**SMART ATTENDANCE SYSTEM USING COMPUTER VISION**

A Project Report submitted to

**Jawaharlal Nehru Technological University, Hyderabad.**

In partial fulfillment for the requirement for the award of B.Tech Degree in Computer

Science and Engineering

By

**S.VISHNUPRIYA 17641A0558**

**T.BINDUMATHI 17641A0519**

**ALAPATI SRIDHAR REDDY 17641A0513**

**A.CHANDU 17641A0533**

Under the guidance of

**Mr. Shekar,**

**Assistant Professor**



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**VAAGDEVI COLLEGE OF ENGINEERING**

(UGC Autonomous, Accredited by NBA, Accredited by NAAC with “A”)

Warangal – 506001

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**VAAGDEVI COLLEGE OF ENGINEERING**

**UGC Autonomous**

**Warangal**



**CERTIFICATE**

This is to certify that the Project entitled **SMART ATTENDANCE SYSTEM USING COMPUTER VISION** is submitted by **S.VISHNUPRIYA** bearing 17641A0558 ,**T.BINDUMATHI** bearing 17641A0519, **A.SRIDHAR REDDY** bearing **17641A0513,A.CHANDU bearing 17641A0533** in partial fulfillment of the requirements for the award of the Degree in Bachelor of Technology in **COMPUTER SCIENCE AND ENGINEERING** during the academic year 2020-2021.

**Guide Head of the Department External Examiner**

**ACKNOWLEDGEMENT**

We would like to express our sincere gratitude to **MR. SHEKAR**, Assistant Professor, whose role as project guide was invaluable for the project. We are extremely thankful for the keen interest he took in advising us, for the books and reference materials provided for the moral support extended to us.

Last but not the least we convey our gratitude to all the teachers for providing us the technical skill that will always remain as our asset and to all non- teaching staff for the gracious hospitality they offered us.

**ABSTRACT**

The conventional attendance system requires manual marking of attendance, which is a time-consuming process and hence is inefficient. So to digitalize the system, we have proposed an automated student attendance system based on face recognition. Face is the representation of one’s identity. Face recognition systems are very useful and powerful in applications especially insecurity control systems in airports and crime investigations.

In our proposed approach, firstly, attendance marking is performed by activating the camera through a user-friendly interface. The system is built by five modules - Student Registration (creating datasets and storing in a database), Training the model with the datasets, Face detection and Recognition, Automatic Attendance Marking and Attendance Report Generation. The system is built using OpenCV, Python and modules like tkinter for UI, openpyxl for automating excel, sqlite database for storing datasets etc.,

The core of the project which is the face recognition can be used in many other applications where face recognition is used for thentication.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| S.No. | Chapters | Page No’s |
| 1. | Introduction | 1 |
| 2. | Design Of the Project | 5 |
| 3. | Implementation | 13 |
| 4. | Testing | 18 |
| 5. | Results | 20 |
| 6. | Conclusion & Future Scope | 23 |
| 7. | Bibliography | 24 |

**CHAPTER-1 INTRODUCTION**

In many institutions and organizations, attendance is a very important factor to maintain the record of lectures, salary and work hours etc. Most of the institutes and organizations follow the manual method using old paper and file methods and some of them have shifted to biometric techniques. The current method that colleges use is that the professor passes a sheet or makes roll calls and marks the attendance of the students and this sheet further goes to the admin department which updates the final excel sheet. This process is quite hectic and time consuming. Also, for professors or employees at institutes or organizations the biometric system serves one at a time. So, why not shift to an automated attendance system, which works on face recognition technique? Be it a classroom or entry gates it will mark the attendance of the students, professors, employees, etc.

Facial image has set out to be an important biometric feature, which easily is acquirable and doesn’t require any special or physical interaction between the subject and the device. As it is observed, image recognition is very complex and challenging one affecting a variety of parameters such as intensity, orientation, expression and size. Face recognition can be applied for a wide variety of problems like image and film processing, human-computer interaction, criminal identification etc. This has motivated researchers to develop computational models to identify the faces, which are relatively simple and easy to implement. The goal is to implement the system (model) for a particular face and distinguish it from a large number of stored faces with some real-time variations as well. The LBPH (Local Binary Patterns Histograms) approach is used for the recognition of the images. It gives us an efficient way to find the lower dimensional space.

**1.1 PROBLEM STATEMENT**

Maintaining attendance is essential in every foundation for checking the performance of students as well as employees. Each organization has its own technique. Traditionally a student's attendance is taken physically on the attendance register or sheet, given by the employee in class. These stamping techniques are repetitive and tedious. Physically recorded participation can be effortlessly controlled. Besides, it is exceptionally hard to confirm one by one student in a substantial classroom environment with disseminated branches whether the verified students are really reacting or not. The previous approach in which manually taking and maintaining the attendance records was a very inconvenient task. Traditionally, student’s attendances are taken manually by using the attendance sheet given by the faculty members in class, which is a time consuming event. Moreover, it is very difficult to verify one by one student in a large classroom environment with distributed branches whether the authenticated students are actually responding or not. The ability to compute the attendance percentage becomes a major task as manual computation produces errors, and also wastes a lot of time. This method could easily allow for impersonation and the attendance sheet could be stolen or lost. An automatic attendance management system using biometrics would provide the needed solution. The results showed improved performance over the manual attendance management system. Biometric Based techniques have emerged as the most promising option for recognizing individuals in recent years since, instead of authenticating people and granting them access to physical and virtual domains based on passwords, PINs, smart cards, plastic cards, tokens, keys and so forth, these methods examine an individual’s physiological and/or behavioural characteristics in order to determine and/or ascertain his identity.

# 1.2 OBJECTIVE

The main goal and objective of this automated Attendance System using Computer Vision is to present face recognition in a real time environment, to see and mark the attendance of their students and employees on a daily basis to keep track of their presence. The system will mark and record the attendance in any environment. This system is automated and users can capture video and accordingly attendance will be marked, improving the accuracy to great extent and finally the attendance report will be generated.

## **1.3 MOTIVATION**

The main motive behind developing this system is to eliminate all the drawbacks, which were associated with manual attendance systems. The drawbacks ranging from wastage of time and paper, till the proxy issues arising in a class, are eliminated. Hence, desired results with a user-friendly interface is expected in the future, from the system. The efficiency of the system could also be increased by integrating various steps and techniques in the future developing stages of the system.

# 1.4 EXISTING SYSTEM

Preserving the attendance is very crucial in all the institutes for checking the overall performance of students. Each institute has its very own method in this regard. A few are taking attendance manually using the old paper or document based approach and some have adopted techniques of automated attendance using a few biometric techniques. There are many computerized methods to be had for this reason i.e. biometric attendance. Biometric based technologies include identification based on physiological characteristics (such as face, fingerprints, finger geometry, hand geometry, hand veins, palm, iris, retina, ear and voice) and behavioural traits (such as gait, signature and keystroke dynamics). Face recognition appears to offer several advantages over other biometric methods, a few of which are outlined here: Almost all these technologies require some voluntary action by the user, i.e., the user needs to place his hand on a hand-rest for fingerprinting or hand geometry detection and has to stand in a fixed position in front of a camera for iris or retina identification.

# 1.5 PROPOSED SYSTEM

The proposed system aims to develop an automated attendance system. To achieve the project objective, firstly, video segments are captured. Pre – Processing of video is done to remove unwanted artifacts i.e. noise and other invariants. The next stage demonstrates detection of faces from the complex backgrounds and recognition of human beings. This system helps to identify students to track his/her presence in the lecture and to avoid proxy attendance caused by unauthorized students. There are four stages of operation to develop the system: Video acquisition, detection of faces and cropping, extraction of features and recognition of face. Students will be first registered in the system with proper details, and facial images captured from different angles and positions. On successful completion of the registration process, store student’s data. Video Acquisition is done by capturing the video of the class being conducted in a classroom. Acquired video is used to detect and recognize faces of different students and differentiate them from background using image processing techniques i.e. Viola – Jones Algorithm for face detection, Cropping of faces and Fisher Face Algorithm for face recognition. Student’s identity verification is done by comparing facial images of the students with the faces stored in the database. If the faces of students are matched, then their attendance is recorded and updated in the system.

# 1.6 SCOPE

The algorithm will be tested with multiple students in the scene and also captured faces at different angles in the scene. The algorithm delivers good results but there is room to improve the algorithm performance in case of large numbers of students and also in case of faces captured in a dark environment, so the proposed system can be extended in the future to cover this aspect. The efficiency of the algorithm also can be increased further so there is also a room for future work in this area. This system can be enhanced further in terms of achieving more efficiency by ease of analysis of patterns in the data. To be used efficiently, all computer software needs certain hardware components or other software resources to be present on a computer. These prerequisites are known as (computer)

The supported Operating Systems for clients include: Windows 2010, windows 2008, windows 2007.

# 

# 1.7 SOFTWARE REQUIREMENTS

### The Software Requirements in this project include:

### a. Python

1. OpenCV framework
2. MS-Excel

Software requirements deal with defining software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application. OpenCV (Open-Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code. The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms.

These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. OpenCV has more than 47 thousand people in the user community. The library is used extensively in companies, research groups and by governmental bodies. As an asynchronous event driven framework.

**1.8 HARDWARE REQUIREMENTS**

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatible, and sometimes incompatible hardware devices for a particular operating system or application.

|  |  |  |
| --- | --- | --- |
| Components | Minimum | Recommended |
| Processor | Intel Core i3 | Intel Core i7 5th GEN |
| RAM | 4GB | 8GB |
| Camera | HD 720p Webcam | Full HD 1080p Webcam |
| Disk | 128Gb | 512Gb |

**CHAPTER-2.DESIGN OF THE PROJECT**

**2.1 FACE DETECTION AND RECOGNITION**

Face recognition has proven to be a good candidate for attendance monitoring and has a room for improvement. Face detection is the first stage of facial recognition . Reliability in face detection remains an issue, with challenges such as multi-view appearance, illumination invariant system, pose estimation, quality of the camera, and resolution of the images. The detection process must be designed in a manner that the cameras would be able to capture every targeted face. The Detection algorithm comprises of the Haar-features,Adaboost,Integral image and cascading classifier

**2.2 ALGORITHMS OF THE PROJECT**

**a) Viola Jones**

Viola Jones was the one to formulate the first ever real-time face detection algorithm which helped to detect faces from the images. In this module, a complete algorithmic description is implemented which includes feature computation using integral images, selection of features with the help of adaboost training and cascading for efficient allocation of computational resources. This algorithm provides fast and accurate detection.

To compute face detection, Viola – Jones algorithm is used. Viola Jones algorithm is divided in four phases:

**1. Haar Feature Selection :**

In Haar feature selection, compute scalar product between the image and Haar templates. Then, calculate the difference between the number of black pixels and number of white pixels to obtain numerous features.All the images are normalized using mean and variance to recoup the effect of different lighting conditions. Images having variance value lower than one with little information of interest are excluded.

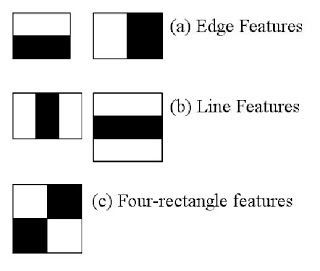


figure 1. Haar features

Fig 1 shows Five Haar patterns that are used to compute various features from the facial images. These haar patterns marked with black or white pixels are moved over an image to compute all the features. These features help to detect faces from images with required computation.

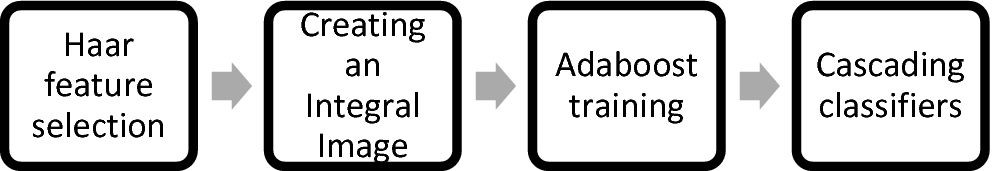


figure 2.Viola Jones Algorithm

# 2. Creating an Integral Image

Integral Image is an effective way of computing the summation of pixel values in a given image. It is mainly used to compute the mean intensity value within a given image. Firstly, create an integral image which helps to compute the value at each pixel (o,p) which is the addition of pixels above (o,p) and pixels to the left of (o,p) inside a rectangular window. For integral image, the value in the

Summed Area Table at (o,p) is simply calculated by :

Q(o,p) = E(o,p) + Q(o-1,p) + Q(o,p-1) – Q(o-1,p-1)

where E(o,p) is the value of the pixel at (o,p), Q(o1,p) is pixel value directly to the left of (o,p), Q(o,p1) is pixel value directly to the right of (o,p) and Q(o1,p-1) is the pixel value to the top -left of (o,p).

# 

# 3.Adaboost Training

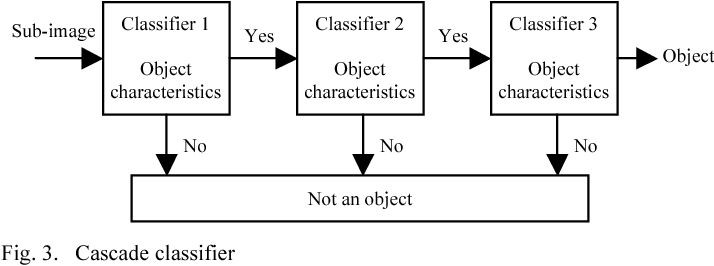
Adaboost training is used to select a subset of features and to construct the classifier. Adaboost refers to a particular method of training a boost classifier. In the Boost classifier every weak learner takes input in the form of object and returns the value showing the class of the object. This technique creates strong classifiers from a number of weak classifiers. To amplify the performance of the system on classification problems, adaboost training is done.

The strong classifier is formulated from the number of weak classifier:

S(x) = m1s1(x) + m2s2(x) + m3s3(x) + ......

where S(x) is a strong classifier and m1s1(x) is a weak classifier

**4.Cascading Classifiers**



Cascading is a peculiar case of learning which concatenates multiple classifiers. All information gathered from the output of the given classifier is used as an additional data for the next classifier in the cascade. Cascading classifiers are trained with several hundreds of "positive" sample images of face and arbitrary "negative" images (i.e. background). Both the positive and negative images must be of the same size. Once the classifier is trained it is applied to an image to detect the faces. To search for the faces from the entire image frame, the search window travels across the image and checks every location of the classifier. ​

**b)Local Binary Pattern algorithm**

Local Binary Pattern (LBP) is a simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number*.*

It was first described in 1994 (LBP) and has since been found to be a powerful feature for texture classification. It has further been determined that when LBP is combined with histograms of oriented gradients (HOG) descriptor, it improves the detection performance considerably on some datasets. Using the LBP combined with histograms we can represent the face images with a simple data vector. As LBP is a visual descriptor it can also be used for face recognition tasks, as can be seen in the following step-by-step explanation.

**STEP 1**:

**Parameters**: the LBPH uses 4 parameters:

**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.

**Neighbors**: 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.

**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.

**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.

**STEP 2:**

**Training the Algorithm**: First, we need to train the algorithm. To do so, we need to use a dataset with the facial images of the people we want to recognize.

**STEP 3:**

**Applying the LBP operation**: The first computational step of the LBPH is to create an intermediate image that describes the original image in a better way, by highlighting the facial characteristics. To do so, the algorithm uses a concept of a sliding window, based on the parameters **radius** and **neighbors**.

The image below shows this procedure:



Suppose we have a facial image in grayscale.

a)We can get part of this image as a window of 3x3 pixels.

It can also be represented as a 3x3 matrix containing the intensity of each pixel (0~255).

b)Then, we need to take the central value of the matrix to be used as the threshold.

c)This value will be used to define the new values from the 8 neighbors.

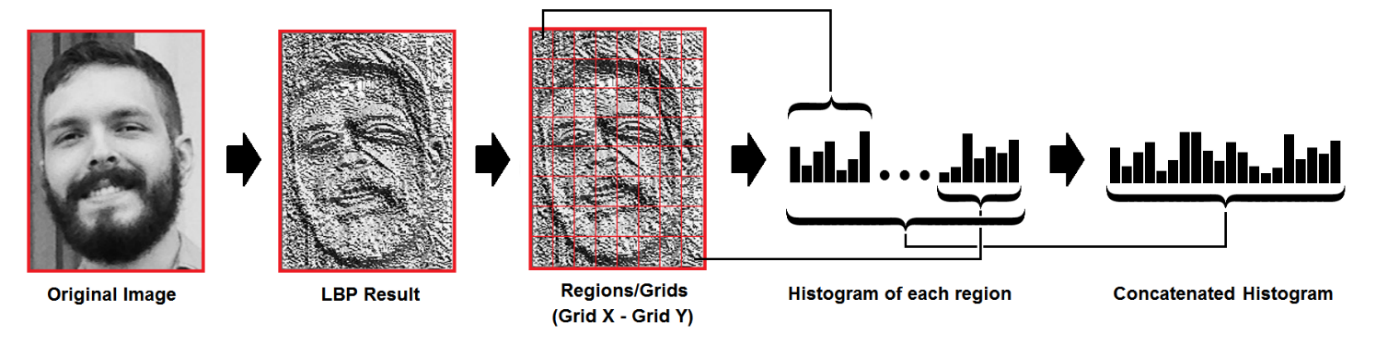
d)For each neighbor of the central value (threshold), we set a new binary value. We set 1 for values equal or higher than the threshold and 0 for values lower than the threshold.

e)Now, the matrix will contain only binary values (ignoring the central value). We need to concatenate each binary value from each position from the matrix line by line into a new binary value (e.g. 10001101

At the end of this procedure (LBP procedure), we have a new image which represents better the characteristics of the original image.

**STEP-4:**

**Extracting the Histograms**: Now, using the image generated in the last step, we can use the **Grid X** and **Grid Y** parameters to divide the image into multiple grids, as can be seen in the following image:



Based on the image above, we can extract the histogram of each region as follows:

As we have an image in grayscale, each histogram (from each grid) will contain only 256 positions (0~255) representing the occurrences of each pixel intensity.

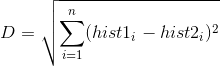
Then, we need to concatenate each histogram to create a new and bigger histogram.Supposing we have 8x8 grids, we will have 8x8x256=16.384 positions in the final histogram. The final histogram represents the characteristics of the image original image.

**STEP 5:**

**Performing the face recognition**: In this step, the algorithm is already trained. Each histogram created is used to represent each image from the training dataset. So, given an input image, we perform the steps again for this new image and creates a histogram which represents the image.

So to find the image that matches the input image we just need to compare two histograms and return the image with the closest histogram.

We can use various approaches to compare the histograms (calculate the distance between two histograms), for example: **euclidean distance**, **chi-square**, **absolute value**, etc. In this example, we can use the Euclidean distance (which is quite known) based on the following formula:



So the algorithm output is the ID from the image with the closest histogram. The algorithm should also return the calculated distance, which can be used as a ‘**confidence**’ measurement. **Note**: don’t be fooled about the ‘confidence’ name, as lower confidences are better because it means the distance between the two histograms is closer.

We can then use a threshold and the ‘confidence’ to automatically estimate if the algorithm has correctly recognized the image. We can assume that the algorithm has successfully recognized if the confidence is lower than the threshold defined.

**Algorithm** -Collecting training images of the students

Input: :Database of enrolled images Y; Threshold *τ*.

Output: : *Y* plus 10 different training images of learned student.

1: NbrImages = 0 // Number of images learned

2: for *NbrImages* ≤ 10 do

3: Capture an image(x)

4: Similarity = D(x,Y)

5: if similarity ≥ *τ* then

6: Y = Y + {x}

7: SNbrImages = NbrImages+1

8: end if

9: end for

**Algorithm** - Detecting faces with an Adaboost trained cascade classifier

Input: : An *M* × *N* grayscale *I* and *L*-layer cascade of shift

Output: : *ρ*, set of windows declared positive by the cascade.

1: Parameter: a window scale multiply by *c*

2: Set *ρ* [*i,i* + *e* − 1] × [*j,j* + *e* − 1] ⊂ *I* : *e* = 24*ckK,k* ∈ *N*

3: for *l* = 1 to *L* do

4: for every window in *ρ* do

5: Remove the windowed image’s mean and compute its standard deviation *σ*

6: if *σ >* 1 then

7: Divide image by *σ*

8: Compute image features required by the shifted classifier at layer *l*

9: if Cascade’s *l* − *th* layer predict negative then

10: Discard this windows from *ρ*

11: end if

12: else

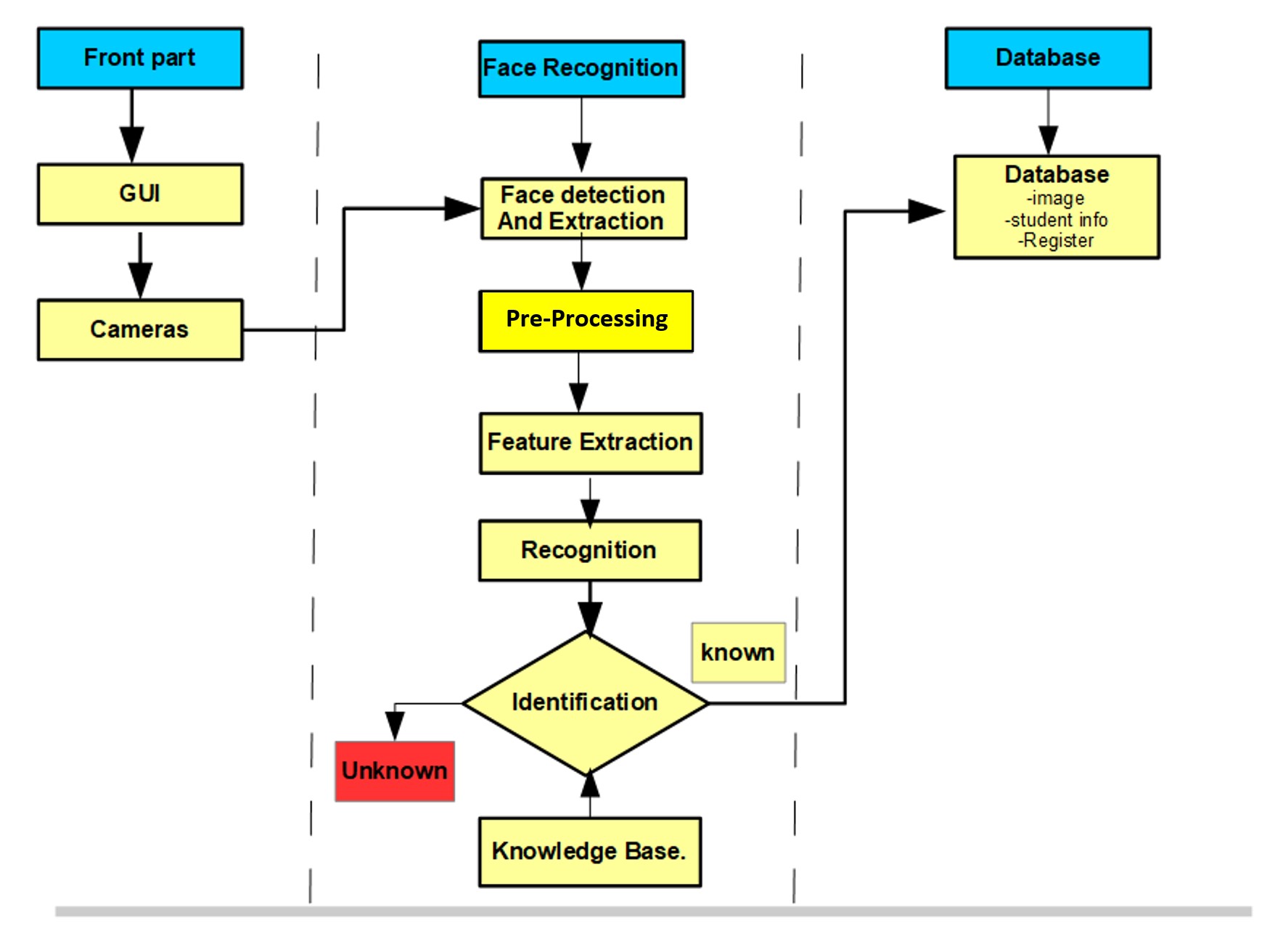
13: Discard this windows from *ρ*

14: end if

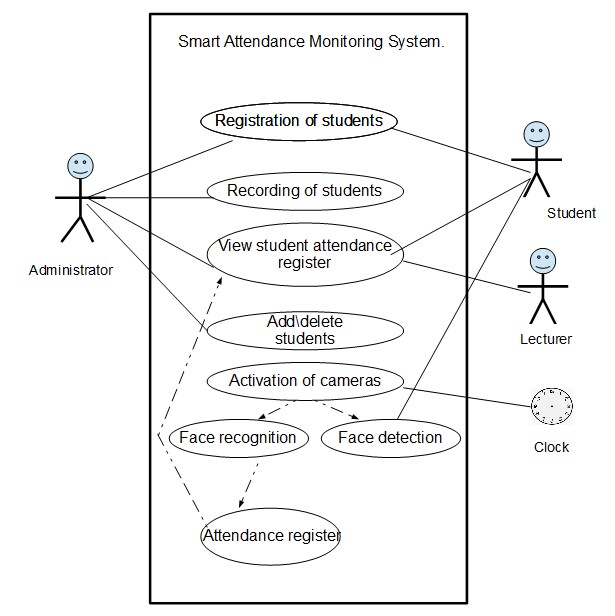
15: end for

16: Return

**2.3 FLOWCHART**



**2.4 UML DIAGRAM**



**CHAPTER-3: IMPLEMENTATION**

**CODE:**

**main.py**

import datasetCreator as d

import detector as det

import tkinter

import att

mywindow = tkinter.Tk()

mywindow.title("Smart Attendance System")

mywindow.configure(bg="white")

head = tkinter.Label(mywindow,text = "Attendance System Using Face Recognition", font= ("times

new roman",20),bg="pink").grid(row=0,rowspan=3)

bt1=tkinter.Button(mywindow,text="StudentRegistration",command=d.datacreation,bg="indigo",fg = "white",height=2, font= ("times new roman",15) ).grid(row=3,padx=5,pady=5,sticky="nesw")

bt2=tkinter.Button(mywindow,text="Mark Attendance" ,command = det.detection,bg="indigo",fg = "white",height=2, font= ("times new roman",15) ).grid(row=4,padx=5,pady=5,sticky ="nesw")

bt3 = tkinter.Button(mywindow,text = "Attendance Report" ,command = att.report,bg="indigo",fg = "white",height=2, font= ("times new roman",15) ).grid(row=5,padx=5,pady=5,sticky ="nesw")

#bt4=tkinter.Button(mywindow,text="About the Developers",bg="lavender",fg = "black",height=2, font= ("times new roman",15) ).grid(row=6,padx=5,pady=5,sticky ="nesw")

#bt5=tkinter.Button(mywindow,text="Exit",command = mywindow.destroy,bg="lavender",fg = "black",height=2, font= ("times new roman",15) ).grid(row=7,padx=5,pady=5,sticky ="nesw")

bt6=tkinter.Button(mywindow,width=20,height=2,text="About",font=("timesnew roman",15),bg="lavender",fg = "black").grid(row=8,column=0,sticky="nw",padx=10,pady=5)

bt7=tkinter.Button(mywindow,width=20,height=2,text="Exit",font=("timesnew roman",15),bg="lavender",fg"black",command=mywindow.destroy).grid(row=8,column=0,sticky="ne" ,padx=10,pady=5)

mywindow.mainloop

**detector.py**

import cv2

import sqlite3

import openpyxl

import numpy as np

def getProfile(id):

conn=sqlite3.connect("studentDb")

cmd="select \* from student where ID="+str(id)

cursor=conn.execute(cmd)

profile=None

for row in cursor:

profile=row

conn.close()

return profile

def detection():

faceDetect=cv2.CascadeClassifier('haarcascade\_frontalface\_default.xml');

vid=cv2.VideoCapture(0);

rec = cv2.face.LBPHFaceRecognizer\_create()

rec.read("recognizer\\trainingData.yml")

id=0

#font = cv2.FONT\_HERSHEY\_SIMPLEX

#font=cv2.cv.InitFont(cv2.cv.CV\_FONT\_COMPLEX\_SMALL,5,1,0,4)

while(True):

ret,img=vid.read();

gray=cv2.cvtColor(img,cv2.COLOR\_BGR2GRAY)

faces=faceDetect.detectMultiScale(gray,1.3,5)

#cv2.imshow(“Face”, img)

for x, y, w, h in faces:

cv2.rectangle(img, (x, y), (x + w, y + h), (0, 0, 255), 2)

id, conf = rec.predict(gray[y:y + h, x:x + w])

profile=getProfile(id)

if(profile!=None):

cv2.putText(img,"ID:"+str(profile[0]),(x,y+h+30),cv2.FONT\_HERSHEY\_SIMPLEX, 1, (255,0,0), 3)

cv2.putText(img,"Name:"+str(profile[1]),(x,y+h+60),cv2.FONT\_HERSHEY\_SIMPLEX,1, (255,0,0),3)

cv2.putText(img,"Dept:"+str(profile[2]),(x,y+h+90),cv2.FONT\_HERSHEY\_SIMPLEX, 1, (255,0,0),3)

cv2.putText(img,"Gender:"+str(profile[3]),(x,y+h+120),cv2.FONT\_HERSHEY\_SIMPLEX,1, (255,0,0), 3)

cv2.imshow("Face",img)

if(cv2.waitKey(1) == ord('q')):

break;

wb = openpyxl.load\_workbook('attendance.xlsx')

sheet = wb.get\_sheet\_by\_name('Sheet1')

for cellObj in list(sheet.columns)[0]:

if (cellObj.value == id):

sheet['B'+str(cellObj.row)]=1

wb.save('attendance.xlsx')

vid.release()

cv2.destroyAllWindows()

#detection()

**trainer.py**

import os

import cv2

import numpy as np

from PIL import Image

def trainingModule():

recognizer = cv2.face.LBPHFaceRecognizer\_create()

path='dataset'

imagePaths=[os.path.join(path,f) for f in os.listdir(path)]

faces=[]

IDs=[]

for imagePath in imagePaths:

faceImg=Image.open(imagePath).convert('L');

faceNp=np.array(faceImg,'uint8')

ID=int(os.path.split(imagePath)[-1].split('.')[0])

faces.append(faceNp)

print(ID)

IDs.append(ID)

cv2.imshow("training",faceNp)

cv2.waitKey(10)

recognizer.train(faces,np.array(IDs))

recognizer.write('recognizer/trainingData.yml')

cv2.destroyAllWindows()

**datasetCreator.py**

import cv2

import sqlite3

import trainer as t

import openpyxl

import numpy as np

def insertOrUpdate(id,name,Dept,gender):

conn = sqlite3.connect("studentDb")

cmd ="select \* from student where ID="+str(id)

cursor = conn.execute(cmd)

isRecordExist=0

for row in cursor:

isRecordExist=1

if(isRecordExist==1):

cmd = "update student set Name=" + str(name)+"where ID="+str(id)

else:

cmd="insertintostudent(ID,Name,Department,Gender) Values("+str(id)+",'"+str(name)+"','"+str(Dept)+"','"+str(gender)+"')"

conn.execute(cmd)

conn.commit()

conn.close()

# creating a dataset dynamically for 1 person

# or start with a static set and add dynamically

# in that case keep in mind of the id's

def datacreation():

facedetectcascade = cv2.CascadeClassifier("haarcascade\_frontalface\_default.xml")

vid = cv2.VideoCapture(0)

id= input("Enter the id ")

name = input("Enter your Name")

Dept = input("Enter your Department Name")

gender = input("Enter you Gender:")

insertOrUpdate(id,name,Dept, gender)

sno=0 #sample number for a given id

wb = openpyxl.load\_workbook('attendance.xlsx')

sheet = wb.get\_sheet\_by\_name('Sheet1')

sheet['A' + str(sheet.max\_row + 1)] = int(id)

wb.save('attendance.xlsx')

while True:

check, frame = vid.read();

gray\_frame = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

faces = facedetectcascade.detectMultiScale(gray\_frame, 1.05, 5)

cv2.imshow("Face Detection", frame)

key = cv2.waitKey(100)

for x, y, w, h in faces:

# after a face is detected, we store it

sno = sno + 1

cv2.imwrite("dataset/"+str(id)+"."+str(sno)+".jpg", gray\_frame[y:y+h,x:x+w])

# or store entire frame in dataset as frame in imwrite( ,frame)

cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 0, 255), 3) # here pass color image i.e., frame

cv2.waitKey(1000) # giving 1 sec time for the person to change expression for dataset creation

#cv2.imshow("Face Detection", frame)

#key = cv2.waitKey(100)

#if (key == ord("q")):

#break

# to get out of loop we dont need infinite loop but only a few samples for recognizing a person..

if(sno==10):

break

vid.release()

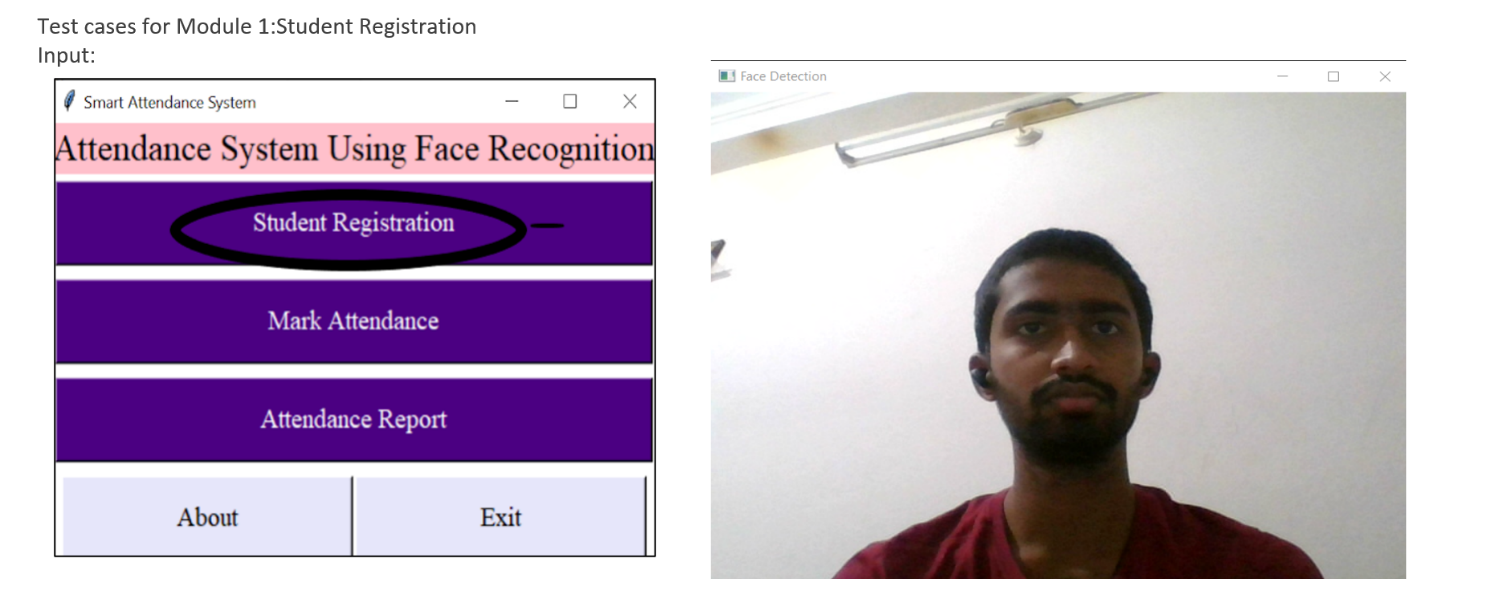
#cv2.destroyAllWindows()

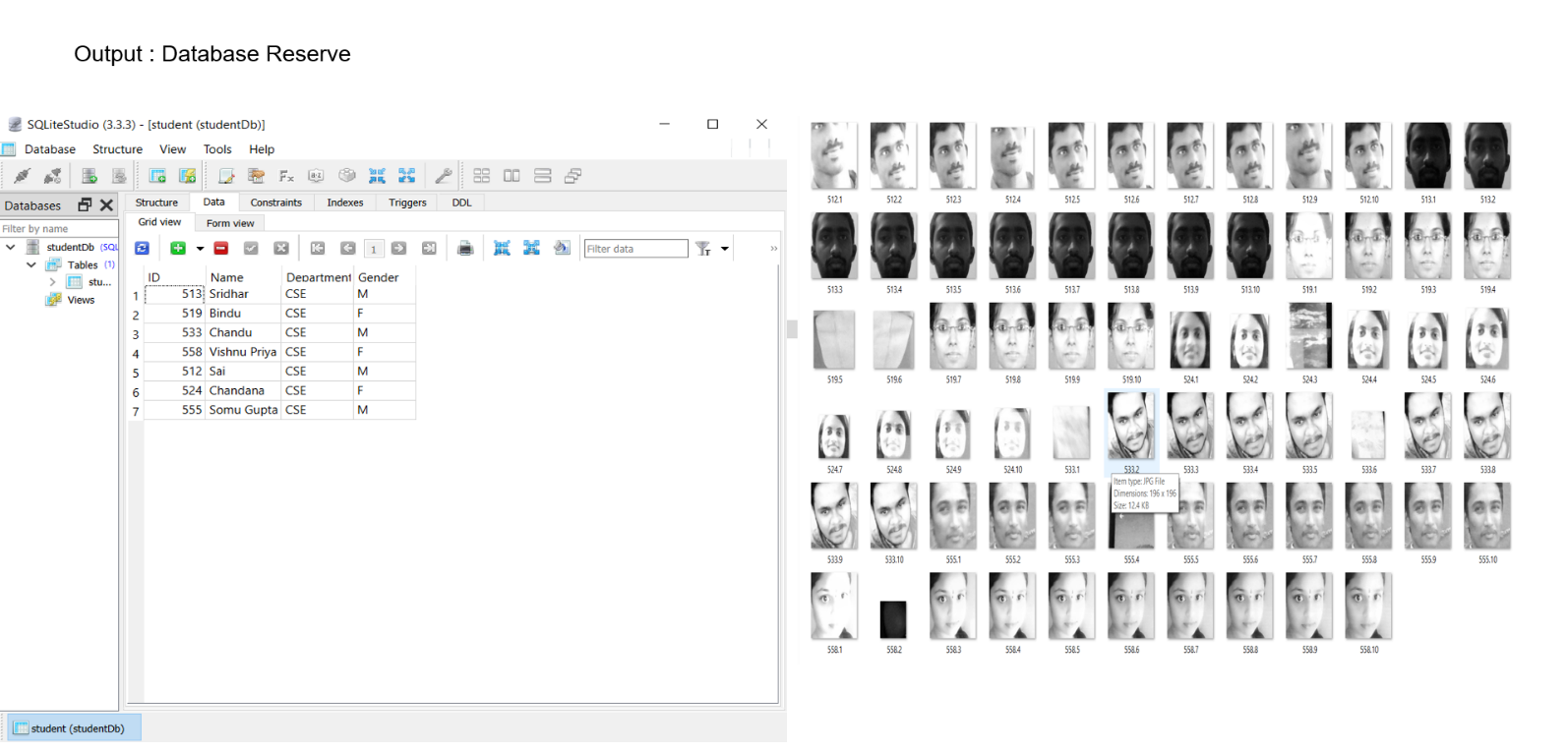
cv2.destroyWindow("Face Detection"

t.trainingModule

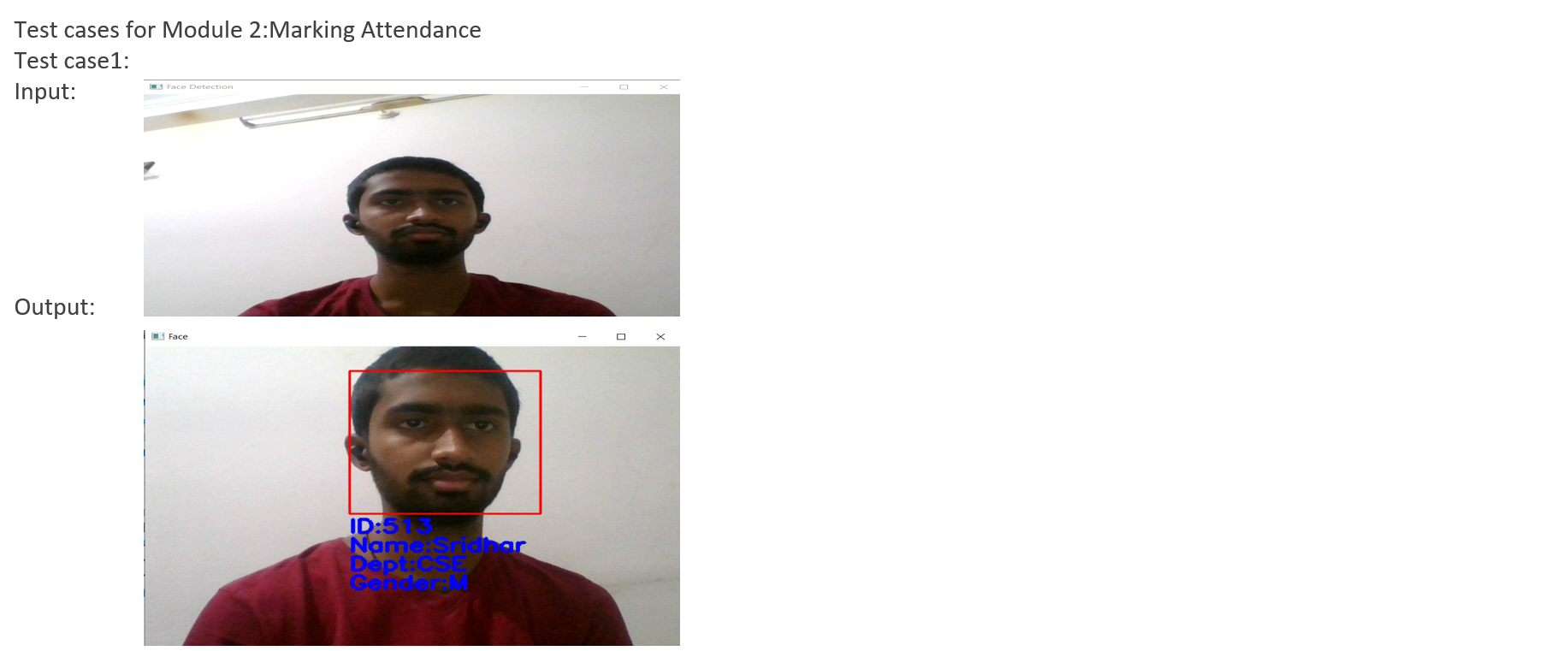
**CHAPTER-4: TESTING**

**TEST CASE-1:**

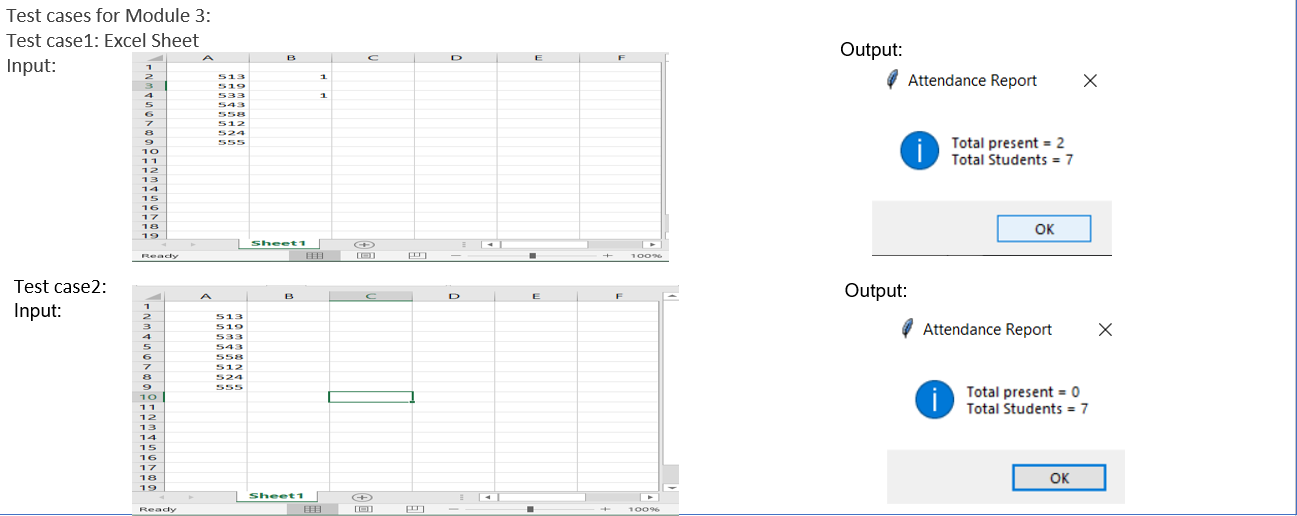




**TESTCASE-2:**

****

**TESTCASE-3:**



**CHAPTER – 5: RESULTS**

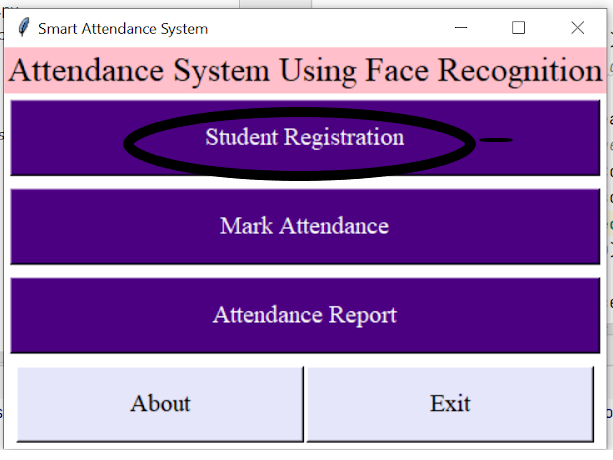
**5.1 MAIN UI INTERFACE WINDOW**

Main UI interface window consist of following buttons: -

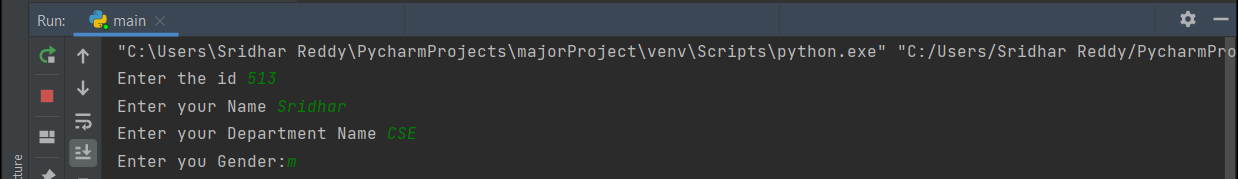
1. Student Registration

2.Mark Attendance

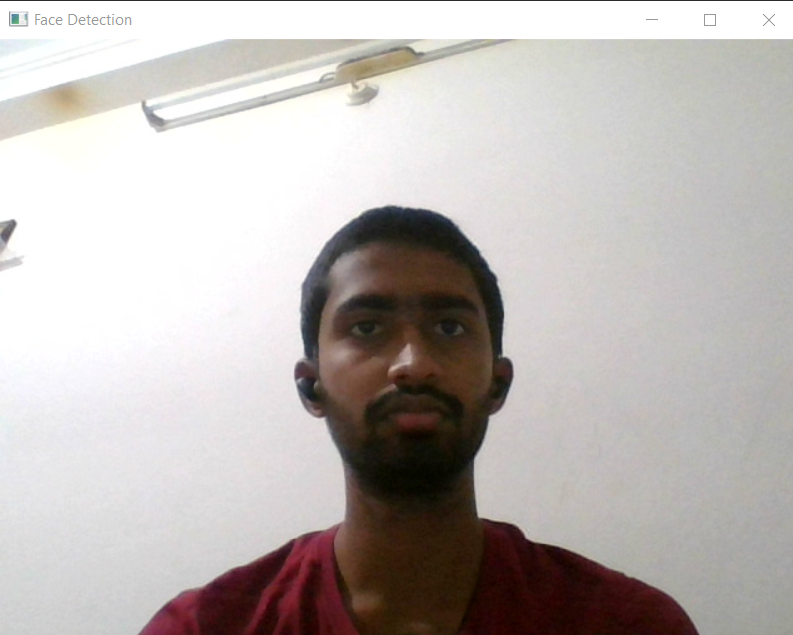
3. Attendance Report



**5.1 STUDENT REGISTRATION**

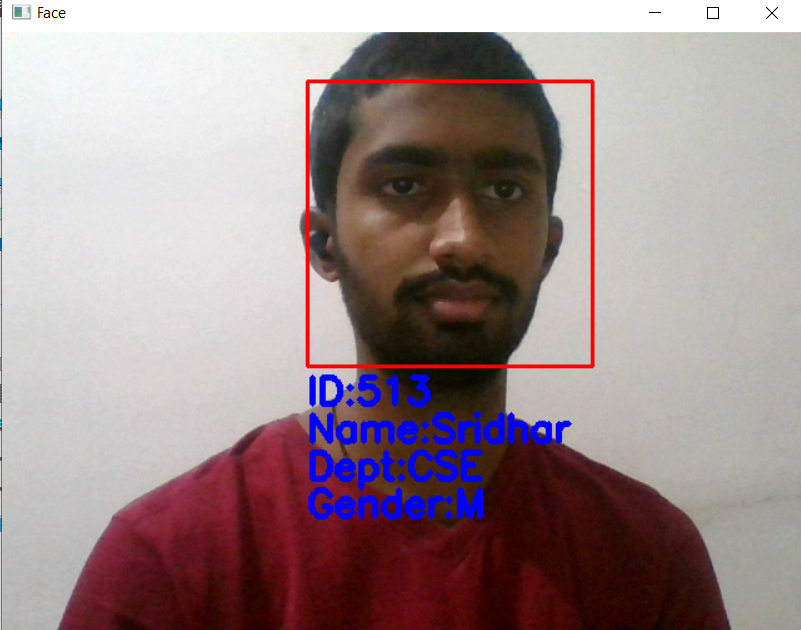


**5.1.1 FACE DETECTION WINDOW**

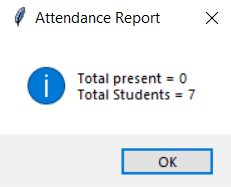
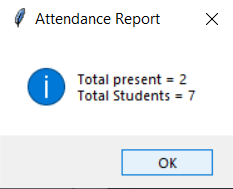


**5.2 MARKING ATTENDANCE**

**5.2.1 Face Recognition Window**



**5.3 Attendance Report**

**CHAPTER – 6: CONCLUSION & FUTURE SCOPE**

**6.1 CONCLUSION:**

Maintaining the attendance is essential in every foundation for checking the performance of students as well as employee in terms of attendance percentage. The task to compute the attendance percentage becomes a major task as manual computation produces errors. So we would like to build a system which is both cost and time efficient and demands minimum voluntary action by the users for both taking attendance and making attendance reports in just one click.

**6.2 FUTURE SCOPE:**

The algorithm will be tested with multiple students in the scene and also captured faces at different angles in the scene. The algorithm delivers good results but there is room to improve the algorithm performance in case of large numbers of students and also in case of faces captured in a dark environment, and detect multiple faces at a time. The efficiency of the algorithm also can be increased further so there is also a room for future work in this area. This system can be enhanced further in terms of achieving more efficiency by ease of analysis of patterns in the data.

**BIBLIOGRAPHY**

1. Face Detection Based on Viola-Jones Algorithm Applying Composite Features by Wen-yao LU and Ming YANG Published in 2019 International Conference on Robots & Intelligent Systems (ICRIS) by IEEE.
2. An Analysis of the Viola-Jones Face Detection Algorithm by ​Yi-Qing Wang
3. Automate the Boring Stuff with Python, 2nd Edition, a Practical Programming book for Total Beginners by Al Sweigart - Chapter 12 : Working with Excel Spreadsheets.

**Web References:**

* https://www.researchgate.net/publication/341876647\_Face\_Recognition\_based\_Attendance \_Management\_System
* Edureka Tkinter Python Course https://www.youtube.com/watch?v=VMP1oQOxfM0