Project Report on

“ATTENDANCE SYSTEM USING FACE DETECTION”



**For Partial Fulfillment of ‘A’ level course**

Under the Guidance of **Mr. Dyutiman Basu**

*Submitted By*

**BIJAYA BEHERA**

COURSE: **‘A’ LEVEL**

REGISTRATION NO: **1346223**

****

**The Institute Of Computer Engineers (INDIA)**

**Kolkata, West Bengal - 700071**

**ATTENDANCE SYSTEM USING**

**FACE DETECTION**

## DECLARATION

This is to certify that the project work entitled **“ATTENDANCE SYSTEM USING FACE DETECTION ” is a bonafide work carried out by Bijaya Behera** as a part of **A Level final year** of **The Institute Of Computer Engineers (INDIA)** during the year **2021-2022.**

I am **Bijaya Behera,** student of Final year student of The Institute Of Computer Engineers (INDIA) from Kolkata, West Bengal, hereby declare that the project work entitled “**ATTENDANCE SYSTEM USING FACE DETECTION**” is carried out by me and submitted in fulfillment of the requirements for the award of NIELIT 'A' Level under Mr. Dyutiman Basu during the Academic year 2020- 2022 and has not been submitted to any other university.

**Bijaya Behera**

**Course: NIELIT ‘A’ Level**

**Registration Number: 1346223**

**ACKNOWLEDGEMENT**

An endeavor over a long period can be advice and support of many well- wishers. I take this opportunity to express our gratitude and appreciation to all of them.

I owe our tributes to Mr. Dyutiman Basu, Software Developer at Techno India Group for his valuable support and guidance during the period of project implementation.

I wish to express our sincere thanks and gratitude for the simulating discussions, in analyzing problems associated with our project work and for guiding us throughout the project. We express our sincere thanks for the encouragement, untiring guidance and the confidence they had shown in us.

I also thank Teaching staff and all non-teaching staff of the department of The Institute Of Computer Engineers (INDIA) for providing resources when required.

**Bijaya Behera**

**Course: NIELIT ‘A’ Level**

**Registration Number: 1346223**

**CONTENT**

Introduction 1

#### Problem definition 4

Proposed system (computerized) 6

Functionality of the proposed system 14

System design 19

Data Base 20

DFD 21

Flowchart 22

Input Face 23

Software/hardware 29

Specification to be used

Feasibility study 32

PERT chart 33

Cost Benefit Analysis 38

Coding 39

Sample input/output screen design 71

System testing 76

User Manual 79

#### Scope & Limitation 81

Bibliography 84

**INTRODUCTION**

The main objective of this project is to develop face recognition based automated student attendance system. In order to achieve better performance, the test images and training images of this proposed approach are limited to frontal and upright facial images that consist of a single face only. The test images and training images have to be captured by using the same device to ensure no quality difference. In addition, the students have to register in the database to be recognized. The enrolment can be done on the spot through the user-friendly interface.

**BACKGROUND**

Face recognition is crucial in daily life in order to identify family, friends or someone we are familiar with. We might not perceive that several steps have actually taken in order to identify human faces. Human intelligence allows us to receive information and interpret the information in the recognition process. We receive information through the image projected into our eyes, by specifically retina in the form of light.

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

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

#### PROBLEM DEFINITION

Traditional student attendance marking technique is often facing a lot of trouble. The face recognition student attendance system emphasizes its simplicity by eliminating classical student attendance marking technique such as calling student names or checking respective identification cards. There are not only disturbing the teaching process but also causes distraction for students during exam sessions. Apart from calling names, attendance sheet is passed around the classroom during the lecture sessions. The lecture class especially the class with a large number of students might find it difficult to have the attendance sheet being passed around the class. Thus, face recognition student attendance system is proposed in order to replace the manual signing of the presence of students which are burdensome and causes students get distracted in order to sign for their attendance. Furthermore, the face recognition based student attendance system able to overcome the problem of fraudulent approach and lecturers does not have to count the number of students several times to ensure the presence of the students.

#### Aims and Objective

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

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

#### Proposed System

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

**Advantages & Disadvantages of Different Biometric System**

|  |  |  |
| --- | --- | --- |
| **System type** | **Advantages** | **Disadvantages** |
| RFID card system | Simple | Fraudulent usage |
| Fingerprint system | Accurate | Time-consuming |
| Voice recognition  system | - | Less accurate compared to others |
| Iris recognition  system | Accurate | Privacy Invasion |

#### Face Detection

Difference between face detection and face recognition are often misunderstood. Face detection is to determine only the face segment or face region from image, whereas face recognition is to identify the owner of the facial image. A few factors which cause face detection and face recognition to encounter difficulties. These factors consist of background, illumination, pose, expression, occlusion, rotation, scaling and translation. The definition of each factor is tabulated in below Table

**Factors Causing Face Detection Difficulties**

|  |  |
| --- | --- |
| Background | Variation of background and environment around people in the image which affect the efficiency of face recognition. |
| Illumination | Illumination is the variation caused by various lighting environments which degrade the facial feature detection. |
| Pose | Pose variation means different angle of the acquired the facial image which cause distortion to recognition process, especially for Eigen face and Fisher face recognition method. |
| Expression | Different facial expressions are used to express feelings and emotions. The expression variation causes spatial relation change and the facial-feature shape change. |
| Occlusion | Occlusion means part of the human face is unobserved. This will diminish the performance of face recognition algorithms due to deficiency information. |
| Rotation, scaling and translation | Transformation of images which might cause distortion of the original information about the images. |

There are a few face detection methods that the previous researchers have worked on. However, most of them used frontal upright facial images which consist of only one face. The face region is fully exposed without obstacles and free from the spectacles.

Algorithm for face detection for student attendance system. They concluded that out of methods such as face geometry- based methods, Feature Invariant methods and Machine learning based methods, algorithm is not only fast and robust, but gives high detection rate and perform better in different lighting condition.

#### Pre-Processing

Cropping of detected face and colour image was converted to grayscale for pre-processing. They also proposed affine transform to be applied to align the facial image based on coordinates in middle of the eyes and scaling of image to be performed.

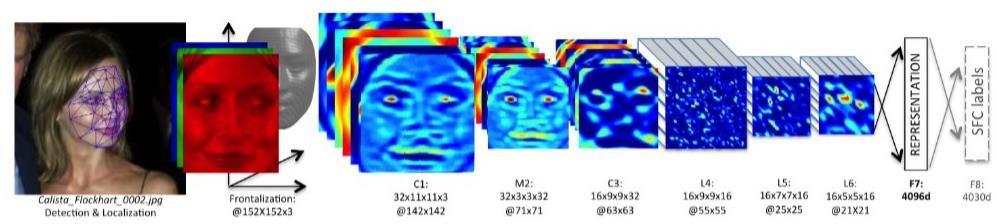
Pre-processing enhances the performance of the system. It plays an essential role to improve the accuracy of face recognition. Scaling is one of the important pre- processing steps to manipulate the size of the image. Scaling down of an image increases the processing speed by reducing the system computations since the number of pixels are reduced. The size and pixels of the image carry spatial information. The size should be same for all the images for normalization and standardization purposes.

Besides scaling of images, colour image is usually converted to grayscale image for pre-processing. Grayscale images are believed to be less sensitive to illumination condition and take less computational time. Grayscale image is 8 bit image which the pixel range from 0 to 255 whereas colour image is 24 bit image which pixel can have 16 77 7216 values. Hence, colour image requires more storage space and more computational power compared to grayscale images.. If colour image is not necessary in computation, then it is considered as noise. In addition, pre-processing is important to enhance the contrast of images. Histogram equalization is one of the methods of pre-processing in order to improve the contrast of the image. It provides uniform distribution of intensities over the intensity level axis, which is able to reduce uneven illumination effect at the same time.

**Functionality of the proposed system**

The feature is a set of data that represents the information in an image. Extraction of facial feature is most essential for face recognition. However, selection of features could be an arduous task. Feature extraction algorithm has to be consistent and stable over a variety of changes in order to give high accuracy result.

There are a few feature extraction methods for face recognition. They proposed PCA for the face recognition. PCA is famous with its robust and high speed computation. Basically, PCA retains data variation and remove unnecessary existing correlations among the original features. PCA is basically a dimension reduction algorithm. It compresses each facial image which is represented by the matrix into single column vector. Furthermore, PCA removes average value from image to centralize the image data. The Principle Component of distribution of facial images is known as Eigen faces. Every single facial image from training set contributes to Eigen faces. As a result, Eigen face encodes best variation among known facial images. Training images and test images are then projected onto Eigen face space to obtain projected training images and projected test image respectively. Euclidean distance is computed by comparing the distance between projected training images and projected test image to perform the recognition. PCA feature extraction process includes all trained facial images. Hence, the extracted feature contains correlation between facial images in the training set and the result of recognition of PCA highly depends on training set image.



#### Types of Feature Extraction

Face recognition system can be categorized into a few Holistic-based methods, Feature-based methods and Hybrid methods. Holistic-based methods are also known as appearance-based methods, which mean entire information about a face patch is involved and used to perform some transformation to obtain a complex representation for recognition. Example of Holistic-based methods are PCA(Principal Component Analysis) and LDA(Linear dependent Analysis).On the other hand, feature-based methods directly extract detail from specific points especially facial features such as eyes, noses, and lips whereas other information which is considered as redundant will be discarded. Example of feature-based method is LBP (Local Binary Pattern). These methods mentioned are usually combined to exist as Hybrid method, for example Holistic-based method combine with Feature-based in order to increase efficiency.

#### Feature Classification & Face Recognition

Classification involves the process of identification of face. Distance classifier finds the distance between the test image and train image based on the extracted features. The smaller the distance between the input feature points and the trained feature points, the higher the similarity of the test image and training image. In other words, the facial images with the smallest/minimum distance will be classified as the same person.

#### Measuring System Performance

Different databases are used in order to evaluate the system performance. The database provided by previous researchers with different variable conditions, for example, lighting and expression will be used to justify the system and for study purpose. Furthermore, our own database will be used to analyze the system for real time application. From the literature review of the previous researchers, the common method to justify the performance of the system is by finding the accuracy of recognition.

#### System design

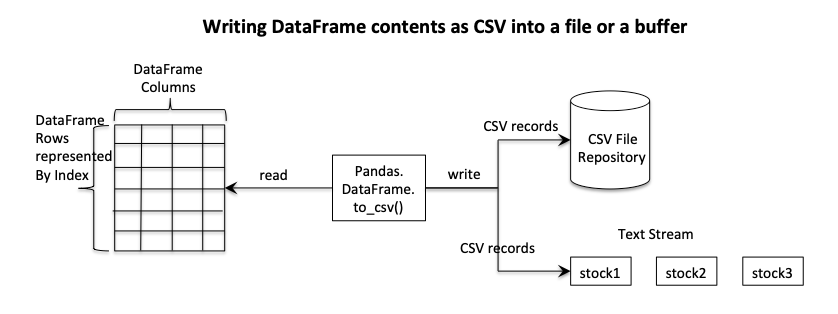
#### Methodology Flow

The approach performs face recognition based student attendance system. The methodology flow begins with the capture of image by using simple and handy interface, followed by pre-processing of the captured facial images, then feature extraction from the facial images, subjective selection and lastly classification of the facial images to be recognized. Both LBP and PCA feature extraction methods are studied in detail and computed in this proposed approach in order to make comparisons. LBP is enhanced in this approach to reduce the illumination effect. An algorithm to combine enhanced LBP and PCA is also designed for subjective selection in order to increase the accuracy. The details of each stage will be discussed in the following sections.

### Data Base

### A database is an organized collection of data, so that it can be easily accessed and managed.You can organize data into tables, rows, columns, and index it to make it easier to find relevant information.

### By using pandas. DataFrame.to\_csv() method you can write/save/export a pandas DataFrame to CSV File. By default to\_csv() method export DataFrame to a CSV file with comma delimiter and row index as the first column. In this article, I will cover how to export to CSV file by a custom delimiter, with or without column header, ignoring index, encoding, quotes, and many more.



#### DFD

#### 48c7d48f-02fe-4a23-93a0-795e0e3bd3eb.jfif

#### 

#### Flow Chart

#### 

#### Input Face

Although our own database should be used to design real time face recognition student attendance system, the databases that are provided by the previous researchers are also used to design the system more effectively, efficiently and for evaluation purposes. face database is used as both training set and testing set to evaluate the performance. Yale face database contains one hundred and sixty-five grayscale images of fifteen individuals. There are eleven images per individual; each image of the individual is in different condition. The conditions included centre-light, with glasses, happy, left-light, without glasses, normal, right-light, sad, sleepy, surprised and wink. These different variations provided by the database is able to ensure the system to be operated consistently in variety of situations and conditions.



**Sample**

**Images in**

**Face Database**

#### Limitations of the Images

The input image for the proposed approach has to be frontal, upright and only a single face. Although the system is designed to be able to recognize the student with glasses and without glasses, student should provide both facial images with and without glasses to be trained to increase the accuracy to be recognized without glasses. The training image and testing image should be captured by using the same device to avoid quality difference. The students have to register in order to be recognized. The enrolment can be done on the spot through the user-friendly interface. These conditions have to be satisfied to ensure that the proposed approach can perform well.

#### Scaling of Image

Scaling of images is one of the frequent tasks in image processing. The size of the images has to be carefully manipulated to prevent loss of spatial information. In order to perform face recognition, the size of the image has to be equalized. This has become crucial, especially in the feature extraction process, the test images and training images have to be in the same size and dimension to ensure the precise outcome. Thus, in this proposed approach test images and train images are standardize at size 250 × 250 pixels

#### Median Filtering

Median filtering is a robust noise reduction method. It is widely used in various applications due to its capability to remove unwanted noise as well as retaining useful detail in images. Since the colour images captured by using a camera are RGB images, median filtering is done on three different channels of the image. The image before and after noise removal by median filtering in three channels. If the input image is a grayscale image, then the median filtering can be performed directly without separating the channels.



Median multy Filtering Median Filteringa Single

#### Conversion to Grayscale Image

Camera captures color images, however the proposed contrast improvement method CLAHE can only be performed on grayscale images. After improving the contrast, the illumination effect of the images able to be reduced. LBP extracts the grayscale features from the contrast improved images as 8 bit texture descriptor. Therefore, color images have to be converted to grayscale images before proceeding to the later steps. By converting color images to grayscale images, the complexity of the computation can be reduced resulting in higher speed of computation.



Conversion of Image to Grayscale Image

#### Feature Extraction

Different facial images mean there are changes in textural or geometric information. In order to perform face recognition, these features have to be extracted from the facial images and classified appropriately. In this project, enhanced LBP and PCA are used for face recognition. The idea comes from nature of human visual perception which performs face recognition depending on the local statistic and global statistic features. Enhanced LBP extracts the local grayscale features by performing feature extraction on a small region throughout the entire image. On the other hand, PCA extracts the global grayscale features which means feature extraction is performed on the whole image

#### 

#### Software/Hardware specification to be used

* + 1. **Software Requirements:**

The software configurations used are

Operating System: Windows10

Programming Language : Python

Audio file format: m4a (any file format is accepted)

### Hardware Requirements:

Processor: INTEL

RAM: Minimum of 256 MB or higher HDD: 10GB or higher

Monitor: 15” or 17” color monitor Keyboard: Standard 110 keys

Mouse Type:USB

#### A fairly simple way is mentioned to implement facial recognition system using Python and OpenCV module along with the explanation of the code step by step in the comments. Before starting we need to install some libraries in order to implement the code. Below you will see the usage of the library along with the code to install it:

#### • OpenCV:

#### OpenCV (Open Source Computer Vision Library) is an opensource computer vision and machine learning software library. which is built to provide a common infrastructure for machine learning algorithms and computer vision. It has thousands of optimized algorithms which can be used different purposes like detecting and recognizing faces, identifying objects and many more. We need it to take pictures using our webcam and some manipulation needed to be done in the image. To install the library you need to install pip in your system after that you can follow the steps in command prompt:

#### Step 1: pip install opencv-python

#### Step 2: pip install opencv-contrib-python

#### Step 3: pip install Pillow

Step 4: pip install pandas

#### • NumPy:

#### NumPy is the fundamental package for scientific computing in Python which provides a multidimensional array object other mathematical operations can be performed using this but simply speaking we just need it to convert our images into some form of an array so that we can store the model that has been trained. To install the library you can type a simple line of code in your command shell: pip install numpy

#### • Haar Cascade:

#### Haar Cascade is basically a classifier which is used to detect the objects for which it has been trained for, from the source. The result is an XML file which stores the trained result. If said simply the Haar Cascade is trained by superimposing the positive image over a set of negative images. The training requires a high spec system and a good internet connection and thousands of training images that is why it is carried out in the server. For increasing the efficiency of the results they use high-quality images and increase the number of stages for which the classifier is trained. We need haar cascade frontal face recognizer to detect the face from our webcam. To download the haar cascade files of different objects you can go the below link: GitHub: HaarCascades Python GUI (tkinter):• Tkinter is a simple GUI module used for implementing fairly simple GUI and helps us to interact with code in a simple way.

### Feasibility Study

### A feasibility study is an assessment of the practicality of a project or system. A feasibility study aims to objectively and rationally uncover the strengths and weaknesses of an existing business or proposed venture, opportunities and threats present in the [natural environment](https://en.wikipedia.org/wiki/Natural_environment), the [resources](https://en.wikipedia.org/wiki/Resources) required to carry through, and ultimately the prospects for success. In its simplest terms, the two criteria to judge feasibility are [cost](https://en.wikipedia.org/wiki/Cost) required and [value](https://en.wikipedia.org/wiki/Value_(economics)) to be attained.

### A well-designed feasibility study should provide a historical background of the business or project, a description of the [product](https://en.wikipedia.org/wiki/Product_(business)) or [service](https://en.wikipedia.org/wiki/Service_(economics)), accounting statements, details of the [operations](https://en.wikipedia.org/wiki/Business_operations) and [management](https://en.wikipedia.org/wiki/Management), [marketing research](https://en.wikipedia.org/wiki/Marketing_research) and policies, financial data, legal requirements and tax obligations. Generally, feasibility studies precede technical development and [project](https://en.wikipedia.org/wiki/Project) implementation. A feasibility study evaluates the project's potential for success; therefore, perceived objectivity is an important factor in the credibility of the study for potential investors and lending institutions.[[citation needed](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)][[5]](https://en.wikipedia.org/wiki/Feasibility_study#cite_note-5) It must therefore be conducted with an objective, unbiased approach to provide information upon which decisions can be based.

**PERT Chart**

A PERT chart is a network diagram used in the Program Evaluation Review Technique (PERT) to represent a project’ timeline. It allows project managers to estimate the duration of projects based on the analysis of task sequences. PERT charts are used by project managers to create realistic schedules by coordinating activities and estimating their duration by assigning three time estimates for each (optimistic, most likely and pessimistic). This makes PERT charts useful when planning projects where the duration of activities is uncertain.

How to make a PERT chart

1. Identify project tasks. The first step in creating a successful

PERT chart involves identifying and collecting necessary

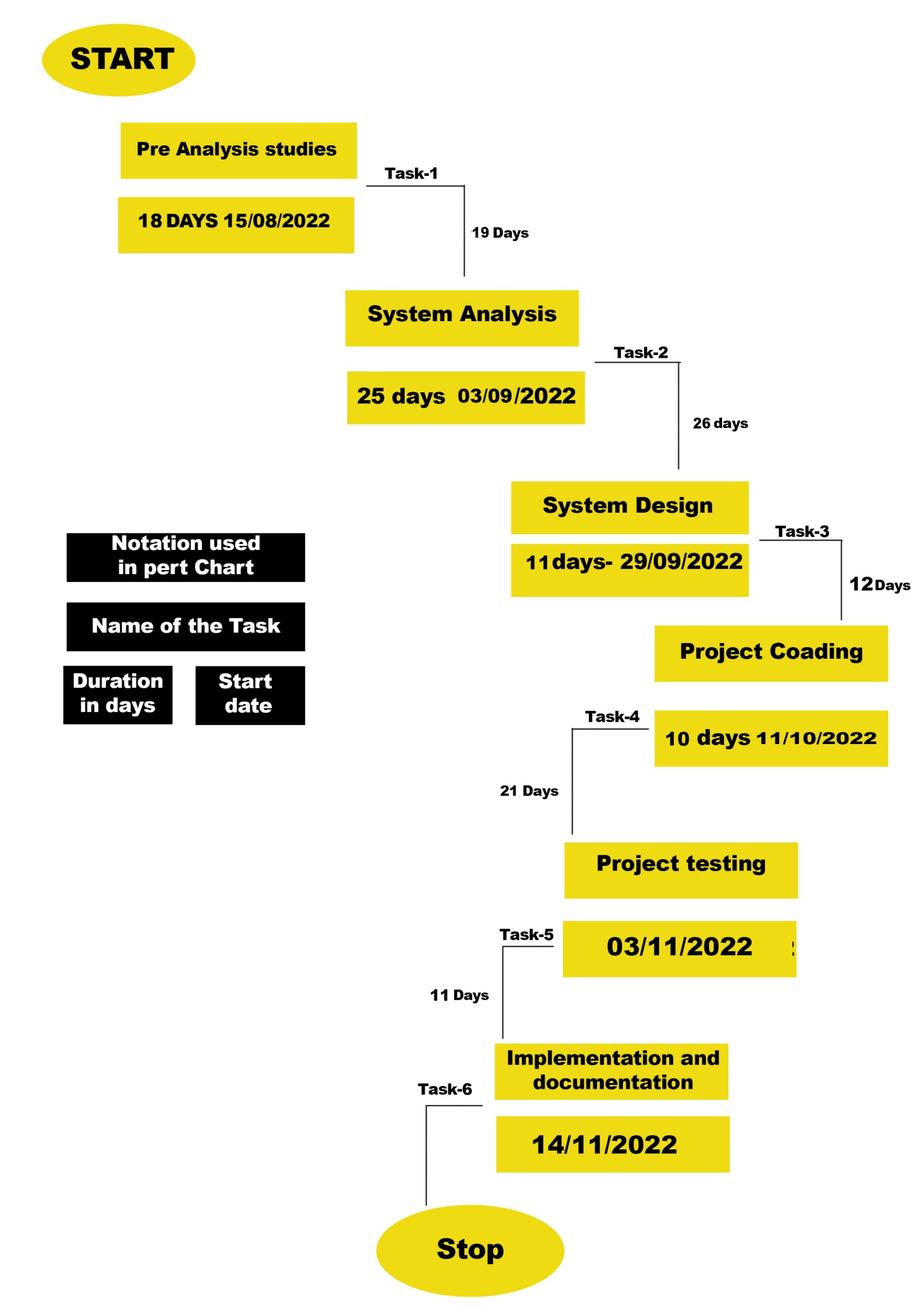
project information and tasks. ...

2. Define task dependencies. ...

3. Connect project tasks. ...

4. Estimate project time frame. ...

5. Manage task progress.



#### Discussion

This proposed approach provides a method to perform face recognition for the student attendance system, which is based on the texture based features of facial images. Face recognition is the identification of an individual by comparing his/her real-time captured image with stored images in the database of that person. Thus, the training set has to be chosen based on the latest appearance of an individual other than taking important factors for instance illumination into consideration.

The proposed approach is being trained and tested on different datasets. Yale face database which consists of one hundred and sixty-five images of fifteen individuals with multiple conditions is implemented. However, this database consists of only grayscale images. Hence, our own database with color images which is further categorized into high quality set and the low quality set, as images are different in their quality: some images are blurred while some are clearer. The statistics of each data set have been discussed in the earlier chapter.

Viola-Jones object detection framework is applied in this approach to detect and localize the face given a facial image or provided a video frame. From the detected face, an algorithm that can extract the important features to perform face recognition is designed.

Some pre-processing steps are performed on the input facial image before the features are extracted. Median filtering is used because it is able to preserve the edges of the image while removing the image noises. The facial image will be scaled to a suitable size for standardizing purpose and converted to grayscale image if it is not a grayscale image because CLAHE and LBP operator work on a grayscale image.

One of the factors that are usually a stumbling stone for face recognition performance is uneven lighting condition. Hence, many alternatives have been conducted in this proposed approach in order to reduce the non-uniform lighting condition. Before feature extraction takes place, pre-processing is performed on the cropped face image (ROI) to reduce the illumination problem.

#### Problems Faced and Solutions Taken

One of the problems in real-time face recognition is the difficulty to obtain sufficient and suitable images for training and testing purpose. It is hard to obtain in real-time databases with a variety of variables, and it is hard to obtain publicly available databases. Face database is one of the databases that could be downloaded by the public. Hence, Yale face database is adopted and used in this proposed approach. However, Yale face database consists of only grayscale images without any background. Hence, our own database consists of colour images which is categorized to high- quality images and low quality-images are also used. Face recognition website, they explained that the Face SDK is a high performance, multi-platform face recognition, identification and facial feature detection solution. For Luxand Face Recognition software, the self-learning Al enables video-based identification and the enrolment can be done at any time as simple as putting a name tag in a video, the system will identify that subject in all past, present and future videos. As a video-based identification software, it is believed to work better than key-frame based identification. Nevertheless, the detailed information of its working principle is unable to be obtained from their sites.

**Cost Benefit Analysis**

### A cost-benefit analysis (CBA) is a process that is used to estimate the costs and benefits of decisions in order to find the most cost effective alternative. A CBA is a versatile method that is often used for the business, project and public policy decisions. An effective CBA evaluates the following costs and benefits:

### Costs

### Direct costs

### Indirect costs

### Intangible costs

### Opportunity costs

### Costs of potential risks

### Practically for your project calculate the software price considering the followings.

### 1. Hardware cost

### 2. Infrastructure cost

### 3. Human labor cost

### 4. Miscellaneous cost

### 5. Profit

|  |  |
| --- | --- |
| Project | Project Cost (INR) |
| Production Cost | 15,000.00 |
| Hardware Cost | 45,000.00 |
| Financial Cost | 5,000.00 |
| Misc. expense | 5,000.00 |
| Cost Price | **70,000.00** |
| Profit | 30,000.00 |
| Selling Price | **1,00,000.00** |

**Codes:**

**IMPORTING** ##############################################

import tkinter as tk

from tkinter import ttk

from tkinter import messagebox as mess

import tkinter.simpledialog as tsd

import cv2,os

import csv

import numpy as np

from PIL import Image

import pandas as pd

import datetime

import time

############################################ FUNCTIONS ############################################

def assure\_path\_exists(path):

dir = os.path.dirname(path)

if not os.path.exists(dir):

os.makedirs(dir)

##################################################################################

def tick():

time\_string = time.strftime('%H:%M:%S')

clock.config(text=time\_string)

clock.afte r(200,tick)

###################################################################################

def contact():

mess.\_show(title='Contact us', message="Please contact us on : 'bijayakumarbehera123.com' ")

###################################################################################

def check\_haarcascadefile():

exists = os.path.isfile("haarcascade\_frontalface\_default.xml")

if exists:

pass

else:

mess.\_show(title='Some file missing', message='Please contact us for help')

window.destroy()

###################################################################################

def save\_pass():

assure\_path\_exists("TrainingImageLabel/")

exists1 = os.path.isfile("TrainingImageLabel\psd.txt")

if exists1:

tf = open("TrainingImageLabel\psd.txt", "r")

key = tf.read()

else:

master.destroy()

new\_pas = tsd.askstring('Old Password not found', 'Please enter a new password below', show='\*')

if new\_pas == None:

mess.\_show(title='No Password Entered', message='Password not set!! Please try again')

else:

tf = open("TrainingImageLabel\psd.txt", "w")

tf.write(new\_pas)

mess.\_show(title='Password Registered', message='New password was registered successfully!!')

return

op = (old.get())

newp= (new.get())

nnewp = (nnew.get())

if (op == key):

if(newp == nnewp):

txf = open("TrainingImageLabel\psd.txt", "w")

txf.write(newp)

else:

mess.\_show(title='Error', message='Confirm new password again!!!')

return

else:

mess.\_show(title='Wrong Password', message='Please enter correct old password.')

return

mess.\_show(title='Password Changed', message='Password changed successfully!!')

master.destroy()

###################################################################################

def change\_pass():

global master

master = tk.Tk()

master.geometry("400x160")

master.resizable(False,False)

master.title("Change Password")

master.configure(background="white")

lbl4 = tk.Label(master,text=' Enter Old Password',bg='white',font=('times', 12, ' bold '))

lbl4.place(x=10,y=10)

global old

old=tk.Entry(master,width=25 ,fg="black",relief='solid',font=('times', 12, ' bold '),show='\*')

old.place(x=180,y=10)

lbl5 = tk.Label(master, text=' Enter New Password', bg='white', font=('times', 12, ' bold '))

lbl5.place(x=10, y=45)

global new

new = tk.Entry(master, width=25, fg="black",relief='solid', font=('times', 12, ' bold '),show='\*')

new.place(x=180, y=45)

lbl6 = tk.Label(master, text='Confirm New Password', bg='white', font=('times', 12, ' bold '))

lbl6.place(x=10, y=80)

global nnew

nnew = tk.Entry(master, width=25, fg="black", relief='solid',font=('times', 12, ' bold '),show='\*')

nnew.place(x=180, y=80)

cancel=tk.Button(master,text="Cancel", command=master.destroy ,fg="black" ,bg="red" ,height=1,width=25 , activebackground = "white" ,font=('times', 10, ' bold '))

cancel.place(x=200, y=120)

save1 = tk.Button(master, text="Save", command=save\_pass, fg="black", bg="#3ece48", height = 1,width=25, activebackground="white", font=('times', 10, ' bold '))

save1.place(x=10, y=120)

master.mainloop()

#####################################################################################

def psw():

assure\_path\_exists("TrainingImageLabel/")

exists1 = os.path.isfile("TrainingImageLabel\psd.txt")

if exists1:

tf = open("TrainingImageLabel\psd.txt", "r")

key = tf.read()

else:

new\_pas = tsd.askstring('Old Password not found', 'Please enter a new password below', show='\*')

if new\_pas == None:

mess.\_show(title='No Password Entered', message='Password not set!! Please try again')

else:

tf = open("TrainingImageLabel\psd.txt", "w")

tf.write(new\_pas)

mess.\_show(title='Password Registered', message='New password was registered successfully!!')

return

password = tsd.askstring('Password', 'Enter Password', show='\*')

if (password == key):

TrainImages()

elif (password == None):

pass

else:

mess.\_show(title='Wrong Password', message='You have entered wrong password')

######################################################################################

def clear():

txt.delete(0, 'end')

res = "1)Take Images >>> 2)Save Profile"

message1.configure(text=res)

def clear2():

txt2.delete(0, 'end')

res = "1)Take Images >>> 2)Save Profile"

message1.configure(text=res)

#######################################################################################

def TakeImages():

check\_haarcascadefile()

columns = ['SERIAL NO.', '', 'ID', '', 'NAME']

assure\_path\_exists("StudentDetails/")

assure\_path\_exists("TrainingImage/")

serial = 0

exists = os.path.isfile("StudentDetails\StudentDetails.csv")

if exists:

with open("StudentDetails\StudentDetails.csv", 'r') as csvFile1:

reader1 = csv.reader(csvFile1)

for l in reader1:

serial = serial + 1

serial = (serial // 2)

csvFile1.close()

else:

with open("StudentDetails\StudentDetails.csv", 'a+') as csvFile1:

writer = csv.writer(csvFile1)

writer.writerow(columns)

serial = 1

csvFile1.close()

Id = (txt.get())

name = (txt2.get())

if ((name.isalpha()) or (' ' in name)):

cam = cv2.VideoCapture(0)

harcascadePath = "haarcascade\_frontalface\_default.xml"

detector = cv2.CascadeClassifier(harcascadePath)

sampleNum = 0

while (True):

ret, img = cam.read()

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

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

for (x, y, w, h) in faces:

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

# incrementing sample number

sampleNum = sampleNum + 1

# saving the captured face in the dataset folder TrainingImage

cv2.imwrite("TrainingImage\ " + name + "." + str(serial) + "." + Id + '.' + str(sampleNum) + ".jpg",

gray[y:y + h, x:x + w])

# display the frame

cv2.imshow('Taking Images', img)

# wait for 100 miliseconds

if cv2.waitKey(100) & 0xFF == ord('q'):

break

# break if the sample number is morethan 100

elif sampleNum > 100:

break

cam.release()

cv2.destroyAllWindows()

res = "Images Taken for ID : " + Id

row = [serial, '', Id, '', name]

with open('StudentDetails\StudentDetails.csv', 'a+') as csvFile:

writer = csv.writer(csvFile)

writer.writerow(row)

csvFile.close()

message1.configure(text=res)

else:

if (name.isalpha() == False):

res = "Enter Correct name"

message.configure(text=res)

########################################################################################

def TrainImages():

check\_haarcascadefile()

assure\_path\_exists("TrainingImageLabel/")

recognizer = cv2.face\_LBPHFaceRecognizer.create()

harcascadePath = "haarcascade\_frontalface\_default.xml"

detector = cv2.CascadeClassifier(harcascadePath)

faces, ID = getImagesAndLabels("TrainingImage")

try:

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

except:

mess.\_show(title='No Registrations', message='Please Register someone first!!!')

return

recognizer.save("TrainingImageLabel\Trainner.yml")

res = "Profile Saved Successfully"

message1.configure(text=res)

message.configure(text='Total Registrations till now : ' + str(ID[0]))

############################################################################################3

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)]

# create empth face list

faces = []

# create empty ID list

Ids = []

# now looping through all the image paths and loading the Ids and the images

for imagePath in imagePaths:

# loading the image and converting it to gray scale

pilImage = Image.open(imagePath).convert('L')

# Now we are converting the PIL image into numpy array

imageNp = np.array(pilImage, 'uint8')

# getting the Id from the image

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

# extract the face from the training image sample

faces.append(imageNp)

Ids.append(ID)

return faces, Ids

###########################################################################################

def TrackImages():

check\_haarcascadefile()

assure\_path\_exists("Attendance/")

assure\_path\_exists("StudentDetails/")

for k in tv.get\_children():

tv.delete(k)

msg = ''

i = 0

j = 0

recognizer = cv2.face.LBPHFaceRecognizer\_create() # cv2.createLBPHFaceRecognizer()

exists3 = os.path.isfile("TrainingImageLabel\Trainner.yml")

if exists3:

recognizer.read("TrainingImageLabel\Trainner.yml")

else:

mess.\_show(title='Data Missing', message='Please click on Save Profile to reset data!!')

return

harcascadePath = "haarcascade\_frontalface\_default.xml"

faceCascade = cv2.CascadeClassifier(harcascadePath);

cam = cv2.VideoCapture(0)

font = cv2.FONT\_HERSHEY\_SIMPLEX

col\_names = ['Id', '', 'Name', '', 'Date', '', 'Time']

exists1 = os.path.isfile("StudentDetails\StudentDetails.csv")

if exists1:

df = pd.read\_csv("StudentDetails\StudentDetails.csv")

else:

mess.\_show(title='Details Missing', message='Students details are missing, please check!')

cam.release()

cv2.destroyAllWindows()

window.destroy()

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)

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

if (conf < 50):

ts = time.time()

date = datetime.datetime.fromtimestamp(ts).strftime('%d-%m-%Y')

timeStamp = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')

aa = df.loc[df['SERIAL NO.'] == serial]['NAME'].values

ID = df.loc[df['SERIAL NO.'] == serial]['ID'].values

ID = str(ID)

ID = ID[1:-1]

bb = str(aa)

bb = bb[2:-2]

attendance = [str(ID), '', bb, '', str(date), '', str(timeStamp)]

else:

Id = 'Unknown'

bb = str(Id)

cv2.putText(im, str(bb), (x, y + h), font, 1, (255, 255, 255), 2)

cv2.imshow('Taking Attendance', im)

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

break

ts = time.time()

date = datetime.datetime.fromtimestamp(ts).strftime('%d-%m-%Y')

exists = os.path.isfile("Attendance\Attendance\_" + date + ".csv")

if exists:

with open("Attendance\Attendance\_" + date + ".csv", 'a+') as csvFile1:

writer = csv.writer(csvFile1)

writer.writerow(attendance)

csvFile1.close()

else:

with open("Attendance\Attendance\_" + date + ".csv", 'a+') as csvFile1:

writer = csv.writer(csvFile1)

writer.writerow(col\_names)

writer.writerow(attendance)

csvFile1.close()

with open("Attendance\Attendance\_" + date + ".csv", 'r') as csvFile1:

reader1 = csv.reader(csvFile1)

for lines in reader1:

i = i + 1

if (i > 1):

if (i % 2 != 0):

iidd = str(lines[0]) + ' '

tv.insert('', 0, text=iidd, values=(str(lines[2]), str(lines[4]), str(lines[6])))

csvFile1.close()

cam.release()

cv2.destroyAllWindows()

######################################## USED STUFFS ############################################

global key

key = ''

ts = time.time()

date = datetime.datetime.fromtimestamp(ts).strftime('%d-%m-%Y')

day,month,year=date.split("-")

mont={'01':'January',

'02':'February',

'03':'March',

'04':'April',

'05':'May',

'06':'June',

'07':'July',

'08':'August',

'09':'September',

'10':'October',

'11':'November',

'12':'December'

}

######################################## GUI FRONT-END ###########################################

window = tk.Tk()

window.geometry("1280x720")

window.resizable(True,False)

window.title("Attendance Management System (BIJAYA BEHERA)")

window.configure(background='#262523')

frame1 = tk.Frame(window, bg="#00aeff")

frame1.place(relx=0.11, rely=0.17, relwidth=0.39, relheight=0.80)

frame2 = tk.Frame(window, bg="#00aeff")

frame2.place(relx=0.51, rely=0.17, relwidth=0.38, relheight=0.80)

message3 = tk.Label(window, text="Face Recognition Based Attendance Management System" ,fg="white",bg="#262523" ,width=55 ,height=1,font=('times', 29, ' bold '))

message3.place(x=10, y=10)

frame3 = tk.Frame(window, bg="#c4c6ce")

frame3.place(relx=0.52, rely=0.09, relwidth=0.09, relheight=0.07)

frame4 = tk.Frame(window, bg="#c4c6ce")

frame4.place(relx=0.36, rely=0.09, relwidth=0.16, relheight=0.07)

datef = tk.Label(frame4, text = day+"-"+mont[month]+"-"+year+" | ", fg="orange",bg="#262523" ,width=55 ,height=1,font=('times', 22, ' bold '))

datef.pack(fill='both',expand=1)

clock = tk.Label(frame3,fg="orange",bg="#262523" ,width=55 ,height=1,font=('times', 22, ' bold '))

clock.pack(fill='both',expand=1)

tick()

head2 = tk.Label(frame2, text=" For New Registrations ", fg="black",bg="#03F2FD" ,font=('times', 17, ' bold ') )

head2.grid(row=0,column=0)

head1 = tk.Label(frame1, text=" For Already Registered ", fg="black",bg="#03F2FD" ,font=('times', 17, ' bold ') )

head1.place(x=0,y=0)

lbl = tk.Label(frame2, text="Enter ID",width=20 ,height=1 ,fg="black" ,bg="#00aeff" ,font=('times', 17, ' bold ') )

lbl.place(x=80, y=55)

txt = tk.Entry(frame2,width=32 ,fg="black",font=('times', 15, ' bold '))

txt.place(x=30, y=88)

lbl2 = tk.Label(frame2, text="Enter Name",width=20 ,fg="black" ,bg="#00aeff" ,font=('times', 17, ' bold '))

lbl2.place(x=80, y=140)

txt2 = tk.Entry(frame2,width=32 ,fg="black",font=('times', 15, ' bold ') )

txt2.place(x=30, y=173)

message1 = tk.Label(frame2, text="1)Take Images >>> 2)Save Profile" ,bg="pink" ,fg="black" ,width=39 ,height=1, activebackground = "gold" ,font=('times', 15, ' bold '))

message1.place(x=7, y=230)

message = tk.Label(frame2, text="" ,bg="#00aeff" ,fg="black" ,width=39,height=1, activebackground = "yellow" ,font=('times', 16, ' bold '))

message.place(x=7, y=450)

lbl3 = tk.Label(frame1, text="Attendance",width=20 ,fg="black" ,bg="#00aeff" ,height=1 ,font=('times', 17, ' bold '))

lbl3.place(x=100, y=115)

res=0

exists = os.path.isfile("StudentDetails\StudentDetails.csv")

if exists:

with open("StudentDetails\StudentDetails.csv", 'r') as csvFile1:

reader1 = csv.reader(csvFile1)

for l in reader1:

res = res + 1

res = (res // 2) - 1

csvFile1.close()

else:

res = 0

message.configure(text='Total Registrations till now : '+str(res))

##################### MENUBAR #################################

menubar = tk.Menu(window,relief='ridge')

filemenu = tk.Menu(menubar,tearoff=0)

filemenu.add\_command(label='Change Password', command = change\_pass)

filemenu.add\_command(label='Contact Us', command = contact)

filemenu.add\_command(label='Exit',command = window.destroy)

menubar.add\_cascade(label='Help',font=('times', 29, ' bold '),menu=filemenu)

################## TREEVIEW ATTENDANCE TABLE ####################

tv= ttk.Treeview(frame1,height =13,columns = ('name','date','time'))

tv.column('#0',width=82)

tv.column('name',width=130)

tv.column('date',width=133)

tv.column('time',width=133)

tv.grid(row=2,column=0,padx=(0,0),pady=(150,0),columnspan=4)

tv.heading('#0',text ='ID')

tv.heading('name',text ='NAME')

tv.heading('date',text ='DATE')

tv.heading('time',text ='TIME')

###################### SCROLLBAR ################################

scroll=ttk.Scrollbar(frame1,orient='vertical',command=tv.yview)

scroll.grid(row=3,column=4,padx=(0,100),pady=(150,0),sticky='ns')

tv.configure(yscrollcommand=scroll.set)

########### BUTTONS ###########

clearButton = tk.Button(frame2, text="Clear", command=clear ,fg="black" ,bg="#ea2a2a" ,width=11 ,activebackground = "white" ,font=('times', 11, ' bold '))

clearButton.place(x=335, y=86)

clearButton2 = tk.Button(frame2, text="Clear", command=clear2 ,fg="black" ,bg="#ea2a2a" ,width=11 , activebackground = "white" ,font=('times', 11, ' bold '))

clearButton2.place(x=335, y=172)

takeImg = tk.Button(frame2, text="Take Images", command=TakeImages ,fg="black" ,bg="#F5FD03" ,width=34 ,height=1, activebackground = "white" ,font=('times', 15, ' bold '))

takeImg.place(x=30, y=300)

trainImg = tk.Button(frame2, text="Save Profile", command=psw ,fg="black" ,bg="#F5FD03" ,width=34 ,height=1, activebackground = "white" ,font=('times', 15, ' bold '))

trainImg.place(x=30, y=380)

trackImg = tk.Button(frame1, text="Take Attendance", command=TrackImages ,fg="black" ,bg="#1AFD03" ,width=35 ,height=1, activebackground = "white" ,font=('times', 15, ' bold '))

trackImg.place(x=30,y=50)

quitWindow = tk.Button(frame1, text="Quit", command=window.destroy ,fg="brown" ,bg="red" ,width=35 ,height=1, activebackground = "white" ,font=('times', 15, ' bold '))

quitWindow.place(x=30, y=450)

##################### END ##########

window.configure(menu=menubar)

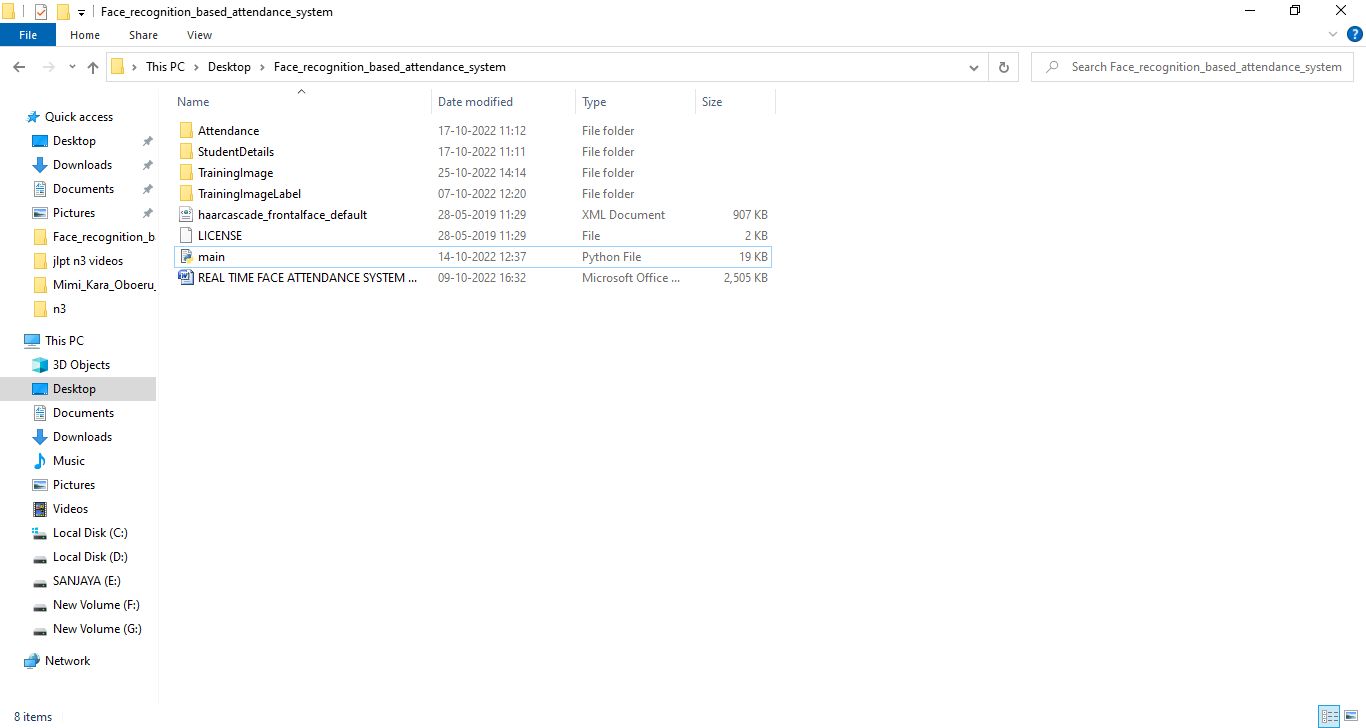
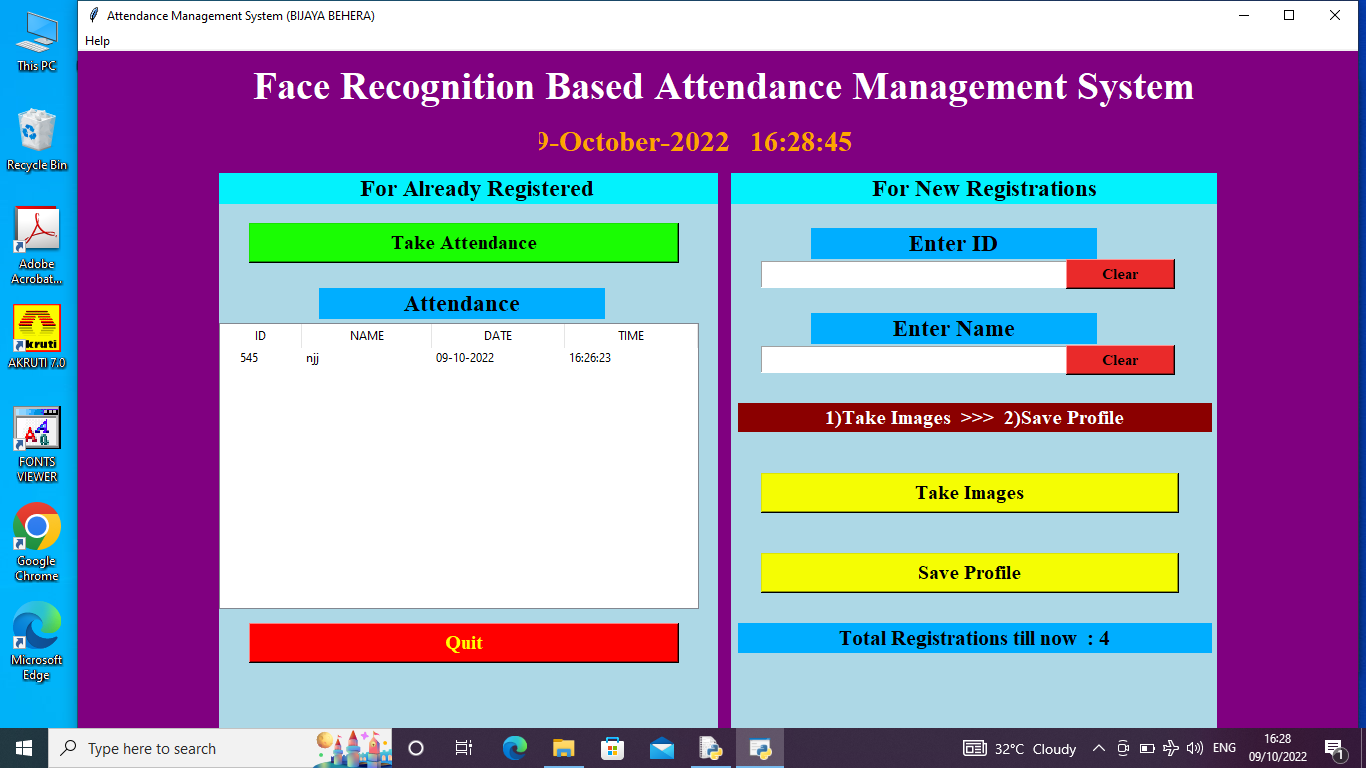
window.mainloop()

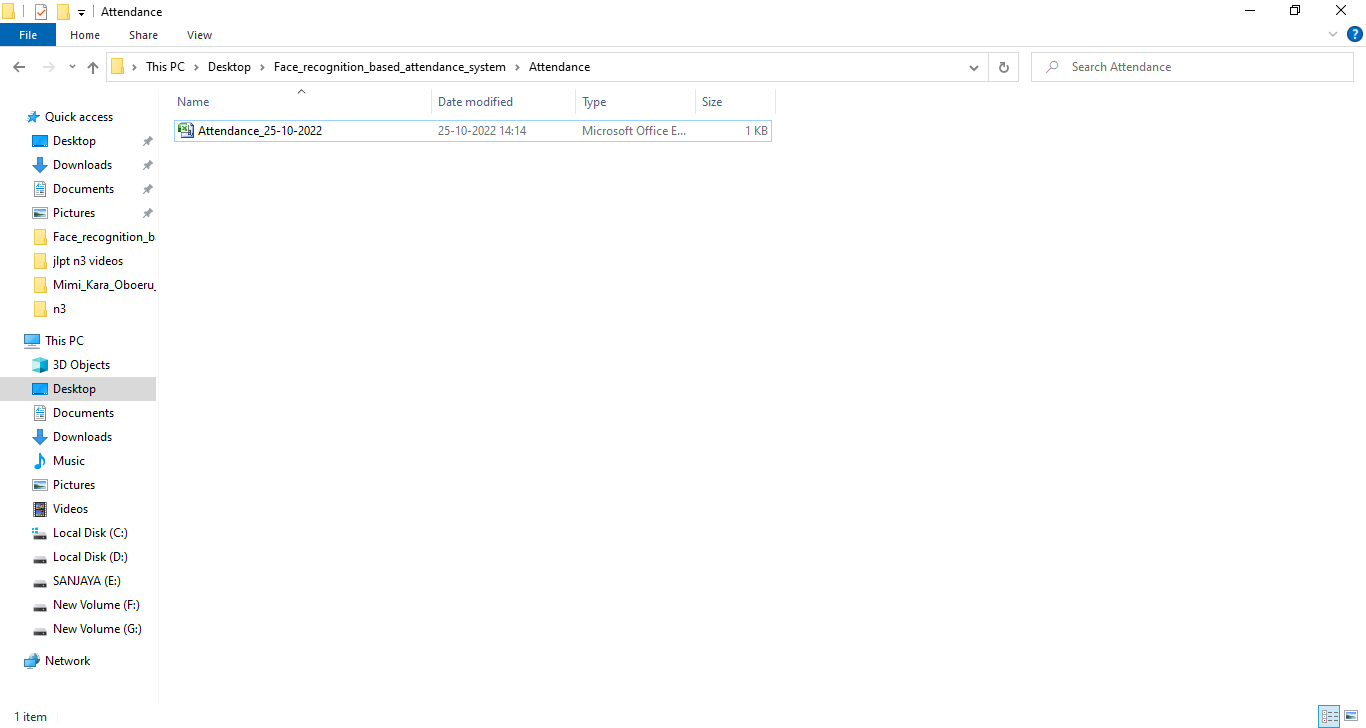
**RESULT AND DISCUSSION**

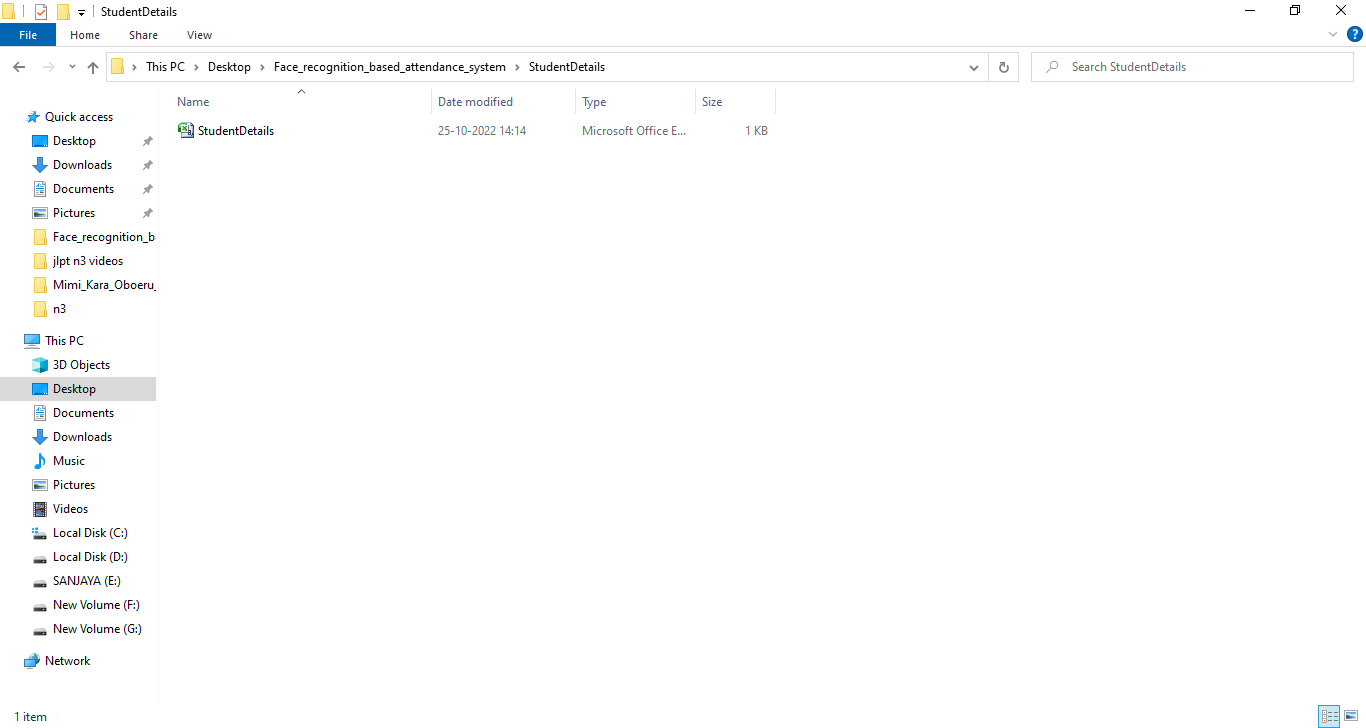
#### Result

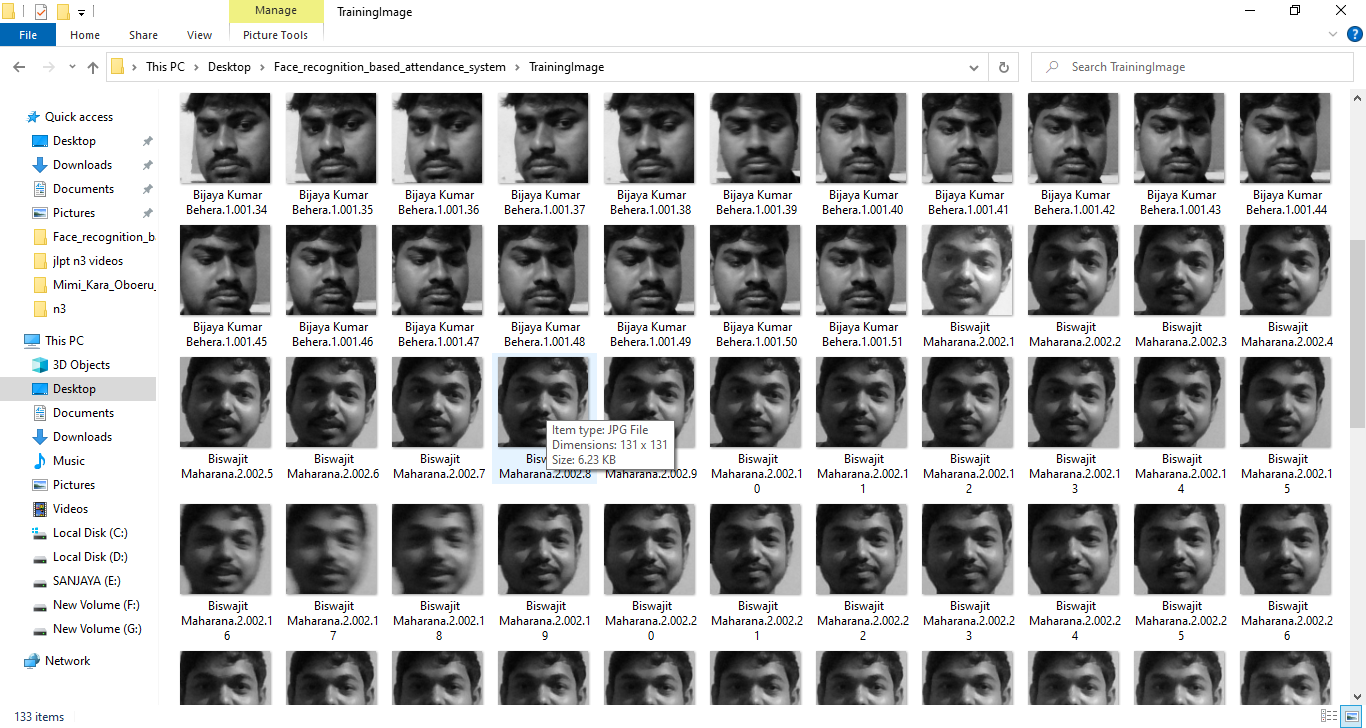
In this proposed approach, face recognition student attendance system with user- friendly interface is designed by using MATLAB GUI(Graphic User Interface). A few buttons are designed in the interface, each provides specific function, for example, start button is to initialize the camera and to perform face recognition automatically according to the face detected, register button allows enrolment or registrations of students and update button is to train the latest images that have been registered in the database. Lastly, browse button and recognize button is to browse facial images from selected database and recognized the selected image to test the functionality of the system respectively. In this part, enhanced LBP with radius two is chosen and used as proposed algorithm. The analysis of choosing the radius size will be further explained in the discussion.

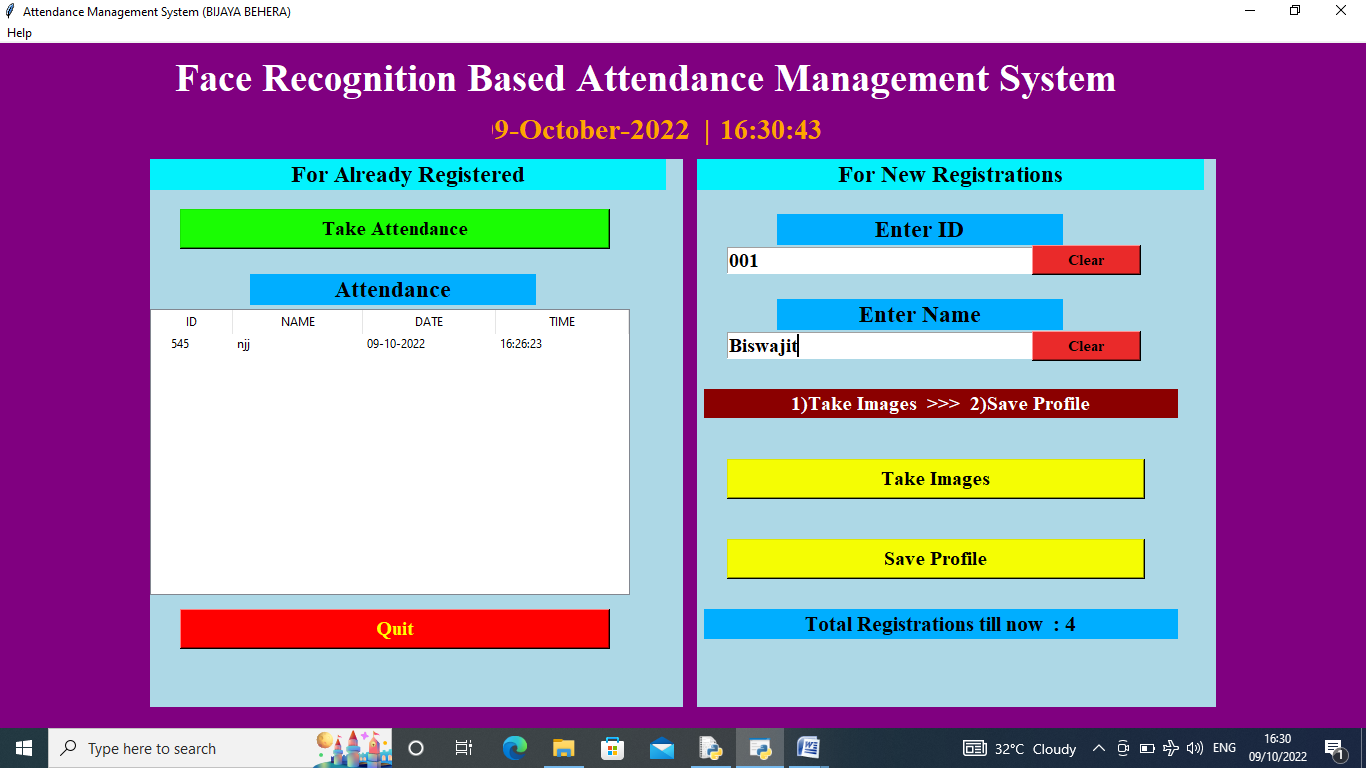
**Sample Input / Output Screen Shots**

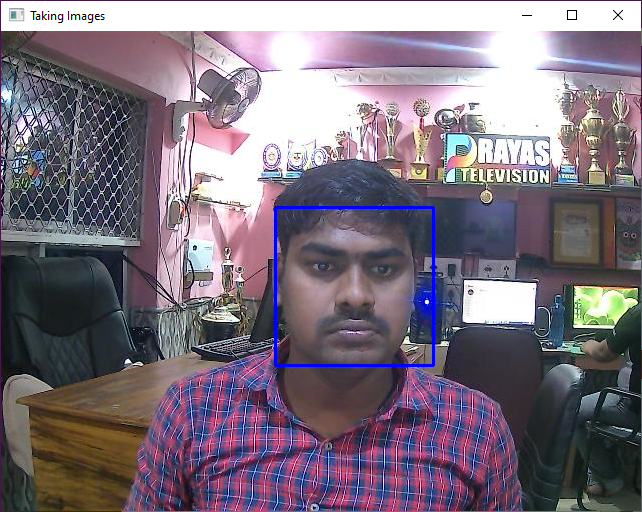
****

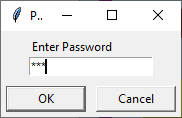


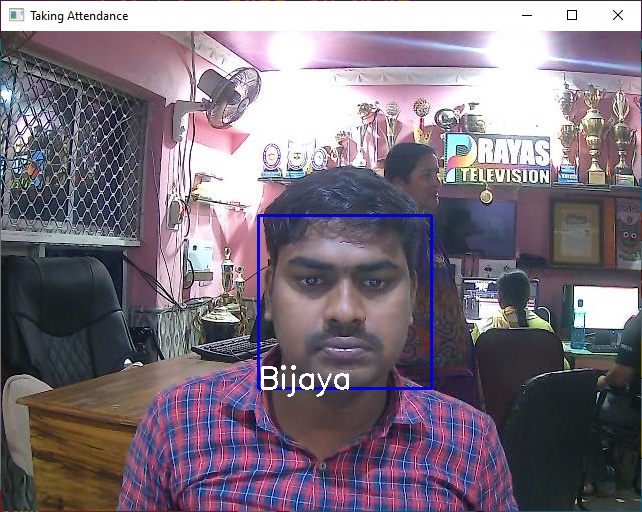


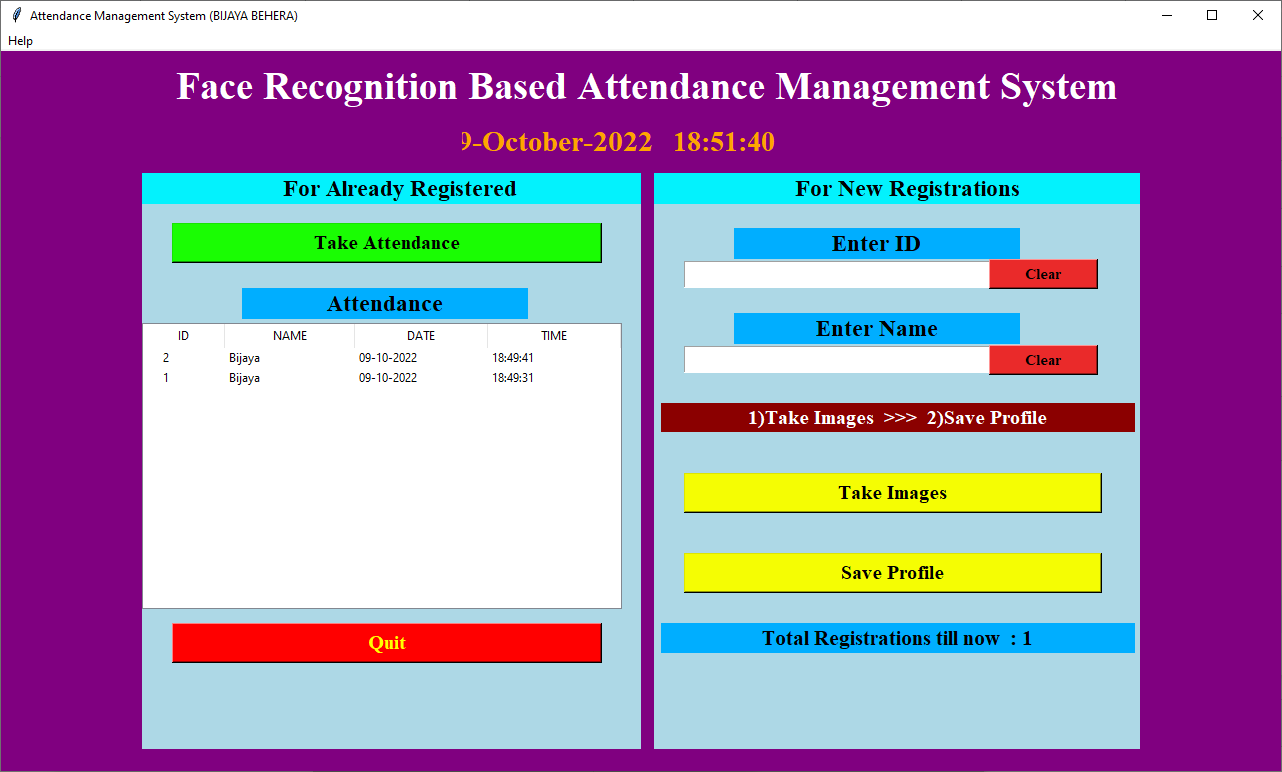












### System Testing

### Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. Testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements.

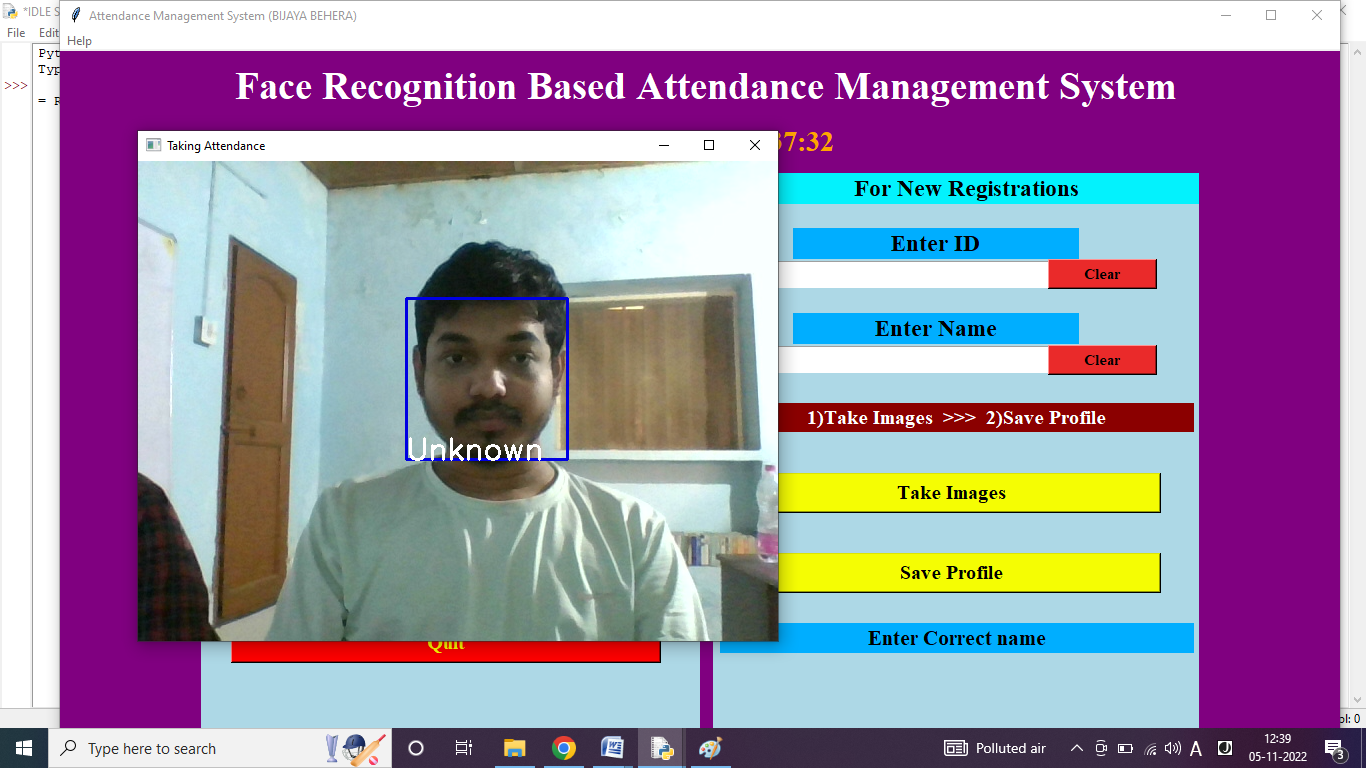
**Unit Testing**

Unit testing involves the testing of each unit or an individual component of the software application. It is the first level of functional testing. The aim behind unit testing is to validate unit components with its performance.The purpose of unit testing is to test the correctness of isolated code. A unit component is an individual function or code of the application. White box testing approach used for unit testing and usually done by the developers.

**Why Unit Testing?**

In a testing level hierarchy, unit testing is the first level of testing done before integration and other remaining levels of the testing. It uses modules for the testing process which reduces the dependency of waiting for Unit testing frameworks, stubs, drivers and mock objects are used for assistance in unit testing.

**Sample Test Output Screen**



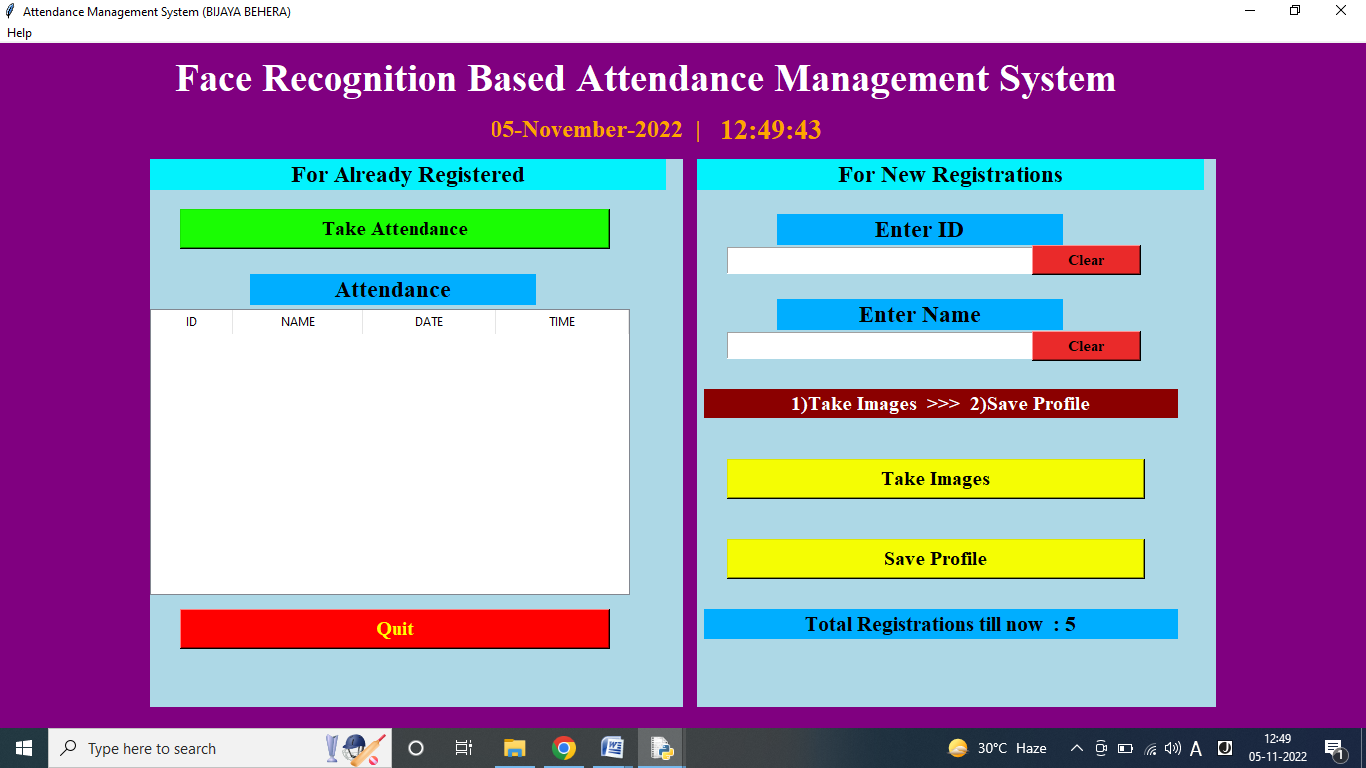
If we try to take attendance of non-existing user (without any registration),it shows “unknown” as result.

### Integration Testing

### Integration testing is the second level of the software testing process comes after unit testing. In this testing, units or individual components of the software are tested in a group. The focus of the integration testing level is to expose defects at the time of interaction between integrated components or units.

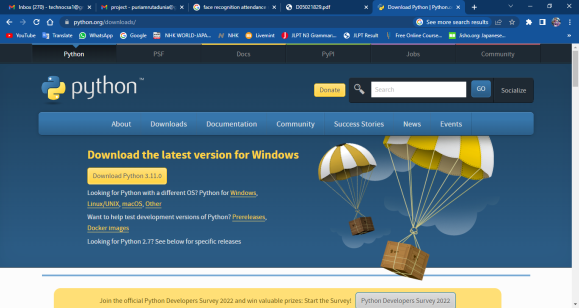
### Unit testing uses modules for testing purpose, and these modules are combined and tested in integration testing. The Software is developed with a number of software modules that are coded by different coders or programmers. The goal of integration testing is to check the correctness of communication among all the modules.

**Sample Test Output Screen**

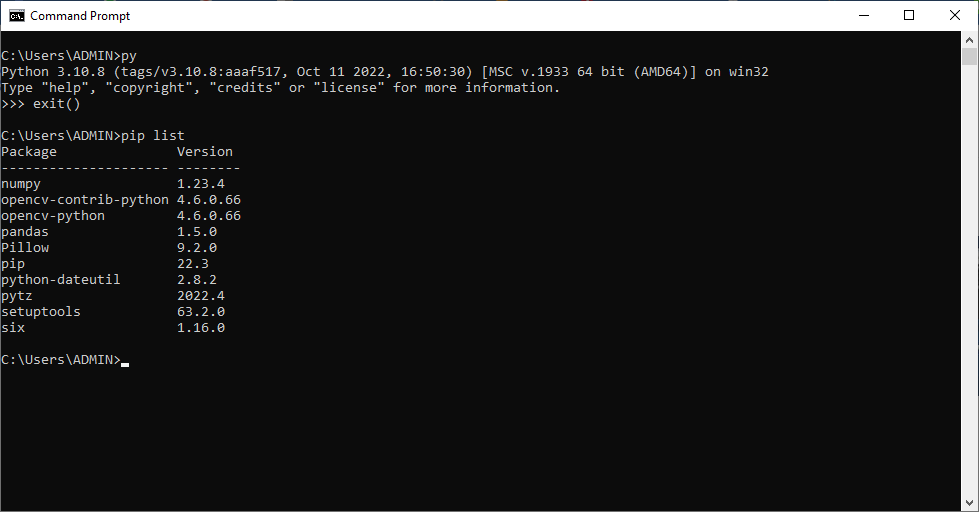


#### User Manual

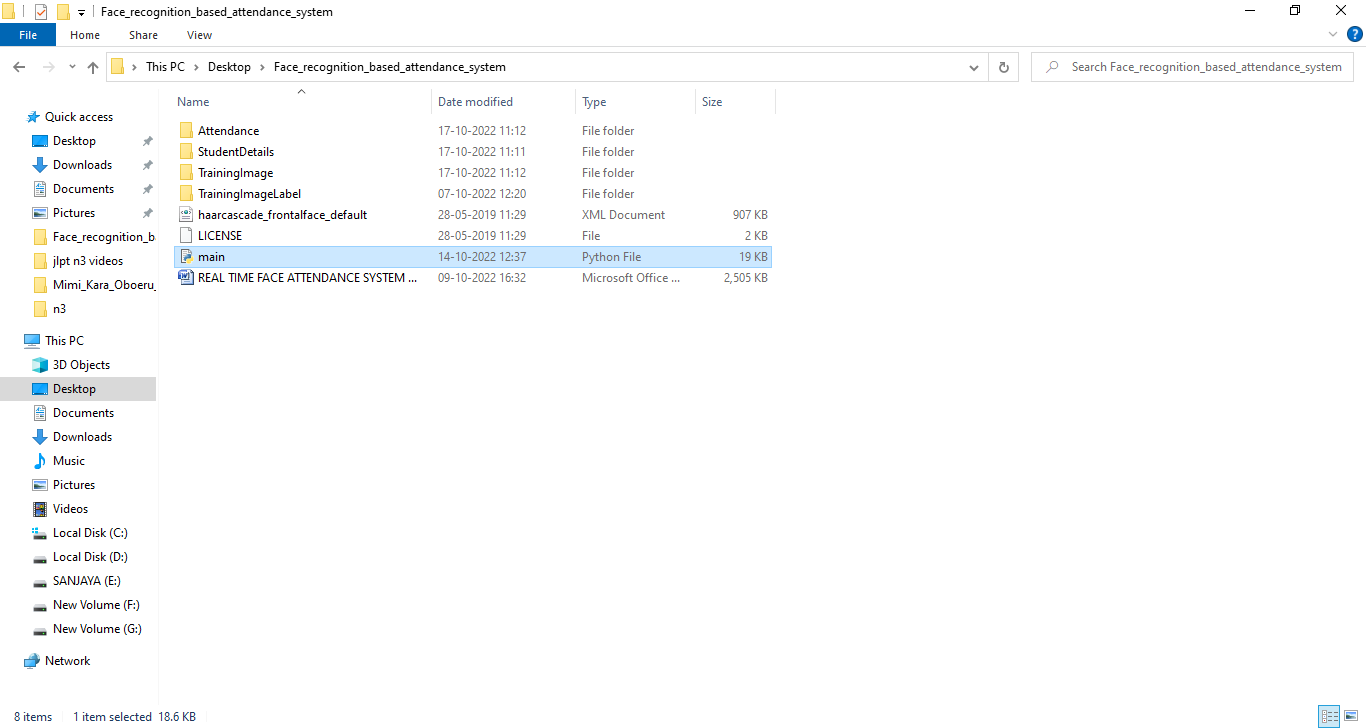
Step 1- Install Python



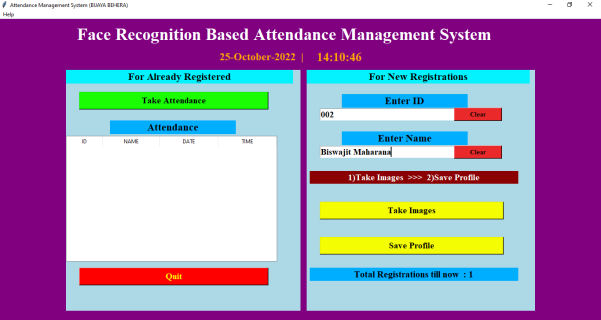
Step 2- Install Pip & All Modules.

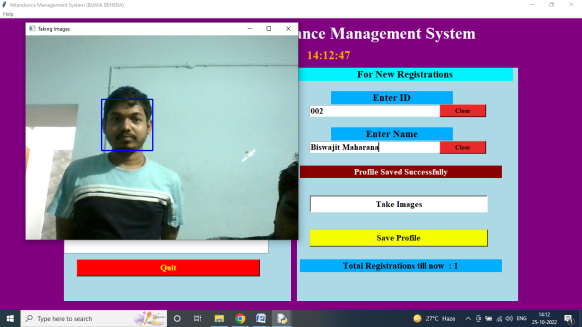


Step 3- Open & Run **main.py**



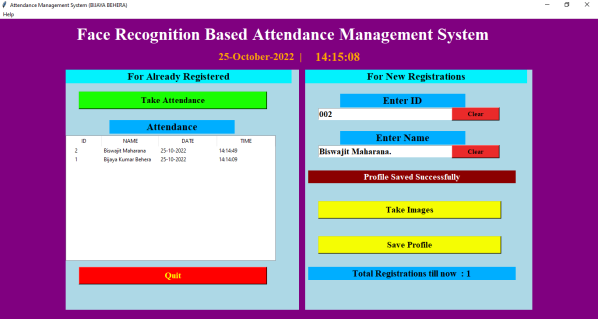
Step 4- Enter ID & Name, then click Take images button. After take images press ‘q’ to close the camera. After all completed then click save profile .





Step 5- Click take Attendance button. A picture of the class is taken, and the RECOGNIZER python file is initiated. Attendance is taken by cropping the faces in the picture and comparing with the faces in the database. - If a face is matched, the responding name with present status is marked in a EXCEL file with the current date and time.





#### Scope & Limitation

In this proposed approach, there are a few limitations. First, the input image has to be frontal and a upright single facial image. Second,the accuracy might drop under extreme illumination problem. Third, false recognition might occur if the captured image is blurred. Besides, LBP is textural based descriptor which extracts local features. Hence, test image and train image have to be the same quality which is captured by using the same device in order to have high accuracy. Lastly, if an individual wears make up in the image for face recognition, the important features will be covered.

In fact, a better camera with a better lighting source able to reduce the illumination problem and also able to avoid the captured of blurred images. In this proposed approach, laptop built in camera is a default device. However the lighting source of the laptop camera is very dim, this cause the system to be unstable. For future work, a better camera and a better lighting source can be used in order to obtain better result. This can reduce the dependency on the brightness of environment, especially the places to capture test and train images. Furthermore, a face recognition system which has more faces other than a single facial image can be designed. This can increase the efficiency of the system. The test image and train image in this approach is highly related to each other and highly dependent on the image captured device. The capture device has to be the same for this approach to perform better. Thus, other algorithms can be used instead of LBP, for example A.I (artificial intelligence) algorithm which can be implemented to perform the face recognition. CNN (Convolution Neural Network) which is a hot topic recently, is a machine deep learning algorithm which is able to perform recognition with less dependency on a particular train image given a large database. However, CNN requires an extremely large database to increase its accuracy or having relatively small class size to have high performance.

In pre-processing stage, an algorithm, for instance affine transform can be applied to align the facial image based on coordinates in the middle of the eyes. This might help, especially in PCA algorithm, which it maps test image to train image to perform face recognition.

**BIBLIOGRAPHY**

Robinson-Riegler, G., & Robinson-Riegler, B. (2008). *Cognitive psychology: applying the science of the mind*. Boston, Pearson/Allyn and Bacon.

Margaret Rouse. (2012). *What is facial recognition? - Definition from WhatIs.com*. [online] Available at:

<http://whatis.techtarget.com/definition/facial-recognition>[Accessed 25 Mar. 2018].

Solon, O. (2017). *Facial recognition database used by FBI is out of control, House committee hears*. [online] the Guardian. Available at: https:/[/www.theguardi](http://www.theguardian.com/technology/2017/mar/27/us-facial-recognition-)a[n.com/technology/2017/mar/27/us-facial-recognition-](http://www.theguardian.com/technology/2017/mar/27/us-facial-recognition-) database-fbi-drivers-licenses-passports [Accessed 25 Mar. 2018].

[Robert Silk](http://www.travelweekly.com/Robert-Silk). (2017). *Biometrics: Facial recognition tech coming to an airport near you: Travel Weekly*. [online] Available at:

<http://www.travelweekly.com/Travel-News/Airline-News/Biometrics-Facial-> recognition-tech-coming-airport-near-you [Accessed 25 Mar. 2018].

[Sidney Fussell.](https://kinja.com/sidneyfussell#_ga%3D2.33904934.647560575.1521956578-2116359232.1521956578) (2018). *NEWS Facebook's New Face Recognition Features: What We Do (and Don't) Know*.

deAgonia, M. (2017). *Apple's Face ID [The iPhone X's facial recognition tech explained]*. [online] Computerworld. Available at: https:/[/www.comput](http://www.computerworld.com/article/3235140/apple-ios/apples-face-id-the-)e[rworld.com/article/3235140/apple-ios/apples-face-id-the-](http://www.computerworld.com/article/3235140/apple-ios/apples-face-id-the-) iphone-xs-facial-recognition-tech-explained.html [Accessed 25 Mar. 2018].

Ashley DuVal. (2012).*Face Recognition Software -History of Forensic Psychology*. [online] Available at:

<http://forensicpsych.umwblogs.org/research/criminal-justice/face-recognition-> software/ [Accessed 12 Apr. 2018].

[Jesse Davis West](https://www.facefirst.com/blog/brief-history-of-face-recognition-software/). (2017). *History of Face Recognition - Facial recognition software*. [online] Available at: https://[www.facefirst.com/blog/brief-history-of-face-](http://www.facefirst.com/blog/brief-history-of-face-) recognition-software/ [Accessed 25 Mar. 2018].

Reichert, C. (2017). *Intel demos 5G facial-recognition payment technology | ZDNet*. [online] ZDNet. Available at:

<http://www.zdnet.com/article/intel-demos-5g-facial-recognition-payment-> technology/ [Accessed 25 Mar. 2018]