

**ATTENDANCE SYSTEM MANAGEMENT USING FACE RECOGNITION**

**BY**

**MOHAMMAD MISHKAT**

**161-15-7359**

**MD. NURA RABBI KHAN**

**161-15-6857**

**AND**

**FHAMIDA AKTHER ARIFA**

**161-15-7353**

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering.

Supervised By

**Shah Md. Tanvir Siddiquee**

Assistant Professor

Department of CSE

Daffodil International University



**DAFFODIL INTERNATIONAL UNIVERSITY**

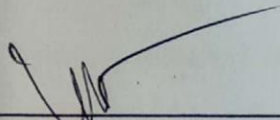
**DHAKA BANGLADESH**

**07 December 2019**

## APPROVAL

This Project titled “**Attendance System Management Using Face Recognition**”, submitted by Mohammad Mishkat, MD. Nura Rabbi Khan, Fhamida Akther Arifa, ID No: 161-15-7359, 161-15-6857, 161-15-7353 to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 07 December 2019.

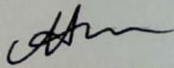
## BOARD OF EXAMINERS



**Dr. Syed Akhter Hossain**  
**Professor and Head**

Department of Computer Science and Engineering  
Faculty of Science & Information Technology  
Daffodil International University

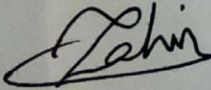
Chairman



**Nazmun Nessa Moon**  
**Assistant Professor**

Department of Computer Science and Engineering  
Faculty of Science & Information Technology  
Daffodil International University

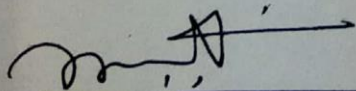
Internal Examiner



**Gazi Zahirul Islam**  
**Assistant Professor**

Department of Computer Science and Engineering  
Faculty of Science & Information Technology  
Daffodil International University

Internal Examiner



**Dr. Mohammad Shorif Uddin**  
**Professor**

Department of Computer Science and Engineering  
Jahangirnagar University

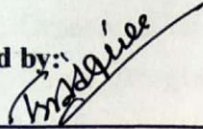
External Examiner



## DECLARATION

We hereby state that, this project has been done by us under the supervision of **Shah Md. Tanvir Siddiquee**, Assistant Professor, and department of CSE Daffodil International University. We also state that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:



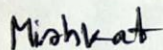
**Shah Md. Tanvir Siddiquee**

Assistant Professor

Department of CSE

Daffodil International University

Submitted by:

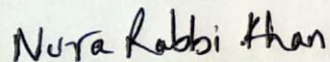


**Mohammad Mishkat**

ID: 161-15-7359

Department of CSE

Daffodil International University

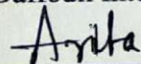


**Md. Nura Rabbi Khan**

ID: 161-15-6857

Department of CSE

Daffodil International University



**Fhamida Akhter Arifa**

ID: 161-15-7353

Department of CSE

Daffodil International University

## ACKNOWLEDGEMENT

We would like to express our sincere gratitude to our parents and thanks and gratefulness to almighty ALLAH. Our project would not have been possible without the blessing of them.

We are highly grateful to our respected supervisor “**Shah Md. Tanvir Siddiquee**”, **Assistant Professor**, Department of **Computer Science and Engineering**, Daffodil International University, Dhaka, Bangladesh. Deep knowledge and his dynamic suggestions helps us to carry out of this project. His endless patience, scholarly direction, continual encouragement, constant and energetic supervision, constructive condemnation, valuable guidance ,reading many inferior draft and correcting them at all stage have made it possible to complete this project.

We would like to express our sincere gratitude to **Dr. Syed Akhter Hossain, Professor and Head**, Department of **Computer Science and Engineering**, Daffodil International University, for his kind help to finish our project and also to other faculty member and the staff of **Computer Science and Engineering** department of Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discuss while completing the course work.

## **ABSTRACT**

**“ATTENDANCE SYSTEM MANAGEMENT USING FACE RECOGNITION”** is a Machine Learning project. This project helps to take attendance by recognizing trained students real time image. It will make taking attendance easy. Its user friendly, this project GUI is so simple so anyone can easily use this. OpenCV and Tkinter is the main library we used to build this system. This project will helpful to research in OpenCV face detection and face recognition. This system recognition accuracy is 80%. This system can be used in school, college, university and office. This system will save time and energy for who takes attendance. We tried our best to make this system user friendly and efficient.

# TABLE OF CONTENTS

<b>CONTENTS</b>	<b>PAGE NO</b>
Board of examiners	I
Declaration	II
Acknowledgements	III
Abstract	IV
List of Figures	VII
List of Table	VIII
<b>CHAPTER</b>	
<b>CHAPTER 1: INTRODUCTION</b>	<b>1-2</b>
1.1 Motivation	1
1.2 Objectives	1
1.3 Description	1
1.4 Expected Outcome	1
1.5 Report Layout	2
<b>CHAPTER 2: BACKGROUND STUDY</b>	<b>3-4</b>
2.1 Introduction	3
2.2 Literature Review	3
2.3 Comparative Studies	4
2.4 Scope of the problem	4
<b>CHAPTER 3: REQUIRMENT SPECIFICATION</b>	<b>5-10</b>
3.1 Introduction	5
3.2 Machine Learning	5
3.3 Machine Learning Libraries	6
3.4 IDE and Python Library	6-9
3.5 Data Flow Diagram	10



<b>CHAPTER 4: DESIGN SPECIFICATION</b>	<b>11-14</b>
4.1 Front End Design	11
4.2 Back End Design	12-13
4.3 Experimental Result	14
<b>CHAPTER 5: IMPLEMENTING AND TESTING</b>	<b>15-23</b>
5.1 Take Image	15
5.2 Implementation of dataset	16-17
5.3 Training Image	18
5.4 Track Image	19
5.5 Test Results and Reports	20-22
5.6 Implementation of Requirement	23
<b>CHAPTER 6: CONCLUSION AND FUTURE WORK</b>	<b>24</b>
6.1 Discussion and Conclusion	24
6.2 Further Plan	24
<b>REFERANCE</b>	<b>25</b>
<b>APPENDEX</b>	<b>26</b>
<b>PLAGARISM</b>	<b>27</b>

## LIST OF FIGURES:

<b>FIGURES</b>	<b>PAGE NO</b>
Figure 2.2.1: OpenCv	4
Figure 3.2.1: Machine Learning	5
Figure 3.4.1: Numpy	8
Figure 3.4.2: Tkinter GUI of our project	9
Figure 3.5.1: DFD	10
Figure 4.1.1: Front End Design	11
Figure 4.2.1: Back End Design	12
Figure 4.2.2: Back End Design	13
Figure 4.2.3: Back End Design	13
Figure 4.3.1: Experimental Result 1	14
Figure 4.3.2: Experimental Result 2	14
Figure 5.1.1: Take Image	15
Figure 5.1.2: Taken Image	16
Figure 5.2.1: Dataset	17
Figure 5.2.2: Image Dataset	17
Figure 5.3.1: Train image	18
Figure 5.4.1: Face recognition	19
Figure 5.5.1: Test result 1	20
Figure 5.5.1: Test result 2	21
Figure 5.5.1: Test result 3	21
Figure 5.5.1: Test result 4	22
Figure 5.5.1: Test result 5	22



## LIST OF TABLES

<b>TABLES</b>	<b>PAGE NO</b>
Table 5.5.1: Test Implementation	22

# CHAPTER 1

## INTRODUCTION

### 1.1 Motivation

“Machine Learning can change jobs that the majority individuals thought might solely be done by people” –Dave Waters

As like we tend to wished to require attending system straightforward by victimization openCV library. OpenCV was designed for procedure competency and with a robust concentrate on real time applications. So, it's good for real time face recognition victimization net camera. Track image then the scholars knowledge can save in an exceedingly surpass sheet. Face recognition is that the task of acknowledge trained pictures.

### 1.2 Objective

Machine learning making our life easy and simple. People starting to depend on it and uses of machine learning rapidly increasing. The objective of our projects are:

- Time Saving.
- Digitalize System.
- User Friendly.
- Automated and real-time face recognizer.

### 1.3 Description

Our project is a machine learning project. Face-Recognition based attendance system is to detect real-time face and recognize trained images and add this person into the dataset. By using openCV and python GUI Tkinter this system build. It's user-friendly and no complexity. This system helps to take attendance easily.

### 1.4 Expected Outcome

Our project is a machine learning project. Machine Learning is just train your machine then we will get our expected outcome. Form our project expected outcome are:

- By using this system we can take attendance.
- This system has real-time face detection and recognition.
- This system accuracy is above 70%.
- We can operate this in web-cam.
- It has user friendly GUI.

## **1.5 Report Layout**

### **Chapter 1: Introduction**

In this chapter we discussed about this project motivation, objective and expected outcome.

### **Chapter 2: Background Study**

This chapter we discussed literature review, comparative studies and scope of the problem of this project.

### **Chapter 3: Requirement Specification**

This chapter we brief our requirement for the system. Requirements are machine learning python libraries.

### **Chapter 4: Design Specification**

This chapter is discussed about front end and back end design of this system.

### **Chapter 5: Implementing and Testing**

This chapter contain the test result and implementing of this system.

### **Chapter 6: Conclusion and Future Work**

This chapter we discussed about this project conclusion and future scope of work.

## CHAPTER 2

### Background Study

#### 2.1 Introduction

OpenCv Face Recognition based attendance system is a machine learning project. In this system we features real-time face detection and recognition then take attendance. This GUI is user friendly and save time. This project developed by using machine learning python libraries. Before going to description of this project, a literature review of openCv is needed. This project is centered machine learning automate attendance system.

#### 2.2 Literature Review

So many projects and thesis papers done on object and face detection. Some of python library are highly used in machine learning for object and face detection also recognition. TensorFlow, OpenCV, SciPy, Keras etc those are used for detection and recognition. After reading some papers we OpenCv will idle for our project.

OpenCV could be a library of python bindings designed to unravel laptop vision. This library contains some machine learning algorithmic rule in order that we will train information simply and also the support of numpy makes the task a lot of easier. Numpy could be an extremely optimized library for numerical operations. All the openCV array structures area unit born-again to-ad-from numpy arrays. Besides that, many different libraries like SciPy, Pandas, Matplotlib that supports openCV and numpy may be used with this. So, openCV is Associate in nursing applicable tool for quick prototyping of laptop vision [1].

For face detection and acknowledge used LBPH face recognizer algorithmic rule. Python offers varied choices for developing GUI (Graphical User Interface).Out of all the GUI systems, tkinter is most typically used library. It's a typical python interface to the Tk GUI toolkit distributed with python. Python with tkinter outputs the quickest and simplest way to create the GUI applications. Import this library and decision the operate and that we will develop responsive program

This two library are needed to develop this project [2].

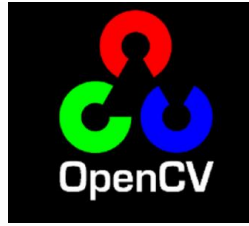


Figure 2.2.1: OpenCV

## 2.3 Comparative Studies

Our implemented project is developed on the basis of machine learning python library. We have studies more throughout developing this project. We have studies each and every requirement and system procedure of that site to work our system in an effective and appropriate way.

We researched and find many similar projects of us but all accuracy and stability is poor. Most of the projects are only face detection and face recognition, those projects has no GUI just has back-end. Our project can detect face and recognize also save his/her id and name in excel sheet. Our project has user friendly GUI [3].

## 2.4 Scope of the Problem

We tried to make this project efficient and accuracy above 80% so that it can recognize face correctly. As we know computer read image data in array. So, some faces decoding array are almost same that time the recognizer accuracy below 60%. Also for this project we have to download this library otherwise this project will not run, such as:

- OpenCV
- Numpy
- Tkinter
- Pandas
- Shutil
- CSV

In 4gb ram laptop this project will take so time to run.

# CHAPTER 3

## REQUIRMENT SEPECIFICATION

### 3.1 Introduction

OpenCv Face Recognition based attendance system is a machine learning project. In this system we features real-time face detection and recognition then take attendance. This GUI is user friendly and save time. Our project needs machine learning and python libraries. So, in this chapter we will go to discuss requirement specification for our project.

### 3.2 Machine Learning

Machine Learning is that the field of study that offers computers the flexibility to be told while not being expressly programmed. They're generally accustomed solve numerous sorts of life issues. Within the older days, individuals accustomed perform Machine Learning tasks by manually secret writing all the algorithms and mathematical and applied mathematics formula. This created the method time overwhelming, tedious and inefficient. However within the fashionable days, it's become abundantly straightforward and economical compared to the past days by numerous python libraries, frameworks, and modules [5].

OpenCV is one of the best machine learning library which used for computer vision such as object, face detection and recognition.

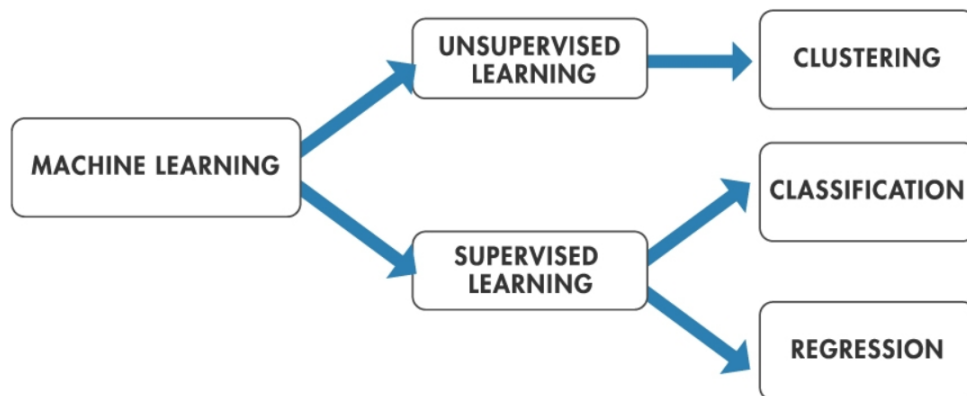


Figure 3.2.1: Machine Learning

### 3.3 Machine learning and libraries

Machine Learning is that the field of study that provides computers the power to find out while not being expressly programmed. They're usually accustomed solve numerous kinds of life issues. Within the older days, folks accustomed perform Machine Learning tasks by manually committal to writing all the algorithms and mathematical and applied mathematics formula. This created the method time overwhelming, tedious and inefficient. However within the fashionable days, it's become pretty much simple and economical compared to the past days by numerous python libraries, frameworks, and modules. Python libraries that used in Machine Learning are:

- Numpy
- Scipy
- Scikit-learn
- TensorFlow
- Keras
- OpenCV
- Pandas
- Matplotlib
- PyTorch

From those OpenCV, Pandas, Numpy we used for our projects [5].

#### **OpenCV**

OpenCV is the massive open-source library for the computer vision, machine learning and image processing and now its plays a key part in real time process which is very vital in today's systems. By using it, one can procedure images and videos to classify objects, faces, or even handwriting of a human. When it combined with numerous for analysis. To identify image design and its various assemblies we use vector space and perform measured operations on these features. By using haarcascade file openCV detect faces.



## **Pandas**

In computer programming, pandas is a software library written for the python programming language for data influence and analysis. In particular, it offers data structures and processes for influencing numerical tables and time series.

When we want to use pandas for data analysis, we usually use it one of three different ways:

1. Convert a python's list, dictionary or numpy array to a panda data frame
2. Open a local file using pandas, usually a CSV file, but could also be a surrounded text file, excel, etc.

We use pandas for save our attendance list in a CSV format in excel.

## **3.4 Ide and python library**

### **IDE**

An integrated development environment (IDE) is a software application that affords complete facilities to computer programmers for software development, An IDE normally contains of at least a source code editor, build automation tools, and a debugger.

For this project we used python IDE and Sublime text editor.

### **Python library**

Python library may be an assortment of functions and strategies that permits you to perform several actions while not writing your code.as an example, the python imaging library (PIL) is one in every of the core libraries for image manipulation in python. Pillow is associate degree actively developed fork of PIL. OpenCV python may be a library of python aimed are real time pc vision and image process. Every library contains a large range of helpful modules that may import in programming [6].

There are some most important library such as:

- Scrapy for Web scarping
- Pillow. Imaging library
- SQLAlchemy. A database library
- Numpy. Provides math fucntions.
- SciPy. Algorithm and mathematical tools.
- Matplotlib. A numerical plotting library.
- pyGtk. Python GUI library
- Tkinter. Python GUI library

- Pandas.
- PIL

From those library we used Pillow, Numpy, Tkinter, Pandas and PIL for our project.

### **Pillow:**

Pillow is a python imaging library, which adds maintenance for opening, manipulating, and saving images. The current version classifies and reads a large number of formats. Write support is purposefully restricted to the most commonly used transaction and presentation formats.

We use this library for manipulate our images.

### **Numpy:**

Numpy is deliberated as one of the most popular machine learning library in python. TensorFlow, OpenCV and other libraries uses Numpy inside for performing multiple operations on Tensors and OpenCV. Array interface is the best and the most important feature of numpy.

### **Features of Numpy**

- **Interactive:** Numpy is very interactive and easy to use.
- **Mathematics:** Varieties complex mathematical operation very simple.
- **Intuitive:** Makes coding actual easy and grasping the ideas is easy.

We use this library for decoding our images and give a semantic array.



Figure 3.4.1: Numpy

## Tkinter:

Python offers multiple choices for developing graphical user interface. Out of all the graphical user interface strategies, tkinter is most typically used technique. It's a customary Python interface to the Tk graphical user interface toolkit shipped with python. Python with tkinter outputs the quickest and easiest method to make the graphical user interface applications. Making a graphical user interface victimization tkinter is a straightforward task.

We use this library for our front end design. It's so user friendly and easy to use.

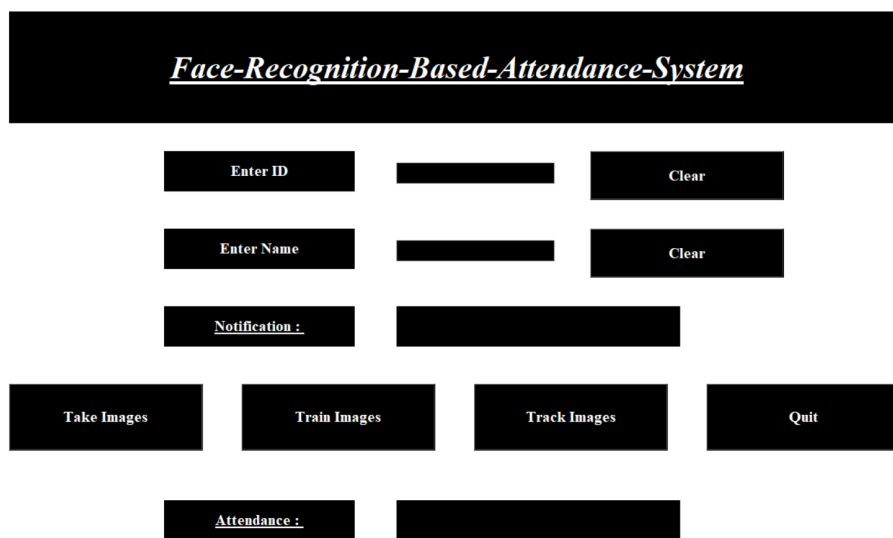


Figure 3.4.2: Tkinter GUI of our project.

### 3.5 Data flow diagram

A data flow sheet (DFD) could be a graphical illustration of the “flow” of knowledge through Associate in nursing data system modeling its method aspects. A DFD is commonly used as a preliminary step to form an outline of the system while not going into nice detail. DFD also can use for the image of knowledge process (structured design).

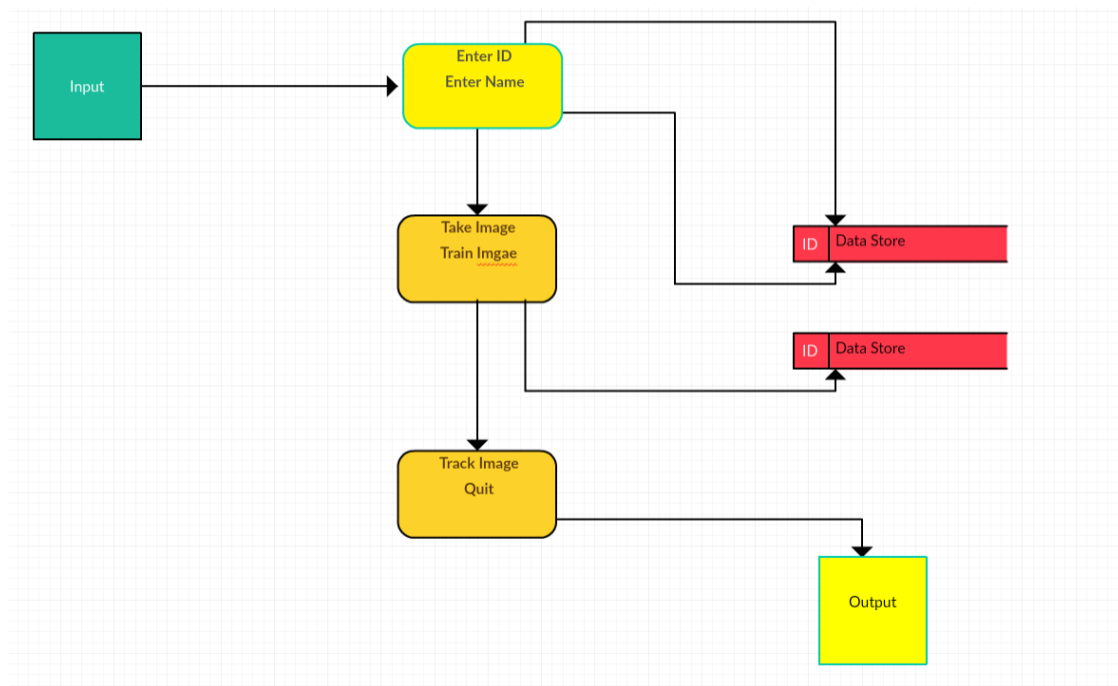


Figure 3.5.1: DFD of this project.

## CHAPTER 4 DESIGN SPECIFICATION

### 4.1 Front-End Design

The GUI design is front end design and that is the main attraction is its front-end design. This is the way of communication between the users and the backend. Front end design also known as client-side development. We made a front-end design for the users to co-operate with the system easily.

In our project we use Tkinter python GUI library for front end design which is very user friendly and easy to use.

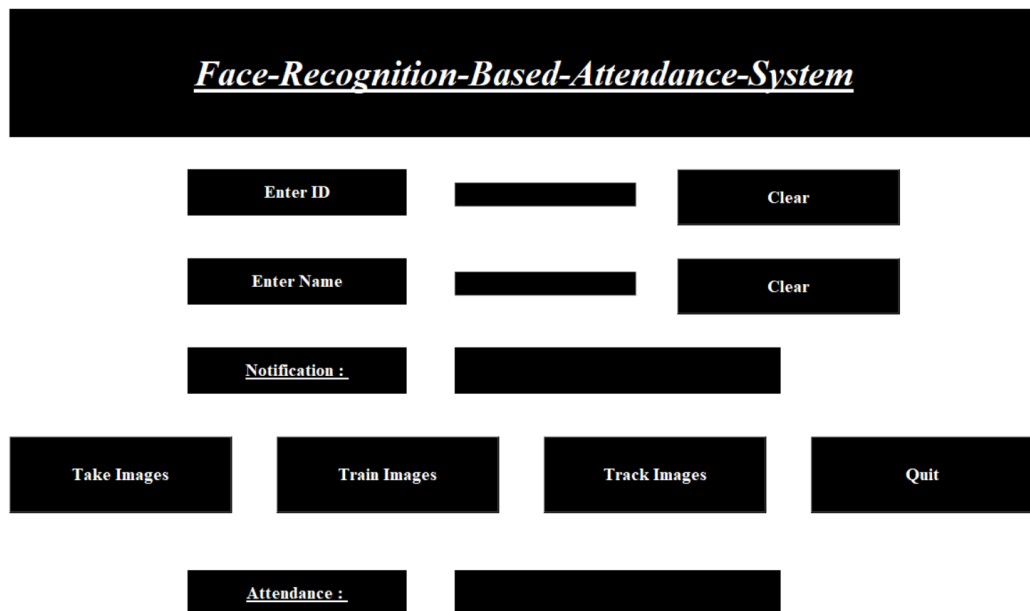
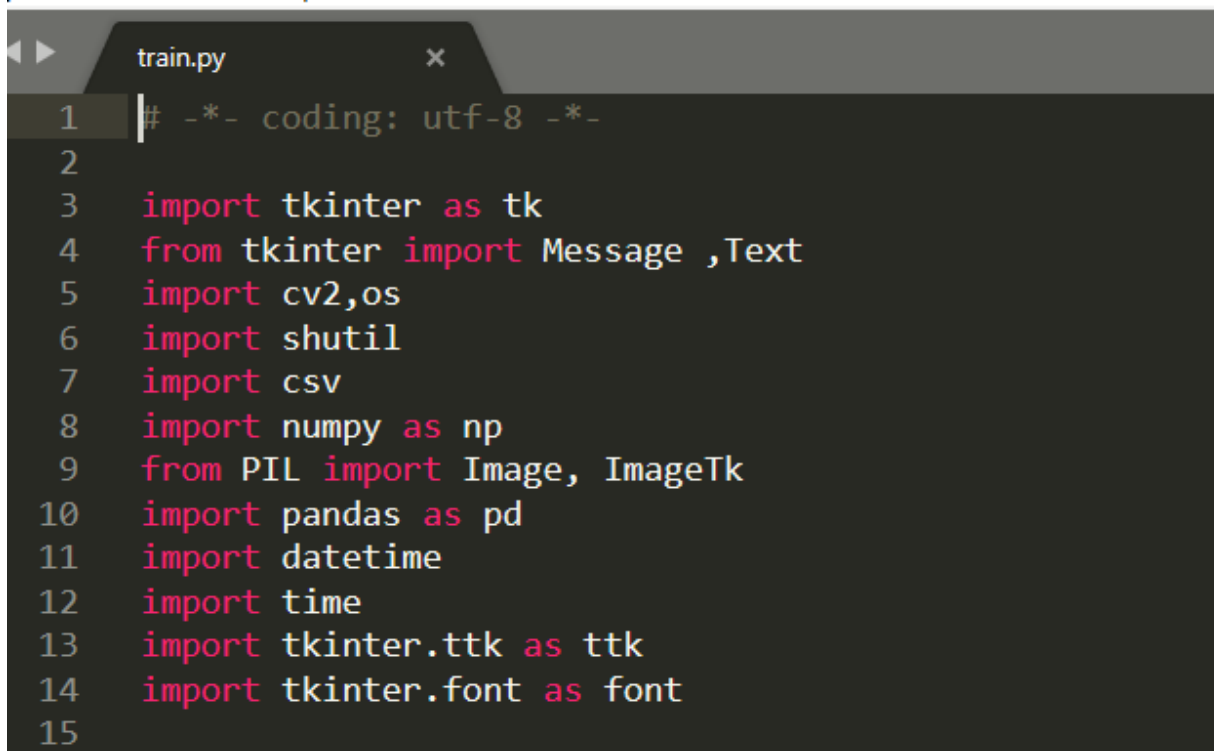


Figure 4.1.1: Front End Design.

## 4.2 Back End Design

The main program of project is back end part. Back end connect with front end and make front end responsive. Back end design is design a program for a system.

In our project we used to design back end machine learning python libraries. Those are openCV, numpy, tkinter, pandas and pillow. By using this libraries we designed our project back end.

A screenshot of a code editor window titled 'train.py'. The code is written in Python and shows the import statements for various libraries. The lines are numbered from 1 to 15. The code is as follows:

```
1  # -*- coding: utf-8 -*-
2
3  import tkinter as tk
4  from tkinter import Message ,Text
5  import cv2,os
6  import shutil
7  import csv
8  import numpy as np
9  from PIL import Image, ImageTk
10 import pandas as pd
11 import datetime
12 import time
13 import tkinter.ttk as ttk
14 import tkinter.font as font
15
```

Figure 4.2.1: Back end design

```

112 def TakeImages():
113     Id=(txt.get())
114     name=(txt2.get())
115     if(is_number(Id) and name.isalpha()):
116         cam = cv2.VideoCapture(0)
117         harcascadePath = "haarcascade_frontalface_default.xml"
118         detector=cv2.CascadeClassifier(harcascadePath)
119         sampleNum=0
120         while(True):
121             ret, img = cam.read()
122             gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
123             faces = detector.detectMultiScale(gray, 1.3, 5)
124             for (x,y,w,h) in faces:
125                 cv2.rectangle(img, (x,y),(x+w,y+h),(255,0,0),2)
126                 #incrementing sample number
127                 sampleNum=sampleNum+1
128                 #saving the captured face in the dataset folder TrainingImage
129                 cv2.imwrite("TrainingImage"+name+"."+Id+'.'+str(sampleNum) + ".jpg", gray[y:y+h,x:x+w])
130                 #display the frame
131                 cv2.imshow('frame',img)
132                 #wait for 100 milliseconds
133                 if cv2.waitKey(100) & 0xFF == ord('q'):
134                     break
135                 # break if the sample number is morethan 100
136                 elif sampleNum>60:
137                     break
138             cam.release()
139             cv2.destroyAllWindows()
140             res = "Images Saved for ID : " + Id + " Name : " + name
141             row = [Id , name]
142             with open('StudentDetails\StudentDetails.csv','a+') as csvFile:
143                 writer = csv.writer(csvFile)
144                 writer.writerow(row)
145             csvFile.close()
146             message.configure(text= res)
147         else:
148             if(is_number(Id)):
149                 res = "Enter Alphabetical Name"
150                 message.configure(text= res)
151             if(name.isalpha()):
152                 res = "Enter Numeric Id"
153                 message.configure(text= res)

```

Figure 4.2.2: Back end program

```

def TrainImages():
    recognizer = cv2.face_LBPHFaceRecognizer.create()#recognizer = cv2.face
    harcascadePath = "haarcascade_frontalface_default.xml"
    detector =cv2.CascadeClassifier(harcascadePath)
    faces,Id = getImagesAndLabels("TrainingImage")
    recognizer.train(faces, np.array(Id))
    recognizer.save("face-trainer.yml")
    res = "Image Trained"#+" ".join(str(f) for f in Id)
    message.configure(text= res)

def getImagesAndLabels(path):
    #get the path of all the files in the folder
    imagePath=[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
    for imagePath in imagePath:
        #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

def TrackImages():
    recognizer = cv2.face.LBPHFaceRecognizer_create()#cv2.createLBPHFaceRe
    recognizer.read("face-trainer.yml")
    harcascadePath = "haarcascade_frontalface_default.xml"
    faceCascade = cv2.CascadeClassifier(harcascadePath);
    df=pd.read_csv("StudentDetails\StudentDetails.csv")
    cam = cv2.VideoCapture(0)
    font = cv2.FONT_HERSHEY_SIMPLEX
    col_names = ['Id','Name','Date','Time']
    attendance = pd.DataFrame(columns = col_names)
    while True:
        ret, im =cam.read()
        gray=cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)
        faces=faceCascade.detectMultiScale(gray, 1.2,5)

```

Figure 4.2.3: Back end program



### 4.3 Experimental Result

After building a project to check his performance and is it working or not it's called experimental result. Our project experimental result given below:

When we run the project this is the first page will show.

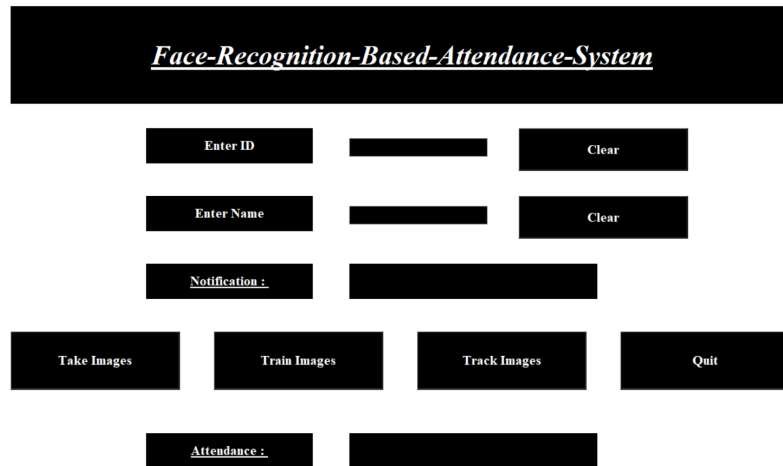


Figure 4.3.1: Experimental result 1

After train image the result is:

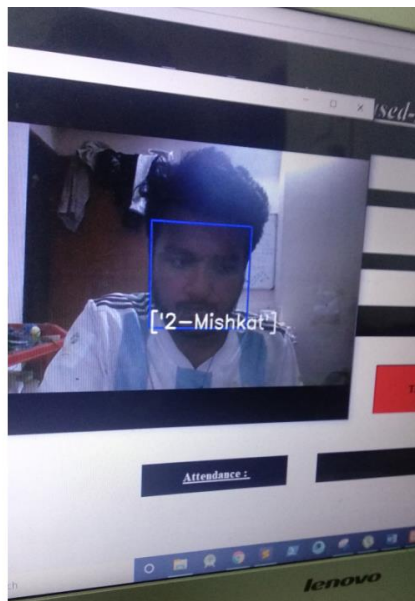


Figure 4.3.2: Experimental Result 2

Our project can detect faces also recognize the trained person images. Our project detect and recognize real time face.

## CHAPTER 5

### IMPLEMENTING AND TESTING

#### 5.1 Take Image

Machine learning is train a machine. For our face recognize we have to train our dataset. By using openCV and Pillow we detect face and capture image and saved in dataset. At a time we set capture image 61 pieces so that accuracy will perfect. Those image will take in gray filter.

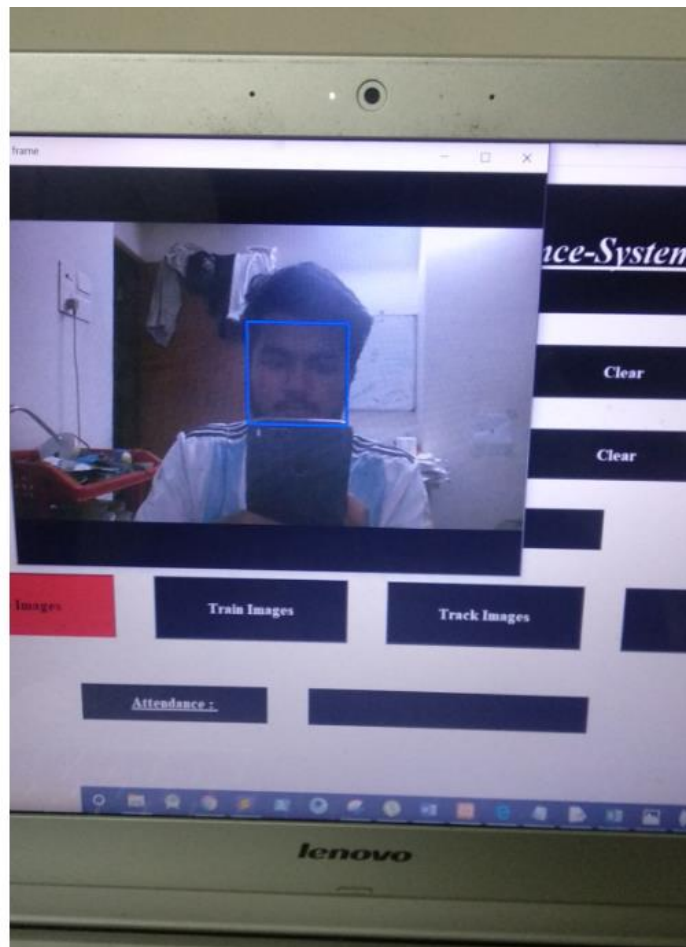


Figure 5.1.1: Take Image



Figure 5.1.2: Taken Images

## 5.2 Implementation of Dataset

A dataset is so important for machine learning projects. Dataset is vast collection of data. If we have more data we learn machine more and efficiency will best. Dataset can create in many format such as CSV, EXCEL, JSON format.

Our project dataset will set automatically after taking image. By using Pillow and pandas library images, name and id will save automatically in Students details and training image folder.

Ciipooara

	A	B
25		
26		
27		
28	35	Zozo
29		
30	36	Arif
31		
32	38	Tonu
33		
34	39	Prome
35		
36	40	Nabila
37		
38	41	Arifa
39		
40	45	Sazal
41	2	Mishkat
42		
43	3	emon
44		
45	5	Ataul
46		
47		

Figure 5.2.1: Dataset



Figure 5.2.2: Image Dataset

### 5.3 Training Image

In machine learning has some algorithm to train image one of them is LBPH face recognizer. This algorithm is the easiest algorithm for image training. For use LBPH algorithm we need grayscale filter facial images. Images can be represented as a 3\*3 matrix and images value decode in binary number. For a face a matrix will generated.

This algorithm built into OpenCV library so that we don't need to implement this algorithm for face recognizer.

After taking image we have to press train images then image will trained.

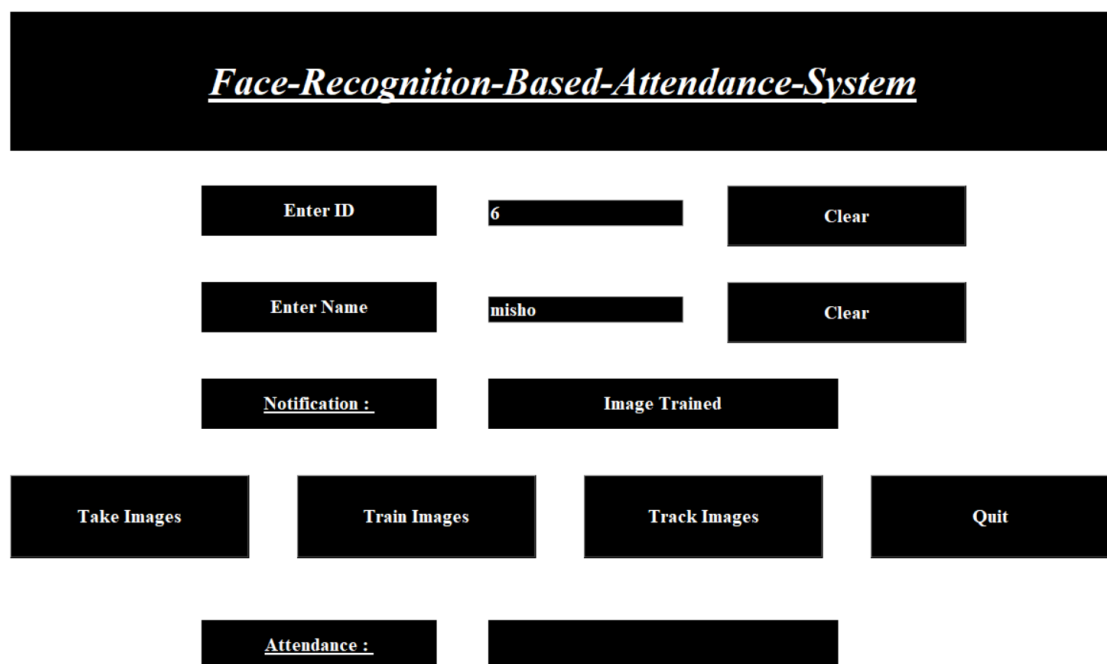


Figure 5.3.1: Train Image

## 5.4 Recognize Image

For face recognize we use OpenCv library. This library has recognition algorithm which is easy to use.

In our project after taking image saved in dataset and trained images, for recognize we use track image button. It recognize real time image and the accuracy above 70%. LBPH algorithm is help to recognize image and also train images. OpenCv and Pillow library helps detect and recognize real time images.

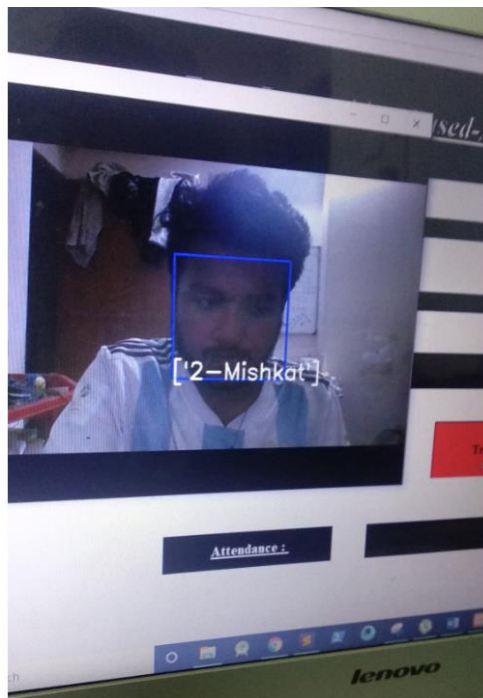


Figure 5.4.1: Face recognize

## 5.5 Test Results and Reports

The test report is wanted to reflect testing results in a formal way, which gives a scope estimate testing results speedily. Our project has need images for training after training that image it can recognize the images. In our GUI take image, train image, track image and quit four button is handling by backend code. When we press take image web cam will open and start capturing pic it will capture 60 picture of person in 10s then picture will saved. Train button will train our images and track button will recognize train images. LBPH algorithm helps to train images and this algorithm built in OpenCV library. This is a user friendly GUI. This can be used in many sectors. This test report is very important and it is needed to know that the system is ready or not ready for implementation. Given some picture of our project test results. The report of our project is it's working perfectly [7].

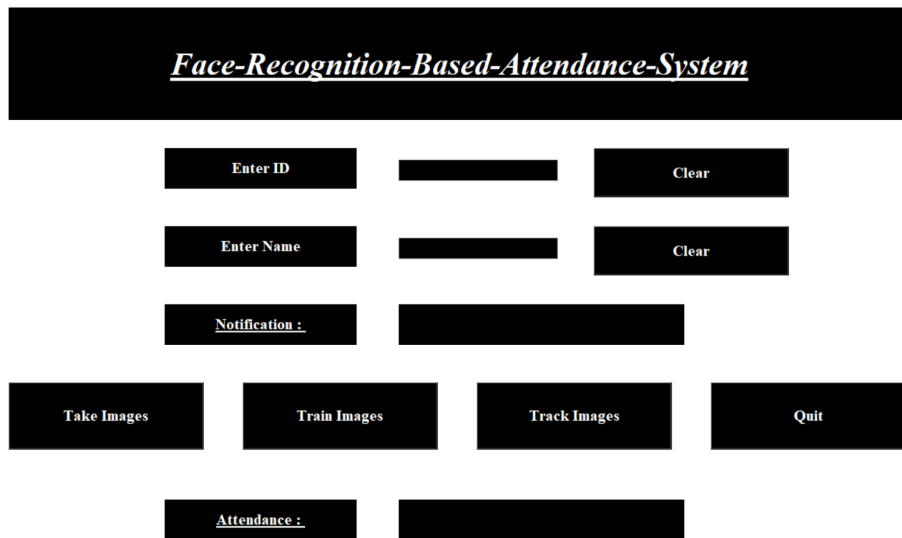


Figure 5.5.1: Test result 1

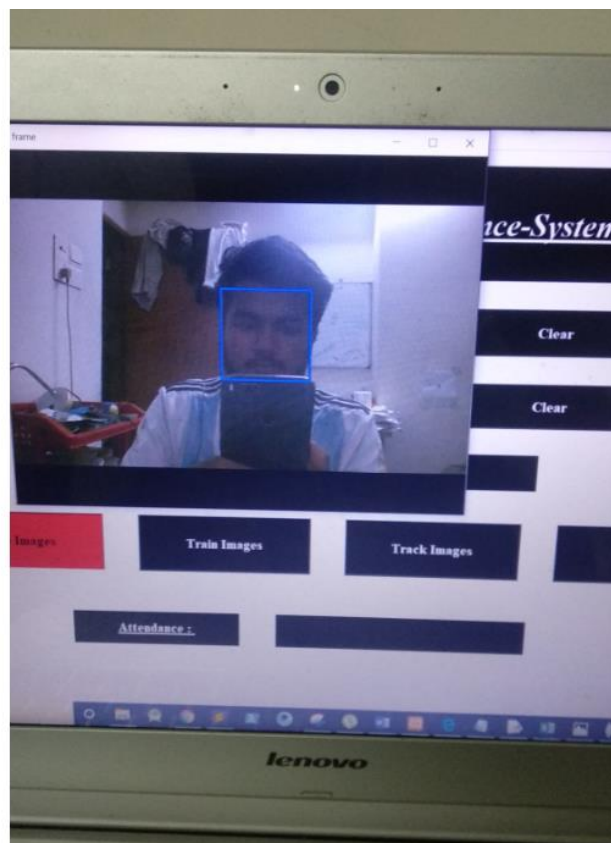


Figure 5.5.2: Test result 2



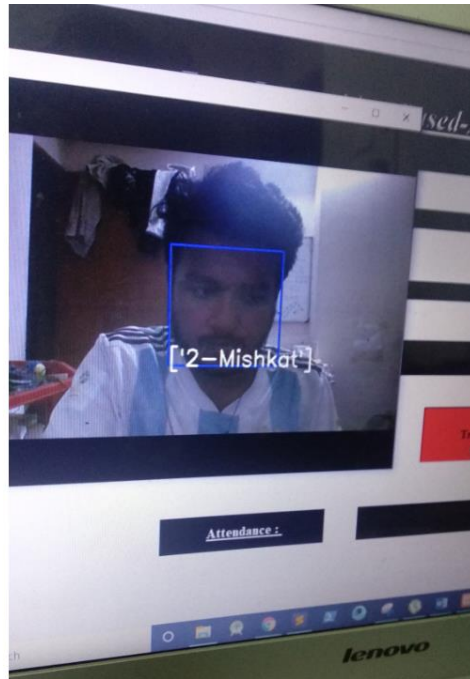


Figure 5.5.3: Test result3

Clipboard	
A	B
25	
26	
27	
28	35 Zozo
29	
30	36 Arif
31	
32	38 Tonu
33	
34	39 Prome
35	
36	40 Nabila
37	
38	41 Arifa
39	
40	45 Sazal
41	2 Mishkat
42	
43	3 emon
44	
45	5 Ataul
46	
47	

Figure 5.5.4: Test Result 4



Figure 5.5.5: Test Result 5

## Test Implementation

Table 5.5.1: Test Implementation

Test Case	Test Inputs	Expected Outcome	Obtained Outcome	Pass/Fail	Tested On
1. Run	In laptop python environment	Show GUI	Not Showed GUI	Fail	24.10.2019
2. Import Library	Train and Track Image	Successfully Train and track image	Successfully Train and track image	Pass	24.10.2019
3. Take image	Save image data	Successfully saved	Successfully saved	Pass	24.10.2019

## 5.6 Implementation of Requirements

Our project is a machine learning project and to run machine learning we need python, R programming language. We built in python environment. For implementation this project we need this requirements.

- Python environment set up.
- Python IDE.
- Download and import Machine learning libraries.
- Download and import Python libraries
- Best quality webcam
- 8gb ram computer with idle processor.

## CHAPTER 6

### CONCLUSION AND FUTURE WORK

#### 6.1 Discussion and Conclusion

This project helps to take attendance easily and it's a modern era project. Now machine learning used in every sector and makes our life easy and simple. Just set a web camera capture image and train those images then those pic will recognize automatically. This project kind of beta project. This project will come with more accuracy of face recognition. At the end we tried to make this project perfect and efficient.

#### 6.2 Further Plan

System features will upgrade day by day for better experience. This era machine learning upgrading rapidly.

In our project we used OpenCV library for recognize person. Our plan to update this system with more accuracy and add more options like:

- Object detection.
- In video detect faces.
- Implement dataset big.
- Train image with more informations.

We are thinking by using react native js we will convert this system into mobile as application.

## REFERENCES

- [1] Alexander Mordvintsev & Abid K, “OpenCV-Python Tutorials Documentation,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, November 5, 2017.
- [2] Face recognition pyimagesearch, available at <<<https://www.pyimagesearch.com/2018/09/24/opencv-face-recognition/>>>, last accessed on 06-06-2019 at 12:00 PM.
- [3] Face detection github, available at <<<https://github.com/mikeckennedy/PySimple>>>, last accessed in 9-10-2019 at 1.00 PM.
- [4] OpenCV geeksforgeek, available at << <https://www.geeksforgeeks.org/opencv-overview/>>>, last accessed in 9-10-2019 at 2.00 PM.
- [5] Machine learning libraries, available at << <https://towardsdatascience.com/top-5-machine-learning-libraries-in-python-e36e3e0e02af>>> last accessed in 9-10-2019 at 2.30 PM.
- [6] Python libraries , geeksforgeek, available at << <https://www.geeksforgeeks.org/best-python-libraries-for-machine-learning/>>> last accessed in 9-10-2019 at 3.00 PM.
- [7] Joseph Howse, “Training detectors and recognizers in Python and OpenCV” 2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), 12 September 2014.

## **APPENDIX**

### **Project Reflection**

In fall 2018 we completed our artificial intelligence course. That course we introduced with OpenCV machine learning library. And we learn that how to use this where to implement this library. Then we decided to make a normal face detection and face recognition desktop application. We built it after showing it to our supervisor sir , he said implement it in any sector. From his advice we tried to make this system and finally we build it. So we believed that this system will help to take attendance in any institute. In a periodic system we will update our system for more efficiency.

## Meshkat

### ORIGINALITY REPORT

<b>20%</b>	<b>4%</b>	<b>1%</b>	<b>20%</b>
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

### PRIMARY SOURCES

<b>1</b>	<b>Submitted to Daffodil International University</b> Student Paper	<b>7%</b>
<b>2</b>	<b>Submitted to Karunya University</b> Student Paper	<b>2%</b>
<b>3</b>	<b>Submitted to North South University</b> Student Paper	<b>1%</b>
<b>4</b>	<b>Submitted to The British College</b> Student Paper	<b>1%</b>
<b>5</b>	<b><a href="http://www.teamindia.com.au">www.teamindia.com.au</a></b> Internet Source	<b>1%</b>
<b>6</b>	<b>Submitted to Heriot-Watt University</b> Student Paper	<b>1%</b>
<b>7</b>	<b>Submitted to UC, Boulder</b> Student Paper	<b>1%</b>
<b>8</b>	<b>Submitted to International University - VNUHCM</b> Student Paper	<b>1%</b>
<b>9</b>	<b>Submitted to University of Missouri, Kansas City</b> Student Paper	<b>1%</b>

