Maharaj Vijayaram Gajapathi Raj
College Of Engineering (Autonomous)

A

Project Report On

"SMART ATTENDANCE SYSTEM"

For the Course

Artificial Intelligence Tools And Techniques

In

Bachelor Of Technology

In

Computer Science And Engineering

BY

BATCH-1

S.No	Regd.No	Name
1	20331A0517	Bethireddy Harshavardhan Reddy
2	20331A0525	Bontha Akash
3	20331A0551	Dunga Likith
4	20331A0563	Gottapu Sai Vignesh
5	20331A0532	Chepa Dhananjai Kumar

UNDER THE SUPERVISION OF

Mr.Anurag De
(Assistant Professor)

Mrs.V.Lavanya
(Assistant Professor)

CERTIFICATE

This is to certify that the project report entitled

S.No	Roll No	Name of the Students	Project Batch	Project Assigned
1.	20331A0517	B.Harshavardhan Reddy	1	Smart Attendance System
2.	20331A0525	B.Akash		
3.	20331A0532	Ch.Dhananjai Kumar		
4.	20331A0551	D.Likith		
5.	20331A0563	G.Sai Vignesh		

respectively, in partial fulfilment for the award of the degree of **"Bachelor of Technology" in Computer Science and Engineering** is a record of work done by them under my supervision during the academic year **2021-2022**.

ACKNOWLEDGEMENTS

We place on record and warmly acknowledge the continuous encouragement, invaluable supervision, timely suggestions, and inspired guidance offered by our guides **Mr.Anurag De,** Assistant Professor, **Mrs.V.Lavanya,** Assistant Professor, Department of Computer Science and Engineering MVGR College of Engineering in bringing this report to successful completion.

We consider it our privilege to express our deepest gratitude to **Dr.P.Ravi Kiran Varma**, Head of the Department for his valuable suggestions and constant motivation that greatly helped the project work to get successfully completed.

We are privileged to express our profound gratitude to **Dr.K.V.L.Raju**, Associate Professor, and Principal for extending his utmost support and cooperation in providing all the provisions for the successful completion ofthe project. We sincerely thank all the members of the staff in the Department of Computer Science & Engineering for their sustained help inour pursuits. We thank all those who contributed directly or indirectly to successfully carrying out this work.

Head Of Department	**Supervisor**	**Supervisor**
Dr. P.Ravi Kiran Varma	**Mr.Anurag De**	**Mrs.Lavanya**
CSE	**CSE**	**CSE**

TABLE OF CONTENT:

ABSTRACT:

Every working area whether its professional, industrial, or educational requires an attendance report. Conventionally, this report is maintained manually through physical means i.e., pen-paper. So if the amount of concerned attendants increases, then, withholding to such attendance procedure will be a tedious job and might result in over-consumption of time. These methods often constitute of human errors resulting in non-verified attendance marking. In recent years, after the advancement of automated environments, many perceptions with different technologies were proposed for instance, biometrics via fingerprint detection, iris detection or by using barcode as an ID. So the idea in fabricating the below project is to generate a time efficient, cost efficient as well as error free mechanism by using real time face detection and updating the attendance automatically inside the excel file. The software constitutes the dataset of students with their images which can be readily edited as well as updated. These images can be uploaded by the user and the mentioned algorithm detects the faces and compares it to the student image dataset in the recognition phase. The corresponding attendance is thus fetched to the excel file. This system rectifies the complications in physical record maintenance and results in effortless yielding of attendance.

INTRODUCTION:

Face recognition is an important application of Image processing owing to its use in many fields. Identification of individuals in an organization for the purpose of attendance is one such application of face recognition. Maintenance and monitoring of attendance records plays a vital role in the analysis of performance of any organization. The purpose of developing attendance management system is to computerize the traditional way of taking attendance. Automated Attendance Management System performs the daily activities of attendance marking and analysis with reduced human intervention. There are many other ways of computerizing the attendance process using other biometric techniques as mentioned below:

1. Signature based System

2. Fingerprint based System

3. Iris Recognition

4. RFID based System

5. Face Recognition

Among all the above techniques the facial recognition technique is unique, efficient, accurate and affordable system. There are many different sub problems in the system which is mentioned step-by-step below.

1. Capture a picture and detect all the faces from it.

2. Focus on a single face and understand it even it

turned into the different direction or bad lighting, still it is a same person.

3. Note the unique characteristics of the face which will help to distinguish it from all other images. The Characteristics can be the nose, depth of eyes, dimensions of face, color of skin etc.

The human brain is capable of recognizing the faces very quickly. The computers can also be implemented to recognize the uniqueness of the faces, so we need to program or make the computer to learn that how to differentiate the faces by their unique characters. The Facial recognition can be divided into two categories as mentioned below:

1. Verification

2. Identification

- Verification is a method of matching of one on one i.e. match or no match. The method can be used for locking and unlocking systems, mobiles etc.
- Identification is a method of identifying a person from a set of people i.e. one on N people.

IMPLEMENTATION AND METHODOLOGY:

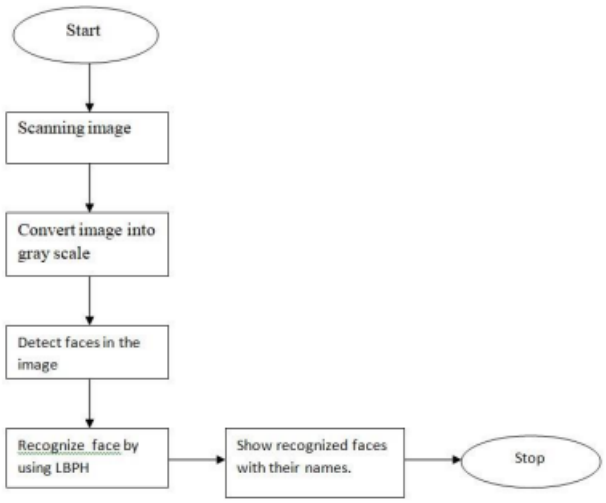
Face recognition can be separated into two significant parts: one is face detection process and other is recognition using feature extraction.

Create Dataset

The first step is to create datasets for different faces that will be used in attendance system. Camera is used for detecting faces and taking images of frontal faces. If number of images are more captured then more will be accuracy level(approx…200). These pictures are put away in an excel file with enrolment ID and name.

Training Process

It begins with traversing of the training data. After the image is captured, the image is converted into grayscale. Then we get an image a window of 3x3 pixels from image which is converted into grayscale. We consider centre pixel as a threshold. If neighbour pixels value exceeded threshold value then put 1 otherwise 0. After concatenating binary value in clockwise direction we get binary value then convert into decimal. This decimal value of image plotted into histogram. Each block of an image is converted into histogram and then concatenate them. After this process images are resized and converted into NumPy array. Faces detected in the image and creating separate list for each image. Faces are affixed with them alongside their individual ID's. Training process are done with their respective ID’s.



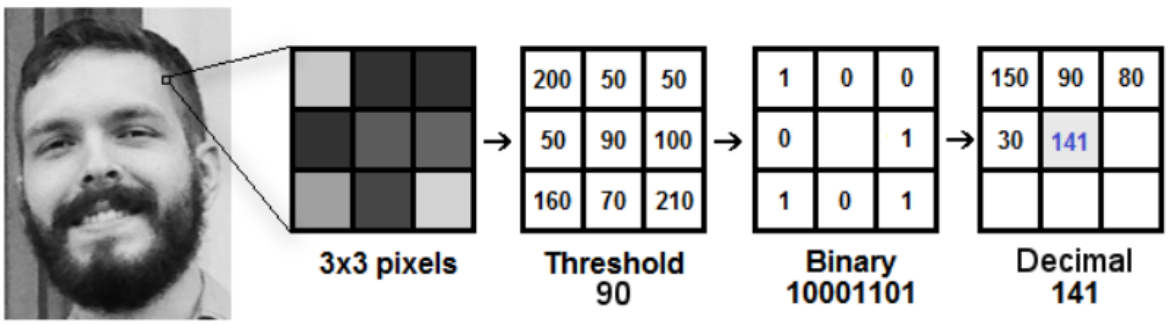
Face Detection - The trained images data is stored in file .yml format. Haar Cascade classifier access those data and detect faces. After detecting, a box is drawn on those faces. The next step in this framework is face recognition.

Face Recognition - After detection, it extracts the features using LBPH then these facial features compared with trained data set. After matching facial features, the image is recognized.

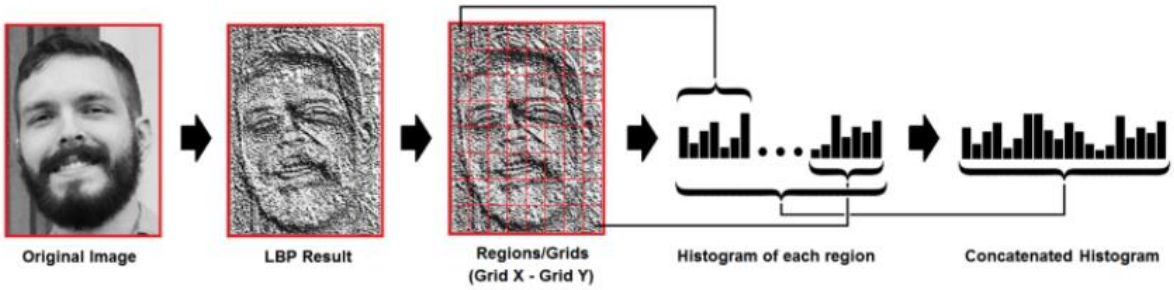
LBPH ALGORITHM –

- For training part this algorithm needs the gray scale Image. It was proposed in 2006.
- LBPH = LBP + HOG(DESCRIPTOR)
- Parameters(radius, neighbours, Grid X, Grid Y)

1. **Radius :** The radius is used to build the circular local binary pattern and represents the radius around the central pixel. It is usually set to 1.
2. **Neighbours:** The number of sample points to build the circular local binary pattern. Keep in mind: the more sample points you include, the higher the computational cost. It is usually set to 8.
3. **Grid X:** The number of cells in the horizontal direction. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector. It is usually set to 8.
4. **Grid Y:** The number of cells in the vertical direction. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector. It is usually set to 8.



Extracting the features(Using Grid X and Grid Y parameters)



Requirements:

Hardware Requirements:

- Processor: Intel
- Webcam
- Ram: 512 MB or more

Software Requirements:

- Python programming language
  Modules:

  1. OpenCV (pip install opencv-python)
  2. Numpy (pip install numpy)
  3. PIL (pip install pillow)
  4. OS (preinstalled)
  5. datetime (preinstalled)
  6. Kivy (pip install kivy) {For GUI}

  Note: Also install (pip install opencv-contrib-python)

Program:

```
import threading
from functools import partial
from kivy.clock import Clock
from kivy.graphics.texture import Texture
from kivy.app import App
from kivy.lang import Builder
from kivy.uix.screenmanager import ScreenManager, Screen
import cv2
import numpy as np
import os
from datetime import datetime
from PIL import Image
from kivy.core.window import Window
import pandas as pd
Window.clearcolor = (.8, .8, .8, 1)
class AttendenceWindow(Screen):
    pass
class DatasetWindow(Screen):
    pass
class WindowManager(ScreenManager):
    pass
kv = Builder.load_file("my.kv")
class MainApp(App):
    running = False
    Dir = os.path.dirname(os.path.realpath(__file__))
    def build(self):
        self.icon = self.Dir + '/webcam.ico'
        self.title = 'Face Detection Attendance System'
        return kv
    def break_loop(self):
```

```
    self.running = False
def startAttendence(self):
    threading.Thread(target=self.Attendence, daemon=True).start()
def startTrain(self):
    threading.Thread(target=self.train, daemon=True).start()
def startDataset(self):
    threading.Thread(target=self.dataset, daemon=True).start()
def StudentList(self):
    os.startfile(self.Dir + '/list/students.csv')
def AttendanceList(self):
    os.startfile(self.Dir + '/Attendance/Attendance.csv')
def Attendence(self):
    self.running = True
    dataset_path = path = os.path.join(self.Dir, 'dataset')
    if not (os.path.isdir(dataset_path)):
        os.mkdir(dataset_path)
    try:
        user_id = int(kv.get_screen('main').ids.user_id.text)
        now = datetime.now()
        date_time = now.strftime("%d/%m/%Y %H:%M:%S")
        date = now.strftime("%d/%m/%Y")
        eye = cv2.CascadeClassifier(self.Dir + '/haarcascade_eye.xml')
        recog = cv2.face.LBPHFaceRecognizer_create()
        recog.read(self.Dir + '/trainer/trainer.yml')
        face = cv2.CascadeClassifier(self.Dir + '/haarcascade_frontalface_default.xml')
        font = cv2.FONT_HERSHEY_DUPLEX
        rec = 0
        id = 0
        face_numbers = 5
        camera = cv2.VideoCapture(0)
        camera.set(3, 1920)
        camera.set(4, 1080)
        minWidth = 0.001*camera.get(3)
```

```
minHeight = 0.001*camera.get(4)
blink = 0
is_eye = False
while self.running:
    rtrn, image=camera.read()
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    faces = face.detectMultiScale(
        gray,
        scaleFactor = 1.3,
        minNeighbors = face_numbers,
        minSize = (int(minWidth), int(minHeight)),
    )
    eyes = eye.detectMultiScale(image,scaleFactor = 1.2, minNeighbors = 5)
    for (x, y, w, h) in eyes:
        cv2.rectangle(image, (x, y),
                (x + w, y + h), (255, 0, 0), 1)
    if len(eyes) >= 2:
        is_eye = True
        cv2.putText(image, "eye detected", (50,50), font, 1, (0,255,0), 1)
    if(len(faces)==0):
        blink = 0
    if len(eyes) < 2:
        blink+=1
    cv2.putText(image, "Blink(16+) : {}".format(blink), (1020,50), font, 1, (0,0,255), 2)
    for(x,y,w,h) in faces:
        id, match = recog.predict(gray[y:y+h,x:x+w])
        if (id == user_id) and (match < 35):
            rec = 1
            cv2.rectangle(image, (x,y), (x+w,y+h), (0,255,0), 2)
            status = "Attandance Recorded"
            cv2.putText(image, str(status), (x,y+h+25), font, 1, (0,255,0), 1)
            try:
                df = pd.read_csv(self.Dir + '/list/students.csv')
```

```
                name = df.loc[df['id'] == id, 'name'].iloc[0]
            except:
                name = "Unknown"
            match = "  {0}%".format(round(100 - match))
        else:
            rec = 0
            cv2.rectangle(image, (x,y), (x+w,y+h), (0,0,255), 2)
            status = "Attandance Not Recorded"
            cv2.putText(image, str(status), (x,y+h+25), font, 1, (0,0,255), 1)
            name = "unknown"
            match = "  {0}%".format(round(100 - match))
        cv2.putText(image, str(name), (x+5,y-5), font, 1, (255,255,255), 2)
        cv2.putText(image, str(match), (x+5,y+h-5), font, 1, (255,255,0), 1)
    Clock.schedule_once(partial(self.display_frame, image))
    k = cv2.waitKey(1)
    if k == 27:
        break
if rec==1 and blink >15:
    df = pd.read_csv(self.Dir + '/Attendance/Attendance.csv')
    coll = ['0']*len(df['id'])
    if date in df.columns:
        if (int(df.loc[df['id'] == id, date].iloc[0]))==0:
            df.loc[df['id'] == id, date]=1
            df.to_csv(self.Dir + '/Attendance/Attendance.csv', index=False)
            kv.get_screen('main').ids.info.text = "Attendence entered successfully."
        else:
            kv.get_screen('main').ids.info.text = "Attendence already exist."
    else:
        df[date] = coll
        df.loc[df['id'] == id, date]=1
        df.to_csv(self.Dir + '/Attendance/Attendance.csv', index=False)
        kv.get_screen('main').ids.info.text = "Attendence entered successfully."
else:
```

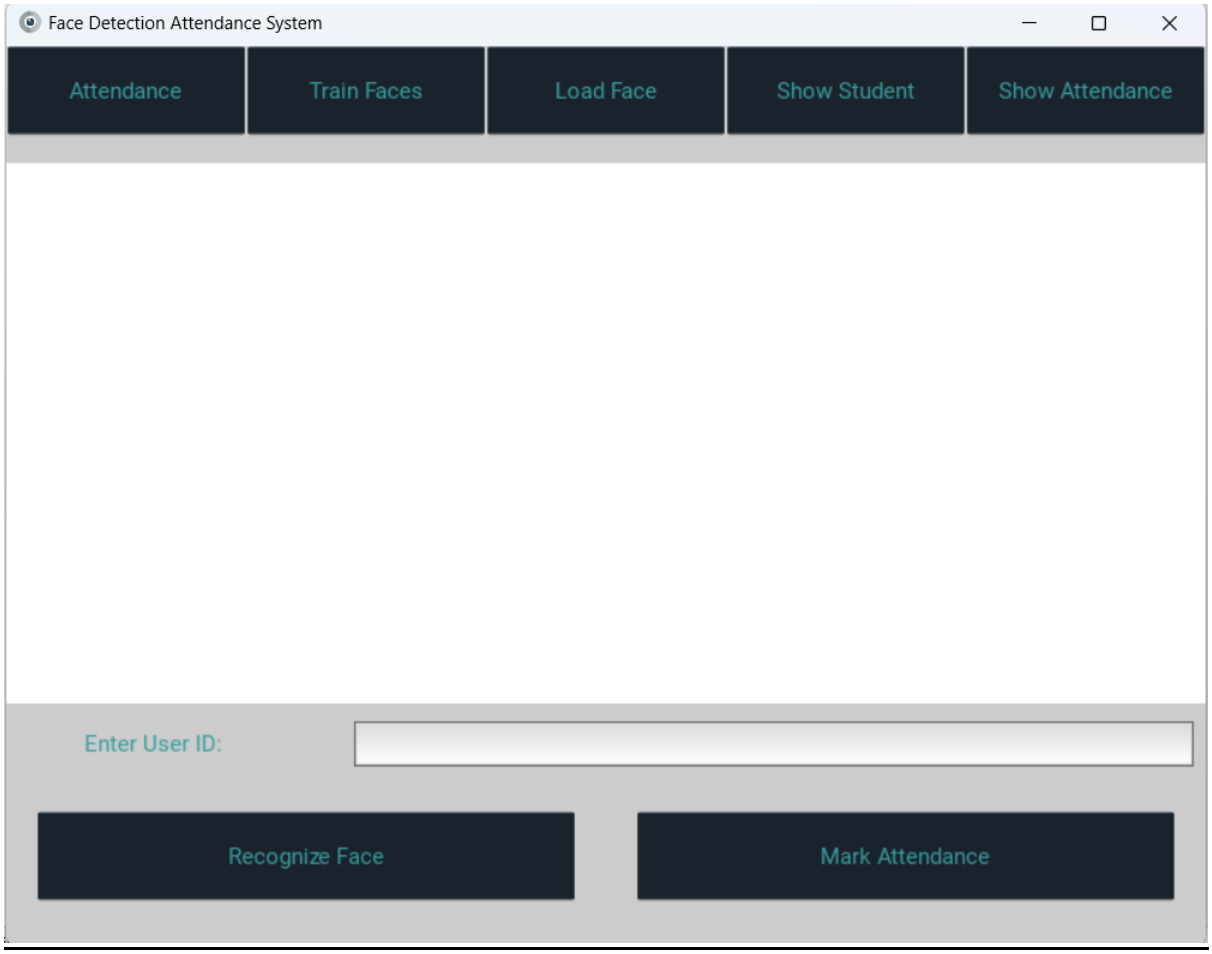
```
            kv.get_screen('main').ids.info.text = "Attendence not entered."
        camera.release()
        cv2.destroyAllWindows()
    except Exception as e:
        kv.get_screen('main').ids.info.text = "Some error occured. Try again!"
        print(e)
def display_frame(self, frame, dt):
    texture = Texture.create(size=(frame.shape[1], frame.shape[0]), colorfmt='bgr')
    texture.blit_buffer(frame.tobytes(order=None), colorfmt='bgr', bufferfmt='ubyte')
    texture.flip_vertical()
    kv.get_screen('main').ids.vid.texture = texture
def dataset(self):
    dataset_path = path = os.path.join(self.Dir, 'dataset')
    if not (os.path.isdir(dataset_path)):
        os.mkdir(dataset_path)
    try:
        name = kv.get_screen('second').ids.user_name.text
        face_id = kv.get_screen('second').ids.user_id.text
        snap_amount = int(kv.get_screen('second').ids.snap.text)
        camera = cv2.VideoCapture(0)
        camera.set(3, 1920)
        camera.set(4, 1080)
        face = cv2.CascadeClassifier(self.Dir + '/haarcascade_frontalface_default.xml')
        if len(face_id)<=0 or len(name)<=0 or snap_amount <=0:
            kv.get_screen('second').ids.info.text = "All Fields Required"
        else:
            count = 0
            while(True):
                rtrn, image=camera.read()
                gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
                faces = face.detectMultiScale(gray, 1.3, 5)
                for(x,y,w,h) in faces:
                    cv2.rectangle(image, (x,y),(x+w,y+h),(255,0,0),2)
```

```
                count+=1
                cv2.imwrite(self.Dir + "/dataset/"+str(name)+"_" + str(face_id) + '_' + str(count) + ".jpg",
gray[y:y+h,x:x+w])
                cv2.imshow('image', image)
            wait = cv2.waitKey(10) & 0xff
            if wait == 27:
                break
            elif count >=snap_amount:
                break
        camera.release()
        cv2.destroyAllWindows()
        try:
            exist = False
            df = pd.read_csv(self.Dir + '/list/students.csv')
            for i in range(len(df['id'])):
                if df['id'].iloc[i] == int(face_id):
                    exist = True
            if not exist:
                df.loc[len(df.index)] = [int(face_id),name]
                df.to_csv(self.Dir + '/list/students.csv', index=False)
            df1 = pd.read_csv(self.Dir + '/Attendance/Attendance.csv')
            for i in range(len(df1['id'])):
                if df1['id'].iloc[i] == int(face_id):
                    exist = True
            if not exist:
                arr = [int(face_id),name]
                arr = np.concatenate((arr,[0]*(len(df1.columns)-2)))
                df1.loc[len(df1.index)] = arr
                df1.to_csv(self.Dir + '/Attendance/Attendance.csv', index=False)
        except Exception as e:
            print(e)
        kv.get_screen('second').ids.info.text = "Face included successfully. Please train the system."
    except:
```

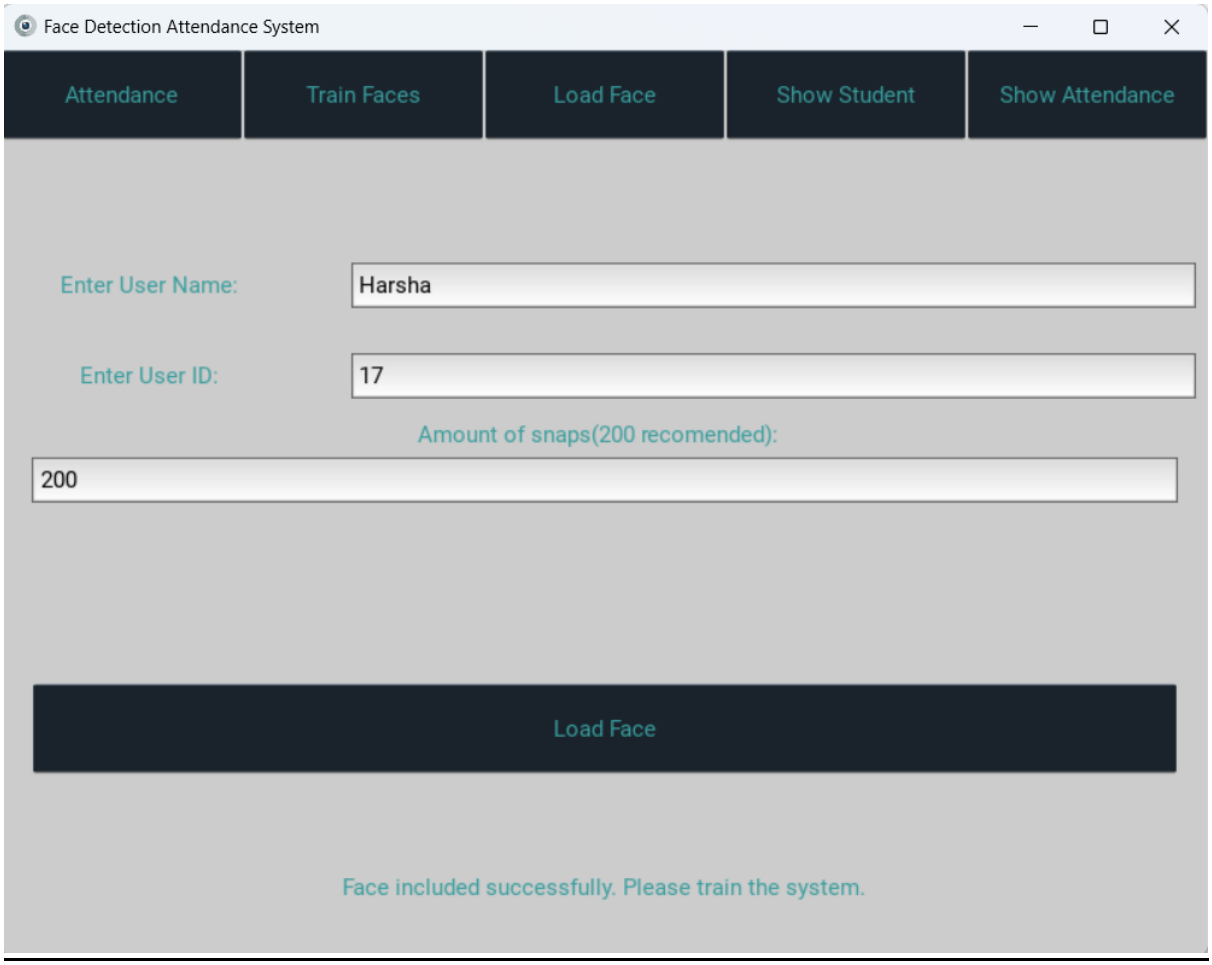
```
            kv.get_screen('second').ids.info.text = "Some error occured. Try again!"
    def getImage_Labels(self, dataset,face):
            imagesPath=[os.path.join(dataset,f) for f in os.listdir(dataset)]
            faceSamples = []
            ids = []
            for imagePath in imagesPath:
                PIL_img=Image.open(imagePath).convert('L')
                img_numpy = np.array(PIL_img, 'uint8')
                id=int(os.path.split(imagePath)[-1].split("_")[1])
                faces = face.detectMultiScale(img_numpy)
                for (x,y,w,h) in faces:
                    faceSamples.append(img_numpy[y:y+h,x:x+w])
                    ids.append(id)
            return faceSamples,ids
    def train(self):
        dataset_path = path = os.path.join(self.Dir, 'dataset')
        if not (os.path.isdir(dataset_path)):
            os.mkdir(dataset_path)
        kv.get_screen('main').ids.info.text = "Training Faces."
        kv.get_screen('second').ids.info.text = "Training Faces."
        dataset = self.Dir + '/dataset'
        recog = cv2.face.LBPHFaceRecognizer_create()
        face = cv2.CascadeClassifier(self.Dir + '/haarcascade_frontalface_default.xml')
        faces,ids=self.getImage_Labels(dataset,face)
        recog.train(faces, np.array(ids))
        recog.write(self.Dir + '/trainer/trainer.yml')
        kv.get_screen('main').ids.info.text = str(len(np.unique(ids))) + " face trained."
        kv.get_screen('second').ids.info.text = str(len(np.unique(ids))) + " face trained."
if(__name__ == "__main__"):
    MainApp().run()
```

Outputs:

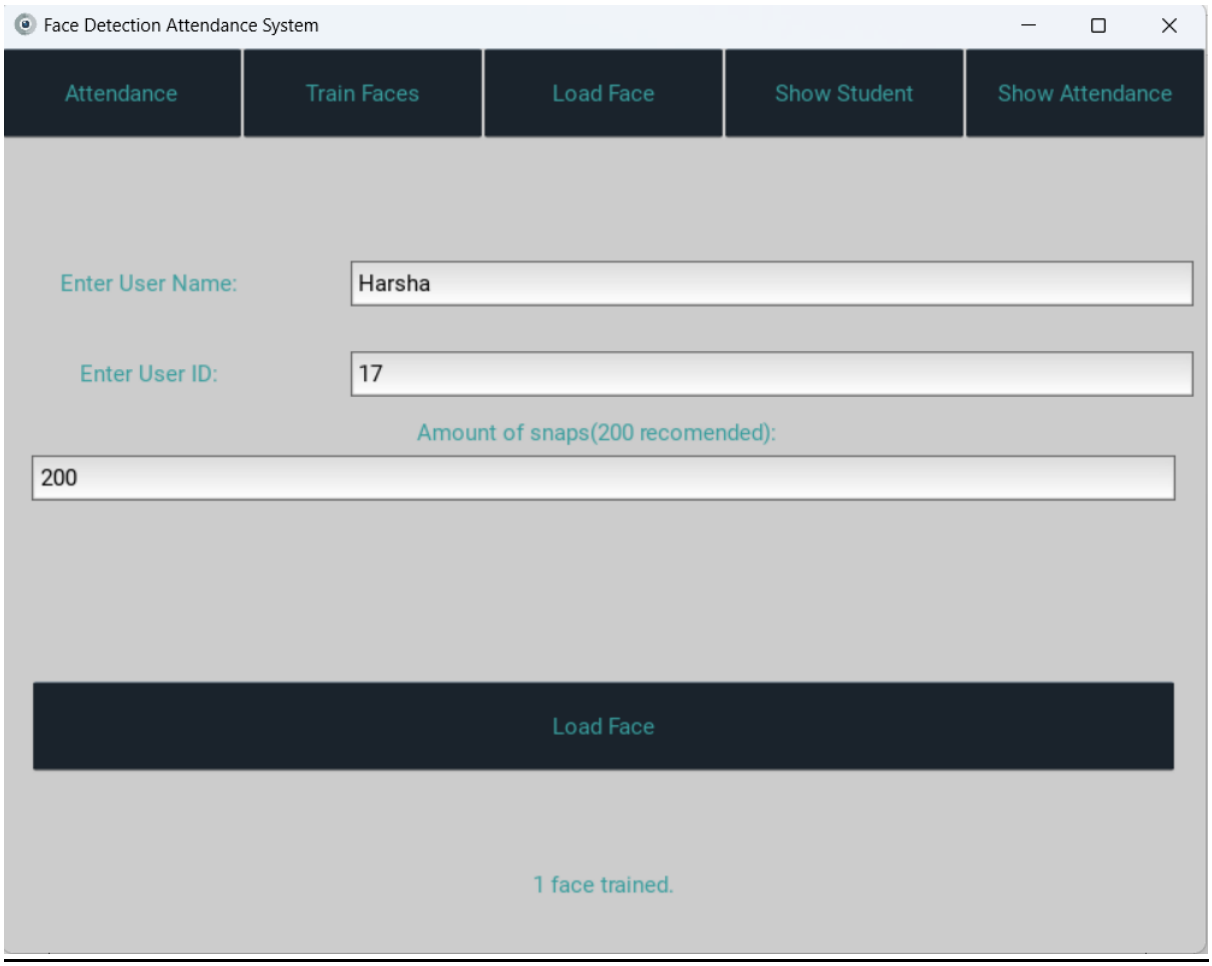
- **GUI SCREEN(Using kivy library)**



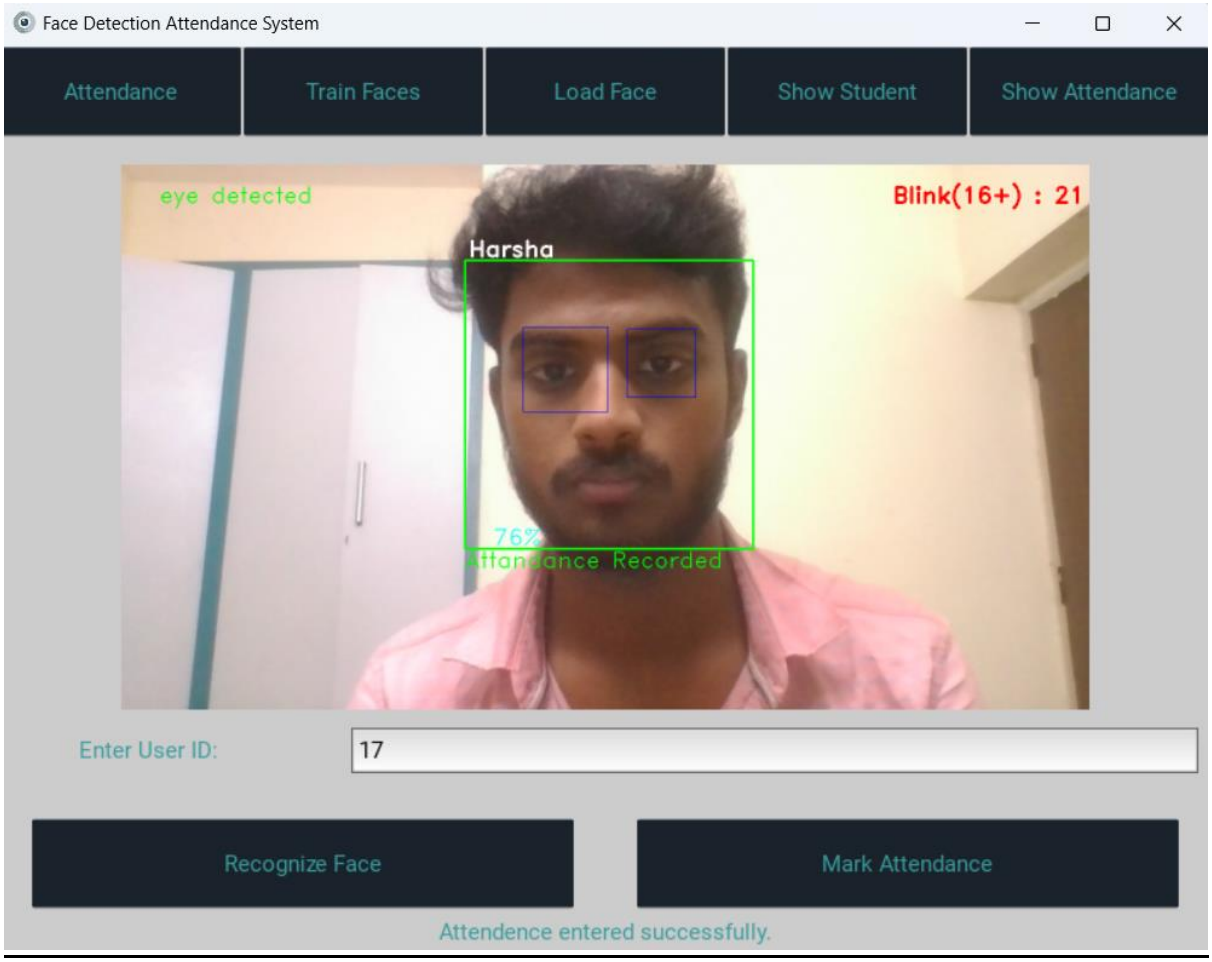
- **Loading Face(Creating dataset)**



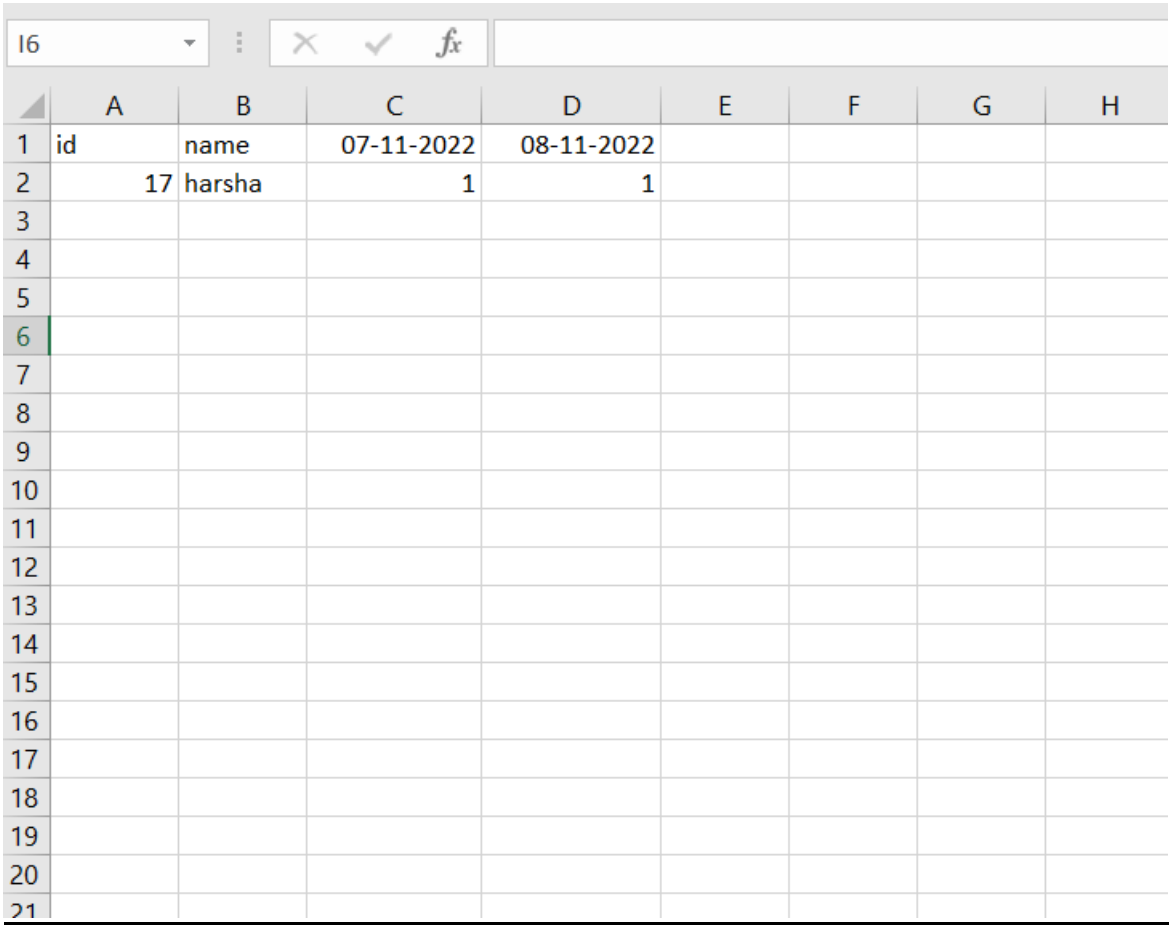
- **Training Face(Using trainer.yml file)**



- **Recognizing Face(face detected using haarcascade classifiers)**



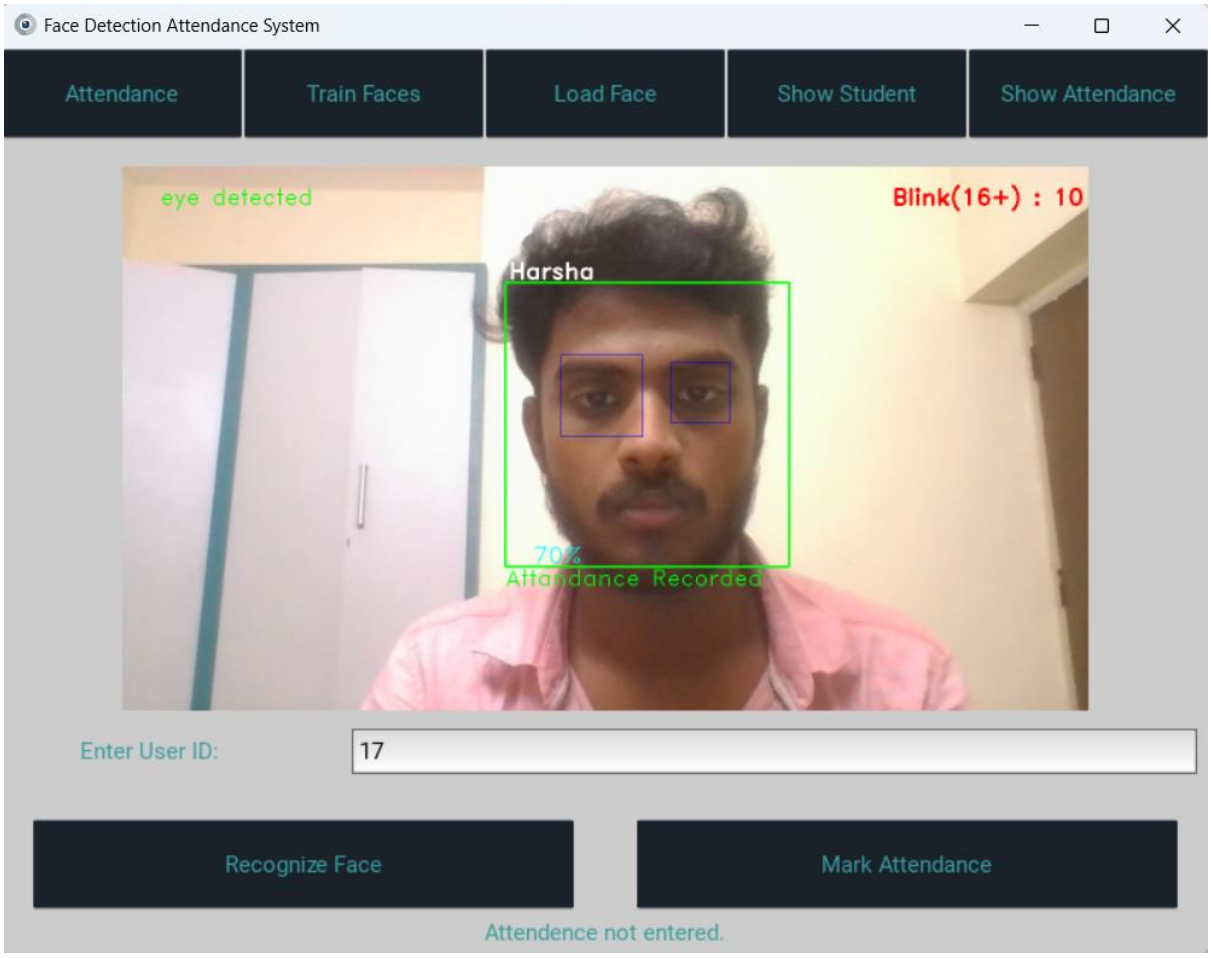
- **Marking Attendance(In attendance.csv file)**



I6

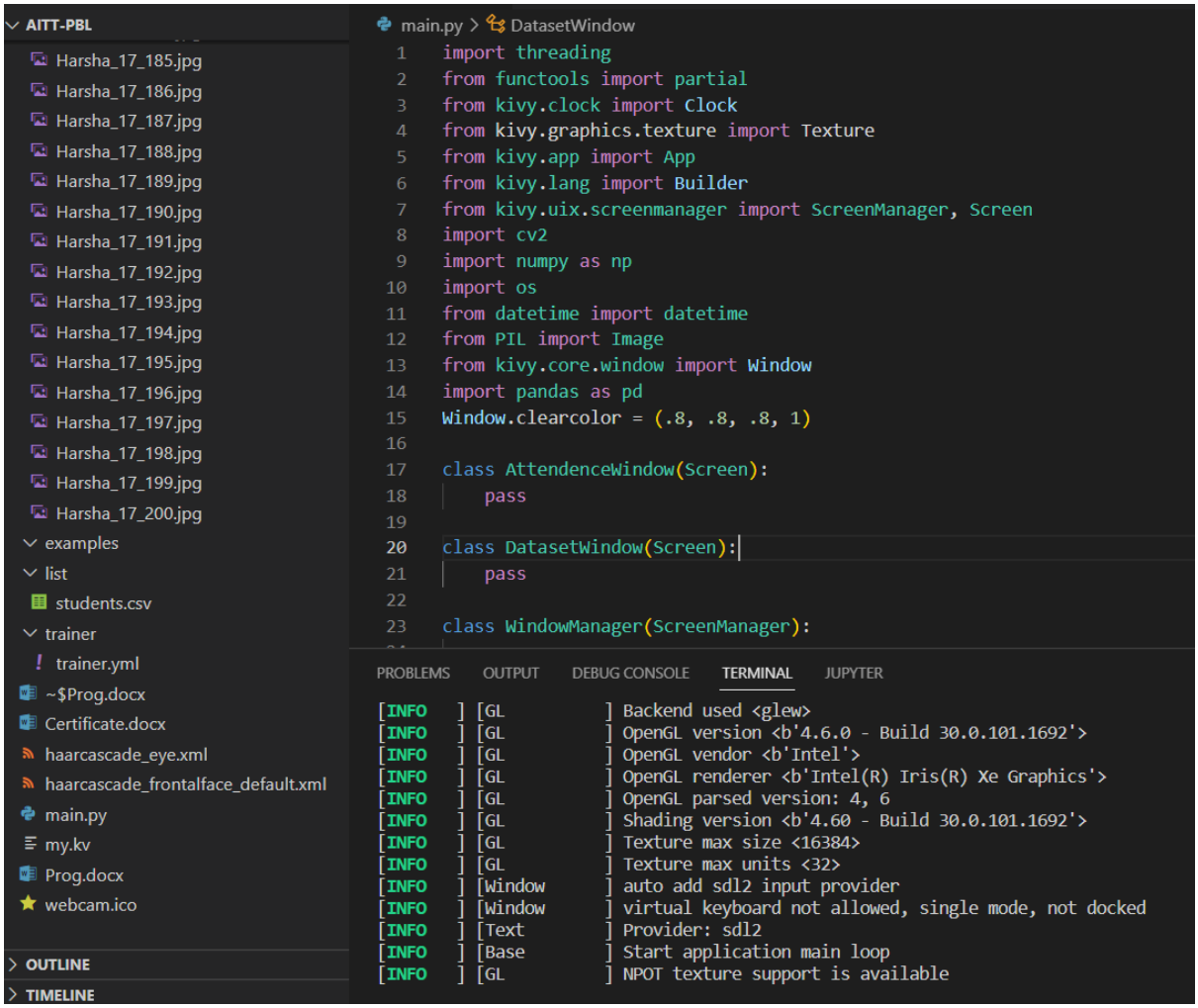
	A	B	C	D	E	F	G	H
1	id	name	07-11-2022	08-11-2022				
2	17	harsha	1	1				
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								

- **Attendance not recorded(if blink < 16)**



Project Outcome:

As you can see here below camera captured the student image individually and stored in training data set . After training the images, data is stored in trainer.yml file. We test the image of the same student and it recognized perfectly with his name. The student attendance was marked in excel sheet.



Conclusion:

This system shows attendance using facial recognition which focuses on saving time and effort. Our system is designed to overcome traditional system which is taken on pen and paper or file system. Face recognition has many applications like it can be used in surveillance, security purpose, law enforcement etc.

References:

[1] Priyanka Wagh, Roshani Thakare, Jagruti Chaudhary, Shweta Patil, “Attendance System based on Eigen face and PCA algorithms” volume 1, Year 2015, 10.1109/ICGCIoT.2015.7380478.

[2] H. K. Nguyen; M. T. Chew, ” RFID-based attendance management system”,2017 IEEE.

[3] Omar Abdul, Rhman Salim, Rashidah Funke Olanrewaju, Wasiu Adebayo Balogun. “ Class Attendance Management System Using Face Recognition." 2018 7th International Conference on Computer and Communication Engineering (ICCCE) IEEE 2018.

[4] Adrian Rhesa Septian Siswanto, Anto Satriyo Nugroho, Maulahikmah Galinium. “Implementation of Face Recognition Algorithm for Biometrics Based Time Attendance System" Center for Information Communication Technology Agency for the Assessment Application of Technology (PTIK-BPPT) Teknologi 3 BId., 3F, PUSPIPTEK Serpong, Tangerang, INDONESIA, 15314.

[5] Rekha AL and Dr. Chethan HK, “Automated Attendance System Using Face Recognition Through Video Surveillance”; International Journal For Technological Research in Engineering”, vol.1, no.11, July 2014, pp.1327-1330

[6]https://towardsdatascience.com/face-recognition-how-lbph-works-90ec258c3d6b.

[7] Automatic Attendance System Using Face Recognition. Ashish Choudhary1,Abhishek Tripathi2,Abhishek Bajaj3,Mudit Rathi4 and B.M Nandini5 1,2,3,4,5 Information Science and Engineering, The National Institute of Engineering.

[8] Shireesha Chintalapati, M.V. Raghunadh, “Automated Attendance Management System Based On Face Recognition Algorithms”, 2013 IEEE

[9] Abhishek Jha,” Class Room Attendance System Using Facial Recognition System”, The International Journal of Mathematics, Science, Technology and Management (ISSN : 2319-8125) Vol. 2 Issue 3.

[10] Jenif D Souza W S, Jothi S, Chandra sekar A,” Automated Attendance Marking and Management System by Facial Recognition Using Histogram”, 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS).