yellow" ,font=('times', 15, ' bold '))
message1.place(x=7, y=230)

message = tk.Label(frame2, text="" ,bg="#00aeff" ,fg="black"  ,width=39,height=1, activebackground = "yellow"
,font=('times', 16, ' bold '))
message.place(x=7, y=450)

lbl3 = tk.Label(frame1, text="Attendance",width=20  ,fg="black"  ,bg="#00aeff"  ,height=1 ,font=('times', 17, ' bold
'))
lbl3.place(x=100, y=115)

res=0
exists = os.path.isfile("StudentDetails\StudentDetails.csv")
if exists:
with open("StudentDetails\StudentDetails.csv", 'r') as csvFile1:
reader1 = csv.reader(csvFile1)
for l in reader1:
res = res + 1
res = (res // 2) - 1
csvFile1.close()
else:
res = 0
message.configure(text='Total Registrations till now  : '+str(res))

###################### MENUBAR #################################

menubar = tk.Menu(window,relief='ridge')
filemenu = tk.Menu(menubar,tearoff=0)
filemenu.add_command(label='Change Password', command = change_pass)
filemenu.add_command(label='Contact Us', command = contact)
filemenu.add_command(label='Exit',command = window.destroy)
menubar.add_cascade(label='Help',font=('times', 29, ' bold '),menu=filemenu)

################## TREEVIEW ATTENDANCE TABLE ###################

tv= ttk.Treeview(frame1,height =13,columns = ('name','date','time'))
tv.column('#0',width=82)
tv.column('name',width=130)
```

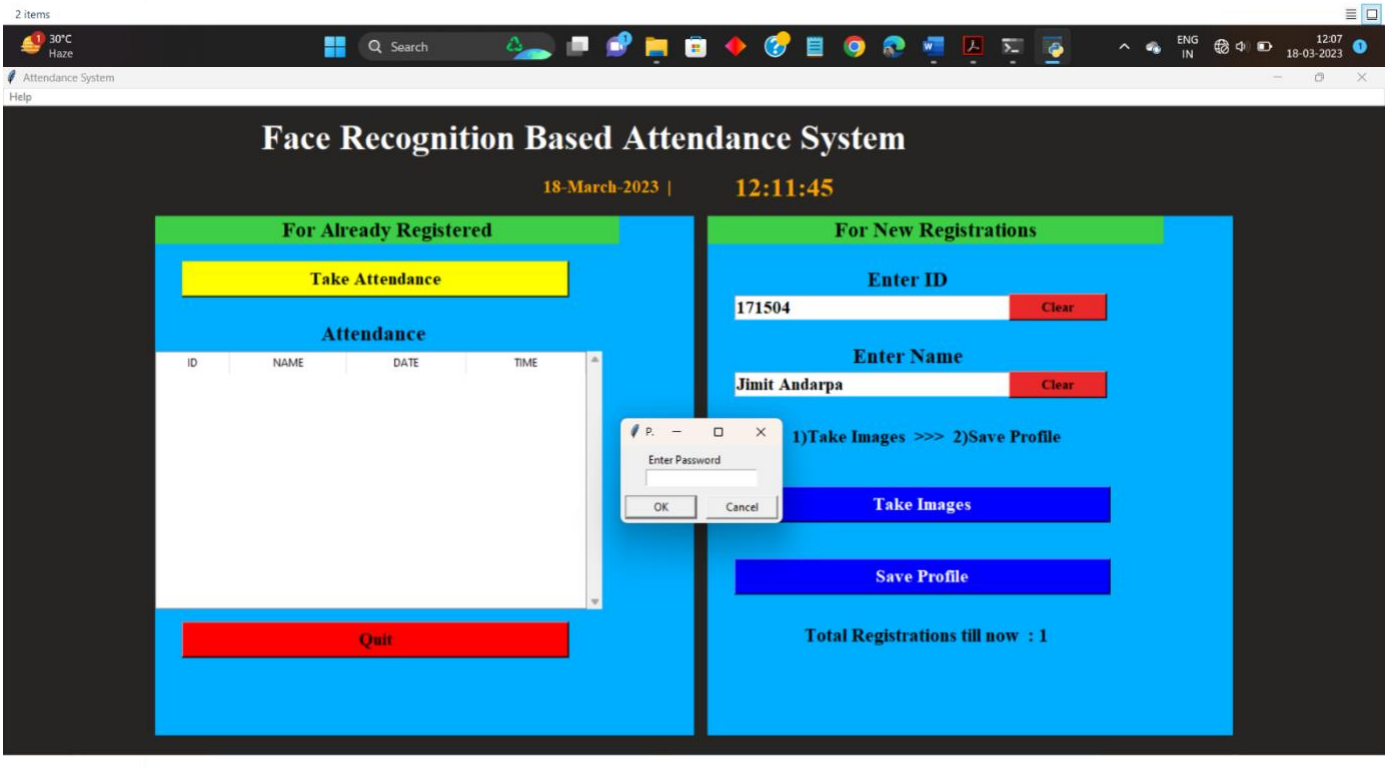
```
tv.column('date',width=133)
tv.column('time',width=133)
tv.grid(row=2,column=0,padx=(0,0),pady=(150,0),columnspan=4)
tv.heading('#0',text ='ID')
tv.heading('name',text ='NAME')
tv.heading('date',text ='DATE')
tv.heading('time',text ='TIME')

###################### SCROLLBAR ################################

scroll=ttk.Scrollbar(frame1,orient='vertical',command=tv.yview)
scroll.grid(row=2,column=4,padx=(0,100),pady=(150,0),sticky='ns')
tv.configure(yscrollcommand=scroll.set)

###################### BUTTONS ##################################

clearButton = tk.Button(frame2, text="Clear", command=clear  ,fg="black"  ,bg="#ea2a2a"  ,width=11
,activebackground = "white" ,font=('times', 11, ' bold '))
clearButton.place(x=335, y=86)
clearButton2 = tk.Button(frame2, text="Clear", command=clear2  ,fg="black"  ,bg="#ea2a2a"  ,width=11 ,
activebackground = "white" ,font=('times', 11, ' bold '))
clearButton2.place(x=335, y=172)
takeImg = tk.Button(frame2, text="Take Images", command=TakeImages  ,fg="white"  ,bg="blue"  ,width=34
,height=1, activebackground = "white" ,font=('times', 15, ' bold '))
takeImg.place(x=30, y=300)
trainImg = tk.Button(frame2, text="Save Profile", command=psw ,fg="white"  ,bg="blue"  ,width=34  ,height=1,
activebackground = "white" ,font=('times', 15, ' bold '))
trainImg.place(x=30, y=380)
trackImg = tk.Button(frame1, text="Take Attendance", command=TrackImages  ,fg="black"  ,bg="yellow"
,width=35  ,height=1, activebackground = "white" ,font=('times', 15, ' bold '))
trackImg.place(x=30,y=50)
quitWindow = tk.Button(frame1, text="Quit", command=window.destroy  ,fg="black"  ,bg="red"  ,width=35
,height=1, activebackground = "white" ,font=('times', 15, ' bold '))
quitWindow.place(x=30, y=450)

###################### END ######################################

window.configure(menu=menubar)
window.mainloop()

#########################################################################################################
######
```

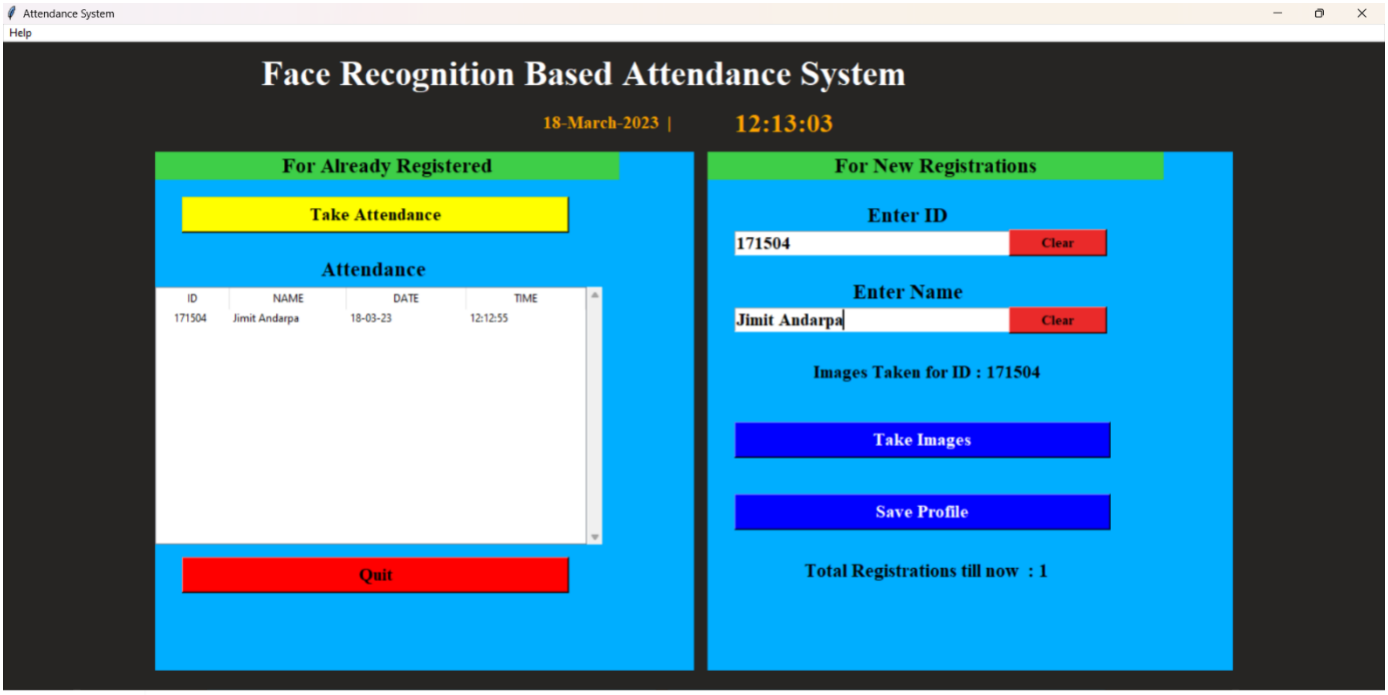
8.OutPut

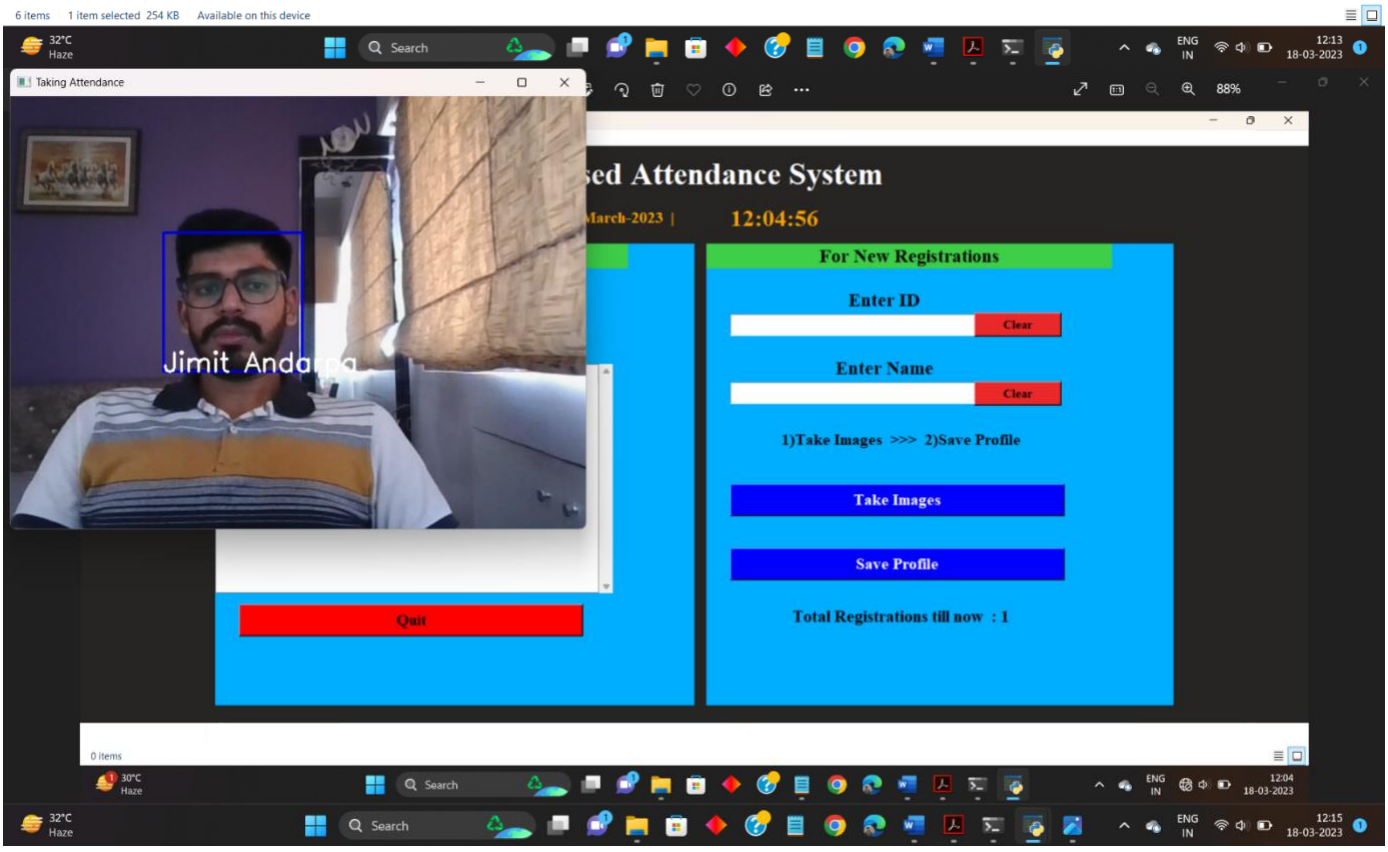

Attendance System
Help
Face Recognition Based Attendance System
18-March-2023 |
12:04:56
For Already Registered
Take Attendance
Attendance
ID
NAME
DATE
TIME
Quit
For New Registrations
Enter ID
Clear
Enter Name
Clear
1)Take Images >>> 2)Save Profile
Take Images
Save Profile
Total Registrations till now : 1

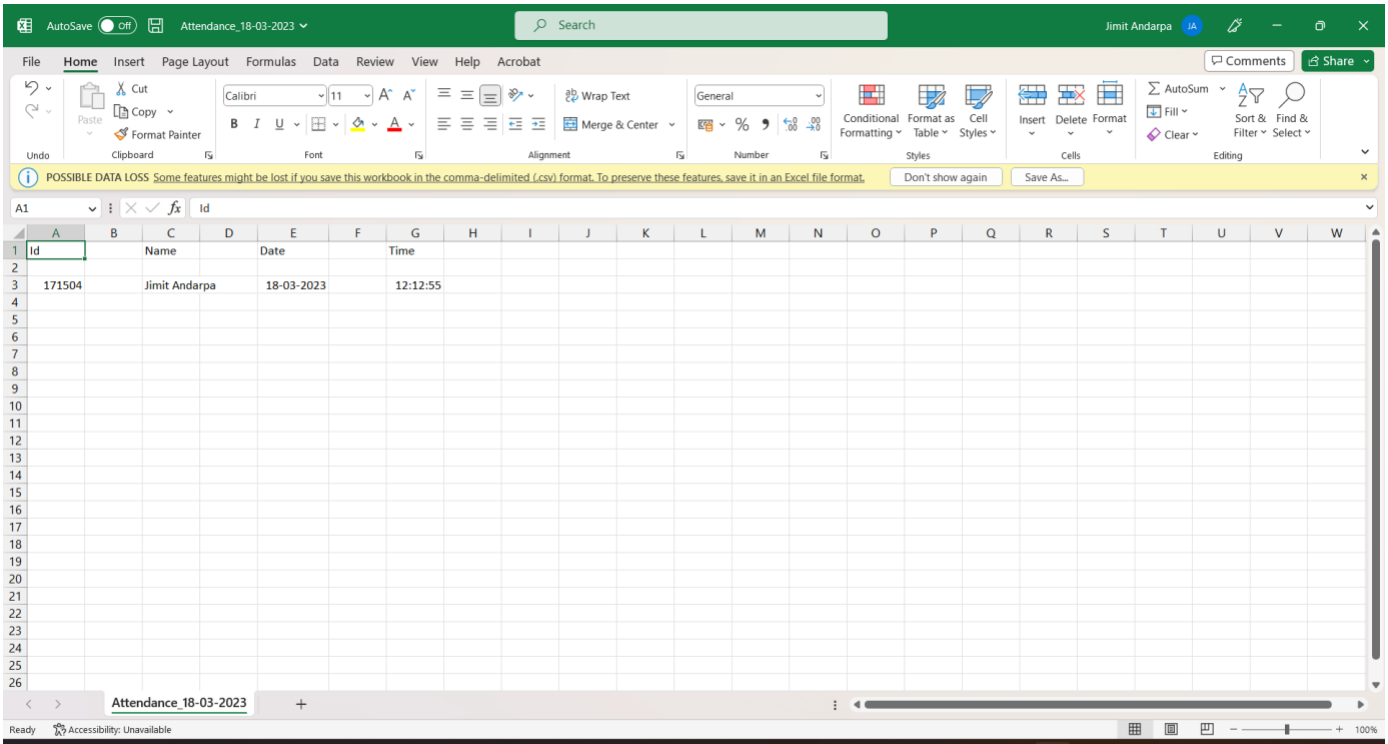

Attendance System
Help
Face Recognition Based Attendance System
18-March-2023 |
12:05:40
Change Password
Enter Old Password
Enter New Password
Confirm New Password
Save
Cancel
Quit
For New Registrations
Enter ID
Clear
Enter Name
Clear
1)Take Images >>> 2)Save Profile
Take Images
Save Profile
Total Registrations till now : 1

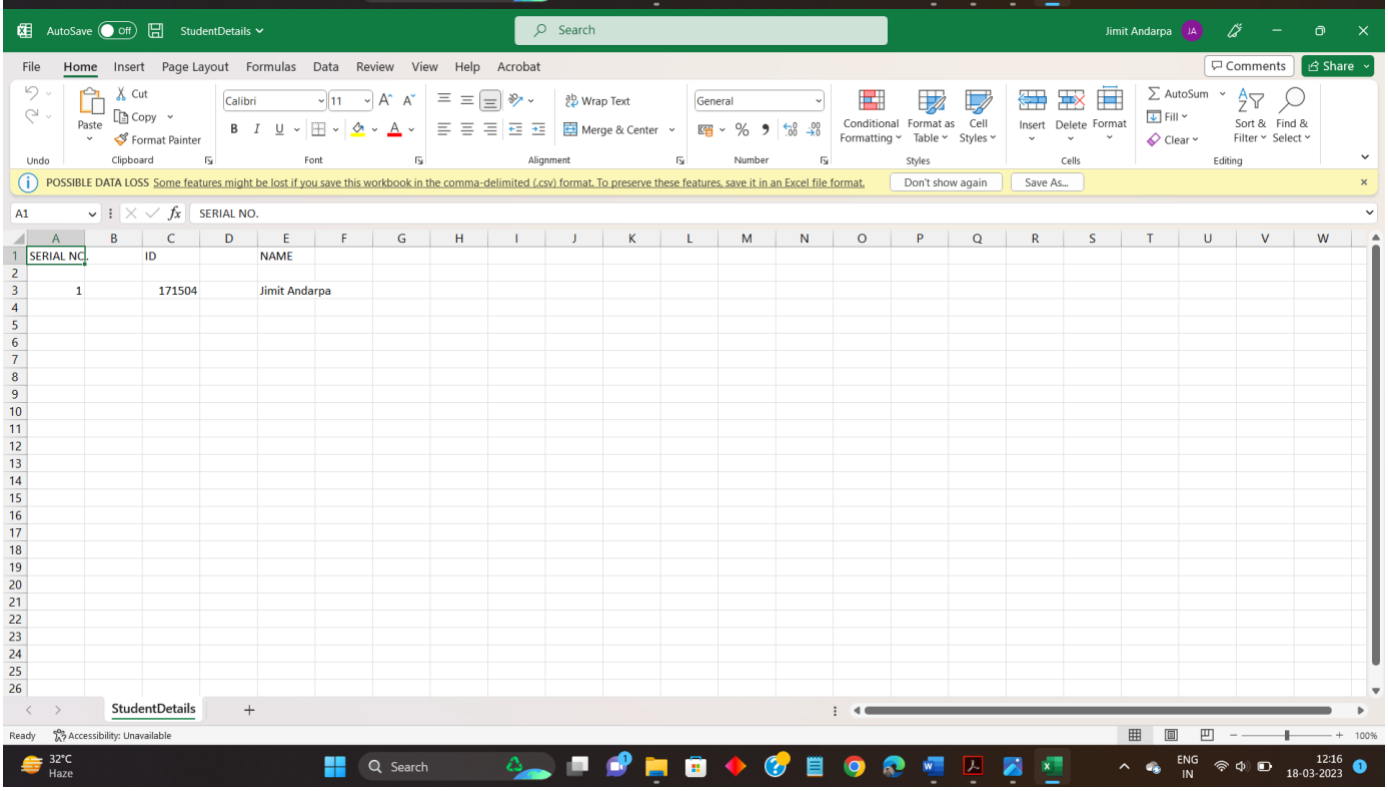

Attendance System
Help
Face Recognition Based Attendance System
18-March-2023 |
12:07:38
For Already Registered
Take Attendance
Attendance
ID
NAME
DATE
TIME
Quit
For New Registrations
Enter ID
Clear
Enter Name
Clear
nages >>> 2)Save Profile
Take Images
Save Profile
Total Registrations till now : 1
Contact us
Please contact us on : 'jimitandarpa26@gmail.com'
OK


Attendance System
Help
Face Recognition Based Attendance System
18-March-2023 |
12:11:45
For Already Registered
Take Attendance
Attendance
ID
NAME
DATE
TIME
Quit
For New Registrations
Enter ID
171504
Clear
Enter Name
Jimit Andarpa
Clear
1)Take Images >>> 2)Save Profile
Take Images
Save Profile
Total Registrations till now : 1
P.
Enter Password
OK
Cancel


Attendance System
Help
Face Recognition Based Attendance System
18-March-2023 |
12:13:03
For Already Registered
Take Attendance
Attendance
ID
NAME
DATE
TIME
171504
Jimit Andarpa
18-03-23
12:12:55
Quit
For New Registrations
Enter ID
171504
Clear
Enter Name
Jimit Andarpa
Clear
Images Taken for ID : 171504
Take Images
Save Profile
Total Registrations till now : 1


Taking Attendance
Jimit Andarpa
sed Attendance System
March-2023 |
12:04:56
For New Registrations
Enter ID
Clear
Enter Name
Clear
1)Take Images >>> 2)Save Profile
Take Images
Save Profile
Total Registrations till now : 1
Quit
Search
ENG
IN
12:13
18-03-2023
12:15
18-03-2023


AutoSave Off
Attendance_18-03-2023
Search
Jimit Andarpa
File Home Insert Page Layout Formulas Data Review View Help Acrobat
Comments
Share
Cut
Copy
Format Painter
Paste
Calibri
11
Wrap Text
Merge & Center
General
Conditional Formatting
Format as Table
Cell Styles
Insert
Delete
Format
AutoSum
Fill
Clear
Sort & Filter
Find & Select
Undo
Clipboard
Font
Alignment
Number
Styles
Cells
Editing
POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format.
Don't show again
Save As...
A1
Id
Id
Name
Date
Time
171504
Jimit Andarpa
18-03-2023
12:12:55
Attendance_18-03-2023
Ready
Accessibility: Unavailable
100%
32°C
Haze
Search
ENG
IN
12:16
18-03-2023


AutoSave Off
StudentDetails
Search
Jimit Andarpa
File Home Insert Page Layout Formulas Data Review View Help Acrobat
Comments
Share
Cut
Copy
Format Painter
Paste
Calibri
11
Wrap Text
Merge & Center
General
Conditional Formatting
Format as Table
Cell Styles
Insert
Delete
Format
AutoSum
Fill
Clear
Sort & Filter
Find & Select
Undo
Clipboard
Font
Alignment
Number
Styles
Cells
Editing
POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format.
Don't show again
Save As...
A1
SERIAL NO.
SERIAL NO.
ID
NAME
1
171504
Jimit Andarpa
StudentDetails
Ready
Accessibility: Unavailable
100%
32°C
Haze
Search
ENG
IN
12:16
18-03-2023

9.CONCLUSION

Automated Attendance System has been envisioned for the purpose of reducing the errors that occur in the traditional (manual) attendance taking system. The aim is to automate and make a system that is useful to the organization such as an institute. The efficient and accurate method of attendance in the office environment that can replace the old manual methods. This method is secure enough, reliable and available for use. No need for specialized hardware for installing the system in the office. It can be constructed using a camera and computer. In this system we have implemented an attendance system for a lecture, section or laboratory by which lecturer or teaching assistant can record students' attendance. It saves time and effort, especially if it is a lecture with huge number of students. Automated Attendance System has been envisioned for the purpose of reducing the drawbacks in the traditional (manual) system. This attendance system demonstrates the use of image processing techniques in classroom. This system can not only merely help in the attendance system, but also improve the goodwill of an institution.

10.REFERENCES

[1] Michael Dobson, Douglas Ahlers, Bernie DiDario, <Attendance Tracking System", United States Patent Application Publication, Pub. No.: US 2006/0035205 A1, Feb.16, 2006.

[2] Naveed Khan Balcoh, M. HaroonYousaf, Waqar Ahmad and M. IramBaig, <Algorithm for Efficient Attendance Management: Face Recognition based approach=, IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 4, No 1, pp.146-150, July 2012.

[3] O. Shoewu and O.A. Idowu, <Development of Attendance Management System using Biometrics ", The Pacific Journal of Science and Technology, Vol. 13, Number1, pp.300-307, May 2012 (Spring).

[4] Damir Demirovic, Emir Skejic , Amira Serfovic, <Performance of some images processing algorithms in TensorFlow=, computer vision and pattern recognition, pp, 799- 778, 2018.

[5] http://www.openCV.org

[6] Hapani, Smit, et al. "Automated Attendance System Using Image Processing." 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA). IEEE, 2018.

[7] S. Bahrampour, N. Ramakrishnan, L. Schott. And M. Shah, M. Comparative Study of Caffe, Neon, Theano, and Torch for Deep Learning<. CoRR, abs/1511.06435. 2015.

[8] I. K. Park, N. Singhal, M. H. Lee, S. Cho and C. Kim, "Design and Performance Evaluation of Image Processing Algorithms on GPUs," in IEEE Transactions on Parallel and Distributed Systems, vol. 22, no. 1, pp. 91-104, Jan. 2011.