FACE RECOGNITION BASED ATTENDANCE SYSTEM

A PROJECT REPORT

Submitted by

Akhil Pandey(211B032)

Bicky Kumar(211B094)

Priyadarshi Kumar(211B227)

Under the guidance of: Dr. P.S Banerjee

January 2024-May 2024

Submitted in partial fulfillment for the award of the degree

Of

BACHELOR OF TECHNOLOGY

IN

Computer Science & Engineering

Department of Computer Science & Engineering

JAYPEE UNIVERSITY OF ENGINEERING & TECHNOLOGY,

AB ROAD, RAGHOGARH, DT. GUNA-473226 MP, INDIA

Declaration by the Student

I hereby declare that the work reported in the B. Tech. project entitled as **Face Recognition Based Attendance System**, in partial fulfillment for the award of degree of B.Tech submitted at Jaypee University of Engineering and Technology, Guna, as per best of my knowledge and belief there is no infringement of intellectual property right and copyright. In case of any violation I will solely be responsible.

Akhil Pandey(211B032)

Bicky Kumar(211B094)

Priyadarshi Kumar(211B227)

Department of Computer Science and Engineering
Jaypee University of Engineering and Technology
Guna, M.P., India

Date:

JAYPEE UNIVERSITY OF ENGINEERING & TECHNOLOGY

Grade 'A+' Accredited with by NAAC & Approved U/S 2(f) of the UGC Act,
1956A.B. Road, Raghogarh, Dist: Guna (M.P.) India, Pin-473226
Phone: 07544 267051, 267310-14, Fax: 07544 267011

Website:www.juet.ac.in

CERTIFICATE

This is to certify that the work titled "**FACE RECOGNITION BASED ATTENDANCE SYSTEM**" submitted by "**AKHIL PANDEY(211B032), BICKY KUMAR(211B094), PRIYADARSHI KUMAR(211B227)**" in partial fulfillment for the award of degree of Bachelor of Technology in Computer Science and Engineering of Jaypee University of Engineering & Technology, Guna has been carried out under my supervision. As per best of my knowledge and belief there is no infringement of intellectual property right and copyright. Also, this work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma. In case of any violation concern students will solely be responsible.

Signature of Supervisor:

Name of Supervisor: Dr. P.S Banerjee

Designation:

Date:

ACKNOWLEDGEMENT

Any endeavor cannot lead to success unless and until a proper platform is provided for the same. This is the reason why: we find ourselves fortunate to complete our work on a minor project under the supervision of Dr. P.S Banerjee. Our sincere gratitude to him for having faith in us and thus allowing us to carry out a project on technology completely new to us, for which we had to research and learn many new things, which will help us deal with advanced work in future. He helped immensely by guiding us throughout the project in any possible ways he could. Last but not the least, we would like to thank Dept. Of Computer Science & Engineering who created this opportunity.

Name of Student ……..
Signature of the student ……..
Date …….

Name of Student ……..
Signature of the student ……..
Date …….

Name of Student ……..
Signature of the student ……..
Date …….

Abstract

Our project delves into the implementation of AI-driven face recognition technology. Leveraging a Convolutional Neural Network trained on a diverse dataset, our aim was to develop a robust model capable of accurately identifying individuals from facial images. Through rigorous experimentation and analysis, our findings underscore the potential of AI in real-world applications, offering promising advancements in security, authentication systems, and beyond. This project represents a significant stride in the field of facial recognition, highlighting the efficacy of modern AI techniques in addressing complex tasks.

I

List of Figures

Table of Contents

Table of Contents

CHAPTER-1 **INTRODUCTION**

1.1 Project Objective:

Attendance is prime important for both the teacher and student of an educational organization. So it is very important to keep record of the attendance. The problem arises when we think about the traditional process of taking attendance in class room.

Calling name or roll number of the student for attendance is not only a problem of time consumption but also it needs energy. So an automatic attendance system can solve all above problems.

There are some automatic attendances making system which are currently used by much institution. One of such system is biometric technique and RFID system. Although it is automatic and a step ahead of traditional methodit fails to meet the time constraint. The student has to wait in queue for giving attendance, which is time taking.

This project introduces an involuntary attendance marking system, devoid of any kind of interference with the normal teaching procedure. The system can be also implemented during exam sessions or in other teaching activities where attendance is highly essential. This system eliminates classical student identification such as calling name of the student, or checking respective identification cards of the student, which can not only interfere withthe ongoing teaching process, but also can be stressful for students during examination sessions. In addition, the students have to register in the databaseto be recognized. The enrolment can be done on the spot through the user- friendly interface

1.1 Background:

Face recognition is crucial in daily life in order to identify family, friends or someone we are familiar with. We might not perceive that several steps have actually taken in order to identify human faces. Human intelligence allows us to receive information and interpret the information in the recognition process. We receive information through the image projected into our eyes, by specifically retina in the form of light. Light is a form of electromagnetic waves which are radiated from a source onto an object and projected to human vision. Robinson-Riegler, G., & Robinson-Riegler, B. (2008) mentioned that after visual processing done by the human visual system, we actually classify shape, size, contour and the texture of the object in order to analyze the information. The analyzed information will be compared to other representations of objects or face that exist in our memory to recognize. In fact, it is a hard challenge to build an automated system to have the same capability as a human to recognize faces. However, we need large memory to recognize different faces, for example, in the Universities, there are a lot of students with different race and gender, it is impossible to remember every face of the individual without making mistakes. In order to overcome human limitations, computers with almost limitless memory, high processing speed and power are used in face recognition systems.

The human face is a unique representation of individual identity. Thus, face recognition is defined as a biometric method in which identification of an individual is performed by comparing real-time capture image with stored images in the database of that person (Margaret Rouse, 2012).

Nowadays, face recognition system is prevalent due to its simplicity and awesome performance. For instance, airport protection systems and FBI use face recognition for criminal investigations by tracking suspects, missing children and drug activities (Robert Silk, 2017). Apart from that, Facebook which is a popular social networking website implement face recognition to allow the users to tag their friends in the photo for entertainment purposes (Sidney Fussell, 2018). Furthermore, Intel Company allows the users to use face recognition to get access to their online account (Reichert, C., 2017). Apple allows the users to unlock their mobile phone, iPhone X by using face recognition (deAgonia, M., 2017).

The work on face recognition began in 1960. Woody Bledsoe, Helen Chan Wolf and Charles Bisson had introduced a system which required the administrator to locate eyes, ears, nose and mouth from images. The distance and ratios between the located features and the common reference points are then calculated and compared. The studies are further enhanced by Goldstein, Harmon, and Lesk in 1970 by using other features such as hair colour and lip thickness to automate the recognition. In 1988, Kirby and Sirovich first suggested principle component analysis (PCA) to solve face recognition problem. Many studies on face recognition were then conducted continuously until today (Ashley DuVal, 2012).

1.2 Problem Statement:

Traditional student attendance marking technique is often facing a lot of trouble. The face recognition student attendance system emphasizes its simplicity by eliminating classical student attendance marking technique such as

calling student names or checking respective identification cards. There are not only disturbing the teaching process but also causes distraction for students during exam sessions. Apart from calling names, attendance sheet is passed around the classroom during the lecture sessions. The lecture class especially theclass with a large number of students might find it difficult to have the attendancesheet being passed around the class. Thus, face recognition attendance system isproposed in order to replace the manual signing of the presence of students whichare burdensome and causes students get distracted in order to sign for their attendance. Furthermore, the face recognition based automated student attendance system able to overcome the problem of fraudulent approach and lecturers does not have to count the number of students several times to ensure the presence of the students.

The paper proposed by Zhao, W et al. (2003) has listed the difficulties of facial identification. One of the difficulties of facial identification is the identification between known and unknown images. In addition, paper proposed by Pooja G.R et al. (2010) found out that the training process for face recognition student attendance system is slow and time-consuming. In addition, the paper proposed by Priyanka Wagh et al. (2015) mentioned that different lighting and head poses are often the problems that could degrade the performance of face recognition based student attendance system.

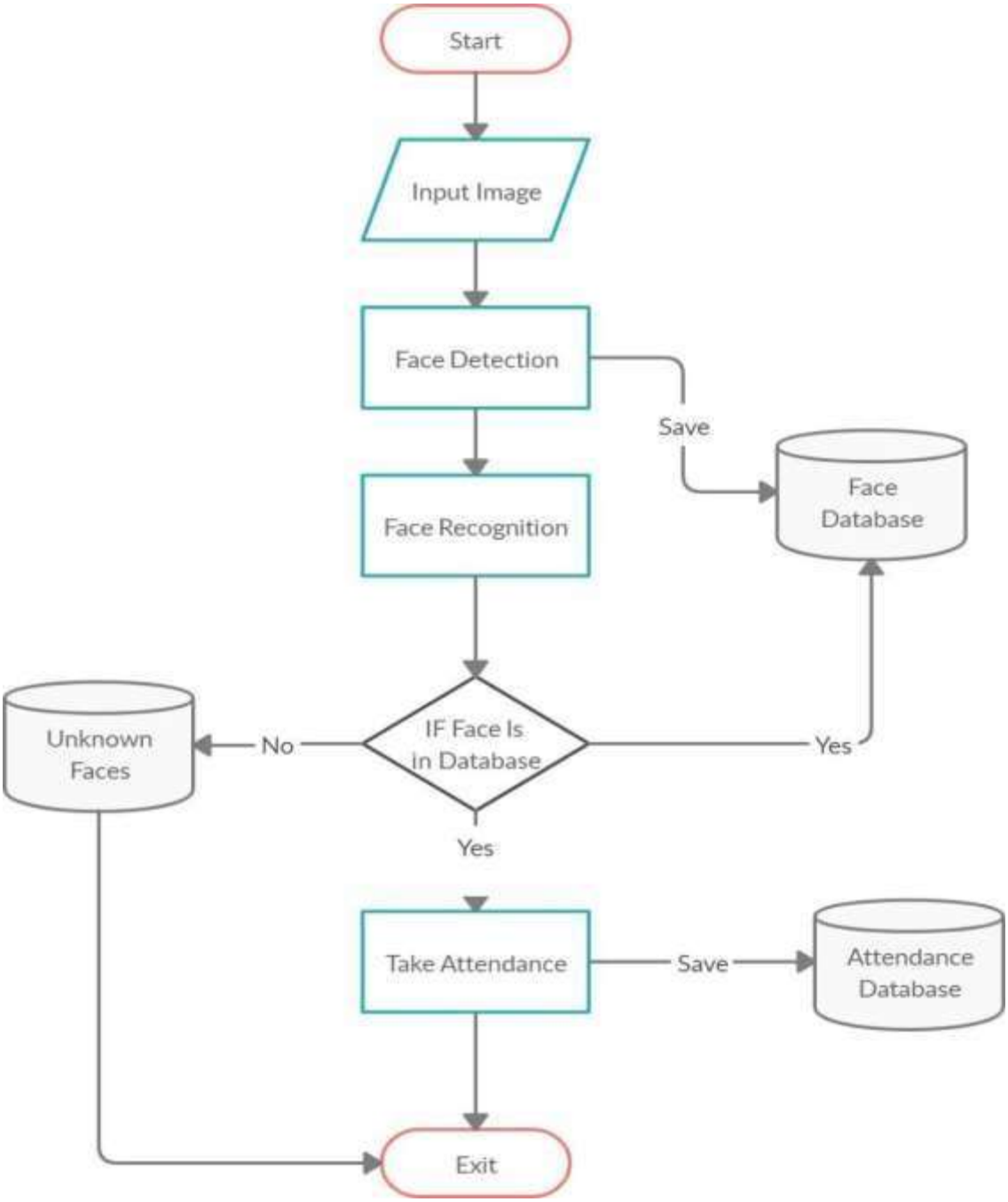
Hence, there is a need to develop a real time operating student attendance system which means the identification process must be done within defined time constraints to prevent omission. The extracted features from facial images which represent the identity of the students have to be consistent towards a change in background, illumination, pose and expression. High accuracy and fast computation time will the evaluation points of the performance

1.3 Aims and Objectives:

The objective of this project is to develop face recognition attendance system. Expected achievements in order to fulfill the objectives are:

- To detect the face segment from the video frame.
- To extract the useful features from the face detected.
- To classify the features in order to recognize the face detected.
- To record the attendance of the identified student.

1.Flow chart


Start
Input Image
Face Detection
Save
Face Database
Face Recognition
Unknown Faces
No
IF Face Is in Database
Yes
Yes
Take Attendance
Save
Attendance Database
Exit

1.4 Scope of the project:

We are setting up to design a system comprising of two modules. The first module (face detector) is a mobile component, which is basically a camera application that captures student faces and stores them in a file using computer vision face detection algorithms and face extraction techniques. The second module is a desktop application that does face recognition of the captured images (faces) in the file, marks the students register and then stores the results in a database for future analysis.

CHAPTER-2 LITERATURE REVIEW

CHAPTER-2

LITERATURE REVIEW

2.1 Student Attendance System:

Arun Katara et al. (2017) mentioned disadvantages of RFID (Radio Frequency Identification) card system, fingerprint system and iris recognition system. RFID card system is implemented due to its simplicity. However, the user tends to help their friends to check in as long as they have their friend's ID card. The fingerprint system is indeed effective but not efficient because it takes time for the verification process so the user has to line up and perform the verification one by one. However for face recognition, the human face is always exposed and contain less information compared to iris. Iris recognition system which contains more detail might invade the privacy of the user. Voice recognition is available, but it is less accurate compared to other methods. Hence, face recognition system is suggested to be implemented in the student attendance system.

System Type	Advantage	Disadvantages
RFID card system	Simple	Fraudulent usage
Fingerprint system	Accurate	Time-consuming
Voice recognition system		Less accurate compared to Others
Iris recognition system	Accurate	Privacy Invasion

Table 2.1: Advantages & Disadvantages of Different Biometric System

2.2 Digital Image Processing:

Digital Image Processing is the processing of images which are digital in nature by a digital computer. Digital image processing techniques are motivated by three major applications mainly:

- Improvement of pictorial information for human perception
- Image processing for autonomous machine application
- Efficient storage and transmission.

2.3 Image Representation in a Digital Computer:

An image is a 2-Dimensional light intensity function

$$\mathbf{f}(\mathbf{x},\mathbf{y}) = \mathbf{r}(\mathbf{x},\mathbf{y}) \times \mathbf{i}(\mathbf{x},\mathbf{y}) \text{ -(2.0)}$$

Where, r (x, y) is the reflectivity of the surface of the corresponding image point. i (x,y) Represents the intensity of the incident light. A digital image f(x, y) is discretized both in spatial co-ordinates by grids and in brightness by quantization. Effectively, the image can be represented as a matrix whose row, column indices specify a point in the image and the element value identifies gray level value at that point. These elements are referred to as pixels or pels.

Typically following image processing applications, the image size which is used is$\mathbf{256} \times \mathbf{256}$, elements, $\mathbf{640} \times \mathbf{480}$ pels or $\mathbf{1024} \times \mathbf{1024}$ pixels. Quantization of these matrix pixels is done at 8 bits for black and white images and 24 bits for colored images (because of the three color planes Red, Green and Blue each at 8 bits)[.

2.4 Steps in Digital Image Processing:

Digital image processing involves the following basic tasks:

- Image Acquisition - An imaging sensor and the capability to digitize the signal produced by the sensor.
- Preprocessing – Enhances the image quality, filtering, contrast enhancement etc.
- Segmentation – Partitions an input image into constituent parts of objects.

- Description/feature Selection – extracts the description of image objects suitable for further computer processing.
- Recognition and Interpretation – Assigning a label to the object based on the information provided by its descriptor. Interpretation assigns meaning to a set of labelled objects.
- Knowledge Base – This helps for efficient processing as well as inter module cooperation.

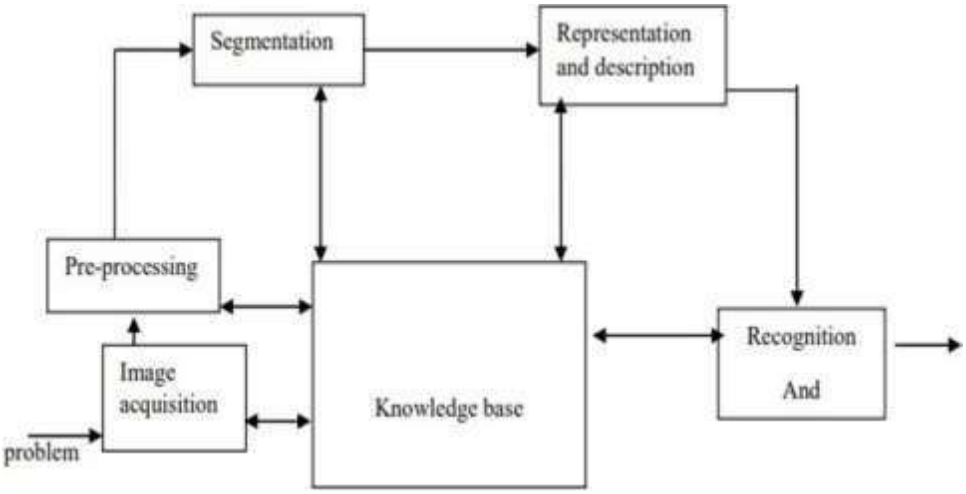


Figure 2.1 : A diagram showing the steps in digital image processing

2.5 Definition of Terms and History:

Face Detection

Face detection is the process of identifying and locating all the present faces in a single image or video regardless of their position, scale, orientation, age and expression. Furthermore, the detection should be irrespective of extraneous illumination conditions and the image and video content[5].

\

2.6 Face Recognition

Face Recognition is a visual pattern recognition problem, where the face, represented as a three dimensional object that is subject to varying illumination pose and other factors, needs to be identified based on acquired images[6].

Face Recognition is therefore simply the task of identifying an already detected face as a known or unknown face and in more advanced cases telling exactly whose face it is[7].

Difference between Face Detection and Face Recognition

Face detection answers the question, Where is the face? It identifies an object as a "face" and locates it in the input image. Face Recognition on the other hand answers the question who is this? Or whose face is it? It decides if the detected face is someone .It can therefore be seen that face detections output (the detected face) is the input to the face recognizer and the face Recognition's output is the final decision i.e. face known or face unknown.

Face Detection

A face Detector has to tell whether an image of arbitrary size contains a human face and if so, where it is. Face detection can be performed based on several cues: skin color (for faces in color images and videos, motion (for faces invideos), facial/head shape, facial appearance or a combination of these parameters.Most face detection algorithms are appearance based without using other cues. Aninput image is scanned at all possible locations and scales by a sub window. Face detection is posed as classifying the pattern in the sub window either as a face or anon-face. The face/nonface classifier is learned from face and non-face training examples using statistical learning methods[9]. Most modern algorithms are basedon the Viola Jones object detection framework, which is based on Haar Cascades

Face Detection Method	Advantages	Disadvantages
Viola Jones Algorithm	1. High detection Speed. 2. High Accuracy.	1. Long Training Time. 2.Limited Head Pose. 3.Not able to detect dark faces.
Local Binary Pattern Histogram	1.Simple computation. 2.High tolerance against the monotonic illumination changes.	1.Only used for binary and grey images. 2.Overall performance is inaccurate compared to Viola-Jones Algorithm.
Ada Boost Algorithm	Need not to have any prior knowledge about face structure.	The result highly depends on the training data and affected by weak classifiers.
SMQT Features and SNOW Classifier Method	1. Capable to deal with lighting problem in object detection. 2. Efficient in computation.	The region contain very similar to grey value regions will be misidentified as face.
Neural-Network	High accuracy only if large size of image were trained.	1. Detection process is slow and computation is complex. 2. Overall performance is weaker than Viola-Jones algorithm.

Table 2.2: Advantages & Disadvantages of Face Detection Methods

CHAPTER-3
MODAL IMPLEMENTATION
AND ANALYSIS

3.1 INTRODUCTION:

Face detection involves separating image windows into two classes; one containing faces (turning the background (clutter). It is difficult because although commonalities exist between faces, they can vary considerably in terms of age, skin color and facial expression. The problem is further complicated by differing lighting conditions, image qualities and geometries, as well as the possibility of partial occlusion and disguise. An ideal face detector would therefore be able to detect the presence of any face under any set of lighting conditions, upon any background. The face detection task can be broken down into two steps. The first step is a classification task that takes some arbitrary image as input and outputs a binary value of yes or no, indicating whether there are any faces present in the image. The second step is the face localization task that aims to take an image as input and output the location of any face or faces within that image as some bounding box with (x, y, width, height).After taking the picture the system will compare the equality of the pictures in its database and give the most related result.

We will use NVIDIA Jetson Nano Developer kit, Logitech C270 HD Webcam, open CV platform and will do the coding in python language.

3.1Modal Implementation:

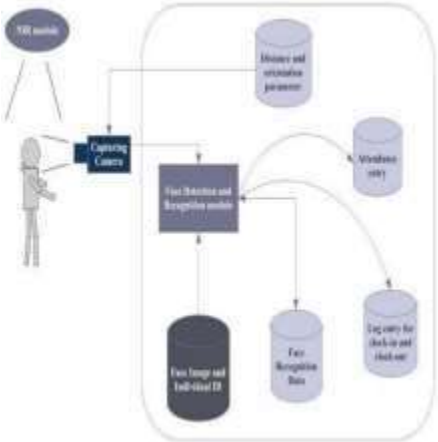


Figure 3.1: Model Implement

The main components used in the implementation approach are open source computer vision library (OpenCV). One of OpenCV's goals is to provide a simple-to-use computer vision infrastructure that helps people build fairly sophisticated vision applications quickly. OpenCV library contains over 500 functions that span many areas in vision. The primary technology behind Face recognition is OpenCV. The user stands in front of the camera keeping a minimum distance of 50cm and his image is taken as an input. The frontal face is extracted from the image then converted to gray scale and stored. The Principal component Analysis (PCA) algorithm is performed on the images and the eigen values are stored in an xml file. When a user requests for recognition the frontal face is extracted from the captured video frame through the camera. The eigen value is re-calculated for the test face and it is matched with the stored data for the closest neighbour.

3.2 Design Requirements:

We used some tools to build the system. Without the help of these tools it would not be possible to make it done. Here we will discuss about the most important one.

3.2.1 Software Implementation:

1. **OpenCV:** We used OpenCV 3 dependency for python 3. OpenCV is library where there are lots of image processing functions are available. This is very useful library for image processing. Even one can get expected outcome without writing a single code. The library is cross-platform and free for use under the open-source BSD license. Example of some supported functions are given bellow:
   - **Derivation**: Gradient/Laplacian computing, contours delimitation
   - **Hough transforms:** lines, segments, circles, and geometrical shapes detection

- **Histograms**: computing, equalization, and object localization with back projection algorithm
- **Segmentation**: thresholding, distance transform, foreground/background detection, watershed segmentation
- **Filtering**: linear and nonlinear filters, morphological operations
- **Cascade detectors**: detection of face, eye, car plates
- **Interest points**: detection and matching
- **Video processing:** optical flow, background subtraction, camshaft (object tracking)
- **Photography**: panoramas realization, high definition imaging (HDR), image inpainting

So it was very important to install OpenCV. But installing OpenCV 3 is a complex process. How we did it is given below:

```
#!/bin/bash
#Usage : sudo bash./installopencv.bash
echo OpenCV 3.0.0 Raspbian Jessie auto install script - Thomas Cyria
echo ============================================================
FILE="/tmp/out.$$"
GREP="/bin/grep"
if [ "$(id -u)" != "0" ]; then
   echo "This script must be run as root" 1>&2
   exit 1
fi
echo installing core dependencies ...
apt-get -y install cmake python3-dev python3.4-dev python3-numpy gcc build-essential cmake-curses-gui
echo installing other dependencies ...
apt-get -y install pkg-config libpng12-0 libpng12-dev libpng++-dev libpng3 libpnglite-dev zlib1g-dbg zlib1g
zlib1g-dev pngtools libtiff5-dev libtiff5 libtiffxx5c2 libtiff-tools libeigen3-dev
echo installing helper apps ...
apt-get -y libav-tools
#apt-get -y ffmpeg libavcodec55 libavformat55
apt-get -y install libjpeg8 libjpeg8-dev libjpeg8-dbg libjpeg-progs libavcodec-dev libavformat-dev libgstreamer0.10-0-dbg
libgstreamer0.10-0 libgstreamer0.10-dev libaine3-ffmpeg libaine1-dev libaine1-bin libunicap2 libunicap2-dev swig libv4l-0 libv4l-dev libpython3.4 libgtk2.0-dev
echo Recovering OpenCV 3.0.0 source...
git clone --branch 3.0.0 --depth 1 https://github.com/Itseez/opencv.git
cd opencv
mkdir release
cd release
echo Preparing compilation, may take a long while...
cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=$(python3 -c "import sys; print(sys.prefix)") -D PYTHON_EXECUTABLE=$(which python3) ..
echo Compiling Open CV 3.0.0, may take 2 to 36 hours
make -j4
echo Compilation Ok, installing...
make install
cd ../..
rm -r opencv
echo Completed !
echo You now can use OpenCV 3.0.0 in both Python 2 and Python 3 !
```

Fig 3.2: Installing OpenCV

We copied this script and place it on a directory on our raspberry pi and saved it. Then through terminal we made this script executable and then ran it.

```
Sudo chmod 755 /myfile/pi/installopencv.bash
sudo /myfile/pi/installopencv.bash
```

these are the command line we used.

2. **Python IDE:** There are lots of IDEs for python. Some of them are PyCharm, Thonny, Ninja, Spyder etc. Ninja and Spyder both are very excellent and free but we used Spyder as it feature- rich than ninja. Spyder is a little bit heavier than ninja but still much lighter than PyCharm. You can run them in pi and get GUI on your PC through ssh-Y. We installed Spyder through the command line below

```
1. sudo apt-get isntall     spyder
```

3.3 Experimental Results:

The step of the experiments process are given below:

Face Detection:

Start capturing images through web camera of the client side: Begin:

- Pre-process the captured image and extract face image
- calculate the eigen value of the captured face image and compared with eigen values of existing faces in the database.
- If eigen value does not matched with existing ones,save the new face image information to the face database (xml file).
- If eigen value matched with existing one then recognition step will done.

End

Face Recognition:

Using PCA algorithm the following steps would be followed in for face recognition:
Begin:

- Find the face information of matched face image in from the database.
- update the log table with corresponding face image and system time that makes completion of attendance for an individua students.

End

This section presents the results of the experiment conducted to capture the face into a grey scale image of 50x50 pixels.

Test data	Expected Result	Observed Result	Pass/ Fail
OpenCAM_CB()	Connects with the installed camera and starts playing.	Camera started.	pass
LoadHaar Classifier()	Loads the HaarClassifier Cascade files for frontal face	Gets ready for Extraction.	Pass
ExtractFace()	Initiates the Paul-Viola Face extracting Frame work.	Face extracted	Pass
Learn()	Start the PCA Algorithm	Updates the facedata. xml	Pass
Recognize()	It compares the input face with the saved faces.	Nearest face	Pass

Table 3.2 Experimental Results-1

Here is our data set sample.

Face Orientations	Detection Rate	Recognition Rate
0º (Frontal face)	98.7 %	95%
18º	80.0 %	78%
54º	59.2 %	58%
72º	0.00 %	0.00%
90º(Profile face)	0.00 %	0.00%

Table 3.3 Experimentaal Results-2

We performed a set of experiments to demonstrate the efficiency of the proposed method. 30 different images of 10 persons are used in training set. Figure 3 shows a sample binary image detected by the ExtractFace() function using Paul-Viola Face extracting Frame work detection method.

CHAPTER-4
CODE IMPLEMENTATION

4.1 Code Implementation:

All our code is written in Python language. First here is our project directory structure and files.

```
FRASJN
    |___[Attendance]
    |___[ImagesUnknown]
    |___[StudentDetails]
    |___[TrainingImage]
    |___[Traininglabel]
    |___main.py
    |___automail.py
    |___CaptureImage.py
    |___check_camera.py
    |___haarcascade_frontalface_default.xml
    |_____recognize.py
```

All those file in the project directory.

```
Note: The names inside square brackets ["folder name"] indicate it is a folder.
[Attendance]          => It contains all the attendance sheets saved after taking attendance.
[ImagesUnknown]  => Unknown images are placed inside this folder to avoid false positives.
[StudentDetails]     => Here we place Studentdetails.csv file to use while recognizing faces.
[Trainingimage]     => After capture dataset of a student, all his/her images are stored here.
```

- main.py

All the work will be done here, Detect the face ,recognize the faces and take attendance.

```
import os  # accessing the os functions
import check_camera
import Capture_Image
import Train_Image
import Recognize


# creating the title bar function

def title_bar():
    os.system('cls')  # for windows

    # title of the program


print("\t**********************************************************************")
    print("\t***** Face Recognition Attendance System using jetson
nano*****")

print("\t**********************************************************************")


# creating the user main menu function

def mainMenu():
    title_bar()
    print()
    print(10 * "*", "WELCOME MENU", 10 * "*")
    print("[1] Check Camera")
    print("[2] Capture Faces")
    print("[3] Train Images")
    print("[4] Recognize & Attendance")
    print("[5] Auto Mail")
    print("[6] Quit")

    while True:
        try:
            choice = int(input("Enter Choice: "))

            if choice == 1:
                checkCamera()
                break
            elif choice == 2:
                CaptureFaces()
                break
            elif choice == 3:
                Trainimages()
                break
            elif choice == 4:
                RecognizeFaces()
                break
            elif choice == 5:
                os.system("py automail.py")
                break
                mainMenu()
            elif choice == 6:
                print("Thank You")
```

```
                break
            else:
                print("Invalid Choice. Enter 1-6")
                mainMenu()
        except ValueError:
            print("Invalid Choice. Enter 1-6\n Try Again")


# ---------------------------------------------------------
# calling the camera test function from check camera.py file

def checkCamera():
    check_camera.camer()
    key = input("Press Enter to return main menu")
    mainMenu()


# ---------------------------------------------------------------
# calling the take image function form capture image.py file

def CaptureFaces():
    Capture_Image.takeImages()
    key = input("Press Enter to return main menu")
    mainMenu()


# ------------------------------------------------------------------
# calling the train images from train_images.py file

def Trainimages():
    Train_Image.TrainImages()
    key = input("Press Enter to return main menu")
    mainMenu()


# ---------------------------------------------------------------------
# calling the recognize_attendance from recognize.py file

def RecognizeFaces():
    Recognize.recognize_attendence()
    key = input("Press Enter to return main menu")
    mainMenu()


# ---------------main driver ------------------
mainMenu()
```

- automail.py

In this project we add an extra feature called auto mail. It can automatically sent the attendance file to specific mail. Auto mail code given below.

```
import yagmail
import os
import datetime
date = datetime.date.today().strftime("%B %d, %Y")
path = 'Attendance'
os.chdir(path)
files = sorted(os.listdir(os.getcwd()), key=os.path.getmtime)
newest = files[-1]
filename = newest
sub = "Attendance Report for " + str(date)
# mail information
yag = yagmail.SMTP(user = "SenderEmailAddress@domain.com", password =
"Password of that Account should be inserted here")

# sent the mail
yag.send(
    to="Receiver email adress",
    subject= sub, # email subject
    contents= "None",  # email body
    attachments= filename  # file attached
)
print("Email Sent!")
```

- Capture_Image.py

This capture_image.py will collect the data set of a student and add his/her name

in tha StudentsDetails.csv

```
import csv

import cv2
import os


# counting the numbers


def is_number(s):
    try:
        float(s)
        return True
    except ValueError:
        pass

    try:
        import unicodedata
        unicodedata.numeric(s)
        return True
    except (TypeError, ValueError):
        pass

    return False


# Take image function

def takeImages():


    Id = input("Enter Your Id: ")
    name = input("Enter Your Name: ")

    if(is_number(Id) and name.isalpha()):
        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,
minSize=(30,30),flags = cv2.CASCADE_SCALE_IMAGE)
            for(x,y,w,h) in faces:
                cv2.rectangle(img, (x, y), (x+w, y+h), (10, 159, 255), 2)
                #incrementing sample number
                sampleNum = sampleNum+1
                #saving the captured face in the dataset folder TrainingImage
                cv2.imwrite("TrainingImage" + os.sep +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(100) & 0xFF == ord('q'):
```

```
                break
            # break if the sample number is more than 100
            elif sampleNum > 100:
                break
        cam.release()
        cv2.destroyAllWindows()
        res = "Images Saved for ID : " + Id + " Name : " + name
        row = [Id, name]
        with open("StudentDetails"+os.sep+"StudentDetails.csv", 'a+') as
csvFile:
            writer = csv.writer(csvFile)
            writer.writerow(row)
        csvFile.close()
    else:
        if(is_number(Id)):
            print("Enter Alphabetical Name")
        if(name.isalpha()):
            print("Enter Numeric ID")
```

- checkcamera.py

This checkcamra.py will check weather the camera is correctly connected or not, if connected whether the face is detecting or not.

```
def camer():
    import cv2
    # Load the cascade
    face_cascade =
cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
    # To capture video from webcam.
    cap = cv2.VideoCapture(0)
    cnt = 0
    while True:
        # Read the frame
        _, img = cap.read()
        # Convert to grayscale
        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        # Detect the faces
        faces = face_cascade.detectMultiScale(gray, 1.3, 5, minSize=(30,
30),flags = cv2.CASCADE_SCALE_IMAGE)
        # Draw the rectangle around each face
        for (x, y, w, h) in faces:
            cv2.rectangle(img, (x, y), (x + w, y + h), (10,159,255), 2)
        # Display
        cv2.imwrite("Avinash"+str(cnt)+".jpg",gray)
        cv2.imshow('Webcam Check', img)
        cnt += 1
        # Stop if escape key is pressed
        if (cv2.waitKey(1) & 0xFF == ord('q')) or cnt==100:
            break
    # Release the VideoCapture object
    cap.release()
    cv2.destroyAllWindows()
```

- Train_Image.py

All the images in the Training Image folder will be accessed here and a model is created by using this trainimage.py file.

```
import os
import time
import cv2
import numpy as np
from PIL import Image
from threading import Thread
# ------------- image labesl -----------------------
def getImagesAndLabels(path):
    # get the path of all the files in the folder
    imagePaths = [os.path.join(path, f) for f in os.listdir(path)]
    # print(imagePaths)
    # create empth face list
    faces = []
    # create empty ID list
    Ids = []
    # now looping through all the image paths and loading the Ids and the
images
    for imagePath in imagePaths:
        # loading the image and converting it to gray scale
        pilImage = Image.open(imagePath).convert('L')
        # Now we are converting the PIL image into numpy array
        imageNp = np.array(pilImage, 'uint8')
        # getting the Id from the image
        Id = int(os.path.split(imagePath)[-1].split(".")[1])
        # extract the face from the training image sample
        faces.append(imageNp)
        Ids.append(Id)
    return faces, Ids
# ---------- train images function --------------
def TrainImages():
    recognizer = cv2.face_LBPHFaceRecognizer.create()
    harcascadePath = "haarcascade_frontalface_default.xml"
    detector = cv2.CascadeClassifier(harcascadePath)
    faces, Id = getImagesAndLabels("TrainingImage")
    Thread(target = recognizer.train(faces, np.array(Id))).start()
    # Below line is optional for a visual counter effect
    Thread(target = counter_img("TrainingImage")).start()
    recognizer.save("TrainingImageLabel"+os.sep+"Trainner.yml")
    print("All Images")
# Optional, adds a counter for images trained (You can remove it)
def counter_img(path):
    imgcounter = 1
    imagePaths = [os.path.join(path, f) for f in os.listdir(path)]
    for imagePath in imagePaths:
        print(str(imgcounter) + " Images Trained", end="\r")
        time.sleep(0.008)
        imgcounter += 1
    return
```

- Recognize.py

When this Recognize.py file is executed, camera will be opened and it will recognize all the students present in this Students.csv file and those who are present it will mark attendance automatically and save in Attendance folder with date and time.

```
import datetime
import os
import time
import cv2
import pandas as pd
#-----------------------
def recognize_attendence():
    recognizer = cv2.face.LBPHFaceRecognizer_create()  #
cv2.createLBPHFaceRecognizer()
    recognizer.read("TrainingImageLabel"+os.sep+"Trainner.yml")
    harcascadePath = "haarcascade_frontalface_default.xml"
    faceCascade = cv2.CascadeClassifier(harcascadePath)
    df = pd.read_csv("StudentDetails"+os.sep+"StudentDetails.csv")
    font = cv2.FONT_HERSHEY_SIMPLEX
    col_names = ['Id', 'Name', 'Date', 'Time']
    attendance = pd.DataFrame(columns=col_names)
    # Initialize and start realtime video capture
    cam = cv2.VideoCapture(0, cv2.CAP_DSHOW)
    cam.set(3, 640)  # set video width
    cam.set(4, 480)  # set video height
    # Define min window size to be recognized as a face
    minW = 0.1 * cam.get(3)
    minH = 0.1 * cam.get(4)
    while True:
        ret, im = cam.read()
        gray = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)
        faces = faceCascade.detectMultiScale(gray, 1.2, 5,minSize =
(int(minW), int(minH)),flags = cv2.CASCADE_SCALE_IMAGE)
        for(x, y, w, h) in faces:
            cv2.rectangle(im, (x, y), (x+w, y+h), (10, 159, 255), 2)
            Id, conf = recognizer.predict(gray[y:y+h, x:x+w])
            if conf < 100:
                aa = df.loc[df['Id'] == Id]['Name'].values
                confstr = "  {0}%".format(round(100 - conf))
                tt = str(Id)+"-"+aa
            else:
                Id = '  Unknown  '
                tt = str(Id)
                confstr = "  {0}%".format(round(100 - conf))
            if (100-conf) > 67:
                ts = time.time()
                date = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-
%d')
                timeStamp =
datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
                aa = str(aa)[2:-2]
                attendance.loc[len(attendance)] = [Id, aa, date, timeStamp]
```

```
            tt = str(tt)[2:-2]
            if(100-conf) > 67:
                tt = tt + " [Pass]"
                cv2.putText(im,str(tt),(x+5,y-5),font,1,(255,255,255),2)
            else:
                cv2.putText(im,str(tt),(x+5,y-5),font,1,(255,255,255),2)
            if (100-conf) > 67:
                cv2.putText(im,str(confstr),(x+5,y+h-5),font,1,(0,255,0),1)
            elif (100-conf) > 50:
                cv2.putText(im,str(confstr),(x+5,y+h-5),font,1,(0,255,255),1)
            else:
                cv2.putText(im,str(confstr),(x+5,y+h-5),font,1,(0,0,255),1)
        attendance = attendance.drop_duplicates(subset=['Id'], keep='first')
        cv2.imshow('Attendance', im)
        if (cv2.waitKey(1) == ord('q')):
            break
    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"+os.sep+"Attendance_"+date+"_"+Hour+"-"+Minute+"-
"+Second+".csv"
    attendance.to_csv(fileName, index=False)
    print("Attendance Successful")
    cam.release()
    cv2.destroyAllWindows()
```

- requirements.txt

This file consists all the required files to be install before executing the codes.

```
pip install opencv-contrib-python
pip install numpy
pip install pandas
pip install Pillow
pip install pytest-shutil
```

```
pip install python-csv
pip install yagmail
```

We can make use of the above commands or we can run a simple command with thw requirements.txt file

```
pip install -r requirements.txt
```

The text file consists:

opencv-contrib-python numpy
pandas

pillow

pytest-shutil

python-csv

CHAPTER-5
WORK PLAN

5.1 Introduction:

A project work plan allows you to outline the requirements of a project, project planning steps, goals, and team members involved in the project.Within each goal, you're going to outline the necessary Key Action Steps in project planning, the requirements, and who's involved in each action step.

Key Action Step:

- Expected Outcome -Add this as a task. The Expected outcome will be the part of Project
- Assignees – Assigning the work to the team members.
- Completion Date -Add a due date and tries to finish the work within the time

Work Breakdown Structure:

In order to develop this system, we gave enormous importance to scheduling because we believed if we want to provide the best of quality in a given period of time then we must give due importance to scheduling which also helped us to achieve a better results.we observe the entire work structure, meaning how the scheduling was maintained throughout the developmental phase. We shall also see the financial foundation of this project and furthermore the feasibility study should be also discussed.

Table 5.1 Work Plan

Activity	status
Selection of project area and Study of the related work.	Completed
Literature Survey and Study of Journals related to the work	Completed
Study on the software implementation works python and image processing	Completed
Study of project related works like face recognition and detection techniques	Completed
Study of the Image processing in python and Open Computer Vision	Completed
Study of hardware and selection of components	Completed
Study of hardware implementation and installation OS	Completed
Study related to creating the environments and working platform	Completed
Study of packages/tools and installation of packages	Completed
Implementation of algorithm in Software.	Completed
Implementation of code in hardware	Completed

CHAPTER-6
PERFORMANCE ANALYSIS

6.1 Introduction:

We conducted a series of experiments to illustrate the system performance under different situations. By carrying out those tests, we were able to get the graph shown above (Distance vs Confidence Level). We may deduce from the graph that when the face is closer to the camera, the confidence level is higher, and vice versa. Therefore, by keeping a threshold for confidence level, we can mark attendance to the person according to the threshold.

6.2 Analysis:

Here we consider one constant parameter intensity of light . we performed different experiments on different distance and different angles. we observed the confidence level at the different positions by gradually increasing the distance .we plotted the graph using the x and y coordinates by considering the x values as the confidence level or accuracy rate. and y values as the distance (cms).

CONCLUSION

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. The face recognition system implementation can be part of Universities. Face Recognition Based Attendance System has been envisioned for the purpose of reducing the errors that occur in the traditional (manual) attendance taking system. The aim is to automate and make a system that is useful to the organization such as an institute. The efficient and accurate method of attendance in the office environment that can replace the old manual methods. This method is secure enough, reliable and available for use. Proposed algorithm is capable of detect multiple faces, and performance of system has acceptable good results.

REFERENCES

[1]. *A brief history of Facial Recognition*, NEC, New Zealand,26 May 2020.[Online]. Available:https://www.nec.co.nz/market-leadership/publications-media/a-brief-history-of-facial- recognition/

[2]. *Face detection,TechTarget Network*, Corinne Bernstein, Feb, 2020.[Online]. Available: https://searchenterpriseai.techtarget.com/definition/face-detection

[3]. Paul Viola and Michael Jones, *Rapid Object Detection using a Boosted Cascade of SimpleFeatures.* Accepted Conference on Computer Vision and Pattern Re cognition, 2001.

[4]. *Face Detection with Haar Cascade*,Towards Data Science-727f68dafd08,Girija Shankar Behera, India, Dec 24, 2020.[Online]. Available:https://towardsdatascience.com/face-detection- with-haar-cascade-727f68dafd08

[5]. *Face Recognition: Understanding LBPH Algorithm,*Towards Data Science-90ec258c3d6b,Kelvin Salton do Prado, Nov 11, 2017.[Online]. Available :https://towardsdatascience.com/face-recognition-how-lbph-works-90ec258c3d6b

[6]. *What is Facial Recognition and how sinister is it*, Theguardian, IanSample, July, 2019. [Online]. Available: https://www.theguardian.com/technology/2019/jul/29/what-is-facial- recognition-and-how-sinister-is-it

[7].Kushsairy Kadir , Mohd Khairi Kamaruddin, Haidawati Nasir, Sairul I Safie, Zulkifli Abdul Kadir Bakti,"A comparative study between LBP and Haar-like features for Face Detection using OpenCV*", 4th International Conference on Engineering Technology and Technopreneuship (ICE2T),* DOI:10.1109/ICE2T.2014.7006273, 12 January 2015.

[8].Senthamizh Selvi.R,D.Sivakumar, Sandhya.J.S , Siva Sowmiya.S, Ramya.S , Kanaga SubaRaja.S,"Face Recognition Using Haar - Cascade Classifier for Criminal Identification", *International Journal of Recent Technology and Engineering(IJRTE)*, vol.7, issn:2277-3878, ,issue-6S5, April 2019.

Student Details

1. Name: Akhil Pandey
   Enrollment Number: 211B032
   Contact: 9305828258
   Email:211B032@juetguna.in

2. Name: Bicky Kumar
   Enrollment Number: 211B094
   Contact: 9801841167
   Email: 211B094@juetguna.in

3. Name: Priyadarshi Kumar
   Enrollment Number: 211B227
   Contact: 9650387593
   Email:211B227@juetguna.in

S