ATTENDANCE MONITORING SYSTEM

SUBMITTED BY-

Sai Teja. K – 18BCE0021

Sri Sai Manikanta Vishnu Shasank -18BCE2131

Vakati Sai Kireeti – 18BCE0087

Sannareddy Vamsi – 19BCE0266

COURSE TITLE : Image Processing

COURSE CODE : CSE4019

Review-3

Under the guidance of

Professor : Rajakumar K

VIT , Vellore.

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

INDEX:

Abstract

Attendance Monitoring System is essential in all organizations for checking the performance of students and it is not easy task to check each and every student is present or not. In all organization attendance are taken manually by calling their register numbers or names and noted in attendance registers issued by the department heads as a proof and in some organizations the students want to sign in these sheets which are stored for future references. This technique is repetitive, complex work and leads to errors as few students regularly sign for their absent students or telling proxy attendance of the absent students. This method additionally makes it more complex to track all the student's attendance and difficult to monitoring the individual student attendance in a big classroom atmosphere. In this article, we use are using the technique of utilization face detection and recognition framework to continuously recognize students going to class or not and marking their attendance by comparing their faces with database to match and marking attendance.

Problem Statement:

Develop an attendance management system (face biometry) which helps management to generalise the process and automate things. This should work on with face and also supports manual entry of attendance details.

General problems faced by traditional methods :

1. Unfair attendance given by students
2. Time consumption
3. More chances of error occurrence.
4. Less generalization implies more hard work.

AIM:

The aim of the project is to develop a software that helps faculty to avoid taking attendance by automatic attendance monitoring system. This Captures the faces of students attending the classes and marks present for them. This simplifies the work consumed by database workers and administration.

This project includes a new approach of detecting faces without using face-recognition module. Everything is discussed below.

1. Introduction:

Face detection using Haar cascades is a machine learning based approach where a cascade function is trained with a set of input data. OpenCV already contains many pre-trained classifiers for face, eyes, smiles, etc. Today we will be using the face classifier. You can experiment with other classifiers as well. You need to download the trained classifier XML file (haarcascade_frontalface_default.xml), which is available in OpenCV's GitHub repository. Save it to your working location.

2. Literature Survey:

- Jadhav, Akshay Jadhav Tushar Ladhe, Krishna Yeolekar "Automated Attendance System Using Face Recognition" Irjet Volume: 04 Issue: 01 | Jan -2017
    1. The system uses the eigenface approach for face recognition. The method analyses and computes eigenfaces which are faces composed of eigenvectors. The method also compares the eigenfaces to identify the presence of a person(face) and its identity. The method involves the following steps [1].As a first step the system should be initialized with a set of training faces.Next,when a face is detected the eigenface is calculated for that face. Then, the system compares the eigenvectors of the current face and the stored ace image and determines whether the face is identified or not. The final step(optional) is that if the unknown face is detected repeatedly the system may learn to recognize it.

- Refik Samet,Muhammed Tanriverdi "Face Recognition-Based Mobile Automatic Classroom Attendance Management System" Published in Ieee 2017 International Conference On Cyberworlds (Cw) (Doi: 10.1109/Cw.2017.34)
    1. In the proposed system, RESTful web services were used for communication among teacher, student, and parent applications

and the cloud server. Attendance results are stored in a database and accessible by the teacher, student and parent mobile applications.

- B. K. Mohamed and C. Raghu - "Fingerprint attendance system for classroom needs", in India Conference (INDICON), 2012 Annual IEEE. IEEE, 2012, pp.433438.
    1. There are some problems in conventional attendance tracking system like one is a student missing out their name, while the other leads to a false attendance record. Another issue of having the attendance record in a hardcopy form is that a lecturer may lose the attendance sheet [7]. For student attendance analysis, to obtain the student attendance percentage, manual computation has to be performed by faculty.

- S. Konatham, B.S. Chalasani, N. Kulkarni, and T.E. Taeib,"Attendance generating system using RFID and GSM," IEEE Long Island Systems, Applications and Technology Conference (LISAT),2016.

- Chaitanya Reddy "Face Recognition Using Artificial Intelligence" Towardsdatascience.Com In April 2017
    1. Over the years there were many methods used to implement facial recognition models but thanks to Artificial Intelligence it made our life easier. Using Deep Learning(part of AI), provided with the sufficient data a Facial Recognition System can be built simply and with a high accuracy. We use the a simple Convolutional Neural Networks(CNN) model in order to build a Facial Recognition System.

- Facial Recognition System Using Local Binary Patterns(LBP) TS Vishnu Priya, G.Vinitha Sanchez, N.R.Raajan School of Electrical & Electronics Engineering,

3.Overview of the work

3.1 Objective:

The student's attendance system using artificial intelligence concept mainly works using the concept of facial recognition system. Face is considered as a primary key feature to identify and talk with other peoples in the world because face considered as a unique identity for each and every person. The facial features will be unique to the other individual. Human distinguish a particular person’s face based on several factors like colour, nose, eyes, ears, etc but for computers, it's difficult to analyse the data so we may use the concept of Computer vision

3.2 Software Requirements:

1. Python 3.8.1
2. Pip install OpenCV (CV2)
3. pip install NumPy
4. TensorFlow
5. Pillow
6. MySQL-connect
7. Pandas

3.3 Hardware Requirements

1. Pc with python installed
2. Web cam

4.System Design

4.1 Methodology:

Functional specifications are the requirements in which requires to operate a system. These requirements are necessary to assemble a system which will be required to attain the objectives embarked on previously was implemented by Jireh Robert Jam.Some of the important functional and non-functional requirements are outlined below by analysing the shop keeper story.

- First capturing the facial image by high quality camera o Facial features should be detected in photo.
- Crop the overall ranges of faces detected.
- Resize all the images until the recognition system takes photo to recognize.
- Calculating the overall attendance percentage based on facial features matched.
- Storing all the detected face images in a folder.
- Loading the images into the database.
- We want to train facial features to computer to recognize.
- Perform recognition for faces stored on database.
- Calculate computer facial recognition speed for effective security.
- Performing face recognition sequentially for each image cropped
- Displaying input and output cropped image side by side on a same slot to recognize and compare the features by machine.
- After recognizing the face displays the name of the output image above the image in the given area to identify easily.

4.2 Proposed algorithm:

1. Capture the Student's Image

2. Apply Face detection algorithms to detect face

3. Extract the Region of Interest in Rectangular Bounding Box

4. Convert to Gray scale, apply histogram equalization and resize to 100x100 i.e., Apply pre-processing

5. if Enrolment Phase then

Store in Database

Else

end if

4.3 FACE DETECTING ALGORITHM:

In order to overcome the problems, the algorithm local binary pattern is proposed. Since face image is composed of several minute patterns this can be efficiently identified by applying the local binary pattern operator. The local binary pattern operator is applied on the given face image.

METHOD OF LOCAL BINARY PATTERNS(LBP):

In local binary pattern the input face is first converted into the grey image and for that image the binary pattern is calculated by comparing the centre pixel with the surrounding pixel. If the centre pixel is greater than that of the neighbouring pixel then it is denoted as 1 and if the neighbouring pixel is smaller than that of the centre pixel it is denoted as 0.This should be done for each and every pixel so that we will get the binary pattern.

4.3.1 ADVANTAGES:

• **Improvement of Security Level**: Every organization needs to secure their premises for the unknown entry into that place. They also wish to monitor the employees and indusial entry into that place. Those who are entering the organization premises without proper access they are capturing in the security surveillance system and notice to the respective person and alerts instantly concerning the person who doesn't have permission.

• **Straightforward Integration method** :The automatic face detection tools works effectively with the current authentication code that organizations have developed.Bascially the technique is a straightforward to code the system to access organizations automatic data processing which makes the method very clear.

• **High Accuracy Rates** :The main advantage is its Accuracy. The system checks and gives the output without any misunderstanding and bad face detection system. The authorized person will be detected at the right time due to the high accuracy levels. The manual recognition, which is done by securities outside of the organization's premises we may use the face recognition technology to automate the process of identification and assures its perfection without changes. We don't want addition employee to monitor the working of cameras 24/7. The main objective of Automation means to reduce human effect and reducing the cost of employees too. Then any organization can recognize the fact that usage of automated face identification is highly secure with accurate data.

5.PROJECT STRUCTURE

• After run you need to give your face data to system so enter your ID and name in box than click on `Take Images` button.

• It will collect 200 images of your faces; it saves a image in `TrainingImage` folder

• After that we need to train a model(for train a model click on `Train Image` button.

• It will take 5-10 minutes for training(for 10 person data).

• After training click on `Automatic Attendance` ,it can fill attendance by your face using our trained model (model will save in `TrainingImageLabel`)

• it will create `.csv` file of attendance according to time & subject.

• You can store data in database (install WampServer),change the DB name according to you in `AMS_Run.py`.

• `Manually Fill Attendance` Button in UI is for fill a manually attendance (without face recognition),it's also created a `.csv` and store in a database.

6.IMPLEMENTATION :

6.1 DESCRIPTION OF MODULES AND PROGRAMS

• AMS_Run

Main module where UI exists and everything about database connectivity and usage of other modules

• Retrain

Retraining the model with images that are present in TestingImage • Training After capturing a set of images every image will be stored in "C:\Users\vamsi\Documents\AllProjects\python projects\Machine learning" , we can use these images for next step

• Testing

Testing of the model is this part

Note : this project uses "haarcascade_frontalface_alt" and "haarcascade_frontalface_default"

6.2 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 :
'vamsisannadi@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=('comic', 12, ' bold '))
    lbl4.place(x=10,y=10)
    global old
    old=tk.Entry(master,width=25 ,fg="black",relief='solid',font=('comic', 12,
' bold '),show='*')
    old.place(x=180,y=10)
    lbl5 = tk.Label(master, text='   Enter New Password', bg='white',
font=('comic', 12, ' bold '))
    lbl5.place(x=10, y=45)
    global new
    new = tk.Entry(master, width=25, fg="black",relief='solid', font=('comic',
12, ' bold '),show='*')
    new.place(x=180, y=45)
    lbl6 = tk.Label(master, text='Confirm New Password', bg='white',
font=('comic', 12, ' bold '))
    lbl6.place(x=10, y=80)
    global nnew
    nnew = tk.Entry(master, width=25, fg="black",
relief='solid',font=('comic', 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=('comic', 10, ' bold '))
    cancel.place(x=200, y=120)
    save1 = tk.Button(master, text="Save", command=save_pass, fg="black",
bg="#00fcca", height = 1,width=25, activebackground="white", font=('comic',
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 > 100:
                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):
    imagePaths = [os.path.join(path, f) for f in os.listdir(path)]
    faces = []
    Ids = []
    for imagePath in imagePaths:
        pilImage = Image.open(imagePath).convert('L')
        imageNp = np.array(pilImage, 'uint8')
        ID = int(os.path.split(imagePath)[-1].split(".")[1])
        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()
    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 < 50):
                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='#2d420a')

frame1 = tk.Frame(window, bg="#c79cff")
frame1.place(relx=0.11, rely=0.17, relwidth=0.39, relheight=0.80)

frame2 = tk.Frame(window, bg="#c79cff")
frame2.place(relx=0.51, rely=0.17, relwidth=0.38, relheight=0.80)

message3 = tk.Label(window, text="Face Recognition Based Attendance Monitoring
System" ,fg="white",bg="#2d420a" ,width=55 ,height=1,font=('comic', 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="#ff61e5",bg="#2d420a" ,width=55 ,height=1,font=('comic', 22, ' bold '))
datef.pack(fill='both',expand=1)

clock = tk.Label(frame3,fg="#ff61e5",bg="#2d420a" ,width=55
,height=1,font=('comic', 22, ' bold '))
clock.pack(fill='both',expand=1)
tick()

head2 = tk.Label(frame2, text="                        For New
Registrations                        ", fg="black",bg="#00fcca" ,font=('comic',
17, ' bold ') )
head2.grid(row=0,column=0)
```

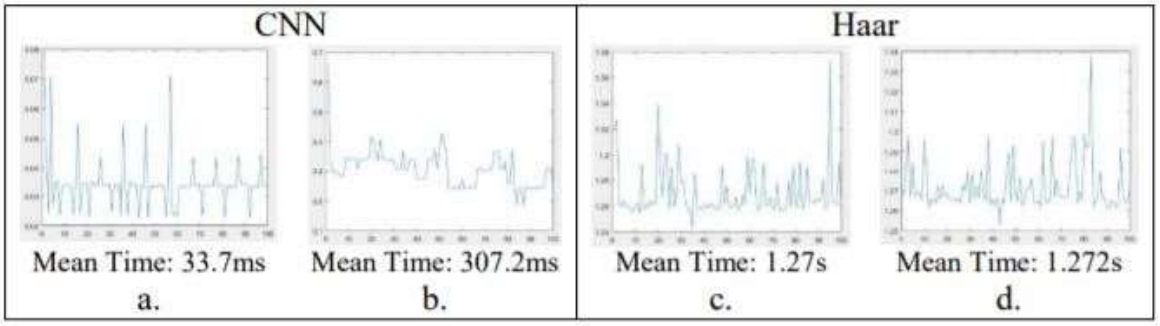
```
head1 = tk.Label(frame1, text="                       For Already
Registered                       ", fg="black",bg="#00fcca" ,font=('comic',
17, ' bold ') )
head1.place(x=0,y=0)

lbl = tk.Label(frame2, text="Enter
ID",width=20  ,height=1  ,fg="black"  ,bg="#c79cff" ,font=('comic', 17, ' bold
') )
lbl.place(x=80, y=55)

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

lbl2 = tk.Label(frame2, text="Enter Name",width=20  ,fg="black"  ,bg="#c79cff"
,font=('comic', 17, ' bold '))
lbl2.place(x=80, y=140)

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

message1 = tk.Label(frame2, text="1)Take Images  >>>  2)Save Profile"
,bg="#c79cff" ,fg="black"  ,width=39 ,height=1, activebackground = "#3ffc00"
,font=('comic', 15, ' bold '))
message1.place(x=7, y=230)

message = tk.Label(frame2, text="" ,bg="#c79cff"
,fg="black"  ,width=39,height=1, activebackground = "#3ffc00" ,font=('comic',
16, ' bold '))
message.place(x=7, y=450)

lbl3 = tk.Label(frame1,
text="Attendance",width=20  ,fg="black"  ,bg="#c79cff"  ,height=1
,font=('comic', 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=('comic', 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="#ff7221"  ,width=11 ,activebackground =
"white" ,font=('comic', 11, ' bold '))
clearButton.place(x=335, y=86)
clearButton2 = tk.Button(frame2, text="Clear",
command=clear2  ,fg="black"  ,bg="#ff7221"  ,width=11 , activebackground =
"white" ,font=('comic', 11, ' bold '))
clearButton2.place(x=335, y=172)
takeImg = tk.Button(frame2, text="Take Images",
command=TakeImages  ,fg="white"  ,bg="#6d00fc"  ,width=34  ,height=1,
activebackground = "white" ,font=('comic', 15, ' bold '))
takeImg.place(x=30, y=300)
trainImg = tk.Button(frame2, text="Save Profile", command=psw
,fg="white"  ,bg="#6d00fc"  ,width=34  ,height=1, activebackground = "white"
,font=('comic', 15, ' bold '))
trainImg.place(x=30, y=380)
```

```
trackImg = tk.Button(frame1, text="Take Attendance",
command=TrackImages  ,fg="black"  ,bg="#3ffc00"  ,width=35  ,height=1,
activebackground = "white" ,font=('comic', 15, ' bold '))
trackImg.place(x=30,y=50)
quitWindow = tk.Button(frame1, text="Quit",
command=window.destroy  ,fg="black"  ,bg="#eb4600"  ,width=35 ,height=1,
activebackground = "white" ,font=('comic', 15, ' bold '))
quitWindow.place(x=30, y=450)


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

7.OUTPUT AND PERFORMANCE ANALYSIS :

The image is captured, recognized and stored in the database along with the student roll number and name. When the next time the image is identified and found a match, the name and roll number of the student pops up and his/her attendance is marked present for the specified subject slot.



Advantages of proposed technique over the existing techniques

Comparison between Haar cascades and Convolutional Neural Networks

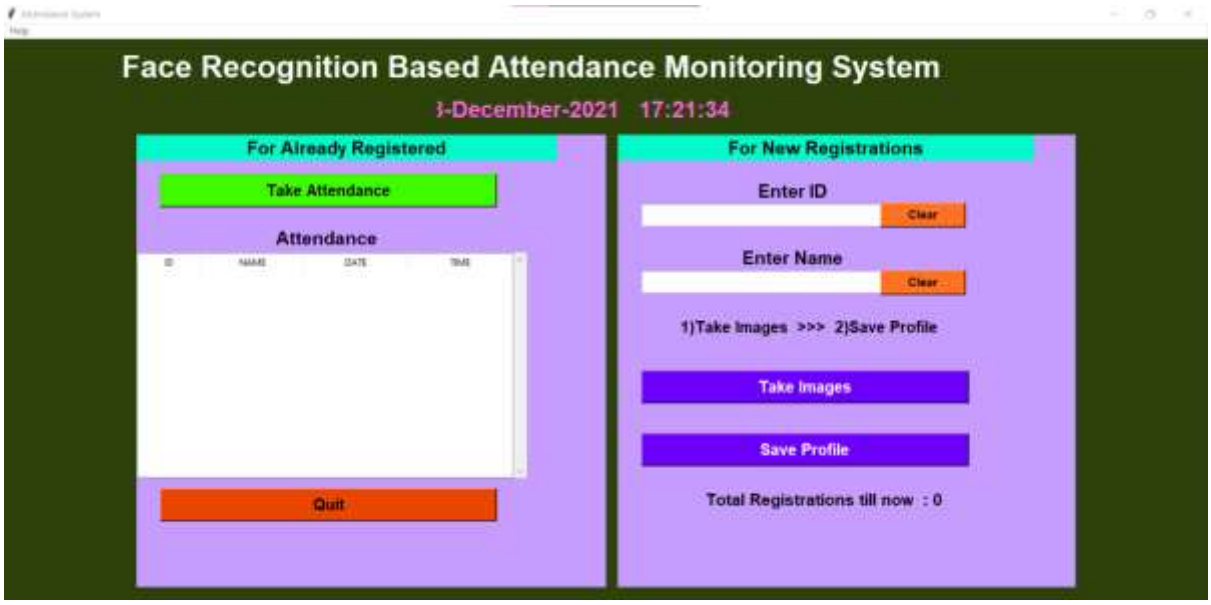
Useche Murillo, Paula & Moreno, Robinson & Pinzón Arenas, Javier. (2017). Comparison between CNN and Haar classifiers for surgical instrumentation classification. Contemporary Engineering Sciences. 10. 1351-1363. 10.12988/ces.2017.711157.

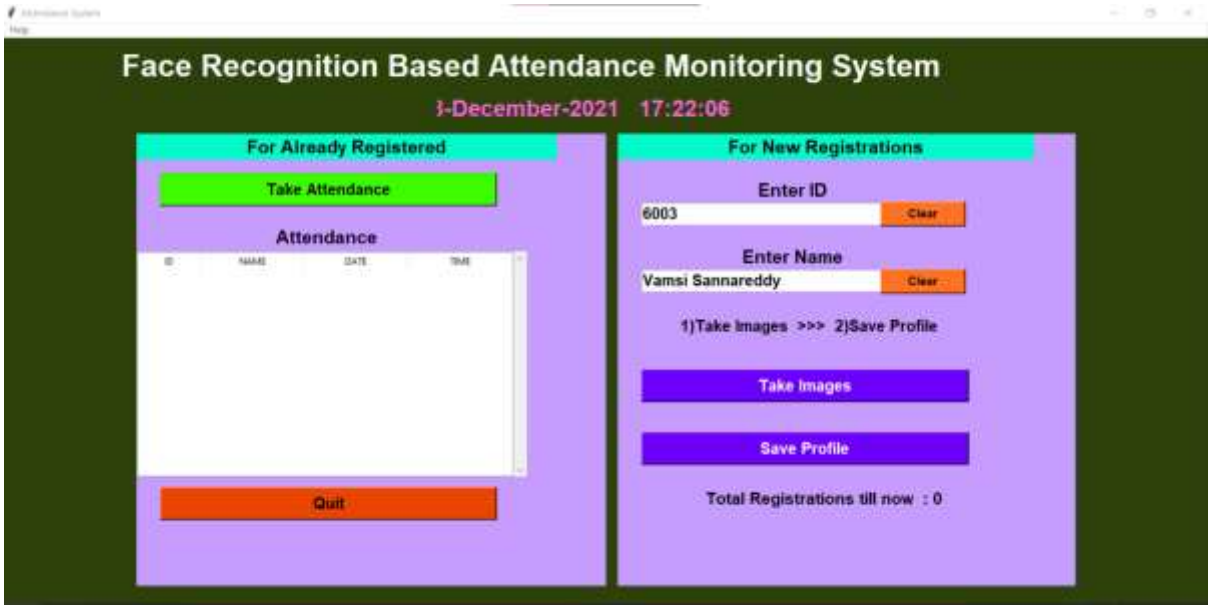
It is possible to adapt the Haar classifiers as detection and classification algorithms for faces, however, it is necessary to train a classifier for each instrument and continuously test the results to choose the best one, while with the CNN, confusion matrices are used to observe the classification of each test image with each of the categories and thus select the one with the best classification percentages.

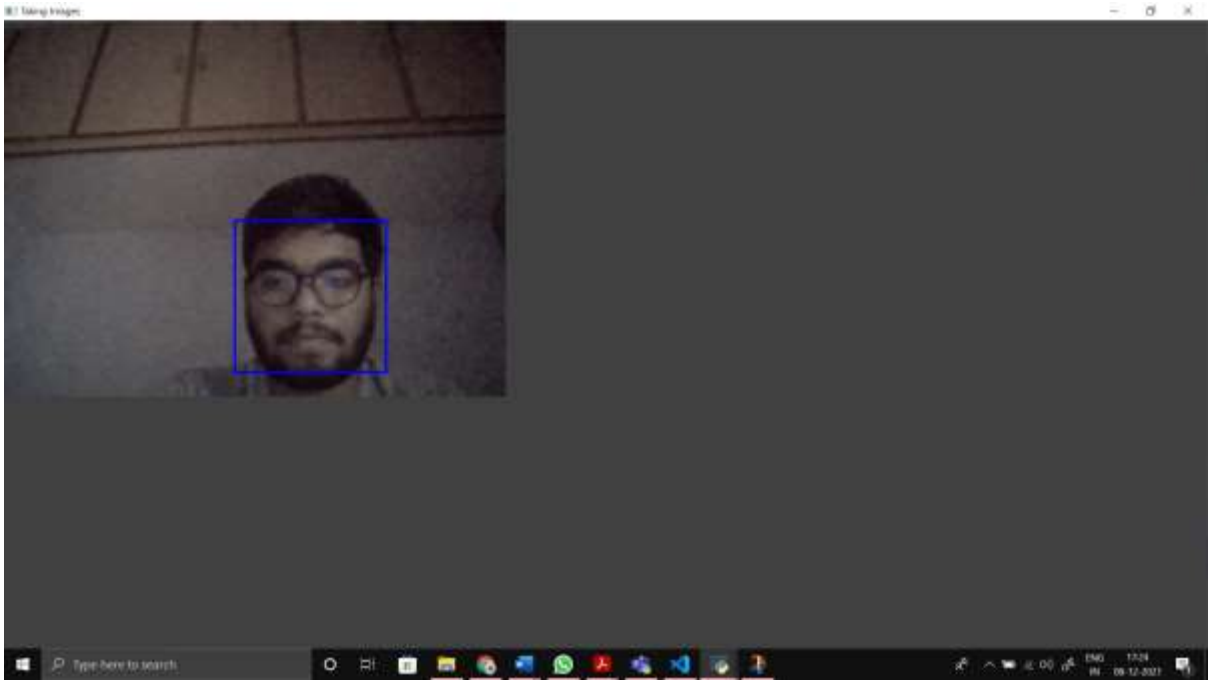
Both the CNNs and the Haar classifiers have drawbacks for the classification of tools against changes in lighting, however, the CNN detection algorithm limits the degree of light variation, given that very abrupt changes cause the entire background to be recognized as an element, while the Haar classifiers continue to recognize faces, but there may be false positives. Haar classifiers, compared to CNNs, have the advantage of having the ability to perform detection and classification simultaneously, which eliminates the need to

design an additional detection algorithm that extracts the image from the tool and enters it into a classifier, where information losses may arise due to similarities between the desired object and the background. Additionally, Haar classifiers do not require changes in their structure to improve the quality of recognition, as it does when define the architecture of a CNN, but a variation in their parameters to change the number of stages and positive and negative training images, making it easier to implement than a CNN.

7.1 EXECUTION SNAP SHOTS :







For New Registrations

Enter ID

6003 Clear

Enter Name

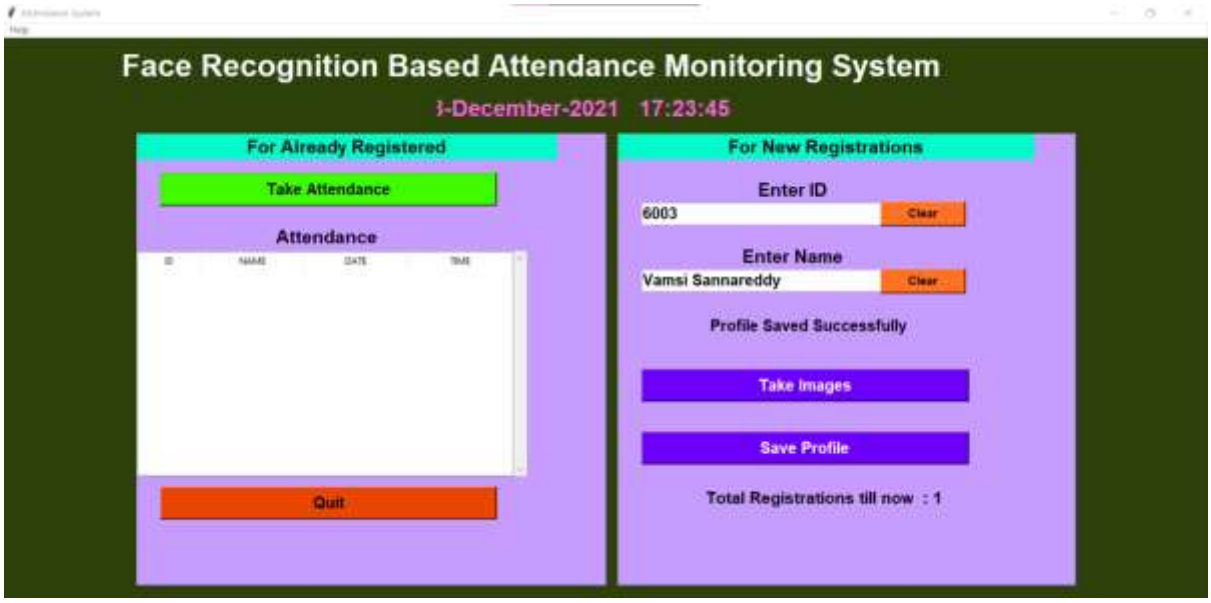
Vamsi Sannareddy Clear

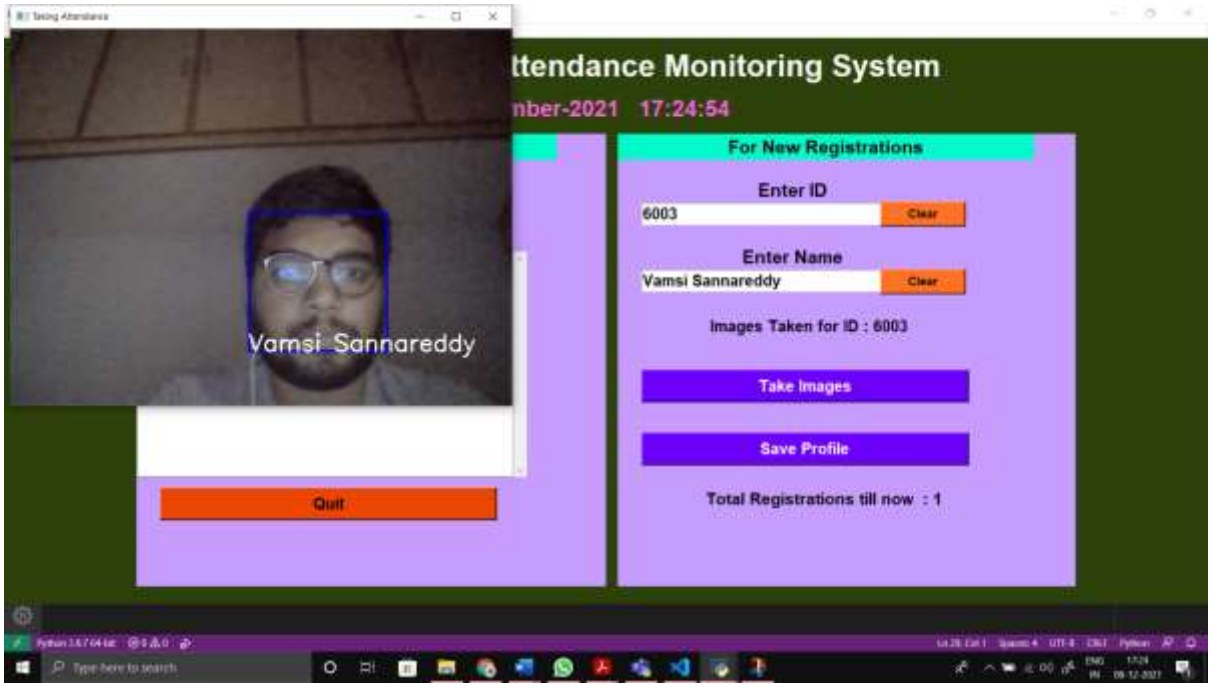
Images Taken for ID : 6003

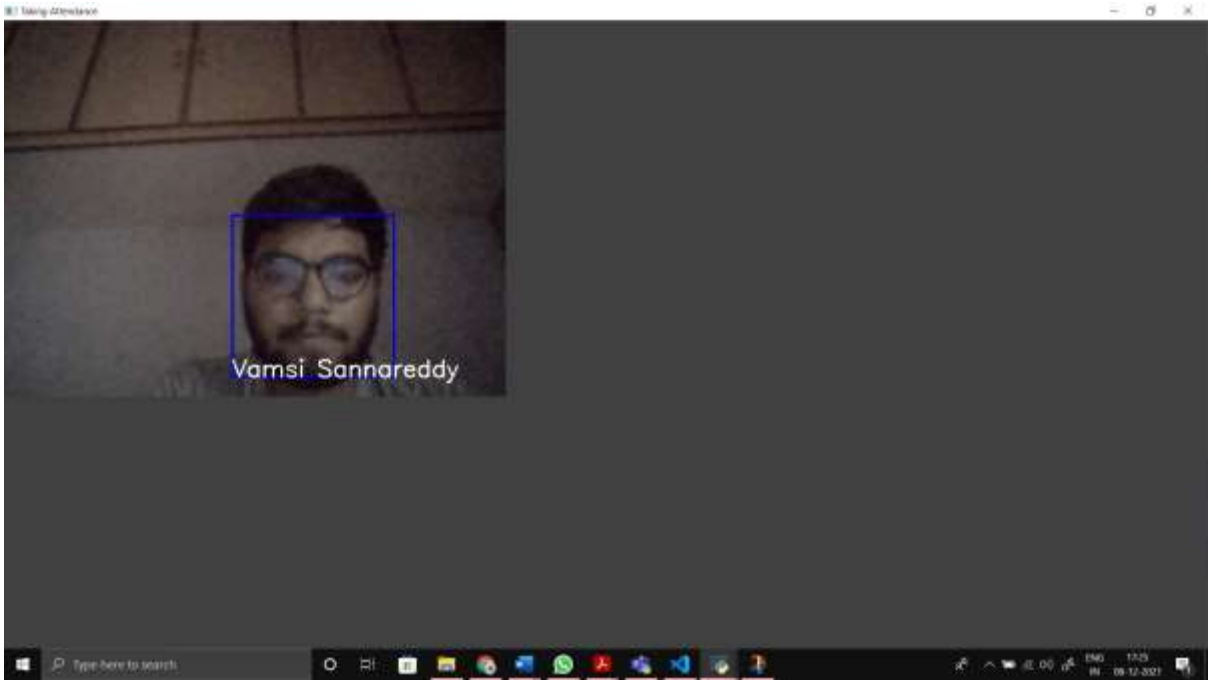
Take Images

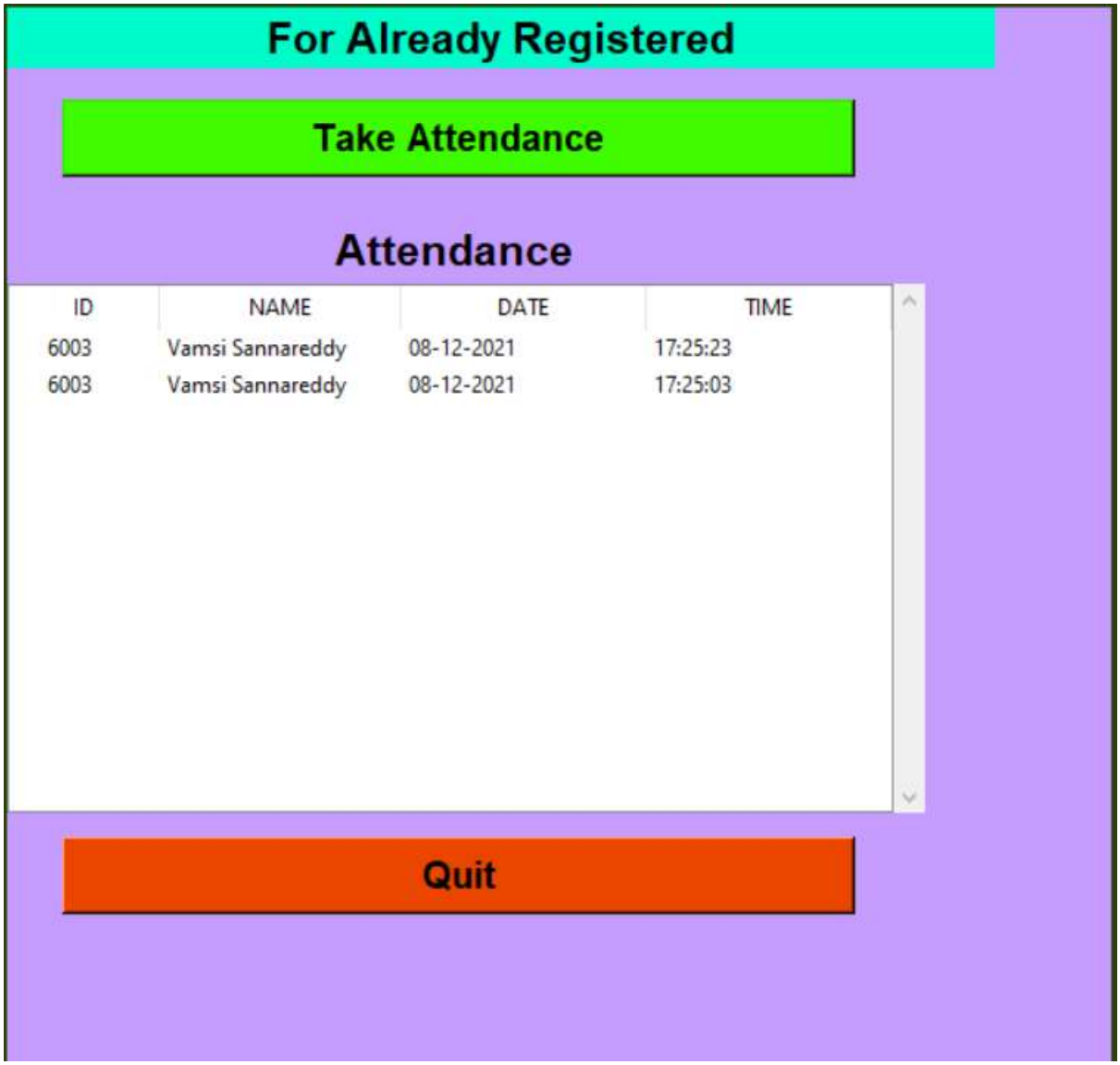
Save Profile

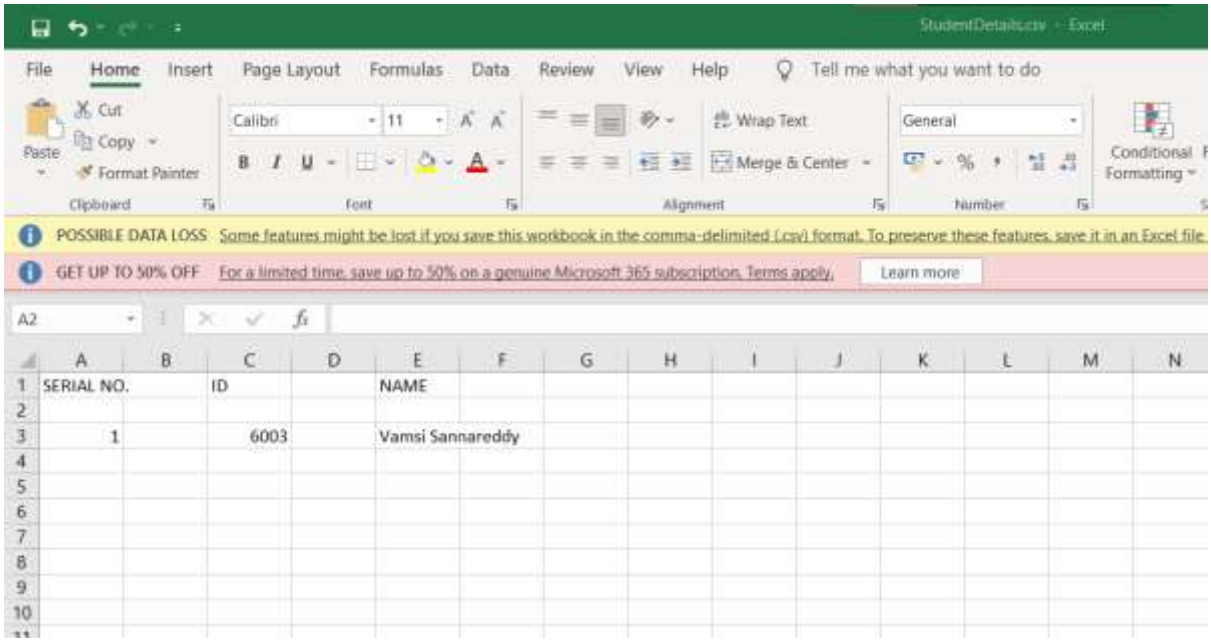
Total Registrations till now : 0

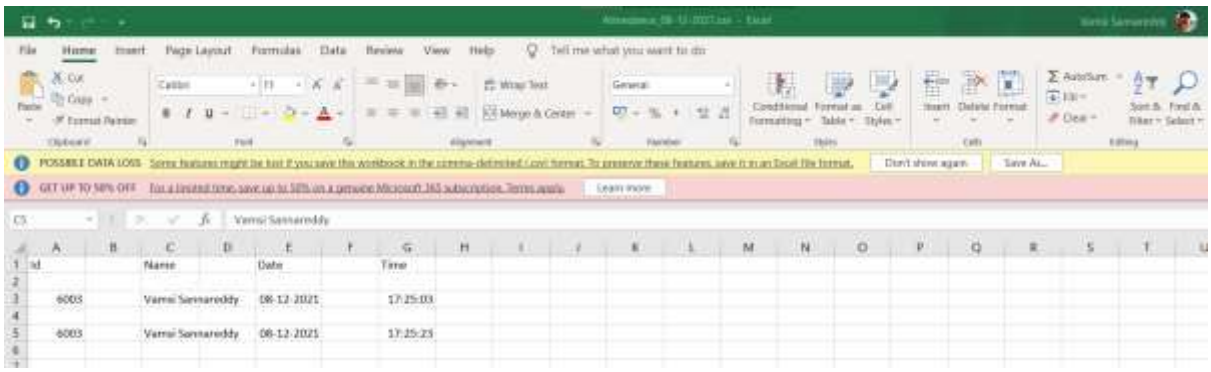

Face Recognition Based Attendance Monitoring System
3-December-2021 17:23:45
For Already Registered
Take Attendance
Attendance
ID
NAME
DATE
TIME
Quit
For New Registrations
Enter ID
6003
Clear
Enter Name
Vamsi Sannareddy
Clear
Profile Saved Successfully
Take Images
Save Profile
Total Registrations till now : 1


Taking Attendance
Vamsi_Sannareddy
ttendance Monitoring System
nber-2021 17:24:54
Quit
For New Registrations
Enter ID
6003
Clear
Enter Name
Vamsi Sannareddy
Clear
Images Taken for ID : 6003
Take Images
Save Profile
Total Registrations till now : 1
Type here to search


Vamsi Sannareddy


For Already Registered
Take Attendance
Attendance
ID
NAME
DATE
TIME
6003
Vamsi Sannareddy
08-12-2021
17:25:23
6003
Vamsi Sannareddy
08-12-2021
17:25:03
Quit


StudentDetails.csv - Excel
File Home Insert Page Layout Formulas Data Review View Help Tell me what you want to do
Cut Copy Format Painter Paste
Calibri 11
Wrap Text Merge & Center
General
Conditional Formatting
Clipboard Font Alignment Number
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.
GET UP TO 50% OFF For a limited time, save up to 50% on a genuine Microsoft 365 subscription. Terms apply. Learn more
A2
SERIAL NO. ID NAME
1 6003 Vamsi Sannareddy


Name Date Time
6003 Vamsi Sannareddy 08-12-2021 17:25:03
6003 Vamsi Sannareddy 08-12-2021 17:25:23

8. FUTURE DIRECTIVES :

This model can be used in automatic attendance filling using face recognition which supports multi face attendance marking. This can drastically improve the amount of time saved in class which is generally used for biometric or manual attendance . This model can be incorporated with cc cameras and directly mark attendance without student or teacher's knowledge. In case of model failure or any bad detection this project can work inefficiently. So, to handle this manual attendance system is also used. Data is stored in database(MySQL) and in form of a file(.csv) . In this way it makes lives easier.

9.REFERENCES :

https://ieeexplore.ieee.org/abstract/document/7006273

https://ieeexplore.ieee.org/abstract/document/8203730

https://www.semanticscholar.org/paper/REAL-TIME-FACE-DETECTION-AND-TRACKING-USING-OPENCV-Kalas/9092d42ad61751274133cd84ad99f7983879bd59?p2df