

PROFESSIONAL TRAINING REPORT

at

**Sathyabama Institute of Science and Technology
(Deemed to be University)**

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Technology Degree in

Information Technology

By

SUBHIKSHA.C
(Reg.No.37120074)



**DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING**

**SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119**

DECEMBER 2020



SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY

(Established under Section 3 of UGC Act, 1956)
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai -600119
www.sathyabama.ac.in



DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **SUBHIKSHA.C** (Reg.no.37120074) who carried out the project entitled “**FACE AND SMILE RECOGNITION**” under our supervision from JUL 2020 to DEC 2020.

Internal Guide

Dr.R.M.GOMATHI, M.TECH.,Ph.D.

Head of the Department

Dr.R SUBHASHINI M.E,Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I , SUBHIKSHA.C (Reg.No.37120074) hereby declare that the Project Report entitled "**FACE AND SMILE RECOGNITION**" done by me under the guidance of **Dr.R.M.GOMATHI, M.TECH.,Ph.D.** is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology Degree in Information Technology .

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D**, Dean, School of Computing , **Dr.SUBHASHINI M.E., Ph.D.** , Head of the Department of Information and Technology for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide Dr. Dr.R.M.GOMATHI, M.TECH.,Ph.D. for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.

July 05, 2019

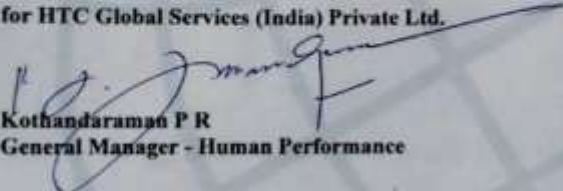
TO WHOMSOEVER IT MAY CONCERN

This is to certify that **Ms. Subhiksha C** has successfully completed her summer internship program with us from 11th November, 2018 to 12th December, 2018.

During the internship tenure, she worked on the domain **R data Science and Networking Tools** under the guidance of Dr. Muthukumaran Balasubramanian A J, General Manager - Training, Institute of Technology Management & Research, a group of HTC Global Services. During the tenure, her conduct, attitude and performance were found to be good.

We wish her all the very best in her future endeavors.

for HTC Global Services (India) Private Ltd.



Kothandaraman P R
General Manager - Human Performance

World Headquarters

3270 West Big Beaver Road, Troy, MI 48064, USA

Phone: 248.786.2500 • Fax: 248.786.2515 • www.htcinc.com

USA • UK • Germany • India • Malaysia • Singapore • UAE • Australia • Indonesia • South Korea

Certificate of Completion

*This is to certify that **Subhiksha.c** successfully
completed 39 total mins of **Python 3 Data Types &
Variables Course** online course on Aug. 13, 2020*

Marufa Yasmin
Marufa Yasmin, Instructor



Certificate ID: 123-456789-1011-1234-5678-9101112345
Certificate URL: [www.udemy.com/certificate/UC-123456789-1011-1234-5678-9101112345](#)

#BeAble

ABSTRACT

The project titled Face And Smile Recognition using python OpenCV. OpenCV-Python is a library of Python bindings designed to solve computer vision problems. OpenCV-Python makes use of Numpy, which is a highly optimized library for numerical operations with a MATLAB-style syntax. All the OpenCV array structures are converted to and from Numpy arrays.. Identifying a person with an image has been popularised through the mass media. However, it is less robust to fingerprint or retina scanning. This report describes the face detection and recognition mini-project undertaken for the visual perception. It reports the technologies available in the Open-Computer-Vision (OpenCV) library and methodology to implement them using Python. For face detection, Haar-Cascades were used and for face recognition Eigenfaces, Fisherfaces and Local binary pattern histograms were used. The methodology is described including flow charts for each stage of the system. Next, the results are shown including plots and screen-shots followed by a discussion of encountered challenges.

TABLE OF CONTENTS

CHAPTER No.	TITLE	PAGE NO
	ABSTRACT	vii
	LIST OF FIGURES	ix
1	INTRODUCTION	
	1.1 INTRODUCTION TO THE PROJECT	9
	1.2 AIM AND SCOPE	9
	1.3 FEASABILITY STUDY	10
	1.4 SYSTEMS	11
2	ALGORITHMS AND METHODS	
	2.1 SDLC METHODOLOGY	13
	2.2 ACTIVITY DIAGRAM	14
3	SYSTEM DESIGN AND TESTING PLAN	
	3.1 MODULES	15
	3.2 INPUT DESIGN	17
	3.3 DATA FLOW DIAGRAM	18
	3.4 UML DIAGRAM	19
	3.5 SYSTEM REQUIREMENTS	20
4	RESULTS AND DISCUSSIONS	
	4.1 SUMMARY	22
	4.2 SCREEN SHOTS	23
5	CONCLUSION AND FUTURE ENHANCEMENTS	
	5.1 CONCLUSION	36
	5.2 FUTURE ENHANCEMENT	37
	5.3 REFERENCES	37
6	APPENDIX	
	6.1 SOURCE CODE	38

LIST OF FIGURES

FIGURE NO.	TITLE OF THE FIGURE	PAGE NO.
2.1	Spiral Model Diagram	6
2.2	Activity Diagram	7
3.1	System Architecture	7
3.2	Data Flow Diagram	9
3.3	Use Case Diagram	10
3.4	Class Diagram	11
3.5	Sequence Diagram	12
3.6	Component Diagram	13

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION TO THE PROJECT

OpenCV is the most popular library for computer vision. Originally written in C/C++, it now provides bindings for Python. OpenCV uses machine learning algorithms to search for faces within a picture. Because faces are so complicated, there isn't one simple test that will tell you if it found a face or not. Instead, there are thousands of small patterns and features that must be matched. The algorithms break the task of identifying the face into thousands of smaller, bite-sized tasks, each of which is easy to solve.

1.2 AIM AND SCOPE

The main aim of the project is "To Create and recognize the face and smile of human" It involved building a system for face detection and face recognition using several classifiers available in the open computer vision library (OpenCV). Face recognition is a non-invasive identification system and faster than other systems since multiple faces can be analysed at the same time. The difference between face detection and identification is, face detection is to identify a face from an image and locate the face. Face recognition is making the decision "whose face is it?", using an image database. In this project both are accomplished using different techniques and are described below. The report begins with a brief history of face recognition. This is followed by the explanation of HAAR-cascades, Eigenface, Fisherface and Local binary pattern histogram (LBPH) algorithms. Next, the methodology and the results of the project are described. A discussion regarding the challenges and the resolutions are described. Finally, a conclusion is provided on the pros and cons of each algorithm and possible implementations.

1.3 Operational Feasibility

Face recognition using OpenCV and Python. First, let us see the libraries we will need and how to install them:

- OpenCV
- dlib
- Face_recognition

OpenCV is an image and video processing library and is used for image and video analysis, like facial detection, license plate reading, photo editing, advanced robotic vision, optical character recognition, and a whole lot more.

The dlib library, maintained by Davis King, contains our implementation of “deep metric learning” which is used to construct our face embeddings used for the actual recognition process.

The face_recognition library, created by Adam Geitgey, wraps around dlib’s facial recognition functionality.

1.4 Economic Feasibility

Economic feasibility attempts to weigh the costs of developing and implementing a new system, against the benefits that would accrue from having the new system in place. This feasibility study gives the top management the economic justification for the new system.

1.5 SYSTEMS

1.5.1 EXISTING SYSTEM

The existing system consists of entering the details in the Microsoft Excel Sheets for the storing of the data. When a manager needs information of the employee he searches for the specified file in the file system. He opens the file and takes the information. Report Generation done manually by copying the content of the different files into another file. The Manually generated report was then printed.

DISADVANTAGES OF EXISTING SYSTEM

- Information retrieval is a very big process.
- Lack of organization of the files may lead to information loss due to accidental deletion of files.
- No security because the files are visible to the users.
- Report generation will be a big task.

1.5.2 PROPOSED SYSTEM

The Proposed system is a browser which is completely related to the programming modules so that it has very good performance and it is very easy to deploy in many operating systems.

ADVANTAGES OF PROPOSED SYSTEM

- The performance is increased due to well designed database.
- Security is increased
- Time saving in report generation
- Easy to update the details

CHAPTER 2

ALGORITHM AND METHODS

2.1. SDLC METHODOLOGY

This document play a vital role in the development of life cycle (SDLC) as it describes the complete requirement of the system. It means for use by developers and will be the basic during testing phase. Any changes made to the requirements in the future will have to go through formal change approval process.

SPIRAL MODEL was defined by Barry Boehm in his 1988 article, "A spiral Model of Software Development and Enhancement. This model was not the first model to discuss iterative development, but it was the first model to explain why the iteration models.

As originally envisioned, the iterations were typically 6 months to 2 years long. Each phase starts with a design goal and ends with a client reviewing the progress thus far. Analysis and engineering efforts are applied at each phase of the project, with an eye toward the end goal of the project.

The steps for Spiral Model can be generalized as follows:

- The new system requirements are defined in as much details as possible. This usually involves interviewing a number of users representing all the external or internal users and other aspects of the existing system.
- A preliminary design is created for the new system.
- A first prototype of the new system is constructed from the preliminary design. This is usually a scaled-down system, and represents an approximation of the characteristics of the final product.
- A second prototype is evolved by a fourfold procedure:
 1. Evaluating the first prototype in terms of its strengths, weakness, and risks.
 2. Defining the requirements of the second prototype.
 3. Planning an designing the second prototype.
 4. Constructing and testing the second prototype.

- At the customer option, the entire project can be aborted if the risk is deemed too great. Risk factors might involved development cost overruns, operating-cost miscalculation, or any other factor that could, in the customer's judgment, result in a less-than-satisfactory final product.
- The existing prototype is evaluated in the same manner as was the previous prototype, and if necessary, another prototype is developed from it according to the fourfold procedure outlined above.
- The preceding steps are iterated until the customer is satisfied that the refined prototype represents the final product desired.
- The final system is constructed, based on the refined prototype.
- The final system is thoroughly evaluated and tested. Routine maintenance is carried on a continuing basis to prevent large scale failures and to minimize down time.

The following diagram shows how a spiral model acts like:

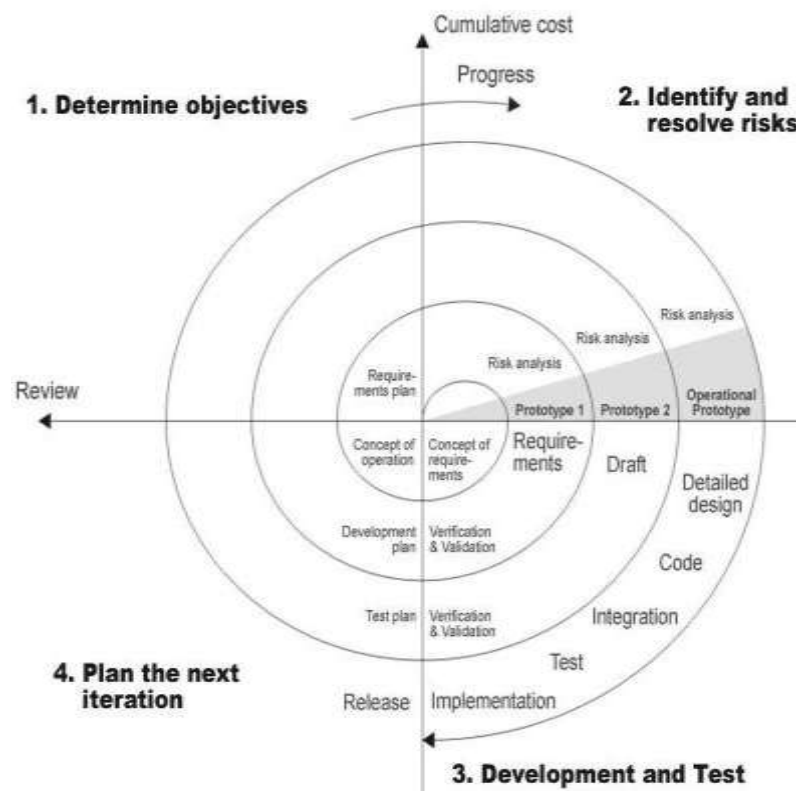


Fig. 2.1 Spiral Model

ADVANTAGES

- Estimates(i.e. budget, schedule etc .) become more realistic as work progresses, because important issues discovered earlier.
- It is more able to cope with the changes that are software development generally entails.
- Software engineers can get their hands in and start working on the core of a project earlier.

2.2 ACTIVITY DIAGRAM

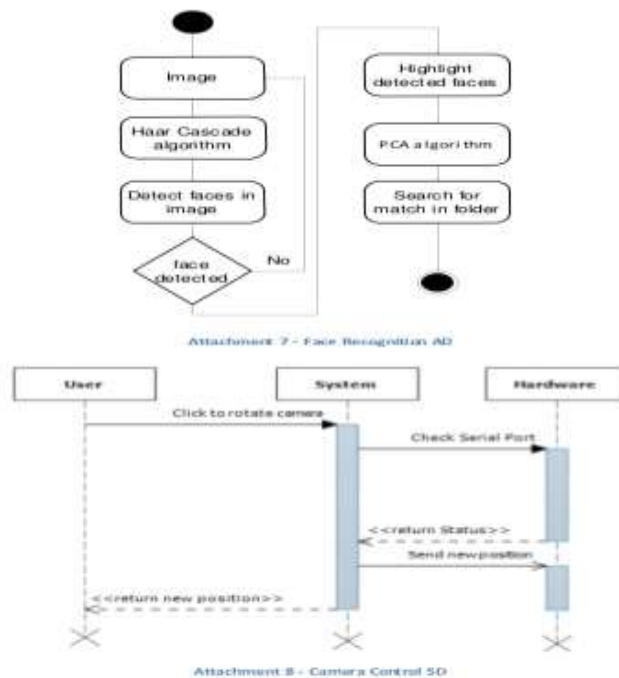


Fig.2.2. Activity diagram

CHAPTER 3

SYSTEM DESIGN AND TESTING PLAN

3.1 MODULES

USER: This module has the entire access to all other modules, admin creates the project and assigning the projects to the created manager, adding members to the managers, assigning Defects based on the priority. Can update the manager, members and access to the particular project data. Generating reports based on the managers report submission.

Developer: Can access the task or Defect assigned by the manager, view assigned projects and resolving the assigned Defect. Developer can view the Defects list assigned by the manager.

Tester: Tester can access to the projects , can view the assigned projects and can add a new Defect to the list and send the bug back to the manager. Tester can login to the system and access the assigned projects list.

Reports: Both Admin and Manager can access this module and generate the reports based on the requirements.

3.2 SYSTEM ARCHITECTURE

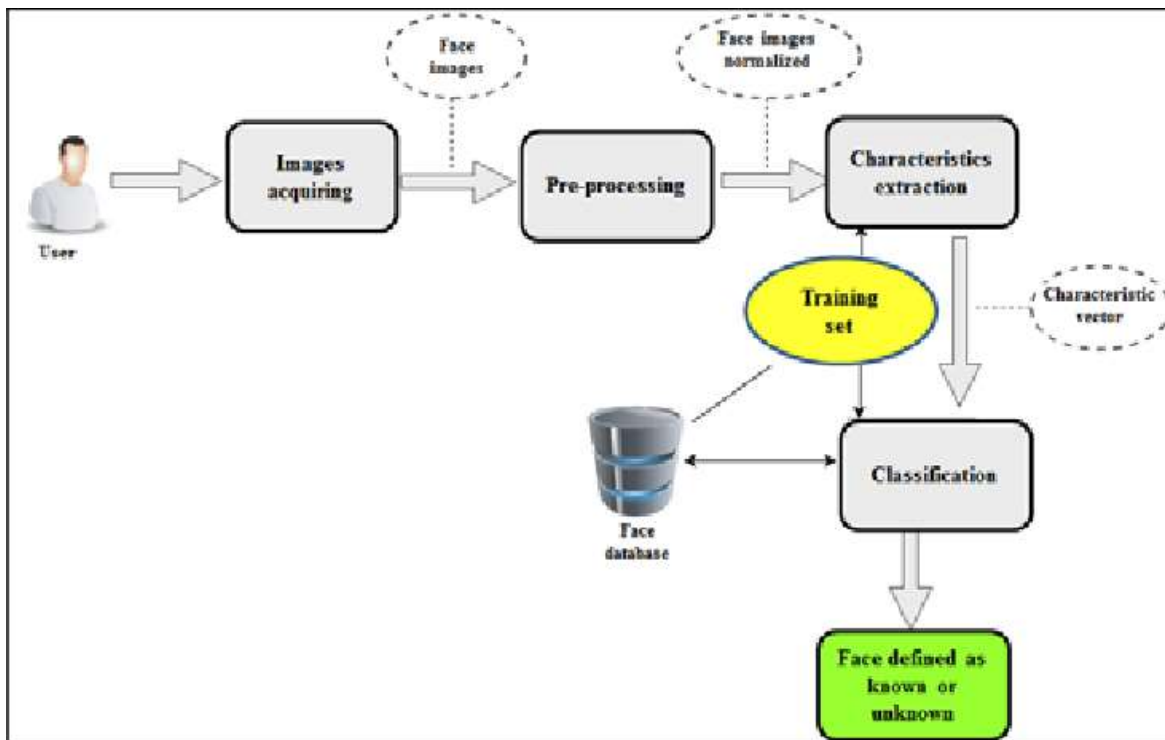


Fig. 3.1 system architecture

3.2 INPUT DESIGN

3.2.1 THE PRESENTATION LAYER

It Also called as the client layer comprises of components that are dedicated to presenting the data to the user. For example: Windows/Web Forms and buttons, edit boxes, Text boxes, labels, grids, etc.

3.2.2 THE BUSINESS RULES LAYER

This layer encapsulates the Business rules or the business logic of the encapsulations. To have a separate layer for business logic is of a great advantage. This is because any changes in Business Rules can be easily handled in this layer. As long as the interface between the layers remains the same, any changes to the functionality/processing logic in this layer can be made without impacting the others. A lot of client-server apps failed to implement successfully as changing the business logic was a painful process

3.2.3 THE DATA ACCESS LAYER

This layer comprises of components that help in accessing the Database. If used in the right way, this layer provides a level of abstraction for the database structures. Simply put changes made to the database, tables, etc do not affect the rest of the application because of the Data Access layer. The different application layers send the data requests to this layer and receive the response from this layer.

3.2.4 THE DATABASE LAYER

This layer comprises of the Database Components such as DB Files, Tables, Views, etc. The Actual database could be created using SQL Server, Oracle, Flatfiles,etc.

In an n-tier application, the entire application can be implemented in such a way that it is independent of the actual Database. For instance, you could change the Database Location with minimal changes to Data Access Layer. The rest of the Application should remain unaffected.

3.3 DATA FLOW DIAGRAM (DFD)

A data flow diagram is a way of representing a flow of data of a process or a system. The DFD also provides the information about the input and output of each entity and the process itself.

DFD- LEVEL

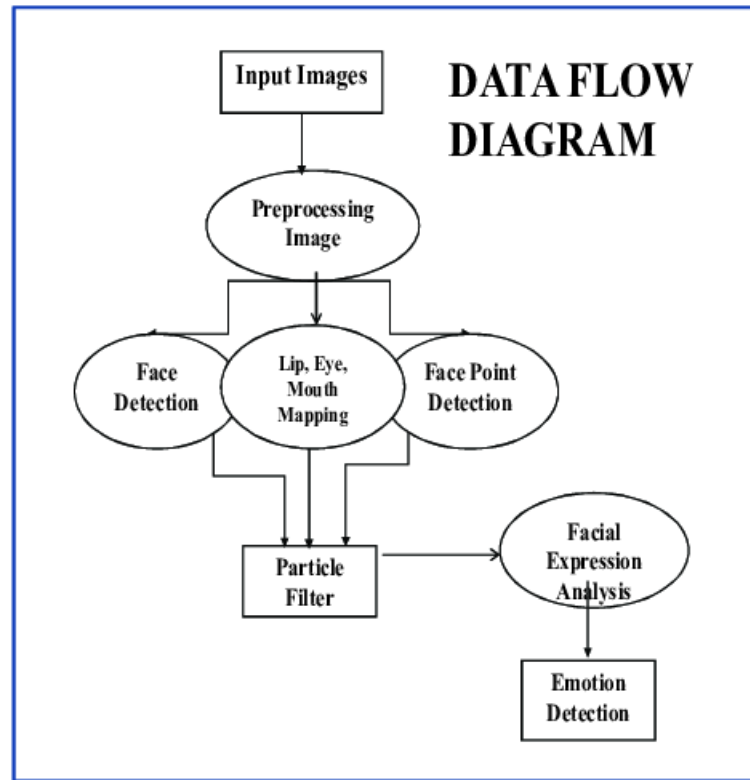


Fig.3.2 Data flow diagram

3.4 UML DIAGRAMS

Unified Modelling language includes a set of graphic notations techniques to create a visual models of object oriented software intensive systems

Use case diagram :

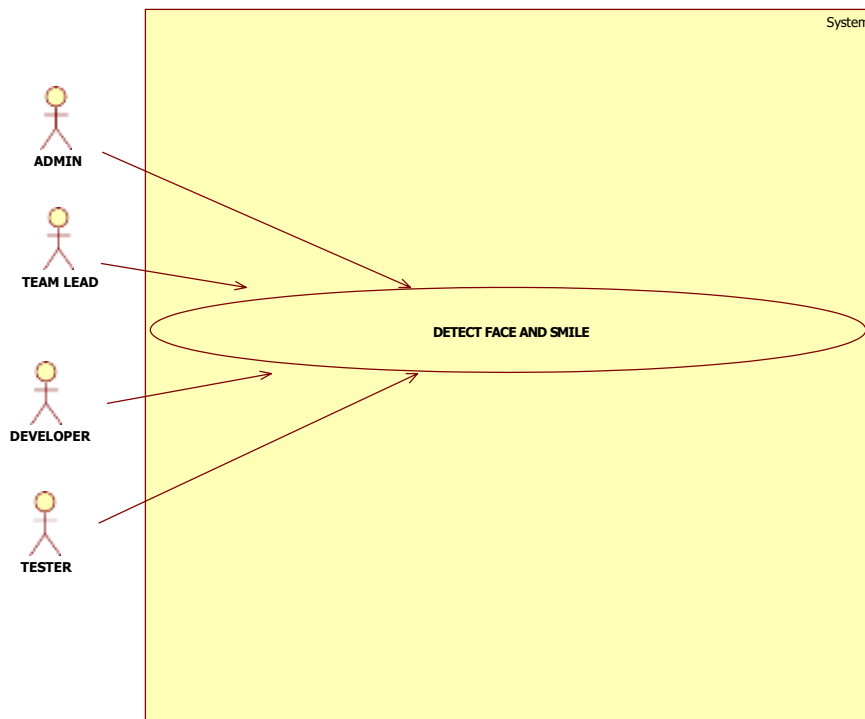


Fig.3.3 Use case diagram

3.5 CLASS DIAGRAM

In software Engineering , a class diagram in the unified modeling language is a type of structure diagram that describes the structure of a system by e systems ,classes ,their structures,operations and their relationships among objects.

In the diagram , classes are represented with bases which contain three parts.

- 3.3.4.1 The top part contains the name of the class.
- 3.3.4.2 The middle point contains the attributes of the class.
- 3.3.4.3 The bottom part gives the methods or operations in the class can take or undertake .

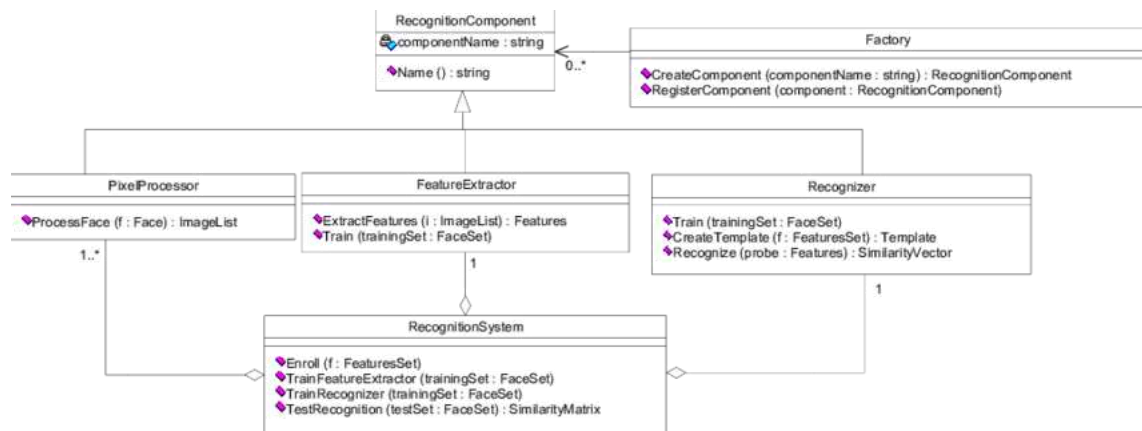


Fig.3.4 Class diagram

3.6 SEQUENCE DIAGRAM

- 3.3.4.4 The Sequence diagram shows the ordering of the messages .
- 3.3.4.5 It shows the order of the messages that the admin should perform.
- 3.3.4.6 The interaction that takes place in a collaboration that either realizes a use case or an operation.
- 3.3.4.7 High-level interactions between user of the system and the system,between the system and other systems,or between subsystems.

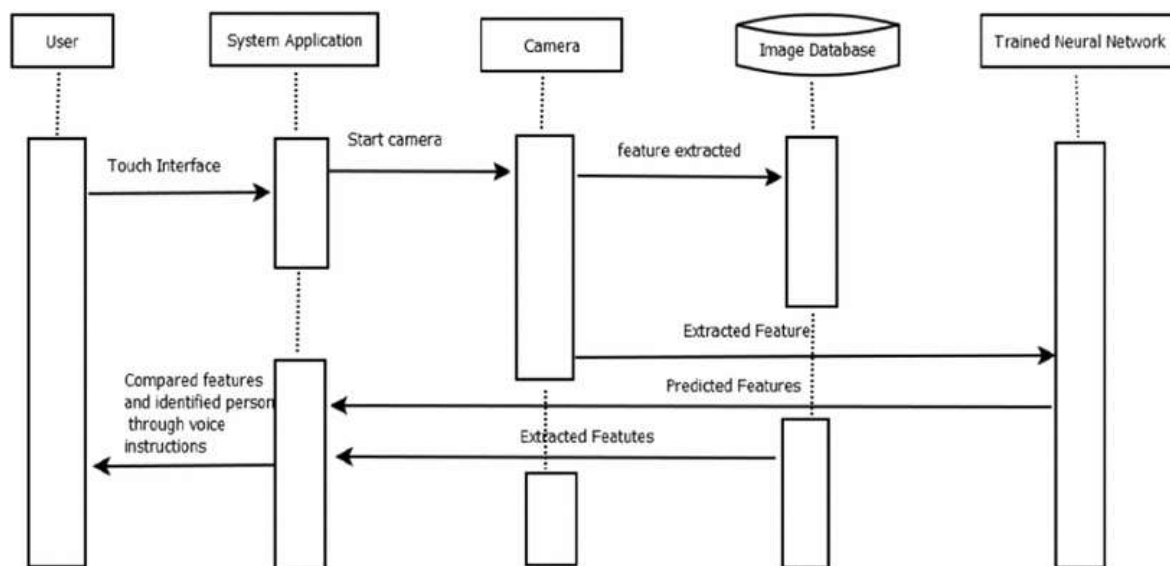


Fig. 3.5 Sequential diagram

3.7 COMPONENT DIAGRAM

1. A component diagram also known as a UML component diagram .
2. It describes the organization and wiring of the physical components in a system.
3. Component diagrams are often drawn to help model implementation details and double check that every aspect of the systems required.

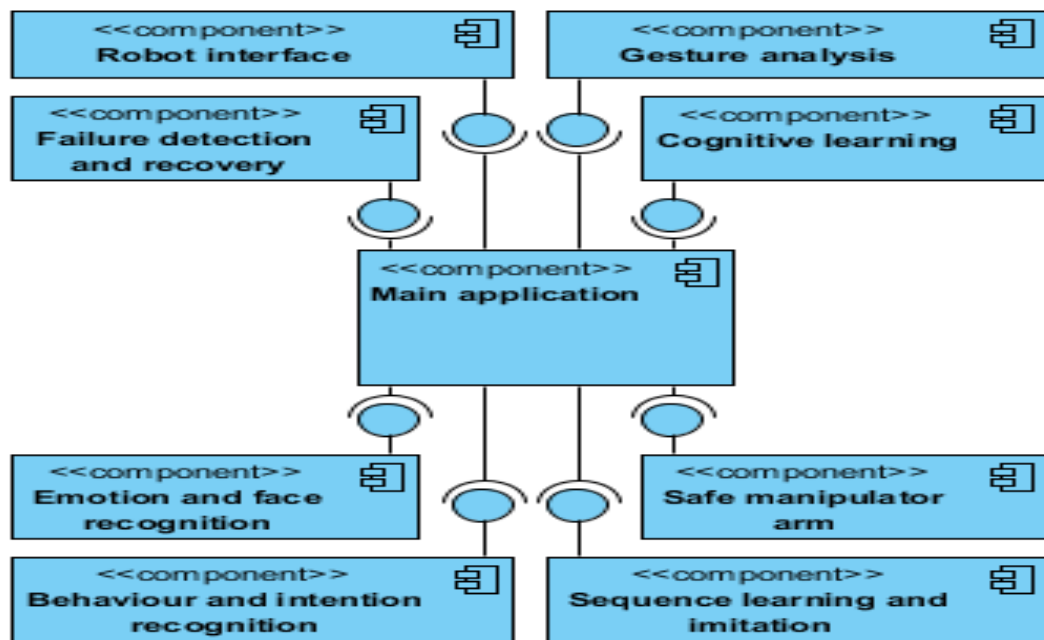


Fig. 3.6 component diagram

3.7 SYSTEM REQUIREMENTS

3.7.1 Software Requirements

Operating System	:	Windows XP/2003 or Linux/Solaris
User Interface	:	HTML, CSS
Programming Language	:	Python
IDE/Workbench	:	Anaconda
Database	:	SQLite
Server Deployment	:	Tomcat6.0

3.7.2 Hardware Requirements

Processor	:	Pentium IV
Hard Disk	:	40GB
RAM	:	256MB

CHAPTER 4

RESULTS AND DISCUSSIONS

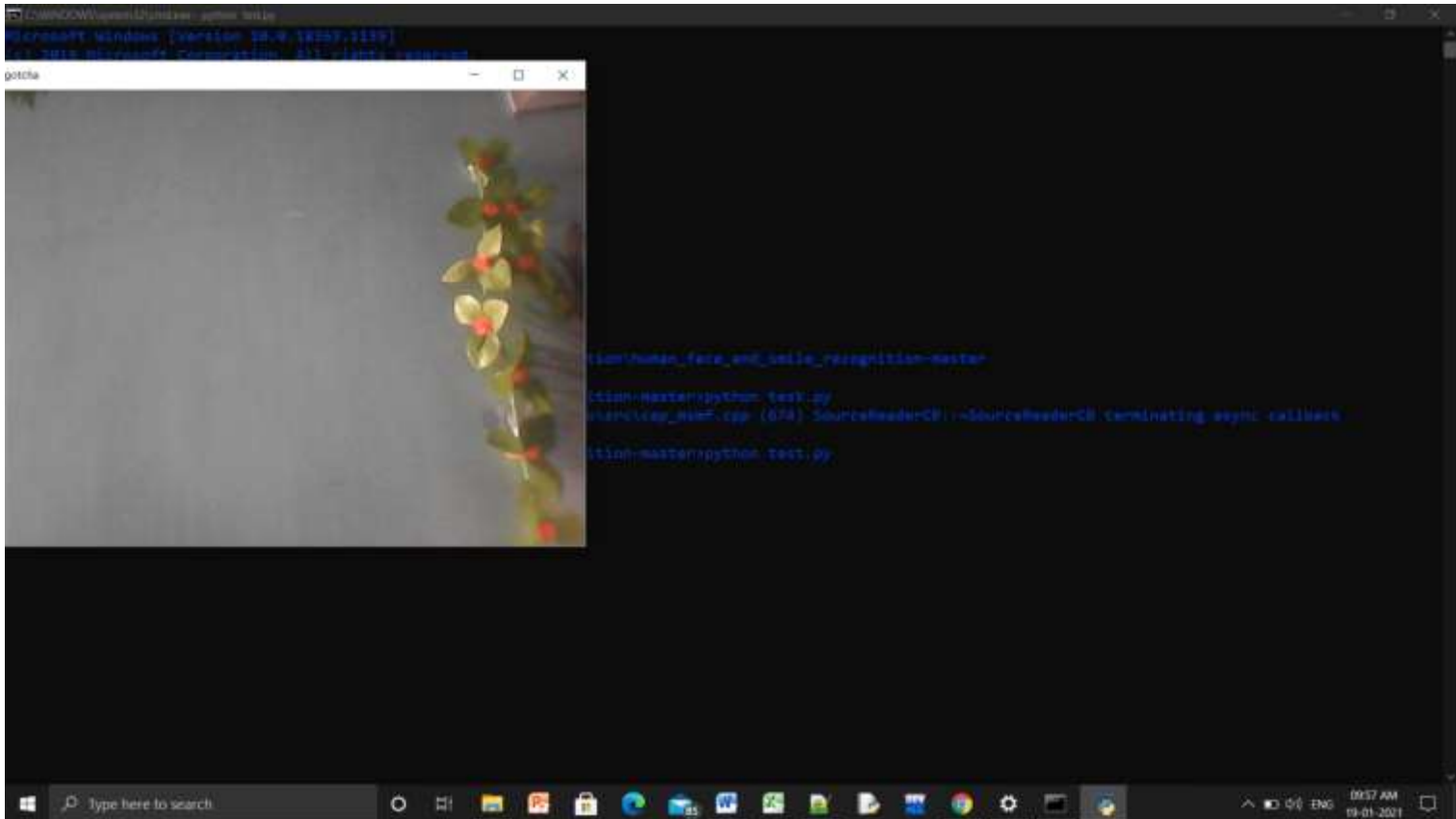
4.1 SUMMARY

With the continuous development of information technology, the demand for security and safety is gradually improving. For the consideration of security, face recognition has been studied for many decades. With the development of information technology, face recognition is widely used in our daily life, especially in security systems, information security, human-computer interaction. Researches are committed to improving the recognition accuracy and response speed of the face recognition system. The state-of-art of face recognition has been significantly improved by the appearance of deep learning. Although these systems perform well on large amounts of web collected facial data, the performance and accuracy are still limited when they are applied in actual scenarios. There is still a long way to go to improve the recognition accuracy of face recognition system in real scenarios.

face recognition is the science which involves the understanding of how the faces are recognized by biological systems and how this can be emulated by computer systems. Biological systems employ different types of visual sensors (i.e., eyes), which have been designed by nature to suit a certain environment where the agent lives. Similarly, computer systems employ different visual devices to capture and process faces as best indicated by each particular application.

4.2 SCREEN SHOTS

MAIN SCREEN



CHAPTER 5

CONCLUSION AND FUTURE ENHANCEMENT

5.1 CONCLUSION

Face recognition technology has one goal: to identify human faces. Out of all biometric systems available, it is the least intrusive and fastest technology. Instead of asking people to scan their fingerprints, hands or irises, face recognition systems can take pictures of people's faces unobtrusively. In most cases, the subjects are not even aware of the process. This removes the feeling of being under surveillance or invasion of privacy. Facial recognition systems are also highly effective in producing results, all thanks to the use of neural network algorithms. Our brain recognizes different features of a face – the space between the eyes, the height of a hairline, width of a nose – subconsciously. The facial recognition algorithms need to do that intentionally.

5.2 FUTURE ENHANCEMENT

A facial recognition system is used to identify and verify a person from an image or video source. It uses biometric software's along with AI enabled devices for mapping facial features and brings out the recognition step. A facial recognition software differentiates a face from rest of the background in the image. The software first recognizes the face then measures different facial features. The software recognizes these features as nodal points. A human face consists of 80 nodal points. After measuring these features a numerical code for the same is created and stored in the database. This is known as the faceprint. Earlier the software relied on 2D image to identify or verify another 2D image from the database but today it uses a 3D model for the same. This 3D model is more reliable, better, effective and accurate than its 2D counterpart. Using the 3D software, the system goes through a series of steps, facial recognition forming the last one.

REFERENCES

1. Wikipedia https://en.wikipedia.org/wiki/Facial_recognition_system. Accessed 18 October 2018.
2. Zhou, E., Cao, Z., Yin, Q.: Naive-deep face recognition: touching the limit of LFW benchmark or not? arXiv preprint arXiv:1501.04690 (2015).
3. Wang, W., Shan, S., Gao, W., Cao, B., Yin, B.: An improved active shape model for face alignment. In: 2002 Proceedings of IEEE International Conference on Multimodal Interfaces, pp. 523–528. IEEE (2002)Google Scholar.
4. Cootes, T.F., Taylor, C.J., Cooper, D.H., et al.: Active shape models-their training and application. *Comput. Vis. Image Underst.* 61(1), 38–59 (1995)CrossRefGoogle Scholar.

APPENDIX

6.1 SOURCE CODE

```
import cv2 ,time

face_cascade=cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
smile_cascade=cv2.CascadeClassifier("smile.xml")

video=cv2.VideoCapture(0)

while True:
    check,frame=video.read()
    gray = cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
    face=face_cascade.detectMultiScale(gray,scaleFactor=1.1,minNeighbors=5)
    for x,y,w,h in face:
        img=cv2.rectangle(frame,(x,y),(x+w,y+h),(0,0,255),3)
        smile=smile_cascade.detectMultiScale(gray,scaleFactor=1.8,minNeighbors=20)
        for x,y,w,h in smile:
            img=cv2.rectangle(frame,(x,y),(x+w,y+h),(255,0,0),3)

    cv2.imshow('gotcha',frame)
    key=cv2.waitKey(1)

    if key==ord('q'):
        break
```

```

video.release()
cv2.destroyAllWindows

# importing libraries
import tkinter as tk
from tkinter import Message, Text
import cv2
import os
import shutil
import csv
import numpy as np
from PIL import Image, ImageTk
import pandas as pd
import datetime
import time
import tkinter.ttk as ttk
import tkinter.font as font
from pathlib import Path

window = tk.Tk()
window.title("Face_Recogniser")
window.configure(background = 'white')
window.grid_rowconfigure(0, weight = 1)
window.grid_columnconfigure(0, weight = 1)
message = tk.Label(
    window, text = "Face-Recognition-System",
    bg = "green", fg = "white", width = 50,

```

```
height = 3, font = ('times', 30, 'bold'))
```

```
message.place(x = 200, y = 20)
```

```
lbl = tk.Label(window, text = "No.",  
width = 20, height = 2, fg = "green",  
bg = "white", font = ('times', 15, ' bold '))  
lbl.place(x = 400, y = 200)
```

```
txt = tk.Entry(window,  
width = 20, bg = "white",  
fg = "green", font = ('times', 15, ' bold '))  
txt.place(x = 700, y = 215)
```

```
lbl2 = tk.Label(window, text = "Name",  
width = 20, fg = "green", bg = "white",  
height = 2, font = ('times', 15, ' bold '))  
lbl2.place(x = 400, y = 300)
```

```
txt2 = tk.Entry(window, width = 20,  
bg = "white", fg = "green",  
font = ('times', 15, ' bold '))  
txt2.place(x = 700, y = 315)
```

```
# The function below is used for checking  
# whether the text below is number or not ?  
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 Images is a function used for creating
# the sample of the images which is used for
# training the model. It takes 60 Images of
# every new user.
```

```
def TakeImages():
```

```
# Both ID and Name is used for recognising the Image
Id =(txt.get())
name =(txt2.get())
```

```
# Checking if the ID is numeric and name is Alphabetical
if(is_number(Id) and name.isalpha()):
    # Opening the primary camera if you want to access
```

```

# the secondary camera you can mention the number
# as 1 inside the parenthesis
cam = cv2.VideoCapture(0)
# Specifying the path to haarcascade file
harcascadePath = "data\haarcascade_frontalface_default.xml"
# Creating the classier based on the haarcascade file.
detector = cv2.CascadeClassifier(harcascadePath)
# Initializing the sample number(No. of images) as 0
sampleNum = 0
while(True):
    # Reading the video captures by camera frame by frame
    ret, img = cam.read()
    # Converting the image into grayscale as most of
    # the the processing is done in gray scale format
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    # It converts the images in different sizes
    # (decreases by 1.3 times) and 5 specifies the
    # number of times scaling happens
    faces = detector.detectMultiScale(gray, 1.3, 5)

    # For creating a rectangle around the image
    for (x, y, w, h) in faces:
        # Specifying the coordinates of the image as well
        # as color and thickness of the rectangle.
        # incrementing sample number for each image
        cv2.rectangle(img, (x, y), (

```

```

        x + w, y + h), (255, 0, 0), 2)
    sampleNum = sampleNum + 1
    # saving the captured face in the dataset folder
    # TrainingImage as the image needs to be trained
    # are saved in this folder
    cv2.imwrite(
        "TrainingImage\ "+name +". "+Id +'. '+ str(
            sampleNum) + ".jpg", gray[y:y + h, x:x + w])
    # display the frame that has been captured
    # and drawn rectangle around it.
    cv2.imshow('frame', img)
    # wait for 100 milliseconds
    if cv2.waitKey(100) & 0xFF == ord('q'):
        break
    # break if the sample number is more than 60
    elif sampleNum>60:
        break
# releasing the resources
cam.release()
# closing all the windows
cv2.destroyAllWindows()
# Displaying message for the user
res = "Images Saved for ID : " + Id + " Name : "+ name
# Creating the entry for the user in a csv file
row = [Id, name]
with open('UserDetails\UserDetails.csv', 'a+') as csvFile:
    writer = csv.writer(csvFile)

```

```

        # Entry of the row in csv file
        writer.writerow(row)
    csvFile.close()
    message.configure(text = res)
else:
    if(is_number(Id)):
        res = "Enter Alphabetical Name"
        message.configure(text = res)
    if(name.isalpha()):
        res = "Enter Numeric Id"
        message.configure(text = res)

# Training the images saved in training image folder
def TrainImages():
    # Local Binary Pattern Histogram is an Face Recognizer
    # algorithm inside OpenCV module used for training the image dataset
    recognizer = cv2.face.LBPHFaceRecognizer_create()
    # Specifying the path for HaarCascade file
    harcascadePath = "data\haarcascade_frontalface_default.xml"
    # creating detector for faces
    detector = cv2.CascadeClassifier(harcascadePath)
    # Saving the detected faces in variables
    faces, Id = getImagesAndLabels("TrainingImage")
    # Saving the trained faces and their respective ID's
    # in a model named as "trainer.yml".
    recognizer.train(faces, np.array(Id))
    recognizer.save("TrainingImageLabel\Trainer.yml")

```

```

# Displaying the message
res = "Image Trained"
message.configure(text = res)

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)]
    faces =[]
    # creating empty ID list
    lds =[]
    # now looping through all the image paths and loading the
    # lds and the images saved in the folder
    for imagePath in imagePaths:
        # loading the image and converting it to gray scale
        pillImage = Image.open(imagePath).convert('L')
        # Now we are converting the PIL image into numpy array
        imageNp = np.array(pillImage, '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)
        lds.append(Id)
    return faces, lds
# For testing phase
def TrackImages():
    recognizer = cv2.face.LBPHFaceRecognizer_create()
    # Reading the trained model

```

```

recognizer.read("TrainingImageLabel\Trainer.yml")
harcascadePath = "data\haarcascade_frontalface_default.xml"
faceCascade = cv2.CascadeClassifier(harcascadePath)
# getting the name from "userdetails.csv"
df = pd.read_csv("UserDetails\UserDetails.csv")
cam = cv2.VideoCapture(0)
font = cv2.FONT_HERSHEY_SIMPLEX
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):
            aa = df.loc[df['Id'] == Id]['Name'].values
            tt = str(Id)+"-"+aa
        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)
    cv2.imshow('im', im)

```

```
        if (cv2.waitKey(1)== ord('q')):  
            break  
    cam.release()  
    cv2.destroyAllWindows()
```

```
takeImg = tk.Button(window, text ="Sample",  
    command = TakeImages, fg ="white", bg ="green",  
    width = 20, height = 3, activebackground = "Red",  
    font =( 'times', 15, ' bold '))  
takeImg.place(x = 200, y = 500)  
trainImg = tk.Button(window, text ="Training",  
    command = TrainImages, fg ="white", bg ="green",  
    width = 20, height = 3, activebackground = "Red",  
    font =( 'times', 15, ' bold '))  
trainImg.place(x = 500, y = 500)  
trackImg = tk.Button(window, text ="Testing",  
    command = TrackImages, fg ="white", bg ="green",  
    width = 20, height = 3, activebackground = "Red",  
    font =( 'times', 15, ' bold '))  
trackImg.place(x = 800, y = 500)  
quitWindow = tk.Button(window, text ="Quit",  
    command = window.destroy, fg ="white", bg ="green",  
    width = 20, height = 3, activebackground = "Red",  
    font =( 'times', 15, ' bold '))  
quitWindow.place(x = 1100, y = 500)
```

