A

Project Report

On

ATTENDENCE PROJECT USING FACIAL RECOGNITION

Submitted in partial fulfillment of the requirements for the award of

DIPLOMA

IN

COMPUTER ENGINEERING

By

VELDOJ SAITEJA	**-**	**18001-CM-057**
ANGOTH SRIDHAR	**-**	**18001-CM-003**
G. DHEERAJ	**-**	**18001-CM-016**

DEPARTMENT OF COMPUTER ENGNEERING

GOVERNMENT POLYTECHNIC

MASABTANK, HYDERABAD- 500 028

TELANGANA

GOVERNMENT POLYTECNIC

MASABTANK, HYDERABAD-500 028

TELANGANA STATE

DEPARTMENT OF COMPUTER ENGNEERING

CERTIFICATE

This is to certify that the project report entitled **"ATTENDENCE PROJECT USING FACIAL RECOGNITION"** being submitted by

VELDOJ SAITEJA	-	18001-CM-057
ANGOTH SRIDHAR	-	18001-CM-003
G. DHEERAJ	-	18001-CM-016

In a partial fulfillment of the requirements for the award of Diploma in Computer Engineering

State Board of Technical Education & Training a record of bonafide work carried out by them.

The results presented in this thesis have been verified and are found to satisfactory. The results embodied in this thesis have not been submitted to any other Board the award of diploma.

Internal Guide **Head of the Department**

(M.SURESH)

Submitted for the viva voice examination held on ______________________

External Examiner

ACKNOWLEDGEMENT

As endeavor of long period can successful only with the advice of my parents and well-wishers. I now take this opportunity to express my deep gratitude and appreciation to all those who encouraged me for successful completion of the project work.

I wish to express our sincere gratitude to **Dr. N.RAJESHWARI DEVI** Ph.D. **Principal.** His consistent help and encouragement to complete the project work.

My special thanks to **Sri G.VIJAY KUMAR** M.TECH **Head of Computer Engineering Department, Government Polytechnic** during the process of project work for his timely suggestions and help in spite of his busy schedule.

I, thankful to my guide **M.SURESH** Lecturer in **Department of Computer Engineering, Government Polytechnic** for this valuable for guidance and suggestions in analyzing and testing throughout the period, till end of project work completion.

I also extend my thanks to the entire Teaching and Non-Teaching faculty of Government polytechnic, who have encouraged the throughout the course of Diploma in Computer Engineering.

Last but not least, I thank my family and all those helped me directly or indirectly for the completion of the project.

TABLES OF CONTENTS

CHAPTERS **PAGE NO**

ABSTRACT

Attendance Management System is an application developed for daily student attendance in colleges or schools. This project attempts to record attendance through face detection. The aim of the project is to create software for attendance management system that can be used by all teachers for their respective courses. The teacher will start the process of taking attendance by clicking a photograph of the class.

The algorithm used to detect face is Voila Jones. The software used to detect faces from image is MATLAB. Histogram of Gradients is used to recognize feature. And for classification the algorithm used is Error Correcting Output Coding which uses Support Vector Machine as a binary classifier.

The administrator maintains the database. Every day at the end attendance is uploaded to the system and the faculty can generate a list of defaulters.

Managing the attendance using traditional approach is really a cumbersome process. The person has to maintain the attendance record to registers and file using pen and paper. The problem with this approach is that it require lots of paper which are the part of our non-renewable natural resources. We are in the age, where we have to think about sustainable development.

In modern biometric systems the wastage of time increases as if there are many number of students. The biometric systems required internet connection and extra connections. To remove all these burdens we developed this project.

So, we also tried to bring the system which enables parent to receive the information of their ward of regularity on daily basis.

PREFACE

Introduction:

This report covers all the information about the working of the project and their benefits. In the introduction all the information about the project is given. It also informs about platform which project has been developed.

Analaysis:

Analysis describes the existing system proposed according to the needs and the feasibility.

Coding:

Coding chapter describe the logic of the coding that has been to implement the project, the code like update, insert, retrieve, session variable to identify the current user login to update and some more things has been explained.

Testing:

System testing informs the testing module by which the system has been test after testing implementation describes how the user will interact with system.

Output GUI:

In the input/output screen layout input and the output are given, which is important to insure that the result obtained from this project are correct and reliable. The next chapter about the scope of development indicates the benefits of the project.

1. INTRODUCTION

Every organization requires a robust and stable system to record the attendance of their students. and every organization have their own method to do so, some are taking attendance manually with a sheet of paper by calling their names during lecture hours and some have adopted biometrics system such as fingerprint, RFID card reader, Iris system to mark the attendance. The conventional method of calling the names of students manually is time consuming event. The RFID card system, each student assigns a card with their corresponding identity but there is chance of card loss or unauthorized person may misuse the card for fake attendance. While in other biometrics such as finger print, iris or voice recognition, they all have their own flaws and also they are not 100% accurate.

Use of face recognition for the purpose of attendance marking is the smart way of attendance management system. Face recognition is more accurate and faster technique among other techniques and reduces chance of proxy attendance. Face recognition provide passive identification that is a person which is to be identified does not to need to take any action for its identity. Face recognition involves two steps, first step involves the detection of faces and second step consist of identification of those detected face images with the existing database. There are number of face detection and recognition methods introduced. Face recognition works either in form of appearance based which covers the features of whole face or feature based which covers the geometric feature like eyes, nose, eye brows, and cheeks to recognize the face .

Our system uses face recognition approach to reduce the flaws of existing system with the help of machine learning, it requires a good quality camera to capture the images of students, the detection process is done by histogram of oriented gradient. And recognizing perform through deep learning.

So, by following the old method of marking the attendance of an individual , it’s a time taking and totally insecure process, so by using the face detection algorithm, we can totally grab the accurate rate of marking the attendance of a student.

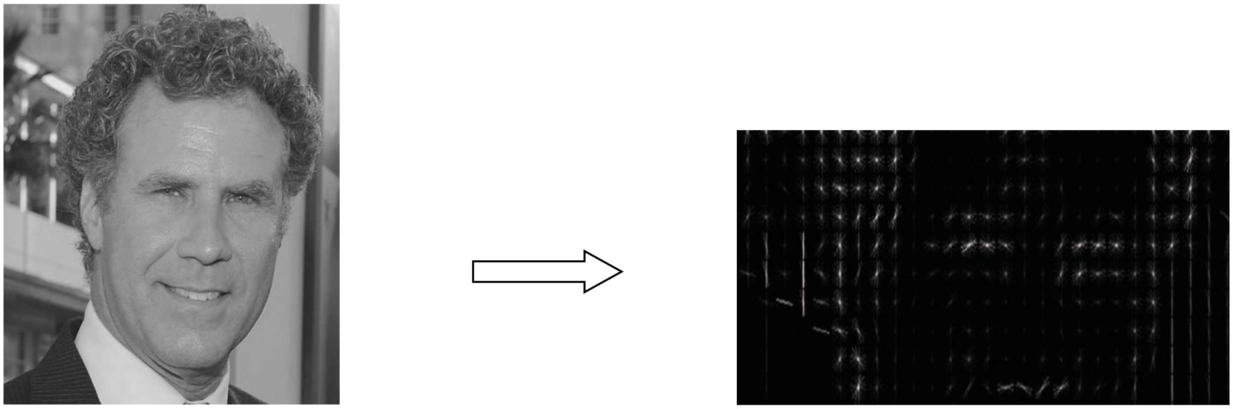
1.2. METHODOLOGY

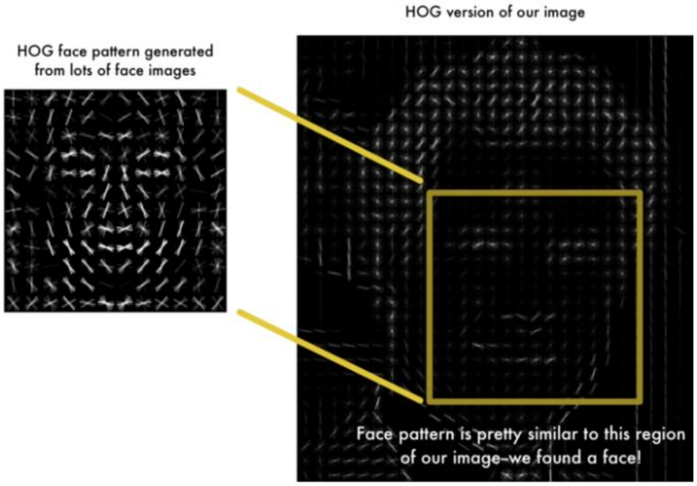
How Face Recognition works?

Step 1: Finding all the Faces

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

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

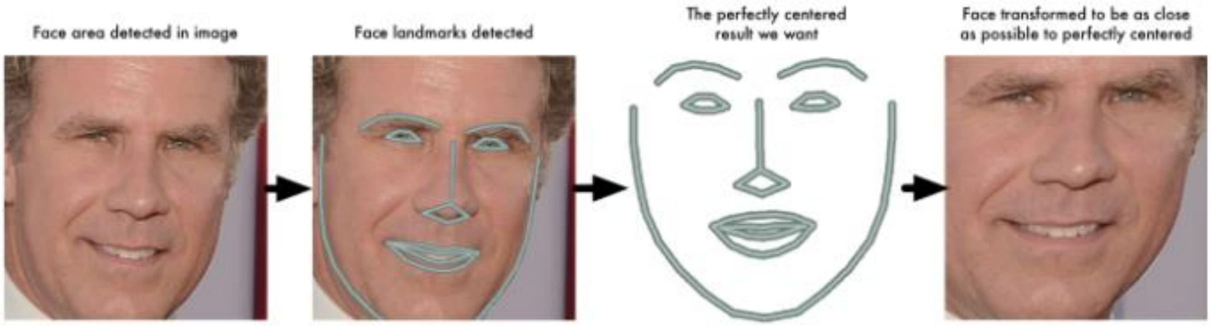



HOG version of our image
HOG face pattern generated from lots of face images
Face pattern is pretty similar to this region of our image–we found a face!

Step 2: Posing and Projecting Faces

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

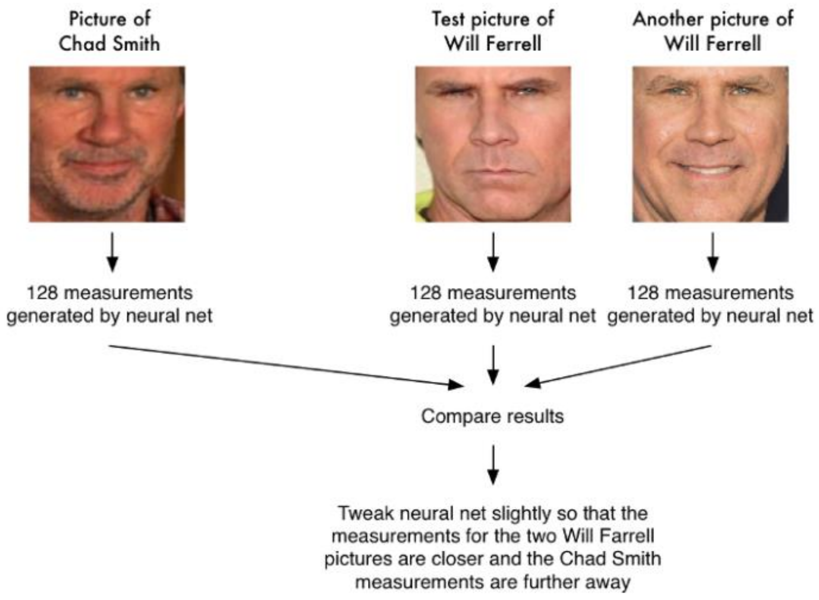
The basic idea is we will come up with 68 specific points (called *landmarks*) that exist on every face — the top of the chin, the outside edge of each eye, the inner edge of each eyebrow, etc. Then we will train a machine learning algorithm to be able to find these 68 specific points on any face


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

Step 3: Encoding Faces

What we need is a way to extract a few basic measurements from each face. Then we could measure our unknown face the same way and find the known face with the closest measurements. For example, we might measure the size of each ear, the spacing between the eyes, the length of the nose, etc.

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

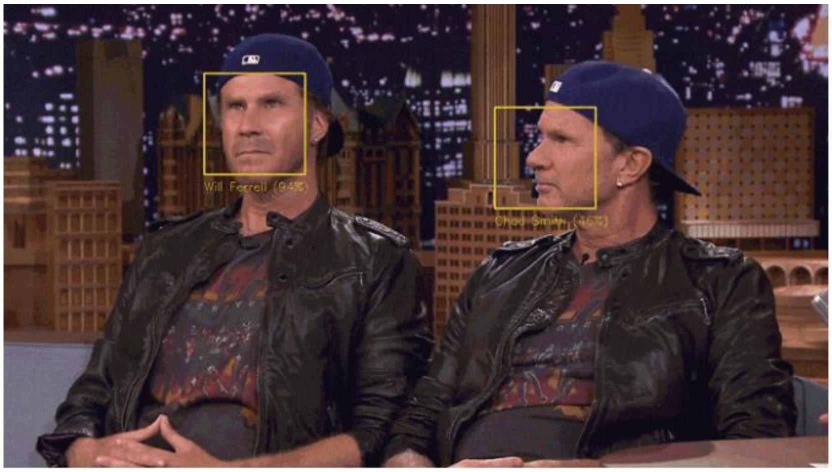

Picture of Chad Smith
Test picture of Will Ferrell
Another picture of Will Ferrell
128 measurements generated by neural net
128 measurements generated by neural net
128 measurements generated by neural net
Compare results
Tweak neural net slightly so that the measurements for the two Will Farrell pictures are closer and the Chad Smith measurements are further away

Step 4: Finding the person’s name from the encoding

This last step is actually the easiest step in the whole process. All we have to do is find the person in our database of known people who has the closest measurements to our test image.

You can do that by using any basic machine learning classification algorithm. No fancy deep learning tricks are needed.

All we need to do is train a classifier that can take in the measurements from a new test image and tells which known person is the closest match. Running this classifier takes milliseconds. The result of the classifier is the name of the person!

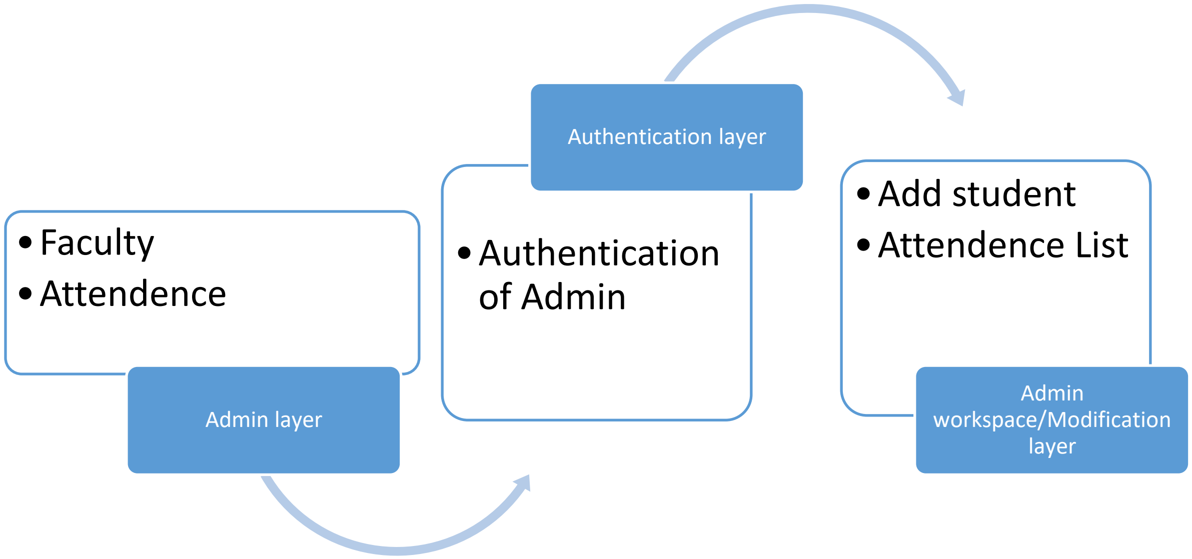


So using above procedure we can detect the face of a person within milli seconds. So coming with a questioning myself that, how it would be by creating a GUI application for attendance management using face detection?

How to implement the GUI for attendance project using Face recognition?

Basically to perform the total GUI for creating the attendance project using face recognition, we must have a complete blue print to design the interface so that we can easily do interlink every Interface diagram and know what comes next after on generating the action done on current UI(page).

These was the Background work done before implementing the project.



Admin Layer:

- This layer can be accessed by both student and faculty where student can only mark his/her attendance and faculty can only proceed with the authentication.

Authentication Layer:

- This layer can only be used/accessed by Faculty where he can login with his userid, password to get into Admin workspace layer.

Admin workspace/ modification layer:

- This layer can only be used by Faculty where they can add the new student for attendance use and they can check the Student Attendance List including the time they given there attendance.

1.3. Requirement specification

Software Requirements

IDE	:	Visual Studio Code/ Pycharm
Language	:	python 3
OS	:	windows 7 above (windows 10 recommended)

Hardware Requirements-

Processor	:	Pentium IV higher
Memory	:	500MB reserved(1GB recommended)
HDD	:	40 GB or more
Mouse	:	Logitech
Webcam	:	Standard high definition webcam

2. SOFTWARE SPECIFICATION

1.1 Face Recognition

Recognize and manipulate faces from Python or from the command line with the world's simplest face recognition library. Built using dlib's state-of-the-art face recognition built with deep learning. The model has an accuracy of 99.38% on the Labeled Faces in the Wild benchmark. This also provides a simple command line tool that lets you do face recognition on a folder of images from the command line!

1.2 OpenCV

Open CV (Open Source Computer Vision Library) is a open source computer vision software library for the purpose of machine learning. Open CV was developed to serve the purpose of computer vision applications and to stimulate the usage of machine perception in the commercially viable products. Open CV is a BSD- licensed product which is easy for the utilization and modification of the code. The library contains more than 2500 advanced algorithms including an extensive set of both typical and state-of-the-art computer vision and machine learning algorithms. These algorithms can be employed for the detection and recognition of faces, identification of objects, extraction of 3 D models of objects, production of 3 D point clouds from stereo cameras, stitching images together for production of a high resolution image of an entire scene, finding similar images from an image database, removing red eyes from images taken using flash, following eye movements, recognition of scenery and establishing markers to overlay it with intensified reality etc.

1.3 Microsoft Excel

Microsoft Excel is a spreadsheet program incorporated in Microsoft Office suite of applications. Spreadsheets prompt tables of values arranged in rows and columns that can be mathematically manipulated using both basic and complex arithmetic functions and operations. Apart from its standard spreadsheet features, Excel also extends programming support via Microsoft's Visual Basic for Applications (VBA), the capacity to access data from external sources via Microsoft's Dynamic Data Exchange (DDE) and extensive graphing and charting abilities. Excel being electronic spreadsheet program can be used to store, organize and manipulate the data. Electronic spreadsheet programs were formerly based on paper spreadsheets used for accounting purpose. The basic layout of computerized spreadsheets is more or less same as the paper ones. Related data can be stored in tables - which are a group of small rectangular boxes or cells that are standardized into rows and columns.

1.4 Tkinter

The tkinter package is a thin object-oriented layer on top of Tcl/Tk. To use tkinter, you don't need to write Tcl code, but you will need to consult the Tk documentation, and occasionally the Tcl documentation. tkinter is a set of wrappers that implement the Tk widgets as Python classes. In addition, the internal module _tkinter provides a threadsafe mechanism which allows Python and Tcl to interact.

1.5 Datetime

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.

1.6 OS

Python OS module provides the facility to establish the interaction between the user and the operating system. It offers many useful OS functions that are used to perform OS-based tasks and get related information about operating system. The OS comes under Python's standard utility modules. This module offers a portable way of using operating system dependent functionality.

1.7 Numpy

NumPy or sometimes is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. The ancestor of NumPy, Numeric, was originally created by Jim Hugunin with contributions from several other developers. In 2005, Travis Oliphant created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. NumPy is open-source software and has many contributors.

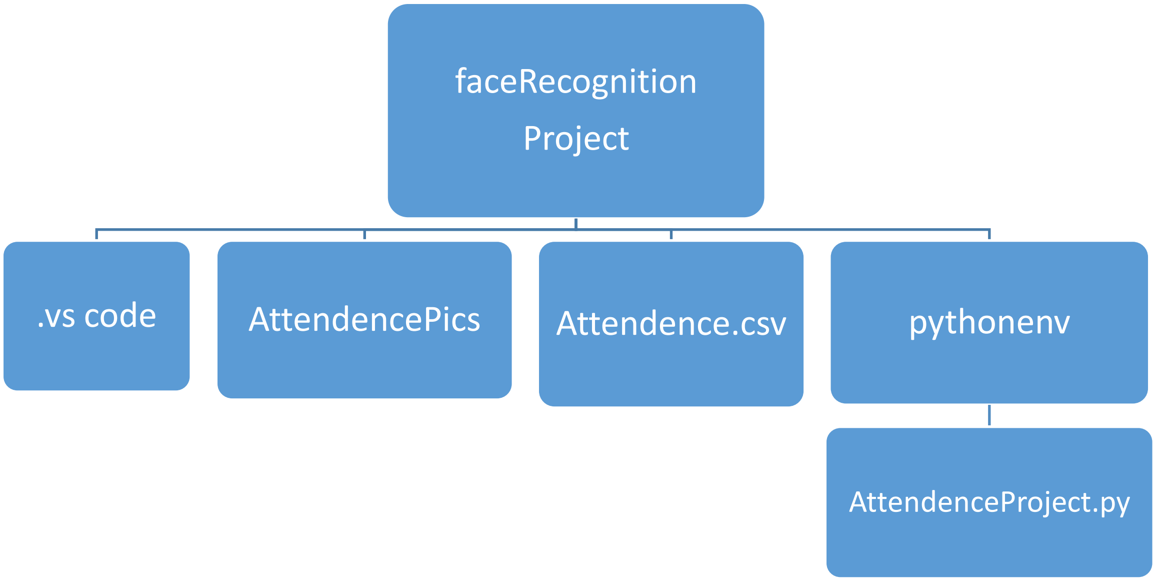
1.8 PIL(Pillow)

Pillow is built on top of PIL (Python Image Library). PIL is one of the important modules for image processing in Python. However, the PIL module is not supported since 2011 and doesn’t support python 3.

Pillow module gives more functionalities, runs on all major operating system and support for python 3. It supports wide variety of images such as “jpeg”, “png”, “bmp”, “gif”, “ppm”, “tiff”. You can do almost anything on digital images using pillow module. Apart from basic image processing functionality, including point operations, filtering images using built-in convolution kernels, and color space conversions.

3. CODING

3.1 File Directory Structure



3.2 Coding Implementation

AttendenceProject.py

```
import cv2
import numpy as np
import face_recognition
import os
from datetime import datetime
from tkinter import *
from PIL import ImageTk, Image  # file upload purpose
from tkinter import filedialog, messagebox  #  file upload purpose
from functools import partial
import csv
import sys
import subprocess, shutil


def functionCall():
    path = 'AttendencePics'
    images = []
    classNames = []
    myList = os.listdir(path)
    print(myList)
    for cl in myList:
        curImg = cv2.imread(f'{path}/{cl}')
        images.append(curImg)
        classNames.append(os.path.splitext(cl)[0])
    print(classNames)

    def markAttendance(name):
        with open('Attendence.csv', 'r+') as f:
            myDataList = f.readlines()
            nameList = []
            for line in myDataList:
                entry = line.split(',')
                nameList.append(entry[0])
            if name not in nameList:
                dtString = datetime.now().strftime('%H:%M:%S')
                f.writelines(f'\n{name},{dtString}')


    def findEncodings(images):
        r = 0
        encodeList = []
        for img in images:
            img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
            encode = face_recognition.face_encodings(img)
            if not len(encode):
                #print(classNames[r])
                messagebox.showwarning( "error while encoding",classNames[r] +
".jpg --> image can be encoded(face is not detected), deleting the image!!")
                os.remove("C:\\New_folder\\masab_tank\\facialRecognitionProjec
t\\AttendencePics\\"+classNames[r]+".jpg")
                del classNames[r]
                print(classNames)
                continue
            else:
                encode = face_recognition.face_encodings(img)[0]
            encodeList.append(encode)
            r = r + 1
        return encodeList

    encodeListKnown = findEncodings(images)
    print(len(encodeListKnown))
    #print(len(findEncodings(images)))
    print("encoding of the images completed successfully!!")

    cap = cv2.VideoCapture(0)
    i = 0
    while i < 2:
        success, img = cap.read()
        #img = captureScreen()
        imgS = cv2.resize(img,(0,0),None,0.25,0.25)
        imgS = cv2.cvtColor(imgS, cv2.COLOR_BGR2RGB)

        facesCurFrame = face_recognition.face_locations(imgS)
        encodesCurFrame = face_recognition.face_encodings(imgS,facesCurFrame)

        for encodeFace,faceLoc in zip(encodesCurFrame,facesCurFrame):
            matches = face_recognition.compare_faces(encodeListKnown,encodeFac
e)
            faceDis = face_recognition.face_distance(encodeListKnown,encodeFac
e)
            print(faceDis)
            matchIndex = np.argmin(faceDis)

            if faceDis[matchIndex]< 0.50:
                name = classNames[matchIndex].upper()
                i += 1
                markAttendance(name)
            else:
                name = 'Unknown'

            print(name)
            y1,x2,y2,x1 = faceLoc
            y1, x2, y2, x1 = y1*4,x2*4,y2*4,x1*4
            cv2.rectangle(img,(x1,y1),(x2,y2),(0,255,0),2)
            cv2.rectangle(img,(x1,y2-35),(x2,y2),(0,255,0),cv2.FILLED)
            cv2.putText(img,name,(x1+6,y2-
6),cv2.FONT_HERSHEY_COMPLEX,1,(255,255,255),2)
        cv2.imshow('OUTPUT',img)
        cv2.waitKey(1)

    cv2.destroyAllWindows()
    messagebox.showinfo("showinfo", name+" , your attendence has been marked s
uccessfully!!")

        #----------------------------------Tkinter Code ----------------------
----------------#



def openfn():
    filename = filedialog.askopenfilename(title='open')
    return filename



def save(source):
    destfile = "C:\\New_folder\\masab_tank\\facialRecognitionProject\\AttendencePics"
    source1 = source.index('.', -8, -1)
    pathmain = source[0:source1]+".jpg"
    ConvertToJPG = Image.open(source)
    Con = ConvertToJPG.convert("RGB")
    Con.save(pathmain)
    shutil.copy(pathmain, destfile)
    messagebox.showinfo("Uploaded..","Your photo has been uploaded successfully!!")



def open_img(root):
    root.destroy()
    window = Tk()
    window.iconbitmap(r'C:\New_folder\masab_tank\facialRecognitionProject\pythonenv\ProjectIcon.ico')

    window_height = 420
    window_width = 400

    screen_width = window.winfo_screenwidth()
    screen_height = window.winfo_screenheight()

    x_cordinate = int((screen_width/2) - (window_width/2))
    y_cordinate = int((screen_height/2) - (window_height/2))

    window.geometry("{}x{}+{}+{}".format(window_width, window_height, x_cordinate, y_cordinate))
    #window.geometry("400x420")
    window.resizable(0,0)
    username = StringVar()
    password = StringVar()
    username.set("admin")
    password.set("admin")
    window.configure(bg="#1D1D1D")
    def call():
```

```python
        x = openfn()
        print(x)
        img = Image.open(x)
        resized = img.resize((220, 220), Image.ANTIALIAS)
        final_resized_image = ImageTk.PhotoImage(resized)
        panel = Label(window , image=final_resized_image)
        panel.image = final_resized_image
        panel.place(x = 90, y = 90)

        Button(window , text = "Upload", command = lambda : save(x), bg = "#f59f02", font = ('Futura',15) ).place(x = 150, y = 325)

       # Button(window , text = "Close", command = lambda : close(window), bg = "#DDAF94", font = ('Futura',15) ).place(x = 210, y = 300)

    BackButton2 = Button(window , text = "Back" , command = lambda : modification(window, username, password) , height = 1, width = 10, bg = "#f59f02", font = ('Futura',10)).place(x = 0, y = 0)

    a = Button(window, text = "Please select the image", command = call, height = 1, width = 20, bg = "#f59f02", font = ('Futura',15)).place(x = 85, y = 40)
    notePoint = Label(window, text = "Note : The filename of an image should be with \n'StudentName'.'extension'", bg = "#1D1D1D",fg="#f59f02", font = ('Futura',13)).place(x = 10, y = 370)

def clear():
    filename = "C:\\New_folder\\masab_tank\\facialRecognitionProject\\Attendence.csv"
    # opening the file with w+ mode truncates the file
    messagebox.askquestion("Clear AttendenceList..", "Are you sure to clear the attendence list?")
    f = open(filename, "w+")
    messagebox.showinfo("Sucessfully", "you have successfully cleared the AttendenceList!!")
    f.close()
def Openattendence():
    os.startfile("C:\\New_folder\\masab_tank\\facialRecognitionProject\\Attendence.csv")




def StudentList(root):
    root.destroy()
    win1 = Tk()

    window_height = 400
    window_width = 400

    screen_width = win1.winfo_screenwidth()
    screen_height = win1.winfo_screenheight()

    x_cordinate = int((screen_width/2) - (window_width/2))
    y_cordinate = int((screen_height/2) - (window_height/2))

    win1.geometry("{}x{}+{}+{}".format(window_width, window_height, x_cordinate, y_cordinate))
    #win1.geometry('400x400')
    win1.title("AttendenceList")
    win1.iconbitmap(r'C:\New_folder\masab_tank\facialRecognitionProject\pythonenv\ProjectIcon.ico')
    win1.configure(bg = "#1D1D1D")
    win1.resizable(0,0)
    username = StringVar()
    password = StringVar()
    username.set("admin")
    password.set("admin")
    BackButton3 = Button(win1 , text = "Back" , command = lambda : modification(win1, username, password) , height = 1, width = 10, bg = "#f9c567", font = ('Futura',10)).place(x = 0, y = 0)

    nameHeading = Label(win1, text = "Student Name", bg = "#f9c567", font = ('Futura',13,'bold')).place(x = 70, y = 60)

    time = Label(win1, text = "Attendent Time", bg = "#f9c567", font = ('Futura',13,'bold')).place(x = 230, y = 60)

    listbox = Listbox(win1, height = 10, width = 20, bg = "#f9c567", font = ('Futura',12, 'bold'), activestyle = None)

    listbox1 = Listbox(win1, height = 10, width = 15, bg = "#f9c567", font = ('Futura',12, 'bold'), activestyle = None)

    listbox.config(highlightbackground = "black")
    listbox1.config(highlightbackground = "black")

    with open("C:\\New_folder\\masab_tank\\facialRecognitionProject\\Attendence.csv", newline='') as f:
        reader = csv.reader(f)
        data = list(reader)
        #print(data)
    divider = "___________________________________________"
    for j in range(len(data)-1):
        print(len(data))
        listbox.insert(END, data[j+1][0])
        listbox.insert(END, divider)
    listbox.place(x = 35, y = 90)
    for j in range(len(data)-1):
        listbox1.insert(END, data[j+1][1])
        listbox1.insert(END, divider)
    listbox1.place(x = 219, y = 90)
    if(len(data)==0):
        present_value = 0
    else:
        present_value = len(data)-1
    list_files = os.listdir("C:\\New_folder\\masab_tank\\facialRecognitionProject\\AttendencePics") # dir is your directory path
    number_files = len(list_files)
    total = Label(win1, text = "Total :", bg = "#f9c567", font = ('Futura',13, 'bold')).place(x = 60, y = 300)

    total_val = Label(win1, text = number_files, bg = "#1D1D1D", fg="#f59f02", font = ('Futura',13, 'bold')).place(x = 125, y = 300)

    Present = Label(win1, text = "Present :", bg = "#f9c567", font = ('Futura',13, 'bold')).place(x = 240, y = 300)

    present_val = Label(win1, text = present_value,bg = "#1D1D1D", fg="#f59f02", font = ('Futura',13, 'bold')).place(x = 330, y = 300)

    clearButton = Button(win1 , text = "Clear All" , command = lambda : clear() , height = 1, width = 15, bg = "#f9c567", font = ('Futura',13)).place(x = 50, y = 340)

    attend = Button(win1 , text = "Open in folder" , command = lambda : Openattendence() , height = 1, width = 15, bg = "#f9c567", font = ('Futura',13)).place(x = 220, y = 340)


def modification(root, user, passw):
    if(user.get()=="admin" and passw.get()=="admin"):
        root.destroy()
        sample = Tk()

        window_height = 400
        window_width = 400

        screen_width = sample.winfo_screenwidth()
        screen_height = sample.winfo_screenheight()

        x_cordinate = int((screen_width/2) - (window_width/2))
        y_cordinate = int((screen_height/2) - (window_height/2))
        sample.geometry("{}x{}+{}+{}".format(window_width, window_height, x_cordinate, y_cordinate))
        sample.iconbitmap(r'C:\New_folder\masab_tank\facialRecognitionProject\pythonenv\ProjectIcon.ico')
        #sample.geometry('400x400')
        sample.configure(bg="#1D1D1D")
        sample.title("Faculty Accessable Tools")
        sample.resizable(0,0)
        BackButton2 = Button(sample , text = "Goto Main" , command = lambda : main(sample) , height = 1, width = 10, bg = "#f59f02", font = ('Futura',10)).place(x = 0, y = 0)

        AddStudent = Button(sample, text = "Add New Student", command = lambda : open_img(sample), height = 1, width = 18, bg = "#f59f02", font = ('Futura',15)).place(x = 90, y = 100)

        AttendenceList = Button(sample, text = "Student AttendenceList", command = lambda : StudentList(sample),height = 1, width = 22, bg = "#f59f02", font = ('Futura',15)).place(x = 70, y = 200)

    else:
        root.destroy()
        useless1 = Tk()
        FVer(useless1, "Note : Invalid Username or Password!!")

def FVer(root, err):
    root.destroy()
    newWindow = Tk()

    window_height = 400
    window_width = 400

    screen_width = newWindow.winfo_screenwidth()
    screen_height = newWindow.winfo_screenheight()
    x_cordinate = int((screen_width/2) - (window_width/2))
    y_cordinate = int((screen_height/2) - (window_height/2))
    newWindow.geometry("{}x{}+{}+{}".format(window_width, window_height, x_cordinate, y_cordinate))
    newWindow.iconbitmap(r'C:\New_folder\masab_tank\facialRecognitionProject\pythonenv\ProjectIcon.ico')
    #newWindow.geometry('400x400')
    newWindow.configure(bg = "#1D1D1D")
    newWindow.title("Faculty Authentication")
    newWindow.resizable(0,0)
    username = StringVar()
    password = StringVar()
    BackButton1 = Button(newWindow , text = "Back" , command = lambda : main(newWindow) , height = 1, width = 10, bg = "#f59f02", font = ('Futura',10)).place(x = 0, y = 0)
    headingLabel = Label(newWindow, text = "Login Here", bg = "#f59f02", font = ('Futura',17,'bold')).place(x = 135, y = 40)

    usernameLabel = Label(newWindow, text = "Username", bg = "#f59f02", font = ('Futura',13)).place(x = 70, y = 102)

    usernameEntry = Entry(newWindow, textvariable = username, width = 20).place(x = 160, y = 102)

    PasswordLabel = Label(newWindow, text = "Password", bg = "#f59f02", font = ('Futura',13)).place(x = 70, y = 180)

    PasswordEntry = Entry(newWindow, textvariable = password, width = 20, show = '*').place(x = 160, y = 180)

    ErrorLabel = Label(newWindow, text = ""+err, font = ('Futura',13), fg = "red", bg = "#1D1D1D").place(x = 70, y = 225)

    Submit = Button(newWindow, text = "Submit", command = lambda: modification(newWindow, username, password), height = 1, width = 15, bg = "#f59f02", font = ('Futura',13)).place(x = 130, y = 280)

def exitfn(closing):
    closing.destroy()

def main(root):
    root.destroy()
    tkwindow = Tk()
    window_height = 400
    window_width = 400

    screen_width = tkwindow.winfo_screenwidth()
    screen_height = tkwindow.winfo_screenheight()

    x_cordinate = int((screen_width/2) - (window_width/2))
    y_cordinate = int((screen_height/2) - (window_height/2))
    tkwindow.geometry("{}x{}+{}+{}".format(window_width, window_height, x_cordinate, y_cordinate))

    tkwindow.iconbitmap(r'C:\New_folder\masab_tank\facialRecognitionProject\pythonenv\ProjectIcon.ico')

    #tkwindow.geometry('400x400')
    tkwindow.configure(bg="#1D1D1D")
    tkwindow.title('ATTENDENCE MANAGEMENT')
    tkwindow.resizable(0,0)
    Heading = Label(tkwindow, text="College Attendence Management", bg = "#f59f02", font = ('Futura',16)).place(x = 50, y = 0)

    Faculty = Button(tkwindow, text="Faculty", command=lambda : FVer(tkwindow, ""),height = 1, width = 16, bg = "#f59f02", font = ('Futura',16)).place(x = 90, y = 100)

    Attendence = Button(tkwindow, text="Attendence", command = functionCall,height = 1, width = 16, bg = "#f59f02", font = ('Futura',16)).place(x = 90, y = 200)

    Close = Button(tkwindow, text="Close", command = lambda : exitfn(tkwindow),height = 1, width = 8, bg = "#CE2424", font = ('Futura',16)).place(x = 140, y =300)

    tkwindow.mainloop()

Main_Window = Tk()
main(Main_Window)
```

4. TESTING

4.1 INTRODUCTION:

Testing is the process of analyzing a system component for finding difference between the expected behavior specified by system models and the observer behavior of the system. It is the process of detecting errors. Testing performs a very critical role for quality assurance and for ensuring the reliability of software. The results of testing are used later on during maintenance also.

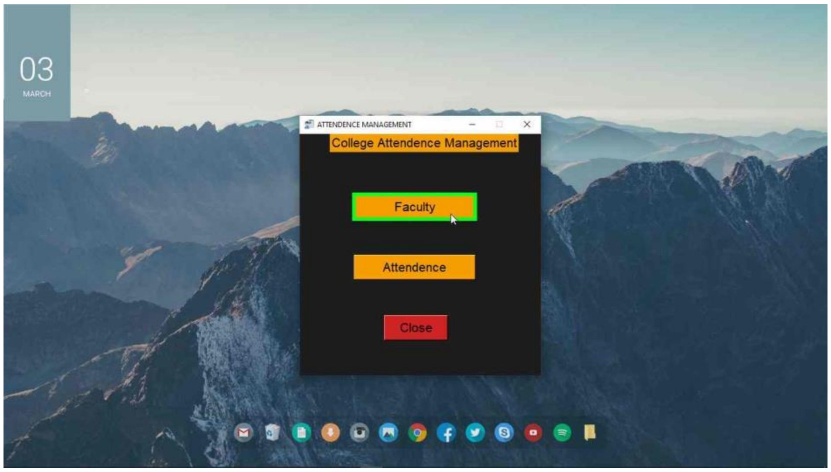
4.2 TESTING METHODS

i. White Box Testing

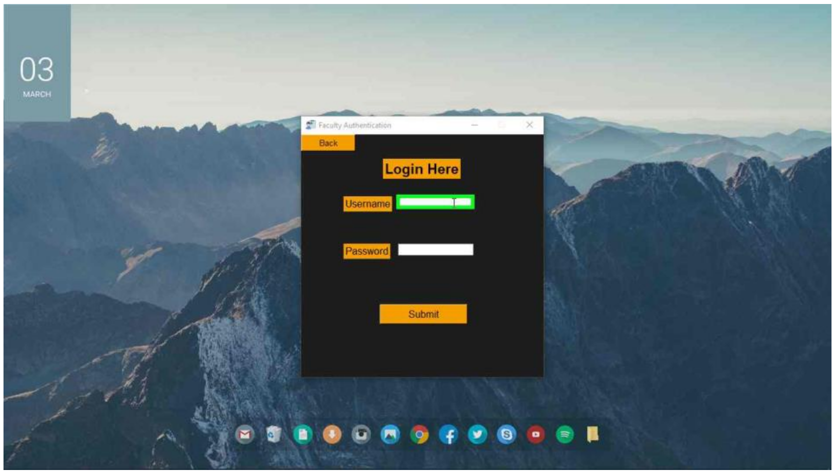
White Box Testing is a testing in which the software tester has knowledge of the inner working structure and language of the software, or at least its purpose. It is used to test areas that cannot be reached from a black box level.

<u>TEST-1</u>

- .This is the the test to check whether the Button called "Faculty" is invoking the new Authentication window or not



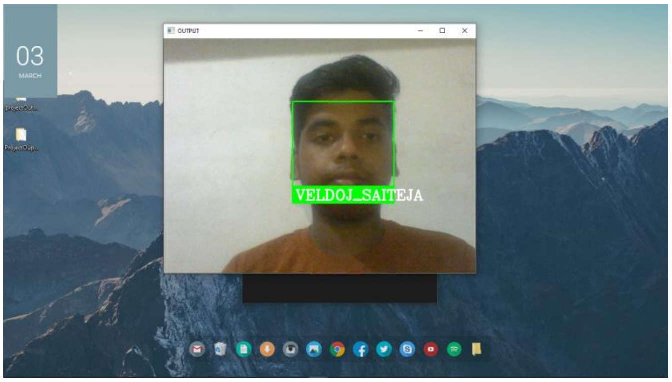
⇩

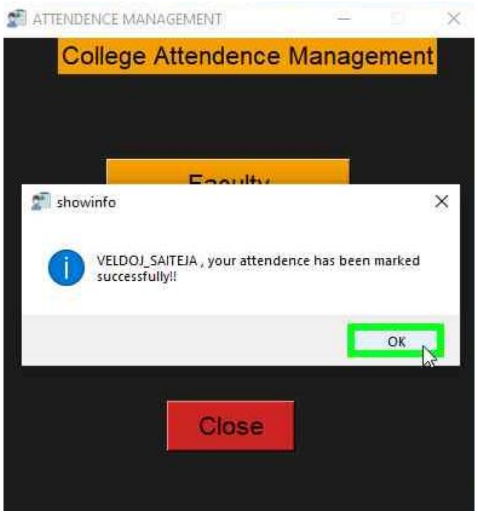


- Here it successfully invoked the Faculty Authentication window for admin login

TEST-2:

- .This is the test to check whether the Button called “Attendence” is opening the webcam of our pc to give the attendance or not.



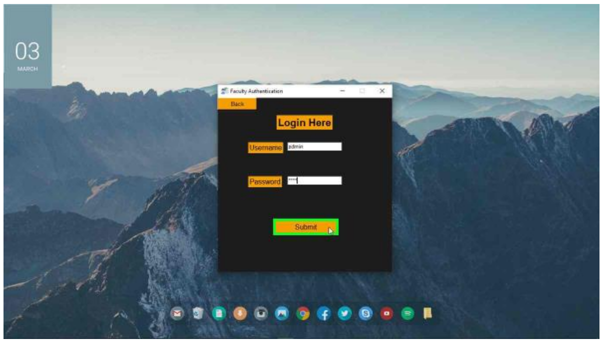


VELDOJ_SAITEJA , your attendence has been marked successfully!!

- Here it successfully mark the student attendance using his face detection.

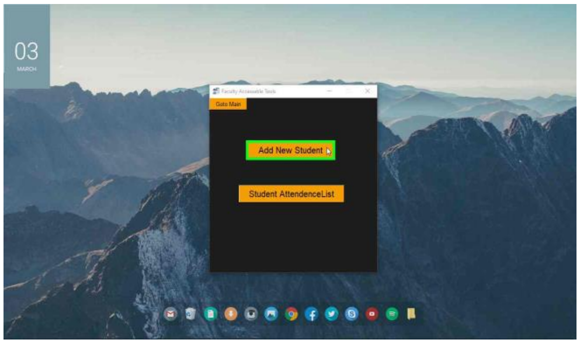
TEST-3

- This is the test for Authentication of Faculty whose been provided with Username and Password and Reflects to Faculty Accessable tools window.



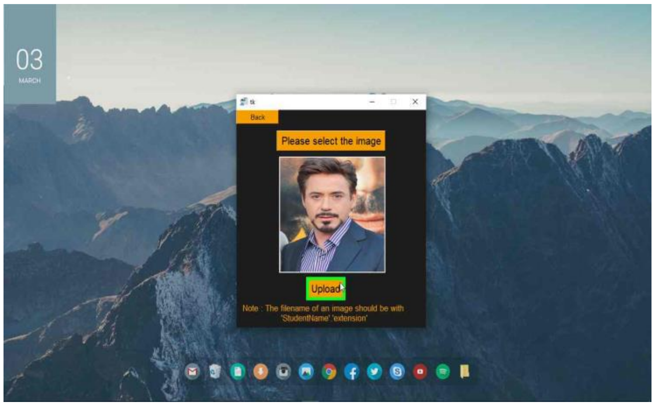
Username admin

Password ****

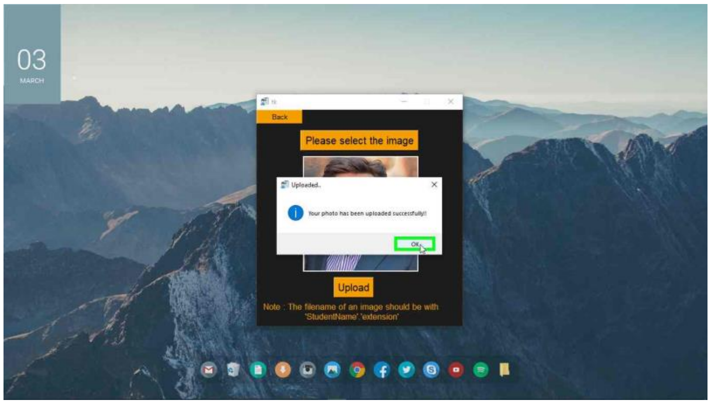


TEST-4

- This is the test to add the new Student Photo for having Attendence.



Note : The filename of an image should be with 'StudentName'.'extension'



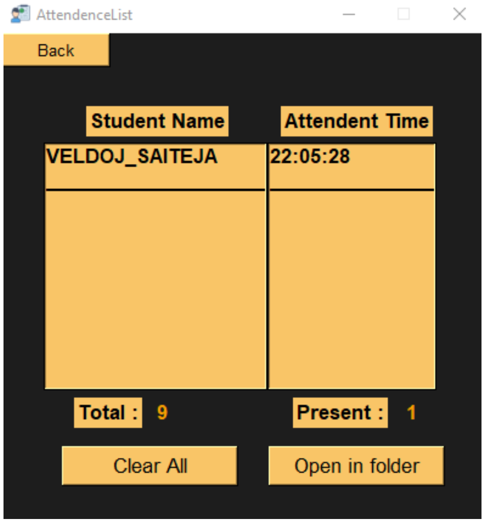
Your photo has been uploaded successfully!!

Note : The filename of an image should be with 'StudentName'.'extension'

- We have successfully uploaded the new student photo and now he can give his attendance using facial recognition

TEST-5

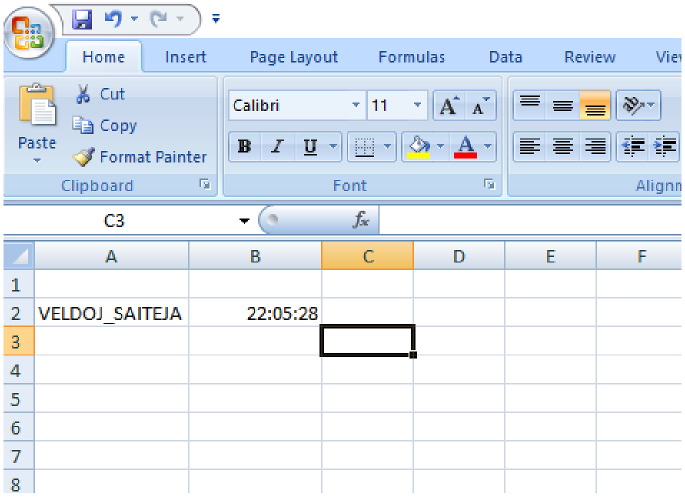
- This is the test to check whether the persons name has been recorded in the attendance sheet or not after he given his attendance using webcam.
- Here in the above **TEST-2** , the person name “VELDOJ_SAITEJA” has been given his attendance but he did not confirm that whether it marked in the attendance sheet or not.



Student Name	Attendent Time
VELDOJ_SAITEJA	22:05:28

Total : 9

Present : 1



	A	B	C	D	E	F
1						
2	VELDOJ_SAITEJA	22:05:28				
3						
4						
5						
6						
7						
8						

ii. Black Box Testing

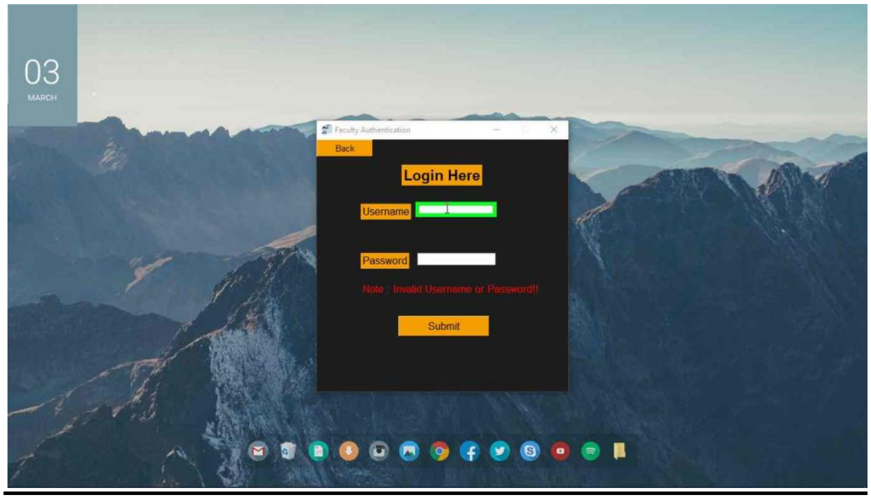
Black Box Testing is testing the software without any knowledge of the inner workings, structure and language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document. It is a testing in which the software under test is treated as a black box. You cannot "see" into it. The test provides inputs and responds to outputs without considering how the

Software works.

TEST-1

- This is the testing where the a person who don’t know any thing about the structural and programming part of the project and trying to login up without knowing correct Username and password.

Wrong Authentication

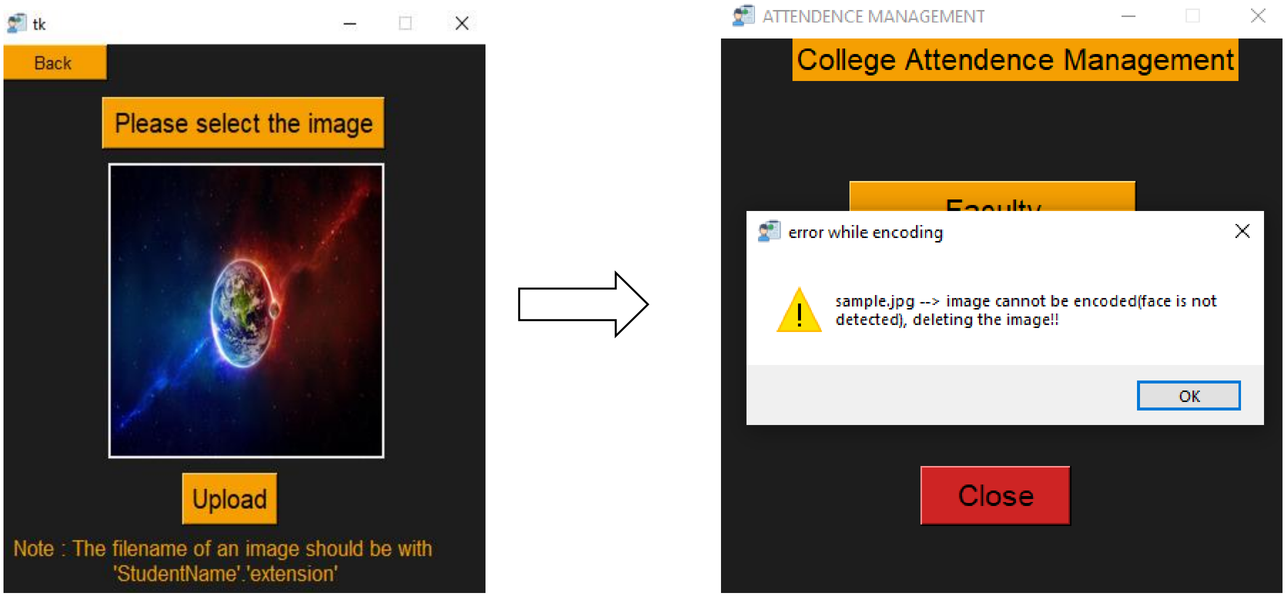


Note : Invalid Username or Password!!

- Here it successfully authenticating with the admin’s username and password.

TEST-2

- This is the testing that not to take a photo uploaded by the admin which contains no proper or No face detection image on it.



Note : The filename of an image should be with 'StudentName'.'extension'

error while encoding

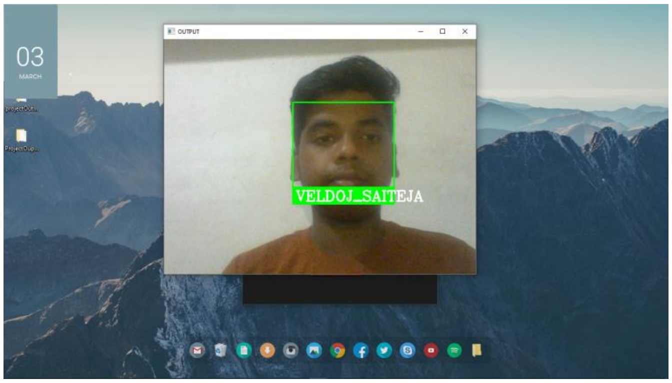
sample.jpg --> image cannot be encoded(face is not detected), deleting the image!!

iii. Load Testing

Load Testing is a non-functional software testing process in which the performance of software application is tested under a specific expected load. It determines how the software application behaves while being accessed by multiple users simultaneously. The goal of Load Testing is to improve performance bottlenecks and to ensure stability and smooth functioning of software application before deployment.

<u>TEST-1</u>

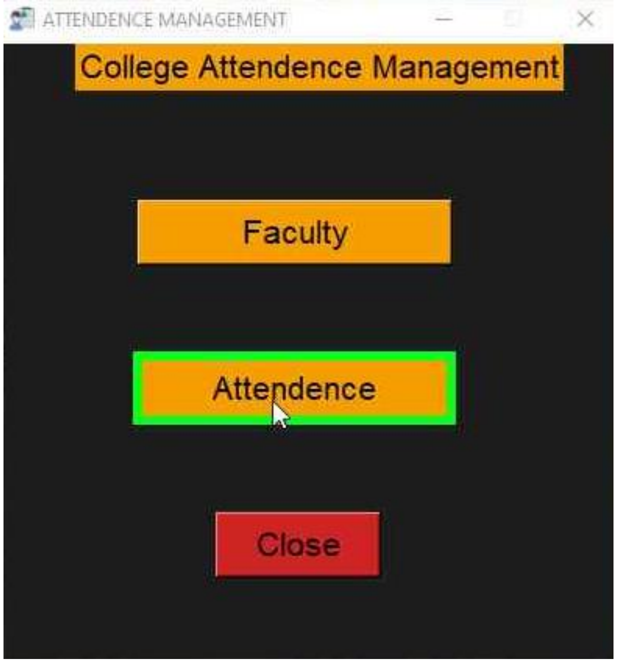
- This is the test describes the how much time it would take to recognize the face from a webcam.

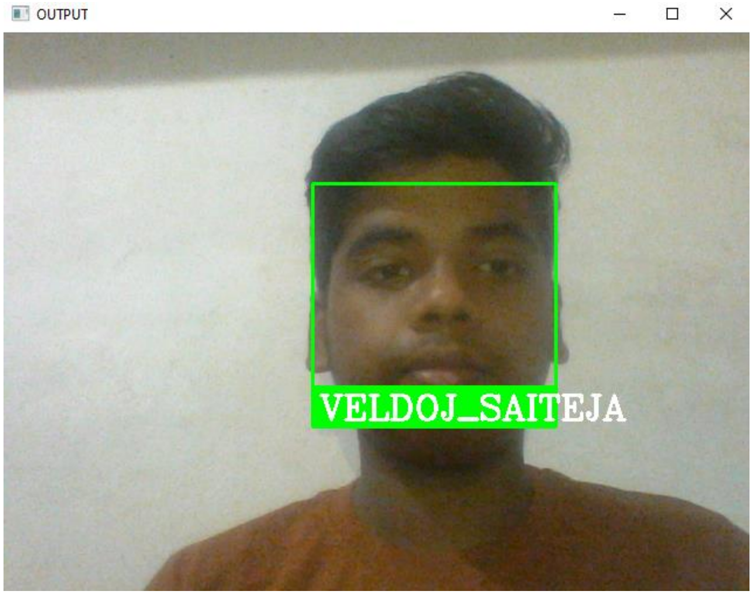


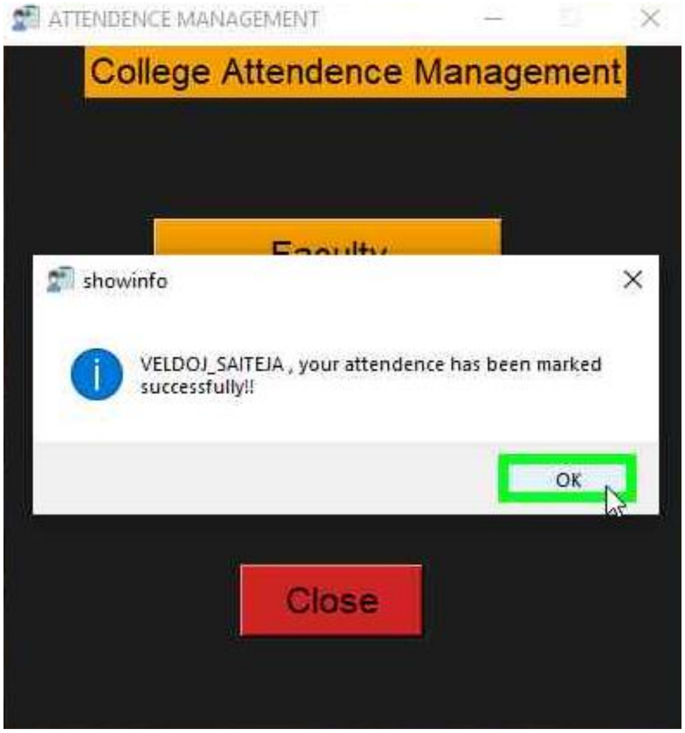
BEST CASE : 1(one) Second

WORST CASE : 8-10 seconds (based on constract and lightening)

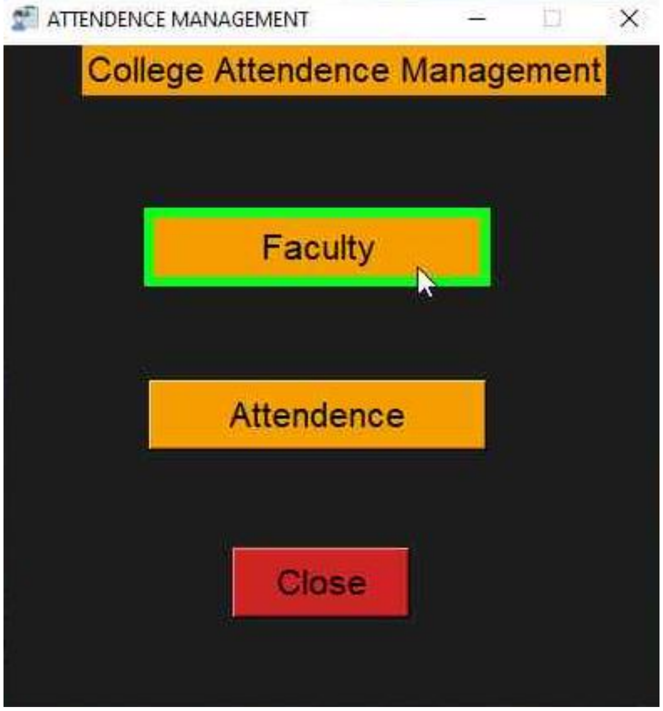
5. SCREENSHOTS

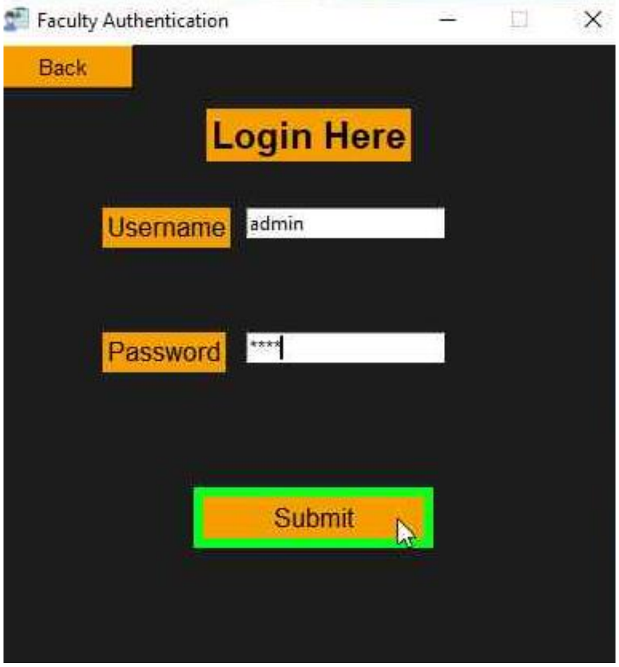






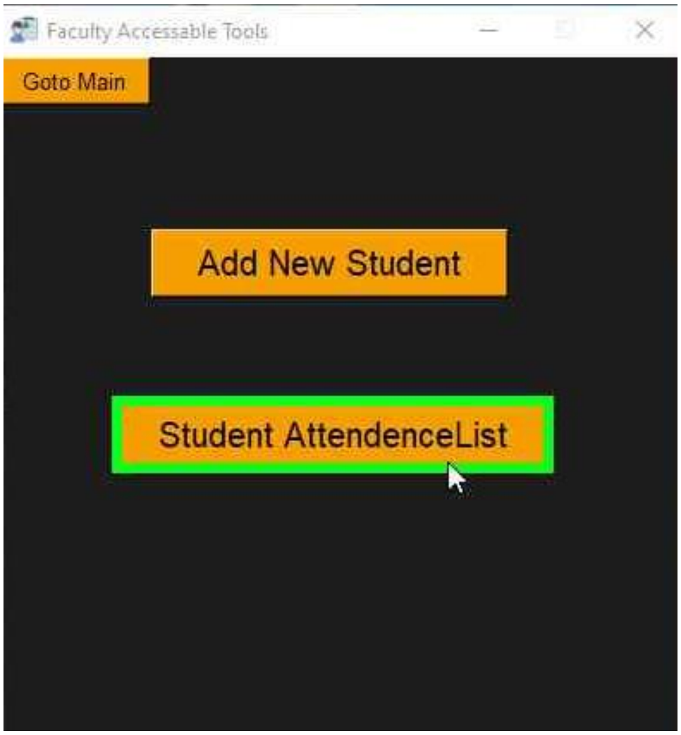
VELDOJ_SAITEJA , your attendence has been marked successfully!!

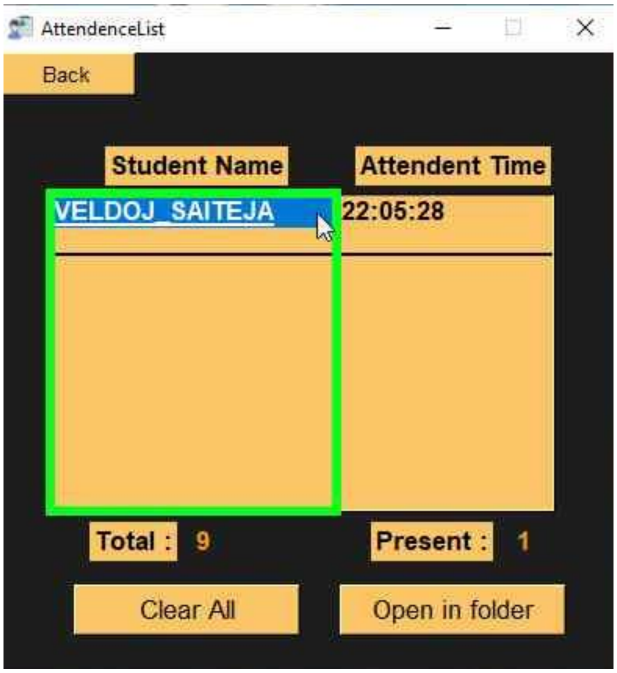




Username admin

Password ****

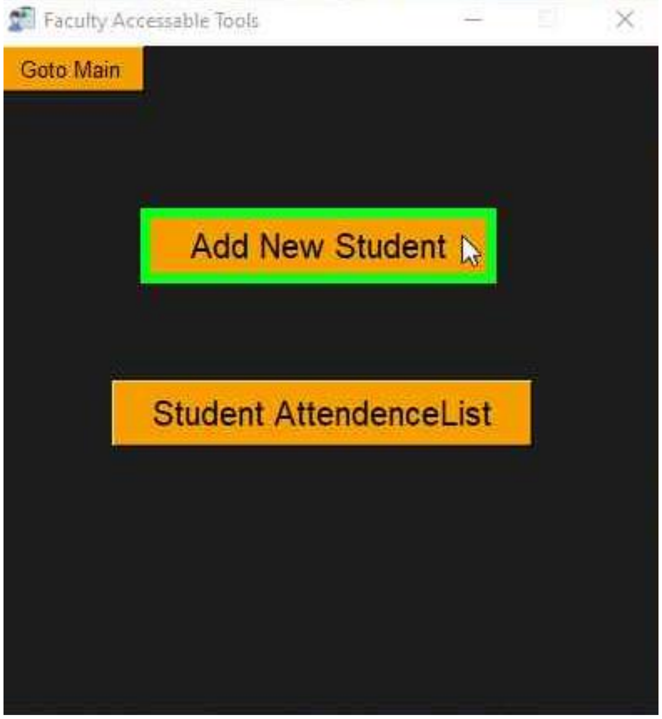


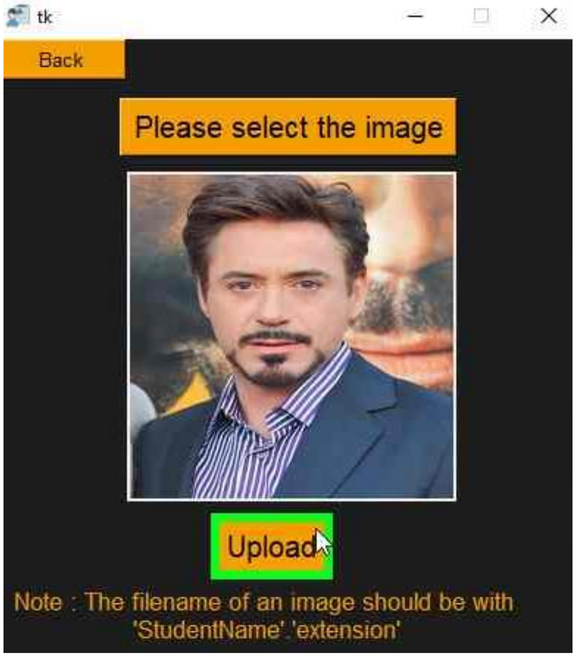


Student Name	Attendent Time
VELDOJ_SAITEJA	22:05:28

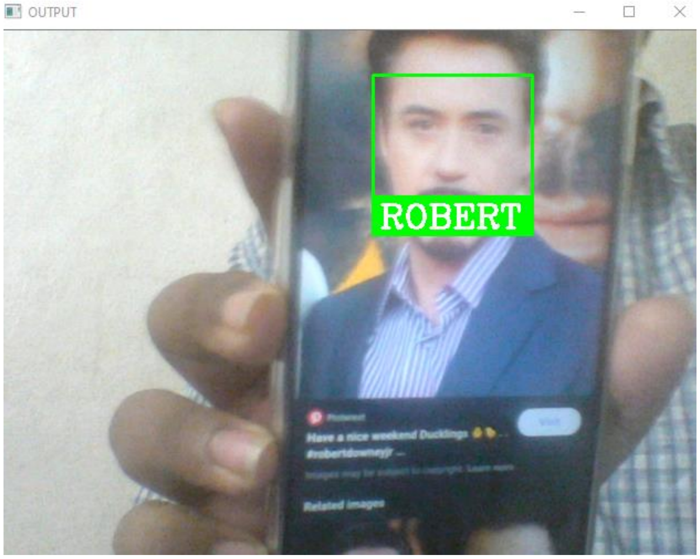
Total : 9

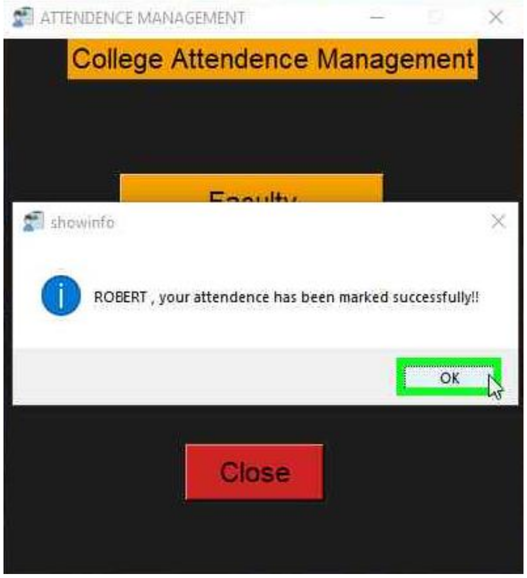
Present : 1





Note : The filename of an image should be with 'StudentName'.'extension'





ROBERT , your attendence has been marked successfully!!

7. FUTURE SCOPE

Almost all academic institutions require attendance record of students and maintaining attendance manually can be hectic as well as time consuming task. Hence maintaining attendance automatically with the help of face recognition will be very helpful and less prone to errors as compared to manual process. This will also reduce manipulation of attendance record done by students and it will save time as well. The future scope of the proposed work can be, capturing multiple detailed images of the students and using any cloud technology to store these images. The system can be configured and used in ATM machines to detect frauds. Also, the system can be used at the time of elections where the voter can be identified by recognizing the face.

6. CONCLUSION

This paper introduces the efficient method of attendance management system in the classroom environment that can replace the old manual methods. This method is secure enough, reliable, accurate and efficient. There is no need for specialized hardware for installing the system in the classroom. It can be constructed using a camera and computer. There is a need to use some algorithms that can recognize the faces in veil to improve the system performance.

Face recognition systems are part of facial image processing applications and their significance as a research area are increasing recently. Implementations of system are crime prevention, video surveillance, person verification, and similar security activities. In this approach, a face recognition based automated student attendance system is thoroughly described. The proposed approach provides a method to identify the individuals by comparing their input image obtained from recording video frame with respect to train image. This proposed approach able to detect and localize face from an input facial image, which is obtained from the recording video frame. Besides, it provides a method in pre-processing stage to enhance the image contrast and reduce the illumination effect. Extraction of features from the facial image is performed by applying both LBP and PCA. The accuracy of this proposed approach is 100 % for high-quality images, 92.31 % for low-quality images and 95.76 % of Yale face database when two images per person are trained.

The goal is reached by face detection and recognition methods. Knowledge-Based face detection methods are used to find, locate and extract faces in acquired images. Implemented methods are skin color and facial features. Neural network is used for face recognition. RGB color space is used to specify skin color values, and segmentation decreases searching time of face images. Facial components on face candidates are appeared with implementation of Log filter. Log filter shows good performance on extracting facial components under different illumination conditions. FFNN is performed to classify to solve pattern recognition problem since face recognition is a kind of pattern recognition. Classification result is accurate. Classification is also flexible and correct when extracted face image is small oriented, closed eye, and small smiled. Proposed algorithm is capable of detect multiple faces, and performance of system has acceptable good results.

8. BIBLOGRAPHY

Reference Book:

1. Python Programming – M. SURESH
2. Tkinter Programmming – GeeksforGeeks

Web Reference:

- **Websites :** https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78
- **Websites :** https://www.geeksforgeeks.org/python-programming-language/
- **YouTube:** https://www.youtube.com/watch?v=_uQrJ0TkZlc