J COMPONENT

PROJECT REPORT

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CSE3088
ARTIFICIAL INTELLIGENCE

PROJECT TITLE:

ATTENDANCE MANAGEMENT SYSTEM

TEAM MEMBERS:

DEEKSHITHA L – 19MIA1030

YUVASHREE R – 19MIA1053

NIHARIKA SAMYUKTHA K – 19MIA1083

FACULTY NAME:

JAYALAKSHMI S

INDEX

ACKNOWLEDGMENT

Primarily, we would like to thank the almighty for all the blessings he showered over us to complete this project without any flaws.

The success and final outcome of this assignment required a lot of guidance and assistance from many people and we are extremely fortunate to have got this all along with the completion of our project. Whatever we have done is only due to such guidance and assistance by our faculty, Dr. S.L.JAYALAKSHMI, to whom we are really thankful for giving us an opportunity to do this project.

Last but not the least, we are grateful to all our fellow classmates and our friends for the suggestions and support given to us throughout the completion of our project.

ABSTRACT

In many educational institutions, managing the attendance of students/candidates is tedious, as there would be a large number of students in the class and keeping track of all is tedious. The conventional attendance system consists of registers marked by teachers which leads to human error and a lot of maintenance. Time consumption is an important point of concern in this system. We have thought of revolutionizing it using available digital tools in the modern era i.e., FACE RECOGNITION.

The project is revolutionized in order to overcome the problems of the conventional system. Our project is about face recognition and then marking the attendance. The dataset is created by capturing images of the students in the class and when the face of the student matches with the images stored in the dataset, attendance is marked and the attendance sheet is generated as CSV file. The system saves the unrecognised faces in another folder as Unknown Images.

INTRODUCTION

Attendance plays an important role in any organization whether it be educational institutions or companies. So, it is very important to keep record of the attendance. Face is a common multidimensional structure and needs great computational investigation for identification. Biometrics strategies have been utilized for a similar purpose since quite a while now. In spite of the fact that it is powerful, it is as yet not totally solid for purpose of recognizing a human. Attendances of each and every individual are being kept up by each institution.

The manual attendance record framework isn't effective and requires more opportunity to arrange record and to measure the normal attendance of every individual. To overcome the problems of the conventional system, we use face recognition. The human face is a unique representation of individual's identity. Thus, the face recognition is a type of biometric method through which identification of an individual is performed by comparing the real time captured image with the stored images of that person in the database.

Currently, Facial Recognition System is widely spread due to its simplicity and fast performance. Face is the representation of one’s identity. So, we have prepared an automated student attendance system based on face recognition.

Why AI for Face Recognition?

- Artificial intelligence is a theory and development of computer systems that can perform tasks that normally require human intelligence.
- Facial recognition is a system built to identify a person from an image.
- AI-based software can instantaneously search databases of faces and compare them to one or multiple faces that are detected in a scene. In an instant, we can get highly accurate results.
- Face recognition AI is applied to many industries nowadays like Health care, Security, Airport boarding, Proctoring, etc.

EXISTING SYSTEMS

	System	Features	Benefits	Limitation
1	Automated attendance management system using face recognition	Use Eigen faces for Recognition.	High accuracy	Multiple faces were not recognized.
2	Face recognition attendance system by nevon	Stores the faces that are detected and automatically marks attendance.	Used for security purposes in organizations.	Don't recognize properly in poor light.

PROPOSED METHODOLOGY

SOLUTION:

- AI powered face recognition, which uses Computer Vision and Machine Learning algorithms to mark the attendance of the employees or students of the organization.

AIM:

- To develop a prototype that will facilitate class attendance by face detection and recognition of students faces in a digital image taken by a web camera or a mobile phone camera.
- The second part of the system will also be able to perform a facial recognition against a small database.

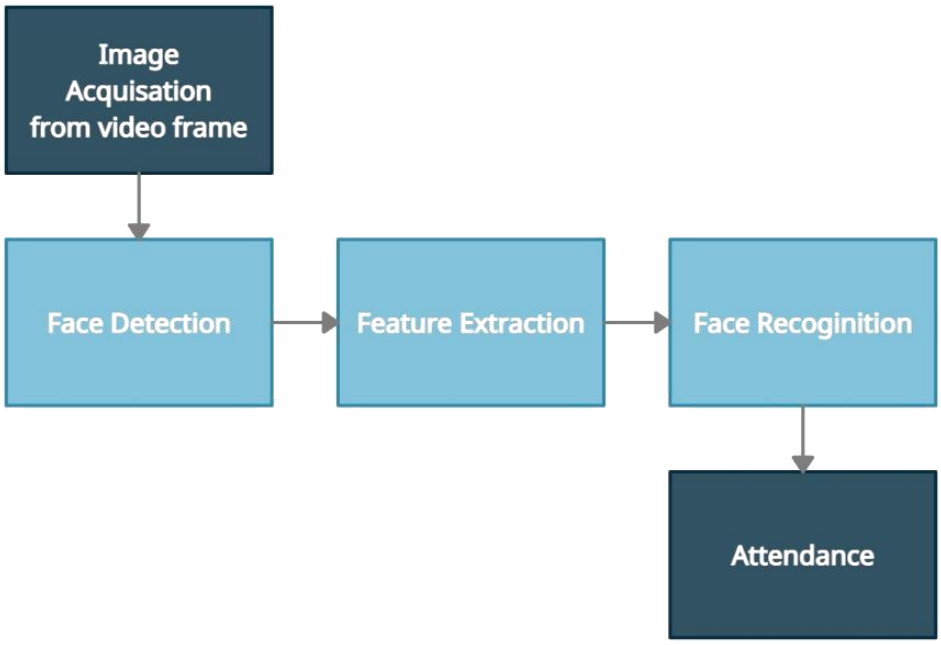
PROPOSED ATTENDANCE SYSTEM

Attendance system through facial recognition is proposed in order to replace the manual marking of student's attendance.

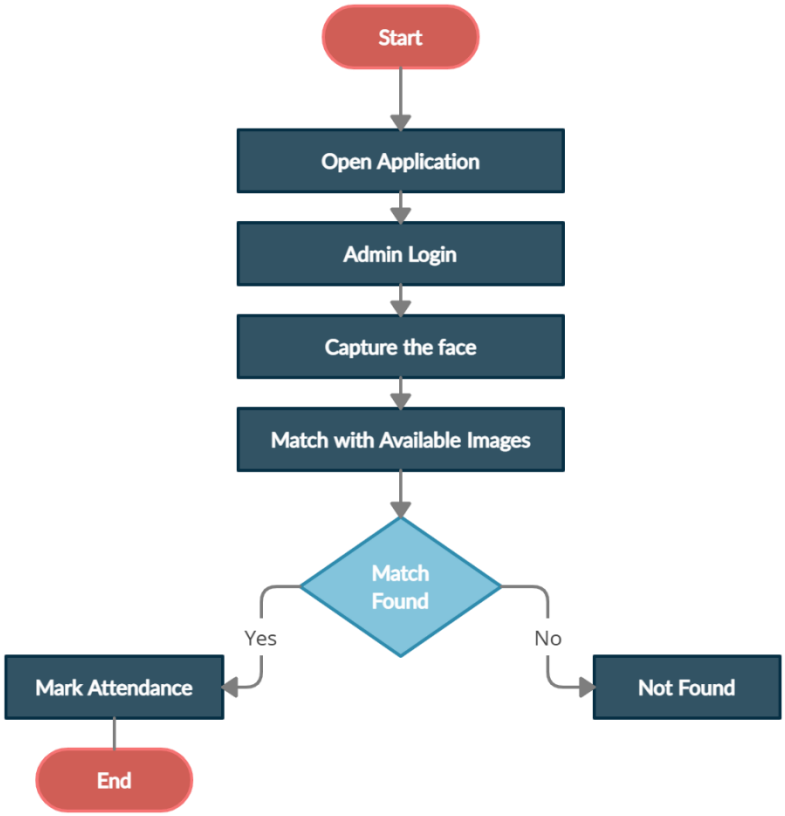
Furthermore, the automated attendance system based on face recognition is able to overcome the problems of fraudulent approach and faculty does not have to count the number of students several time to verify the presence of students.

The objective of this project is to design an automated attendance system based on face recognition. Some expected steps in order to fulfil the objectives are as follows:

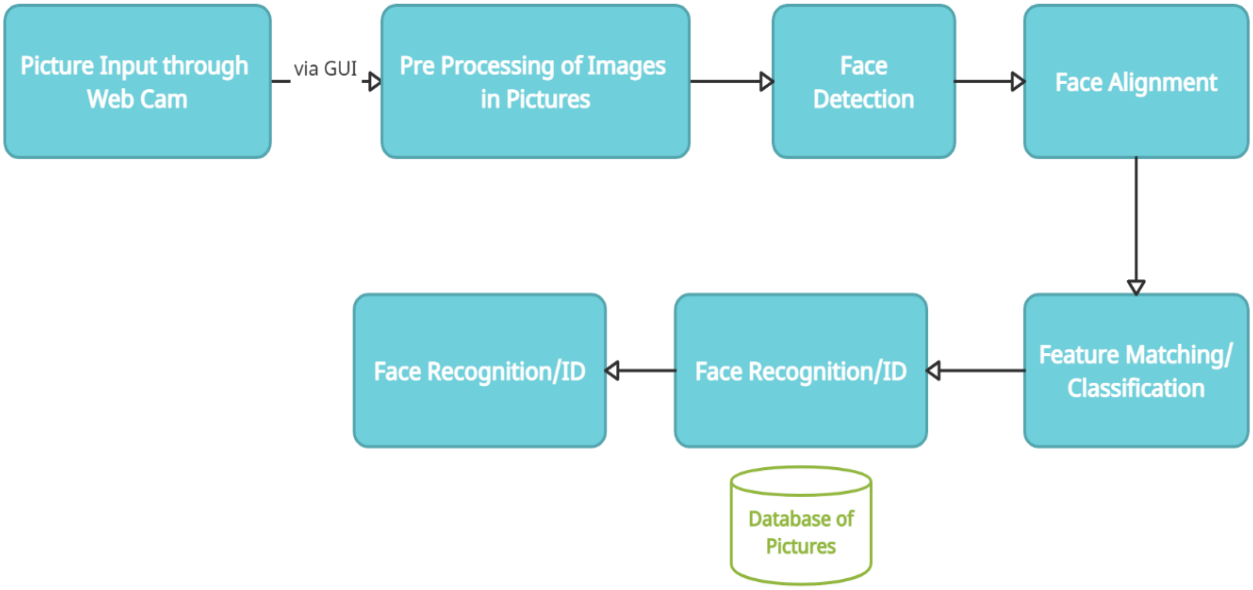
- ♣ To identify the face segment from the video frame.
- ♣ To extract the useful features from the identified face.
- ♣ To classify the features in order to recognize the identified face.
- ♣ To record the attendance of the identified students.



ARCHITECTURE DIAGRAM



FACE RECOGNITION PROCESS:



MODULES

IMAGE CAPTURING:

Image is captured from the live video when webcam is turned on

Faces are detected using Haar Cascade Algorithm.

Haar – Cascade:

- It is an Object Detection Algorithm used to identify faces in an image or a real time video. The algorithm uses edge or line detection features.
- The algorithm is given a lot of positive images consisting of faces, and a lot of negative images not consisting of any face to train on them.
- Haar-Cascades can be used to detect any types of objects as long as you have the appropriate XML file for it.
- The algorithm can be explained in four stages:

  (i) Calculating Haar Features

  (ii) Creating Integral Images

  (iii) Using Adaboost

  (iv) Implementing Cascading Classifiers

TRAINING THE IMAGES:

The captured images are trained using LBPH algorithm.

LBPH Algorithm:

- Local Binary Patterns Histogram algorithm is based on local binary operator.
- The LBPH algorithm is a part of opencv.
- LBPH is an excellent feature for the classification of certain textures like faces.
- Steps involved in LBPH:
    1. Parameters: LBPH uses 4 parameters; Radius, Neighbours, Grid X and Grid Y.
    2. Training the Algorithm.
    3. Applying the LBP operation.
    4. Extracting the Histograms.
    5. Performing the face recognition.

TRACKING THE IMAGES:

It recognizes similar faces from the Training Images dataset.

MARKING THE ATTENDANCE:

On marking the attendance, CSV file is generated consisting of Id, Name, Date and Time.

STEPS INVOLVED IN FACE RECOGNITION

FACE DETECTION:

To begin with, the camera will detect and recognize a face.

FACE ANALYSIS:

Then the photo of the face is captured and analyzed.

IMAGE TO DATA CONVERSION:

Now the captured image is converted to a mathematical formula and these facial features become numbers. This numerical code is known a face print. The way every person has a unique fingerprint, in the same way, they have unique face print.

MATCH FINDING:

Then the code is compared against a database of other face prints. This database has photos with identification that can be compared. The technology then identifies a match for your exact features in the provided database. It returns with the match and attached information such as name and college ID.

PYTHON LIBRARIES USED

- **Numpy**

Numpy provides support of highly optimized multidimensional arrays.

- **Pandas**

Pandas is the most popular Python library for manipulating data.

- **Tkinter**

Tkinter is Python's de-facto standard GUI (Graphical User Interface) package. This framework provides a simple way to create GUI elements using the widgets found in the Tk toolkit. Tk widgets can be used to construct buttons, menus, data fields, etc. in a Python application.

- **CSV**

CSV module implements classes to read and write tabular data in CSV format

- **Datetime**

The datetime module supplies classes for manipulating dates and times in many ways.

- **OpenCV**

Used to count the number of people.

SOFTWARE USED

- **PyCharm**

PyCharm is a hybrid-platform developed by JetBrains as an IDE for Python. It is commonly used for Python application development.

- **Microsoft excel**

Microsoft Excel is a spreadsheet program incorporated in Microsoft Office suite of applications.

SAMPLE CODE

Code to detect face using Haar-Cascade:

```
import cv2

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 + "." + Id + '.' + str(sampleNum) + ".jpg", gray[y:y + h, x:x + w])
        # display the frame
        cv2.imshow('frame', img)
        # wait for 100 miliseconds
        if cv2.waitKey(30) & 0xFF == ord('q'):
            break
        # break if the sample number is morethan 100
        elif sampleNum > 60:
            break
cam.release()
cv2.destroyAllWindows()
```

Training the captured images using LBPH algorithm.

```
import cv2
import numpy as np
import os
from PIL import Image

def getImagesAndLabels(path):
    imagePaths = [os.path.join(path, f) for f in os.listdir(path)]
    faces = []
    Ids = []

    for imagePath in imagePaths:
        # loading the image and converting it to gray scale
        pilImage = Image.open(imagePath).convert('L')
        # Now we are converting the PIL image into numpy array
        imageNp = np.array(pilImage, 'uint8')
        # getting the Id from the image
        Id = int(os.path.split(imagePath)[-1].split(".")[1])
        # extract the face from the training image sample
        faces.append(imageNp)
        Ids.append(Id)
    return faces, Ids

recognizer = cv2.face_LBPHFaceRecognizer.create()
faces, Id = getImagesAndLabels("TrainingImage")
recognizer.train(faces, np.array(Id))
recognizer.save("TrainingImageLabel\Trainner.yml")
print('Completed', 'Your model has been trained successfully!!')
```

Tracking the images:

```
def TrackImages():
    recognizer = cv2.face.LBPHFaceRecognizer_create()#cv2.createLBPHFaceRecognizer()
    recognizer.read("TrainingImageLabel\Trainner.yml")
    harcascadePath = "haarcascade_frontalface_default.xml"
    faceCascade = cv2.CascadeClassifier(harcascadePath);
    df=pd.read_csv("StudentDetails\StudentDetails.csv")
    cam = cv2.VideoCapture(0)
    font = cv2.FONT_HERSHEY_SIMPLEX
    col_names =  ['Id','Name','Date','Time']
    attendance = pd.DataFrame(columns = col_names)
```

```
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)
        Id, conf = recognizer.predict(gray[y:y+h,x:x+w])
        if(conf < 50):
            ts = time.time()
            date = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d')
            timeStamp = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
            aa=df.loc[df['Id'] == Id]['Name'].values
            tt=str(Id)+"-"+aa
            attendance.loc[len(attendance)] = [Id,aa,date,timeStamp]

        else:
            Id='Unknown'
            tt=str(Id)
        if(conf > 75):
            noOfFile=len(os.listdir("ImagesUnknown"))+1
            cv2.imwrite("ImagesUnknown\Image"+str(noOfFile) + ".jpg", im[y:y+h,x:x+w])
        cv2.putText(im,str(tt),(x,y+h), font, 1,(255,255,255),2)
    attendance=attendance.drop_duplicates(subset=['Id'],keep='first')
    cv2.imshow('im',im)
```

```
  ts = time.time()
  date = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d')
  timeStamp = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
  Hour,Minute,Second=timeStamp.split(":")
  fileName="Attendance\Attendance_"+date+"_"+Hour+"-"+Minute+"-"+Second+".csv"
  attendance.to_csv(fileName,index=False)
  cam.release()
  cv2.destroyAllWindows()
  res=attendance
  message2.configure(text= res)
  res = "Attendance Taken"
  message.configure(text= res)
  tk.messagebox.showinfo('Completed','Congratulations ! Your attendance has been marked successfully for the day!!')
```

```
def quit_window():
   MsgBox = tk.messagebox.askquestion ('Exit Application','Are you sure you want to exit the application',icon = 'warning')
   if MsgBox == 'yes':
       tk.messagebox.showinfo("Greetings", "Thank You very much for using our software. Have a nice day ahead!!")
       window.destroy()

takeImg = tk.Button(window, text="IMAGE CAPTURE ", command=TakeImages  ,fg="white"  ,bg="tomato"  ,width=25  ,height=2, activebackground = "pink" ,f
takeImg.place(x=245-x_cord, y=425-y_cord)
trainImg = tk.Button(window, text="MODEL TRAINING ", command=TrainImages  ,fg="white"  ,bg="royalblue1"  ,width=25  ,height=2, activebackground = "p
trainImg.place(x=645-x_cord, y=425-y_cord)
trackImg = tk.Button(window, text="MARKING THE ATTENDANCE ", command=TrackImages  ,fg="white"  ,bg="forest green"  ,width=30  ,height=2, activebackg
trackImg.place(x=1075-x_cord, y=425-y_cord)
quitWindow = tk.Button(window, text="QUIT", command=quit_window  ,fg="white"  ,bg="red"  ,width=10  ,height=2, activebackground = "pink" ,font=('Tim
quitWindow.place(x=1070, y=575-y_cord)

window.mainloop()
```

RESULTS AND DISCUSSIONS

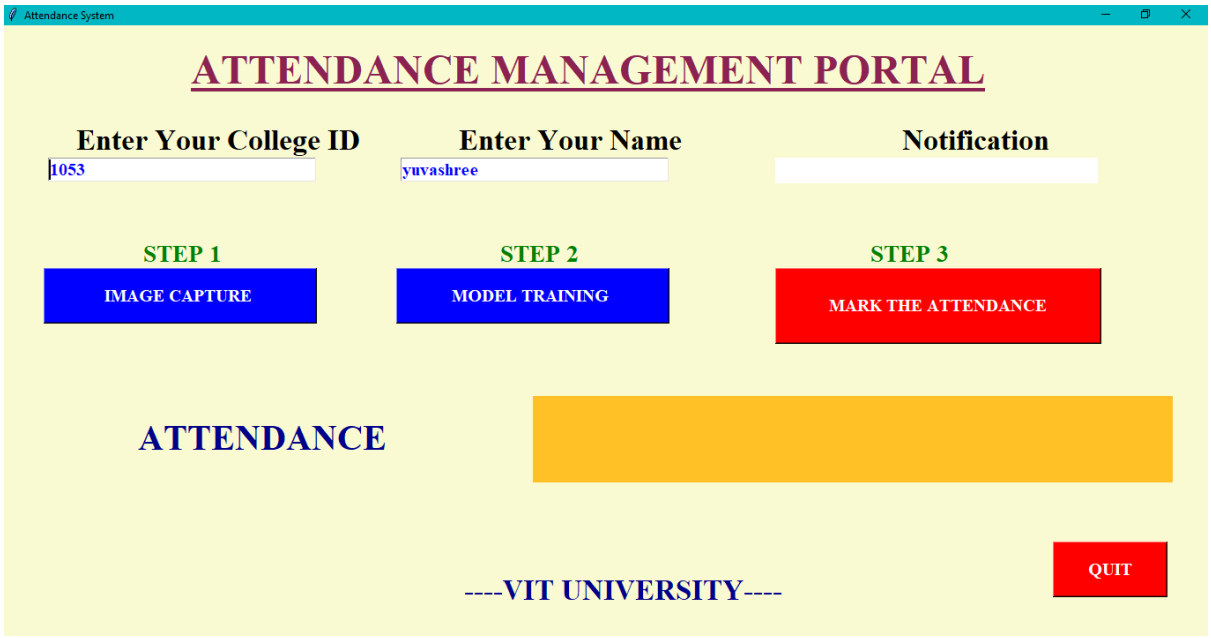


IMAGE CAPTURING:

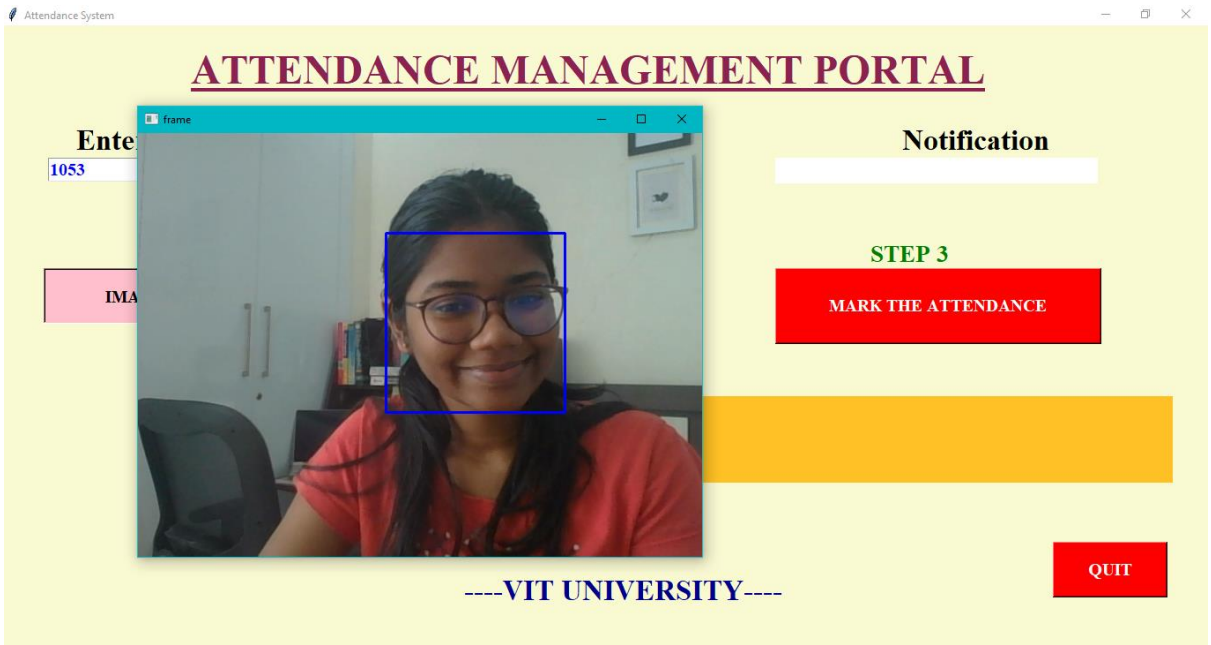
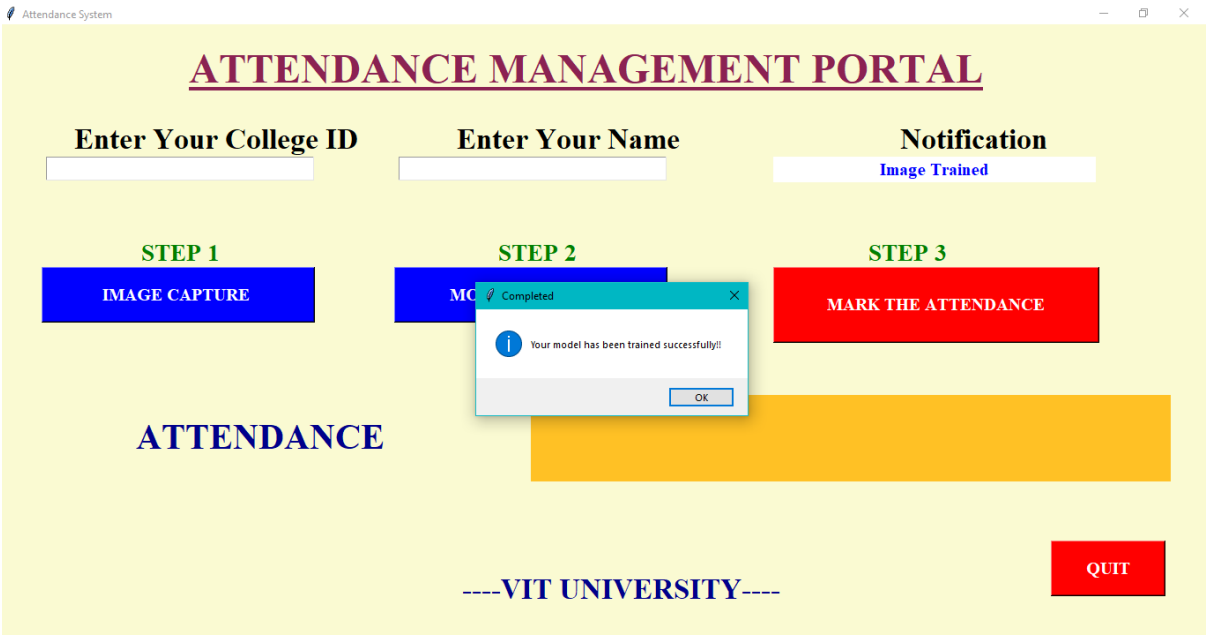
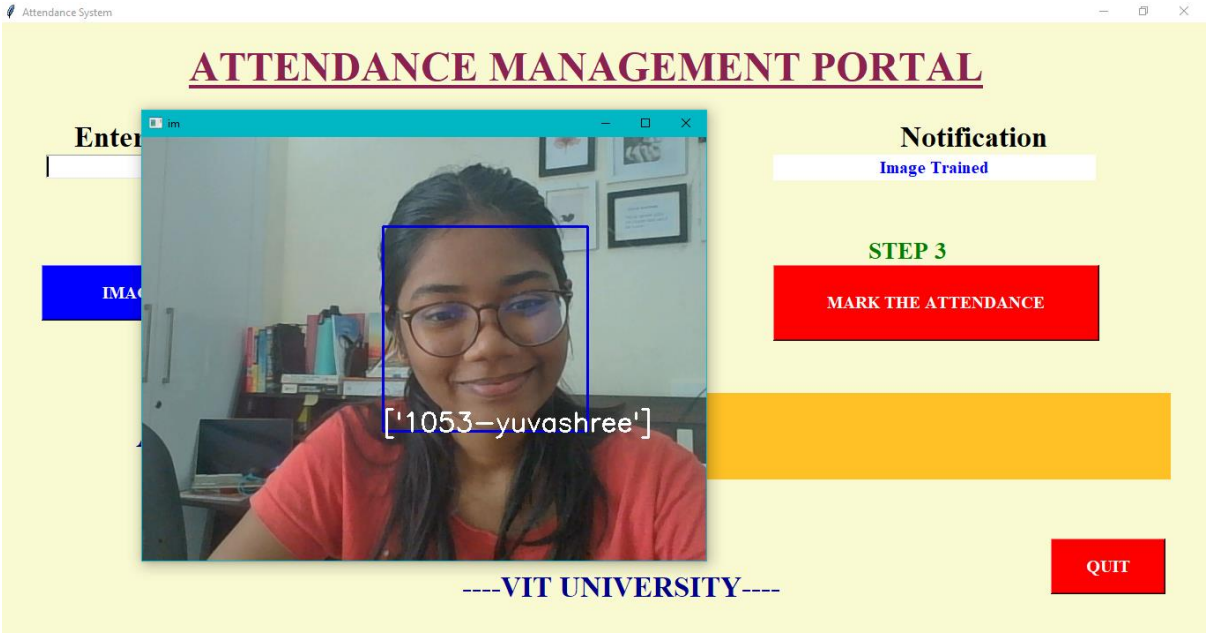


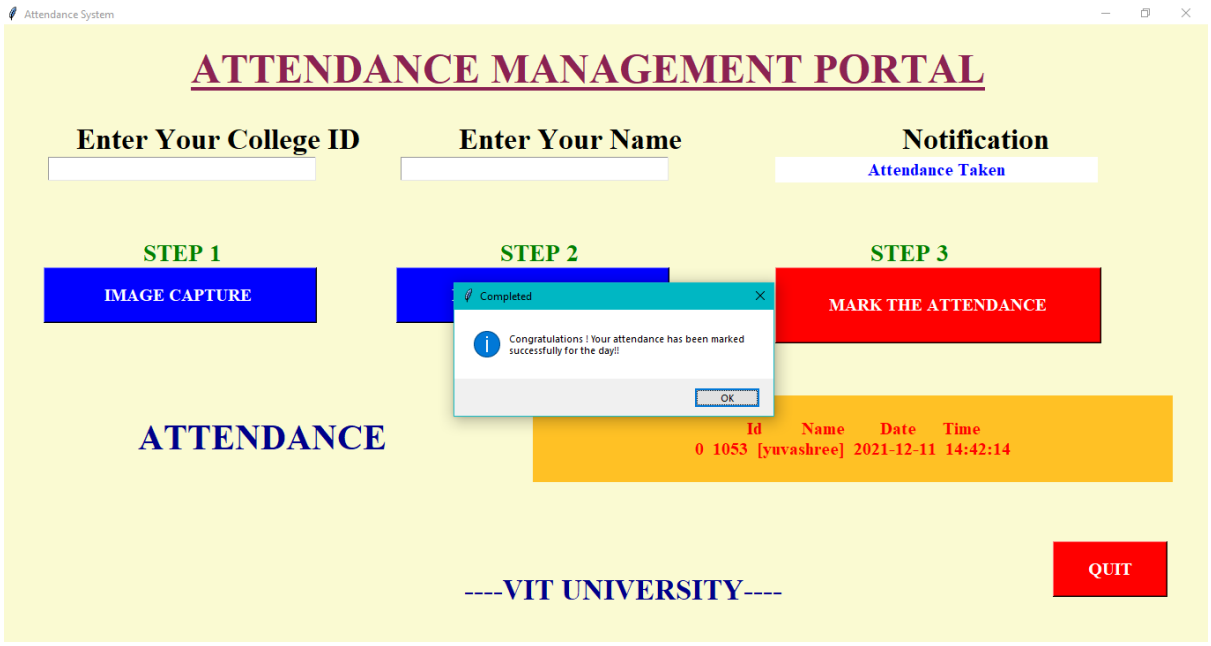
IMAGE IS TRAINED:



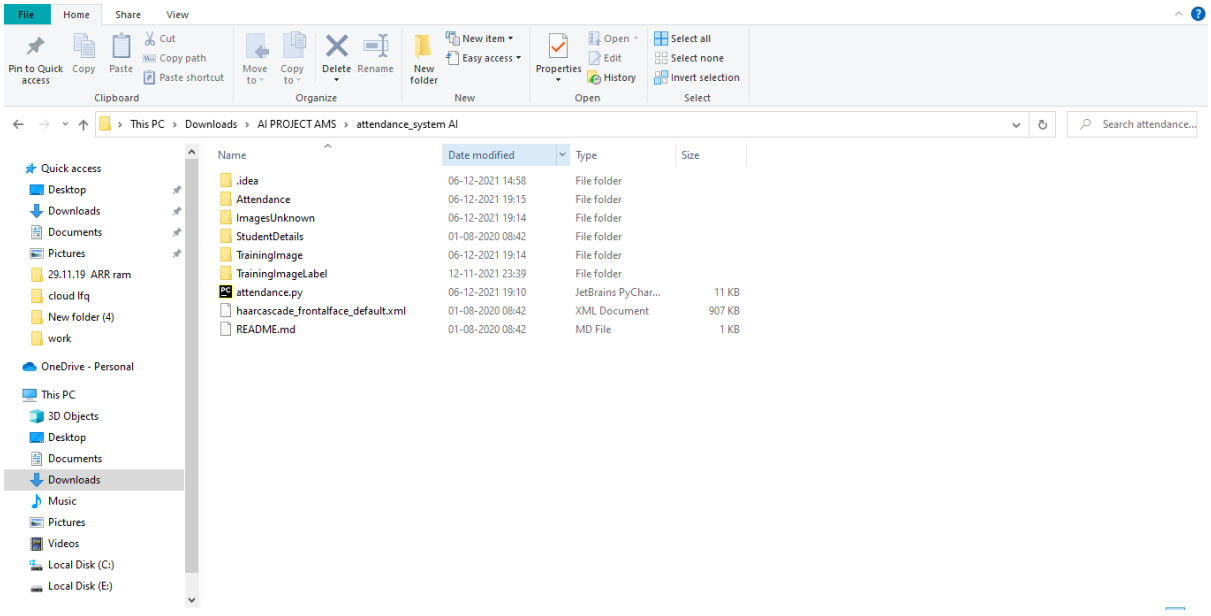
TRACKING THE IMAGE:

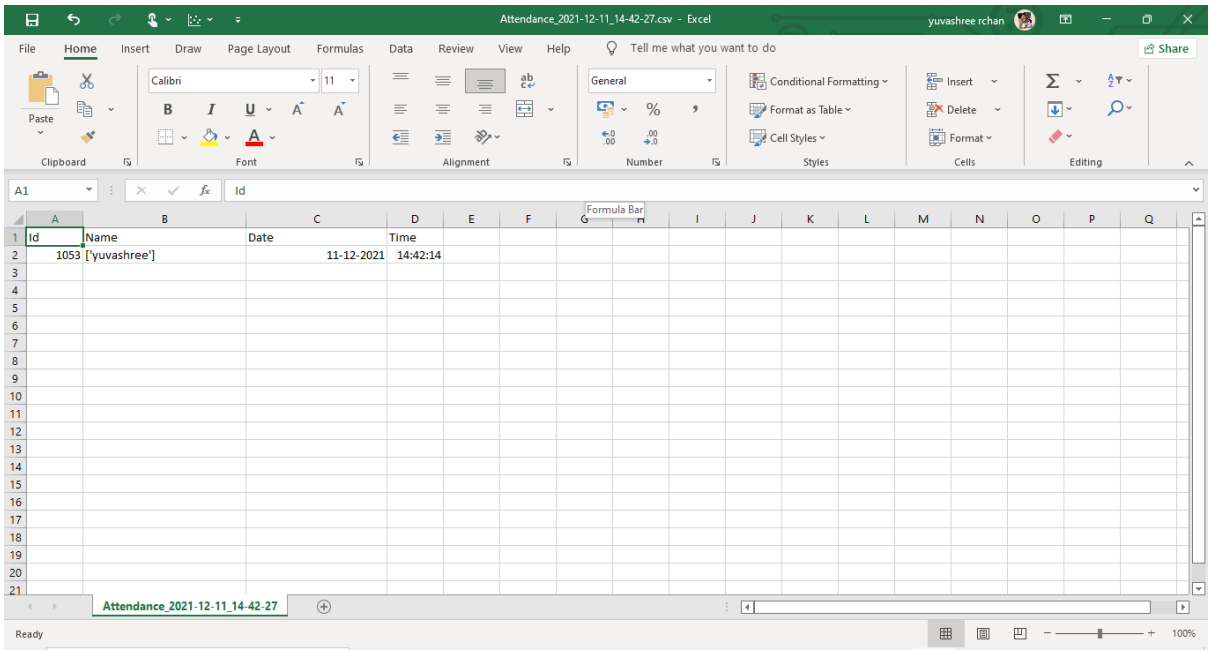


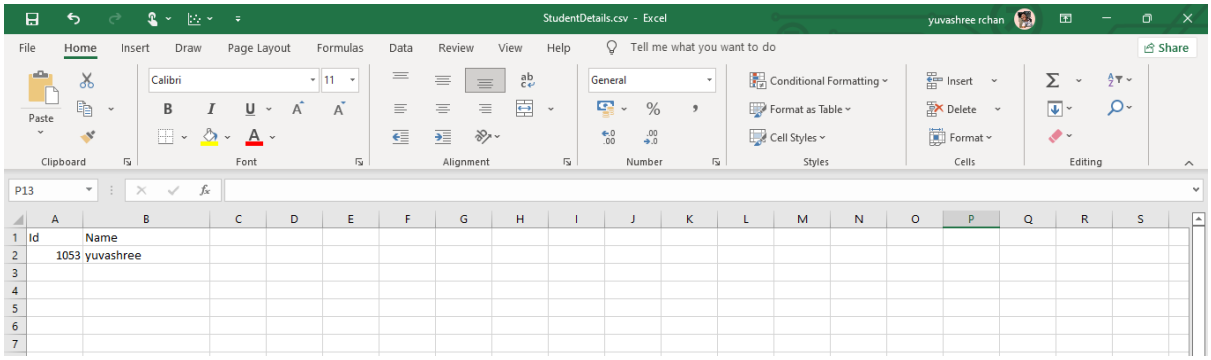
MARKING THE ATTENDANCE:



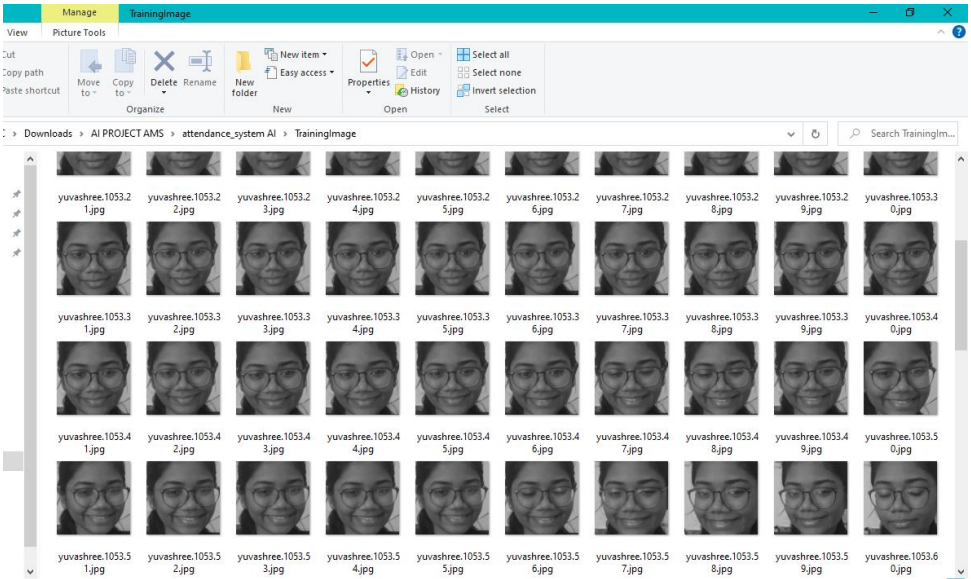
THE FOLDERS CREATED:







TRAINING IMAGES:



CONCLUSION

The Face Recognition based Automated Attendance System is simple, accurate and works efficiently. Our project uses the most productive Open CV face recognition method accessible for Attendance Management.

The system has been implemented using the LBPH algorithm. LBPH is the most authentic and competent face recognition algorithm found in Open CV for the identification of the students in an educational institute and marking their attendance by avoiding proxies.

The proposed approach provides a method to identify the individuals by comparing their input image obtained from image in folder. From this model we can recognize the faces of students and can mark their attendance automatically in real time without human intervention. The feasibility of the model can be increased if a cloud can be hired to store details.

FUTURE ENHANCEMENTS

- The project has a very vast scope in future. The project can be implemented in colleges with extra technologies. Project can be updated in near future as it is very flexible in terms of expansion. The face recognition model would be done more precisely so that maximum accuracy can be achieved.

- The following are the future scope for the project:

  - Discontinue of particular student eliminates potential attendance.
  - Attendance management system can be accessed through mobile devices.
  - Individual Attendance system using login for only faculty and admin and rest others can view the attendance.

REFERENCES

- https://pythonprogramming.net/haar-cascade-face-eye-detection-python-opencv-tutorial/
- https://towardsdatascience.com/face-recognition-how-lbph-works-90ec258c3d6b
- https://en.wikipedia.org/wiki/OpenCV
- https://www.mdpi.com/1424-8220/20/2/342/htm
- https://www.geeksforgeeks.org/python-gui-tkinter/
- https://issuu.com/grdjournals/docs/automatic_face_recognition_attendance_system_using
- https://www.researchgate.net/publication/341870242_Smart_Attendance_System_using_OPENCV_based_on_Facial_Recognition
- https://www.pyimagesearch.com/2018/09/24/opencv-face-recognition/
- https://www.researchgate.net/figure/Face-Detection-and-Recognition-Flow-Diagram_fig2_326986115