Facendance

(Attendance Software Using Face Recognition and Detection)

A MINOR PROJECT

Submitted in Partial Fulfillment of the Requirement for the Award of the Degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING

SUBMITTED TO

Rajiv Gandhi Proudyogiki Vishwavidyalaya , Bhopal (M.P.)

SUBMITTED BY

Prince Dwivedi (0176CS191122)

Lakshmi Narain College of Technology Excellence, Bhopal (M.P.), India

UNDER THE SUPERVISION OF

Prof . Sitaram Patel

Asst. Professor, Department of Computer Science & Engineering

Lakshmi Narain College of Technology Excellence, Bhopal (M.P.), India

Department of Computer Science & Engineering

Lakshmi Narain College of Technology Excellence,

Bhopal (M.P.)

Session: June 2022

CANDIDATE'S DECLARATION

I **Prince Dwivedi,** student of **Bachelor of Technology, Computer Science & Engineering, Lakshmi Narain College of Technology Excellence, Bhopal** hereby declare that the work presented in the project entitled "**Facendance"** is outcome of my own bona-fide work, which is correct to the best of my knowledge and this work has been carried out taking care of Engineering Ethics. The work presented does not infringe any previous work and has not been submitted to any University for the award of any degree.

The project entitled **"Facendance"** being submitted by "**Prince Dwivedi" (0176CS191122)** has been examined by us and is hereby approved for the award of degree **"Bachelor of Technology in Computer Science & Engineering"**, for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein, but approve the project only for the purpose for which it has been submitted.

Prince Dwivedi
0176CS191122

CERTIFICATE

This is to certify that the project entitled **"Facendance"** is the bona-fide work carried out independently by **Prince Dwivedi,** student of **Bachelor of Technology**, Department of Computer Science & Engineering from Rajiv Gandhi Prodyogiki Vishwavidyalaya, Bhopal.

In the partial fulfillment of the requirement for the award of the degree of Bachelor ofTechnology, and this project has not formed previously the basis for the award of any degree associate ship, fellowship or any other similar title according to our knowledge.

FORWARDED BY:

Dr. ANIL KUMAR SAXENA
Principal
LNCTE, Bhopal

ACKNOWLEDGEMENT

At the outset, we would like to link to thank the Almighty who made all the things possible. Writing this project report would not have been possible without the support of several people whom we need to wholeheartedly thank. A special thank to **Dr. Megha Kamble** Professor and Head, Department of Computer Science and Engineering, LNCTE, Bhopal who helped us in completing the project and exchanged her interesting ideas thoughts and made this project easy accurate .

We are equally grateful to **Dr. Anil kumar Saxena**, principal, LNCTE for providing all the necessary resources to carry out this project work. We would like to thank all staff members of CSE, L.N.C.T.E. and friends, for their moral and psychological support.

Prince Dwivedi
0176CS191122

ABSTRACT

STATEMENT OF PROBLEM:

Attendance of students in a large classroom is hard to be handled by the traditional system, as it is time-consuming and has a high probability of error during the process of inputting data into the computer. Our project proposed automated attendance marking system using face recognition technique.

RESULT:

The system deployed Haar cascade classifier to find the positive and negative of the face and LBPH (Local binary pattern histogram) algorithm for face recognition by using python programming and OpenCV library. Here we use the tkinter GUI interface for user interface purpose. Firstly, our app asks to fill the details of the student and take image of the particular student. It takes 60 images as sample and store them in folder Training Image. After completion it notify that images saved. After taking image sample we have to click Train Image button. Now it takes few seconds to train machine for the images that are taken by clicking Take Image button and creates a Trainner.yml file and store in TrainingImageLabel folder. Now all initial setups are done. By clicking Track Image button camera of running machine is opened again. If face is recognized by system then Id and Name of person is shown on Image. Press Q (or q) for quit this window. The attendance of the student was updated to the Excel sheet after student's face has been recognized.

INDEX

LIST OF FIGURES

LIST OF TABLES

CHAPTER-1

INTRODUCTION

1.Introduction

1.1 Introduction

In Face Recognition based attendance management systems, the flow process starts by being able to detect and recognize frontal faces from an input dataset present in a database. In today's world, it has been proven that students engage better during lectures only when there is effective classroom control. The need for high level student engagement is very important. An analogy can be made with that of pilots as described by Mundschenk et al (2011 p101)" Pilots need to keep in touch with an air traffic controller, but it would be annoying and unhelpful if they called in every 5 minutes". In the same way students need to be continuously engaged during lectures and one of the ways is to recognize and address them by their names. Therefore, a system like this will improve classroom control. In my own view based on experience, during my time as a teacher, I realized calling a student by his/her name gives me more control of the classroom and this draws the attention of the other students in the classroom to engage during lectures.

Face detection and recognition is not new in our society we live in. The capacity of the human mind to recognize particular individuals is remarkable. It is amazing how the human mind can still persist in identification of certain individuals even through the passage of time, despite slight changes in appearance.

Anthony (2014 p1) reports that, due to the remarkable ability of the human mind to generate near positive identification of images and facial recognition of individuals, this has drawn considerable attention for researchers to invest time in finding algorithms that will replicate effective face recognition on electronic systems for use by humans.

Wang et al (2015 p318) states that" the process of searching a face is called face detection. Face detection is to search for faces with different expressions, sizes and angles in images in possession of complicated light and background and feeds back parameters of face".

Face recognition processes images and identifies one or more faces in an image by analyzing patterns and them. This process uses algorithms which extracts features and compare them to a database to find a match. comparing Furthermore, in one of most recent research, Nobel (2017, p. 1), suggest that DNA techniques could transform facial recognition technology, by the use of video analysis software which can be improved thanks to a completely advance in research in DNA analysis. By so doing, camera-based surveillance systems software to analyze DNA sequences, by treating a video as a scene that evolves the same way DNA does, to detect and recognize human face.

Problem Definition:

This project is being carried out due to the concerns that have been highlighted on the methods which lectures use to take attendance during lectures. The technology aims in imparting a tremendous knowledge oriented technical innovation these days. Machine learning is one among the interesting domain that enables the machine to train itself by providing an appropriate output during testing by applying different learning algorithms. Nowadays Attendance is considered as an important for both the students as well as the teacher of an educational organization. With the advancements of the machine learning technology the machine automatically detects the attendance performance of the students and maintains a record of those collected data. The motivations for organizing this special section

were to better address the challenges of face recognition in real – world scenarios, to promote systematic research and evaluation of promising methods and systems, to provide a snapshot of where we in this domain, and to stimulate discussion about future directions.

In general, the attendance system of the student can be maintained in two different forms namely,

- Manual Attendance system (MAS)
- Automated Attendance System (AAS)

Manual Student Attendance Management system is a process is a process where a teacher concerned with the particular subject need to call the students name and mark the attendance manually. Manual attendance may be considered as a time–consuming process or sometimes it happens for the teacher to miss someone or students may answer multiple times on the absence of their friends. So, the problem arises when we think about the traditional process of taking attendance in the classroom. To solve all these issues, we go with Automatic Attendance System.

Automated Attendance System (AAS) is a process to automatically estimate the presence or the absence of the student in the classroom by using face recognition technology. It is also possible to recognize whether the student is sleeping or awake during the lecture and it can also be implemented in the exam sessions to ensure the presence of the student. The presence of the students can be determined by capturing their faces on to a high–definition monitor video streaming service, so it becoming highly reliable for the machine to understand the presence of all the students in the classroom. The two common Human Face Recognition techniques are,

- Feature-based approach
- Brightness-based approach.

The Feature-based approach also known as local face recognition system, used in pointing the key features of the face like eyes, ears, nose, mouth, etc, Whereas the brightness-based approach also termed as the global face recognition system used in recognizing all parts of the image.

1.2 Importance of Face Recognition System:

Importance of Face Recognition System as a Security Solution Face is considered as the most important part of human body. Research shows that even face can speak and it has different words for different emotions. It plays a very crucial role for interacting with people in the society. Its covey's people's identity, so it can be used as a key for security solutions in many organizations. Nowadays, face recognition system is getting increasing trend across the world for providing extremely safe and reliable security technology. It is gaining significant importance and attention by thousands of corporate and government organizations only because of its high level of security and reliability. Moreover, this system is providing vast benefits when compared to other biometric security solutions like palm print and finger print.

The system captures biometric measurements of a person from a specific distance without interacting with the person. With its crime deterrent purpose, this system can help many organizations to identify a person who is having any kind of criminal record or any legal issues .thus this technology is becoming very important for numerous residential buildings and corporate organizations .This technique is based on the ability to recognize a human face and then compare the different features of the face with previously recorded one.

Along with it, it is developed by having user friendly features and operations that includes different nodal points of the face. There are approximately 80-90 unique nodal points of a face. From these nodal points, it measures some major points like distance between the eyes, length of the jaw line, shape of the cheek bones, depth of the eyes etc. These points are measured by creating a code called the face print which represents the identification of the face in the computer database.

Face recognition is the most popular system that is widely used by millions of corporate offices for maintaining their human resources. Without any errors and faults, the system recognizes the employees and also records their entry as well as exit time in its computer database.

CHAPTER-2

LITERATURE SURVEY

2.LITERACTURE SURVEY

In this chapter, a brief overview of studies made on face recognition will be introduced alongside some popular face detection and recognition algorithms. This will give a general idea of the history of systems and approaches that have been used so far.

Overview of Face Recognition:

Most face recognition systems rely on face recognition algorithms to complete the following functional task as suggested by Shang-Hung Lin. (2000, p.2). The figure below shows a simplified diagram from the framework for face recognition from the study suggested by Shang-Hung Lin.

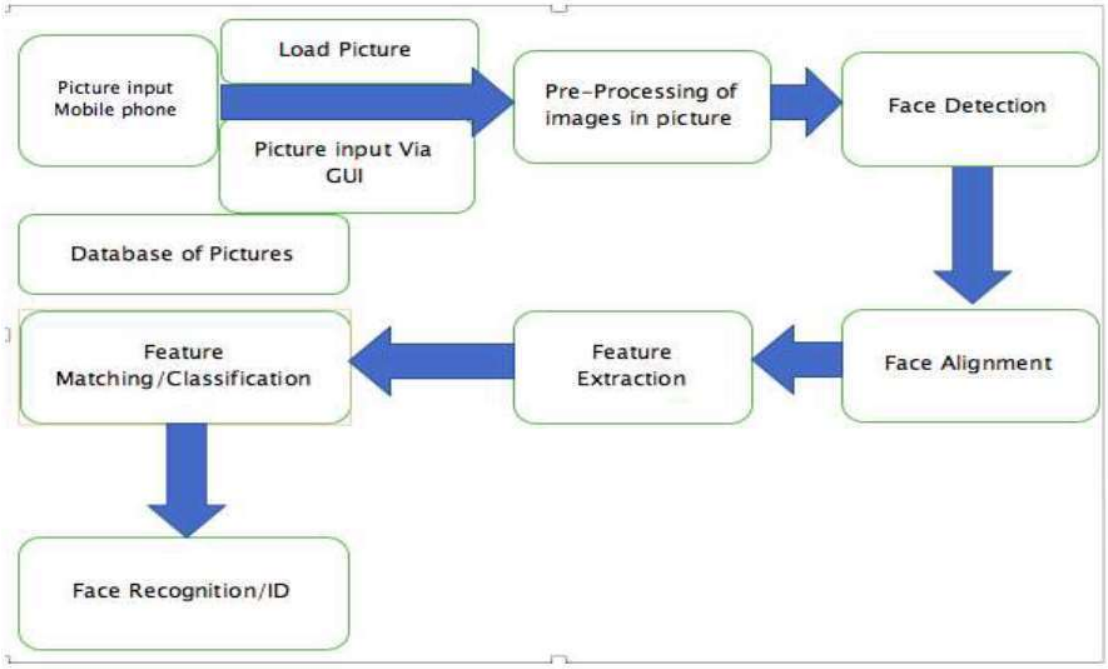


Figure: Face Detection and Recognition Flow Diagram.

From the figure, above, **Face Detection** or face detector will detect any given face in the given image or input dataset. **Face localization**, will detect where the faces are located in the given image/input dataset by use of bounding boxes. **Face Alignment** is when the system will find a face and align landmarks such as nose, eyes, chin, and mouth for feature extraction. Feature extraction, extracts key features such as the eyes, nose, and mouth to undergo tracking. Feature matching and classification. Matches a face based on a trained data set of pictures from a database of about minimum number of pictures. Face recognition, gives a positive or negative output of a recognized face based on feature matching and classification from a referenced facial image. Face detection is the process of locating a face in a digital image by any special computer software build for this purpose. Feraud et al (2000 p.77) discuss face detection as to detect a face in an image means to find its position in the image plane and its size or scale. As figure shows, the detection of a face in a digital image is a prerequisite to any further process in face recognition or any face processing software. The

idea of the technology namely Student Attendance System has been implemented with a machine learning approach. This system automatically detects the student's performance and maintains the student's record like attendance. Therefore, the attendance of the student can be made available by recognizing the face. On recognizing, the attendance details.

Automated Attendance System using Face Recognition proposes that the system is based on face detection and detection and recognition algorithms, which is used to automatically detects the students face he/she enters the class and the system is capable to marks the attendance by recognizing him. The effectiveness of the pictures is also being discussed to enable the faster recognition of the image.

The original LBP (Local binary patterns) operator was introduced by the paper of Timo Ojala et al (2002).In paper by Md. Abdur Rahim et al. (2013), they proposed LBP to extract both texture details and contour to represent facial images divides each facial image into smaller regions and histogram of each region is extracted. The histograms of every region are concatenated into a single feature vector. This feature vector is the representation of the facial image and Chi square statistic is used to measure similarities between facial images. The smallest window size of each region is 3 by 3. It is computed by thresholding each pixel in a window where middle pixel is the threshold value. The neighborhood large than threshold value is assigned to 1 whereas the neighborhood lower than threshold value is assigned to 0. Then the resulting binary pixels will form a byte value representing center pixel.

Fig:LBPOperator(Md. Abdur Rahim et.al,2013)

LBP has a few advantages which make it popular to be implemented. It has high tolerance against the monotonic illumination changes and it is able to deal with variety of facial expressions, image rotation and aging of persons. These overwhelming characteristics cause LBP to be prevalent in real-time applications.

FACE RECOGNITION:

In the early 90s numerous algorithms were developed for face recognition and increase in the need for face detection. Systems were designed to deal with video streaming. The past few years has proven to have developed more research and systems to deal with such challenges. Dodd, (2017 p.1) reported that in the recent Notting Hill carnival, some arrest resulted due to trial of facial recognition systems. Hence the reason why there is still on-going research on this system. In contrast, the 2011 London riots had just one arrest contributed by facial recognition software out of the 4962 that took place. With the most recent technology of facial **Face recognition** can be defined as the method of identifying an individual based on biometrics by way of comparing a digital captured image with the stored record of the person in question. Recognition and detection techniques, commercial products have emerged on the markets. Despite the commercial success a few issues are still to be explored.

Jafri and Arabnia (2009) in their study discuss Face Recognition in two primary tasks. **Verification**; a one-to-one matching of an unknown face alongside a claim of identity, to ascertain the face of the individual claiming to be the one on the image. **Identification** which is also a one-to-one matching, given an input image of a face for an individual (unknown), to determine their identity by comparing the image against a database of images with known individuals. However, Face Recognition can also be used in numerous applications such as Security, Surveillance, General Identity Verification (electoral registration, national ID cards, passports, driving licenses, student IDs), Criminal Justice systems, Image Database Investigations, Smart Card, Multi-media Environments, Video Indexing and Witness face reconstruction. Face Recognition in most common form is its frontal view which is not unique or rigid as numerous factors cause its appearance to vary. Variations in facial appearance has been categorized in two groups of intrinsic factors (physical nature of the face which is independently of the observer) and extrinsic factors (illumination, pose, scale and imaging parameters such as resolution, noise, focus, imaging) as discussed by Gong et al. (200) and supported by Jafri and Arabnia (2009 p.42).

Lenc and Král (2014 pp.759-769) classify face recognition into various approaches; **Correlation Method**, compares two images by computing the correlation between them, with the images handled as one-dimensional vectors of intensity values. The images are normalized to have zero mean and unit variance with the nearest neighbor classifier used in the image directly. With these considerations stated, the light source intensity and characteristics of the camera are suppressed. The limitations of this method are; Large amount of memory storage needed, the corresponding points in the image space may not be tightly clustered and it is computationally expensive.

Neural Networks; performs based on neural networks with the images sampled into a set of vectors. The vectors created from the labeled images are used as a training set for Self-Organized Map. In other study carried out by Dhanaseely et al. (2012), discuss the neural Network Classifiers as an Artificial Neural Network (ANN) that comprises of artificial neurons that uses a computational model to process information. They further conducted an experiment based on their proposed system, to measure the performance recognition rate of two of the neural networks, the Feed Forward Neural Network and Cascade Neural Network.

Hidden Markov Models; associated with the states of the HMM are the subdivided regions of the face (eyes, nose, mouth etc.). the images in this method are sampled with a rectangular window of the same width as the image and shifted downward with a specific block overlap. This is done thanks to the representation of boundaries between regions which are represented by probabilistic transition between the states of the HMM.

Local Binary Patterns; first used in texture as texture descriptor, the operator uses the value of the central pixel to threshold a local image region. The pixels are labelled either as 0 or 1 depending on whether the value is lower or greater than the threshold. Linna et al. (2015) in their study, proposed a system (Online Face Recognition System) that is based on LBP and Facial Landmarks, which uses nearest neighbor classifier in LBP histogram matching. They experimented the system on the videos of Honda/UCSD video database. They used both Offline and Online testing for different distance thresholds and achieved recognition rates of 62.2%,64.0% and 98.6% respectively for the Offline test. The recognition rate was calculated based on a confusion matrix that is shown in Figure 2.10 below, obtained as a screenshot from this paper. The online test performed at a recognition rate of 95.9%. The high achieved recognition rates as per their experiment is based on longer search strategy. The detected face

tracked, is used to find the nearest neighbor match and the number of frames from the start of the face tracking are used in the database. This shows that the number of frames decreases as the database gets larger and hence increase in search time. This is because more time is needed to find the nearest match for a single frame. However, as more time is needed to find the nearest match, although recognition rate may be high, it is still not robust enough to compete with other methods.

2.1 Applications of face Recognition:

1.Fraud Detection for Passports and Visas

According to reports, experts using the automatic face-recognition software in Australian Passport office are 20 percent more efficient as compared to average people detecting fraud. An effective tool to detect fraud, face recognition is increasingly being used to identify documents such as driving license and immigration visas.

2.ATM and Banks

China started using the first face recognition technology in the ATMs. The new cash machine developed using this technology ensured increased security of the card user and worked by mapping facial data, matching it against the database. As a part of the biometric authentication, it used the data from facial features and iris recognition.

3.Identification of Criminals

An increasingly popular tool among the law enforcement agencies, face recognition technology has significantly contributed in the domain of investigation and crime detection. Several countries including the USA is building the facial recognition database, to improve the quality of the investigation. According to a report released by the Center for Privacy and Technology at Georgetown University law school, the law enforcement database in the U.S includes 117 million individuals.

4.Prevent Fraud Voters

Face detection was used in 2000 presidential election in Mexico to prevent duplicate voting. Several individuals had attempted to vote multiple times using different names. The duplicate votes were prevented to a great extent, thanks to the face recognition technology.

5.Track Attendance

Face recognition system is being used by some organization to track the attendance of the employees. The system collects and records the facial fine points of the employees in the database. Once the process is done, the employee only needs to look at the camera and the attendance is automatically marked in the face recognition attendance system.

6.Keep Track ofthe Members

Several churches across the world are using face recognition technology to keep a track of the church-goers. The places include India, Indonesia, and Portugal. The CCTV footages of the churchgoers are matched against a database of high-resolution images which a church has to compile on its own.

7. Threats and Concerns

According to several civil right groups and privacy campaigners, the face identification takes away the right of the people to remain anonymous. According to these people, the government agencies and private companies are unwilling to accept the fact that they should necessarily seek permission first before using data like face recognition as these will leave people identifiable wherever they go. Not just this, all the scattered bits of data left behind everywhere due to our digital presence can be put together to find out every minute detail of us as a person, which may include our taste, preference, friends, habits and movement. In Europe and Canada, the organizations will have to seek permission before using face recognition technology.

Chapter-3

REQUIREMENTS AND TECHNICAL DESCRIPTION

3.1 System Requirements

3.1.1 System Requirements

System requirements are the configuration that a system must have in order for a hardware or software application to run smoothly and efficiently. Failure to meet these requirements can result in installation problems or performance problems. The former may prevent a device or application from getting installed, whereas the latter may cause a product to malfunction or perform below expectation or even to hang or crash.

We can specify the system requirements in terms of hardware and software system requirements. Hardware system requirements often specify the operating system version, processor type, memory size, available disk space and additional peripherals, if any, needed. Software system requirements, in addition to the aforementioned requirements, may also specify additional software dependencies (e.g., libraries, driver version, framework version).

HARDWARE REQUIREMENTS:

Processor	intel core i5 8th Gen
Graphics Processing Unit (GPU)	NVIDIA GEFFORCE
Random Access Memory (RAM)	4GB
Hard Disk	500GB

Table: 3.a Hardware Used

SOFTWARE DEPENDENCIES:

Requirement	Version
Open CV	4.3.0
tkinter GUI	8.6
NumPy	1.18.1
pandas	1.0.3
PIL	1.1.7

Table: 3.b Dependencies Used

SOFTWARE REQUIREMENTS:

Operating system	Windows 10
Language	Python 3.6 version
Editor	ANACONDA NAVIGATOR Spyder
Design Tool	Star UML

Table: 3.c Software Used

3.2 Technological description

The technology selected for implementing Face Recognition based attendance management systems are open CV, python and tkinter GUI interface.

3.2.1 OpenCV:

- OpenCV (Open Source Computer Vision) is a library for computer vision that includes numerous highly optimized algorithms that are used in Computer vision tasks.
- OpenCV supports a wide variety of programming languages such as C++, Python, Java etc. Support for multiple platforms including Windows, Linux, and MacOS.
- OpenCV Python is nothing but a wrapper class for the original C++ library to be used with Python. Using this, all of the OpenCV array structures gets converted to/from NumPy arrays.
- This makes it easier to integrate it with other libraries which use NumPy. For example, libraries such as SciPy and Matplotlib.

Basic operations of OpenCV:

- Access pixel values and modify them
- Access image properties
- Setting Region of Image (ROI)
- Splitting and Merging images
- Change an image colour

Face detection using OpenCV:

This seems complex at first but it is very easy. Let me walk you through the entire process and you will feel the same.

Step 1: Considering our prerequisites, we will require an image, to begin with. Later we need to create a cascade classifier which will eventually give us the features of the face.

Step 2: This step involves making use of OpenCV which will read the image and the features file. So, at this point, there are NumPy arrays at the primary data points.

All we need to do is to search for the row and column values of the face NumPy ndarray. This is the array with the face rectangle coordinates.

Step 3: This final step involves displaying the image with the rectangular face box.

Check out the following image,

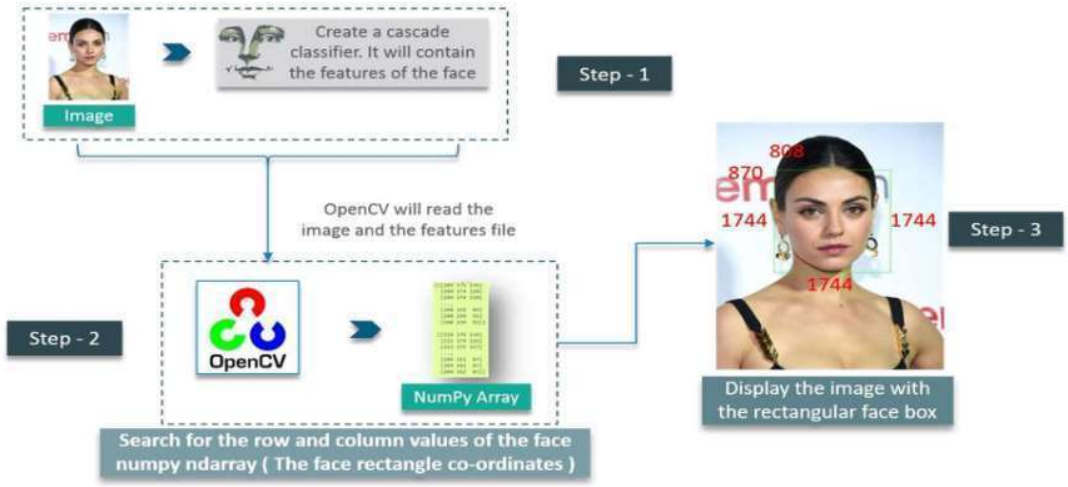


Fig 3.2.1: Face detection using OpenCV

3.2.2 Python:

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics.

Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together.

The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Python is a multi-paradigm programming language because of we can use in Web development, AI, ML, Data analyst, Data science and networking

Python is a popular selection for use as a scripting language for many software developments processes. Similar to many other interpretative languages, Python provides more flexibility than compiled languages, and it can be economically utilized to integrate disparate systems together.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

Python Syntax compared to other programming languages

- Python was designed for readability, and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
- Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

Python Features

Python's features include –

Easy-to-learn − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the languagequickly.

Easy-to-read − Python code is more clearly defined and visible to the eyes.

Easy-to-maintain − Python's source code is fairly easy-to-maintain.

A broad standard library − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.

Interactive Mode − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.

Programming − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system ofUnix.

Scalable − Python provides a better structure and support for large programs than shell scripting.

Portable − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.

Extendable − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.

Databases − Python provides interfaces to all major commercial databases.

Python Indentation:

Indentation refers to the spaces at the beginning of a code line.

Where in other programming languages the indentation in code is for readability only, the indentation in Python is very important.

Python uses indentation to indicate a block of code.

Ex: if 5>2:
    print (“5 is greater than 2”)

Python will give you an error if you skip the indentation

Import module

Import in python is similar to #include header file in C/C++. Python modules can get access to code from another module by importing the file/function using import. The import statement is the most common way of invoking the import machinery, but it is not the only way.

import module name

When import is used, it searches for the module initially in the local scope by calling import () function. The value returned by the function are then reflected in the output of the initial code.

```
Ex: import math
    print (pi)
```

Installing and using Python on Windows 10 is very simple. The installation procedure involves just three steps:

1. Download thebinaries
2. Run the Executableinstaller
3. Add Python to PATH environmentalvariables

To install Python, you need to download the official Python executable installer. Next, you need to run this installer and complete the installation steps. Finally, you can configure the PATH variable to use python from the commandline.

3.2.3 tkinter GUI interface:

Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is the most commonly used method. **Tkinter** is actually an inbuilt Python module used to create simple GUI apps. It is the most commonly used module for GUI apps in the Python.

Importing tkinter is same as importing any other module in the Python code.

Ex: Import tkinter

Creating a GUI using tkinter is an easy task.

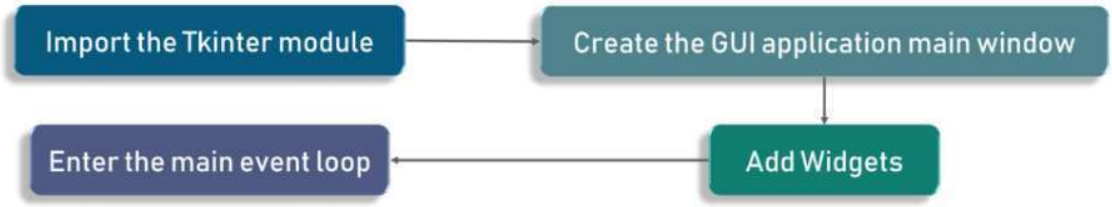


Fig 3.2.3: steps for creating tkinter GUI

To start out with, we first import the Tkinter model. Followed by that, we create the main window. It is in this window that we are performing operations and displaying visuals and everything basically. Later, we add the widgets and lastly, we enter the main event loop.

There are 2 main key words used which the user needs to remember while creating interface with this GUI. They are Widgets and main event loop.

- **Widgets:** Widgets are something like elements in the HTML. You will find different types of widgets to the different types of elements in the tkinterlike buttons, labels, radio buttons, checkbox, entry button…etc.
- **Main event loop:**There is a method known by the name mainloop () is used when your application is ready to run. mainloop () is an infinite loop used to run the application, wait for an event to occur and process the event as long as the window is not closed.

Tk (screenName=None, baseName=None, className='Tk', useTk=1): To create a main window, tkinter offers a method 'Tk (screenName=None, baseName=None, className='Tk', useTk=1)'. To change the name of the window, you can change the className to the desired one. The basic code used to create the main window of the application is:

Ex: m=tkinter.Tk()

#where m is the name of the main window object

Sample example:

```
import tkinter
m = tkinter. Tk()
'''
widgets are added here
'''
m.mainloop()
```

CHAPTER-4

DESIGN

4.DESIGN

4.1 Objective

The overall design objective is to provide an efficient, modular design that will reduce the system's complexity, facilitate change and result in an easy implementation. This will be accomplished by designing strongly cohesion system with minimal coupling. In addition, this document will provide interface design models that are consistent, user friendly and will provide straight forward transition through the various system functions.

The purpose of the design phase is to develop a clear understanding of what the developer wants people to gain from his/her project. As the developer work on the project, the test for every design decision should be "Does this feature fulfill the ultimate purpose of the project?". The design document will verify that the current design meets all of the explicit requirements contained in the system model as well as the implicit requirements desired.

4.2 UML Diagrams

Unified Modelling Language (UML) is a general-purpose modelling language. The main aim of UML is to define a standard way to visualize the way a system has been designed. It is quite similar to blueprints used in other fields of engineering.

UML is not a programming language; it is rather a visual language. We use UML diagrams to portray the behaviour and structure of a system. UML helps software engineers, businessmen and system architects with modelling, design and analysis. The Object Management Group (OMG) adopted Unified Modelling Language as a standard in 1997. It's been managed by OMG ever since. International Organization for Standardization (ISO) published UML as an approved standard in 2005. UML has been revised over the years and is reviewed periodically.

Goals of UML:

The primary goals in the design of the UML were:

1. Provide users with a ready-to-use, expressive visual modelling language so they can develop and exchange meaningful models.
2. Provide extensibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development processes.
4. Provide a formal basis for understanding the modelling language.
5. Support higher-level development concepts such as collaborations, frameworks, patterns and components.
6. Integrate best practice

4.2.1 The conceptual model of UML

A conceptual model can be defined as a model which is made of concept and their relationships.

A conceptual model is the first step before drawing UML diagrams. It helps to understand the entities in the real world and how they interact with each other.

To understand how the UML works, we need to know the three elements:

1. UML basic buildingblocks
2. Rules to connect the building blocks (Rules for how these building blocks may be put together).
3. Common mechanisms that apply throughout in theUML.

4.2.2 Types of Diagrams

UML diagrams are divided into three different categories such as,

- Structural diagram
- Behavioral diagram
- Interaction diagram

Structural diagrams

Structural diagrams are used to represent a static view of a system. It represents a part of a system that makes up the structure of a system. A structural diagram shows various objects within the system.

Following are the various structural diagrams in UML:

- Class diagram
- Object diagram
- Package diagram
- Component diagram
- Deployment diagram

Behavioral diagrams

Any real-world system can be represented in either a static form or a dynamic form. A system is said to be complete if it is expressed in both the static and dynamic ways. The behavioural diagram represents the functioning of a system.

UML diagrams that deals with the static part of a system are called structural diagrams. UML diagrams that deals with the moving or dynamic parts of the system are called behavioural diagrams.

Following are the various behavioural diagrams in UML:

- Activity diagram
- Use case diagram
- State machine diagram

Interaction diagrams

Interaction diagram is nothing but a subset of behavioural diagrams. It is used to visualize the flow between various use case elements of a system. Interaction diagrams are used to show an interaction between two entities and how data flows within them.

Following are the various interaction diagrams in UML:

- Timing diagram
- Sequence diagram
- Collaboration diagram

4.3 Use case diagram:

Use Case Diagram captures the system's functionality and requirements by using actors and use cases. Use Cases model the services, tasks, function that a system needs to perform. Use cases represent high-level functionalities and how a user will handle the system. Use-cases are the core concepts of Unified Modelling language modeling.

A Use Case consists of use cases, persons, or various things that are invoking the features called as actors and the elements that are responsible for implementing the use cases. Use case diagrams capture the dynamic behavior of a live system. It models how an external entity interacts with the system to make it work. Use case diagrams are responsible for visualizing the external things that interact with the part of the system.

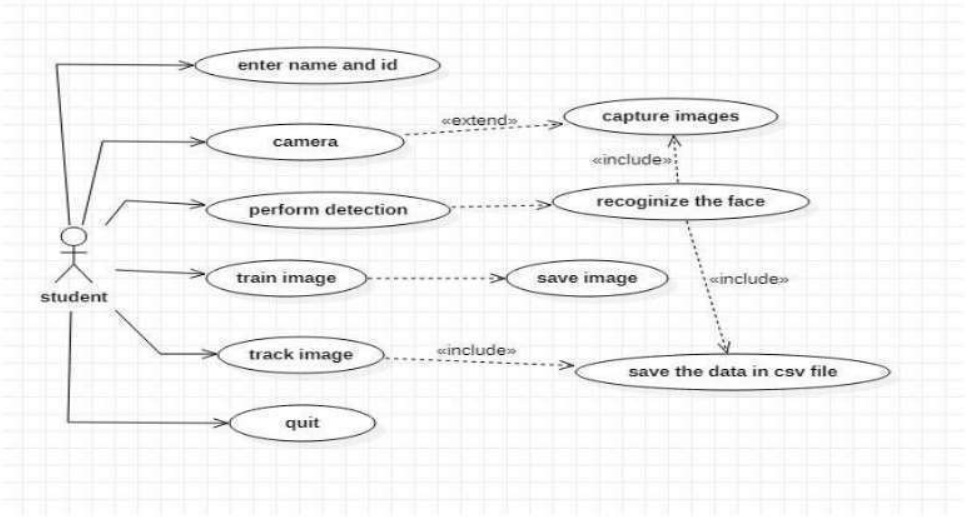


Fig 5.1: Use Case Diagram

4.4 Class diagram:

Class diagram gives an overview of a software system by displaying classes, attributes, operations, and their relationships. This Diagram includes the class name, attributes, and operation in separate designated compartments.

Class Diagram defines the types of objects in the system and the different types of relationships that exist among them. It gives a high-level view of an application. This modelling method can run with almost all Object-Oriented Methods. A class can refer to another class. A class can have its objects or may inherit from other classes.

Class Diagram helps construct the code for the software application development.

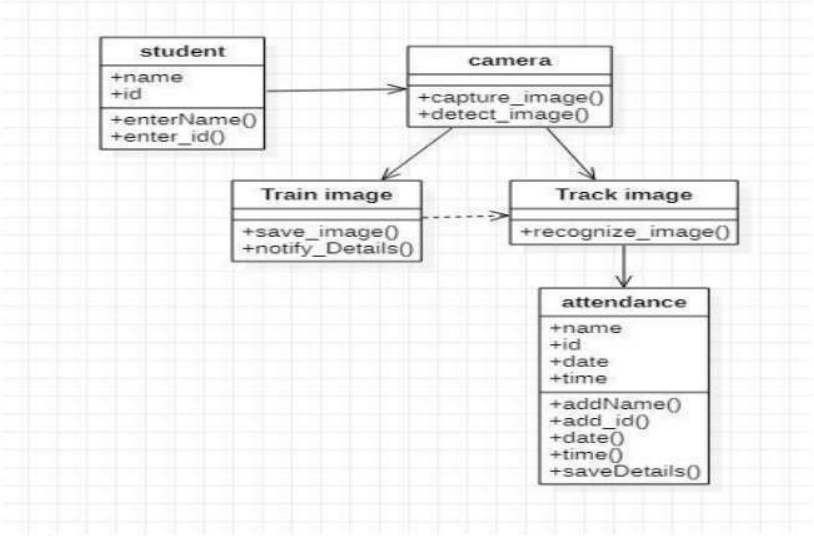


Fig 5.2: Class Diagram

4.5 Sequence Diagram:

UML Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. Sequence Diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent and when.

Sequence Diagrams captures:

- The interaction that takes place in a collaboration that either realizes a use case or an operation (instance diagrams or generic diagrams)
- High-level interactions between user of the system and the system, between the system and other systems, or between subsystems (sometimes known as system sequence diagrams)

Sequence diagrams can be useful references for businesses and other organizations. Try drawing a sequence diagram to:

- ★ Represent the details of a UML use case.
- ★ Model the logic of a sophisticated procedure, function, or operation.
- ★ See how objects and components interact with each other to complete a process.
- ★ Plan and understand the detailed functionality of an existing or future scenario.

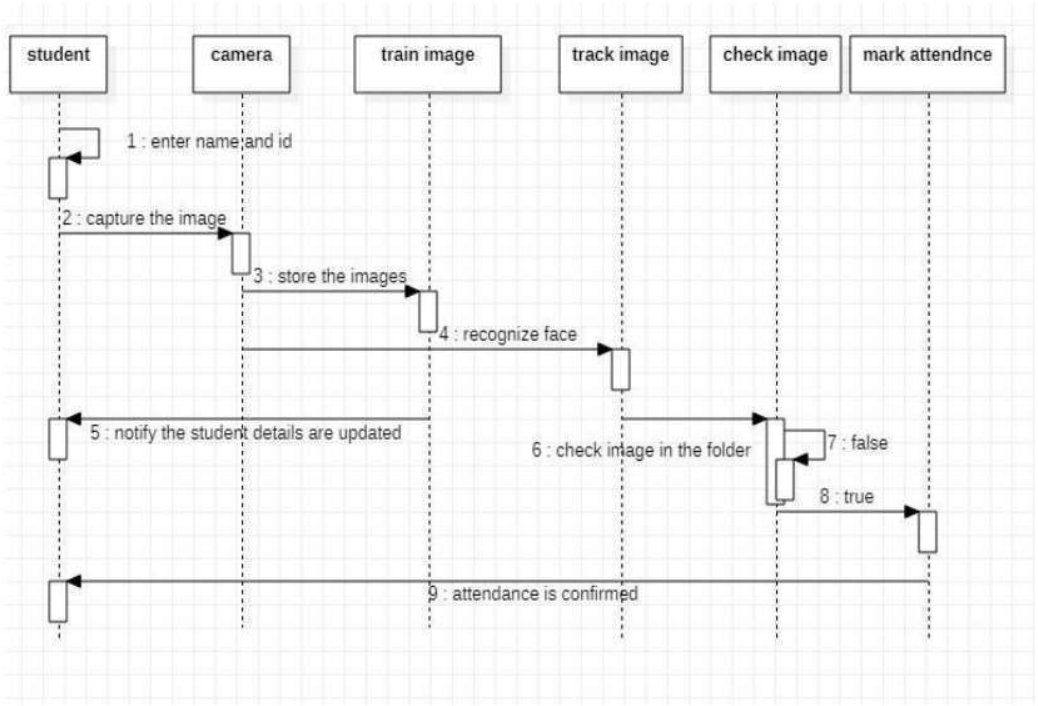


Fig 5.3: Sequence Diagram

4.6 Activity Diagram:

Activity diagram is defined as a UML diagram that focuses on the execution and flow of the behavior of a system instead of implementation. It is also called object-oriented flowchart. Activity diagrams consist of activities that are made up of actions which apply to behavioral modeling technology.

An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed. We can depict both sequential processing and concurrent processing of activities using an activity diagram. They are used in business and process modelling where their primary use is to depict the dynamic aspects of a system. An activity diagram is very similar to a flowchart.

The basic usage of activity diagram is similar to other four UML diagrams. The specific usage is to model the control flow from one activity to another. This control flow does not include messages.

Activity diagram is suitable for modelling the activity flow of the system. An application can have multiple systems. Activity diagram also captures these systems and describes the flow from one system to another. This specific usage is not available in other diagrams. These systems can be database, external queues, or any other system

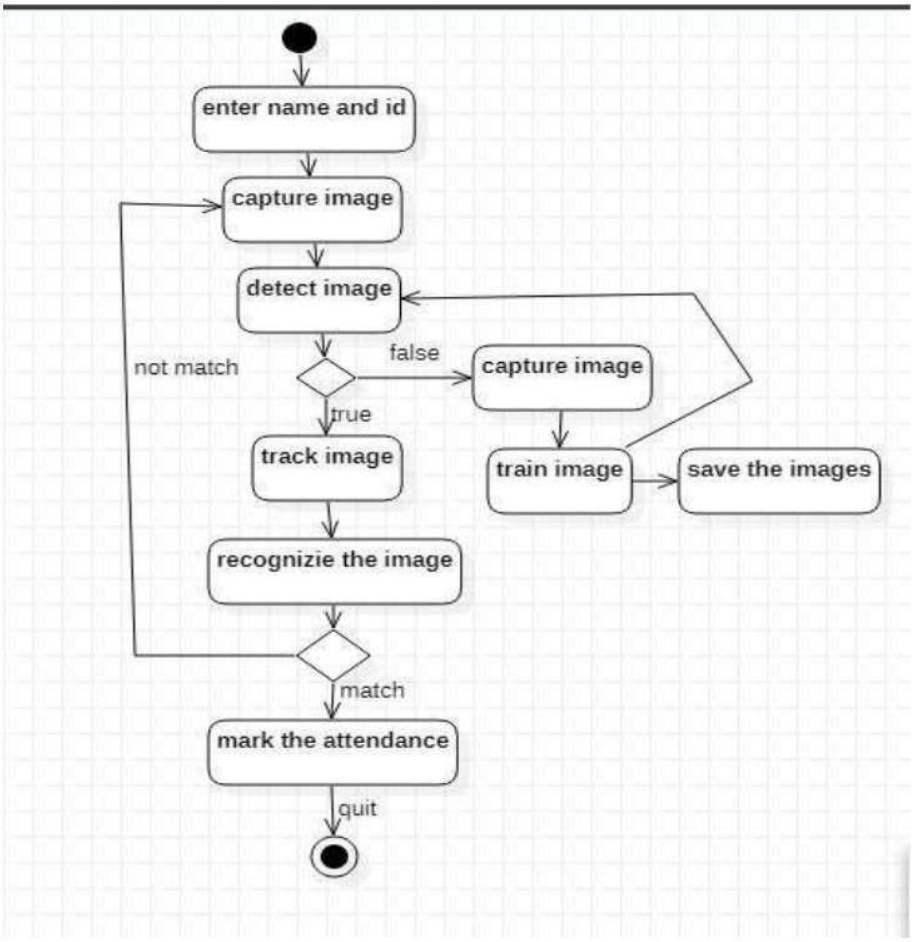


Fig 5.4: Activity Diagram

CHAPTER-5

IMPLEMENTATION

5.Implementation

Methodology:

a) Local Binary Pattern Histogram (LBPH)
b) Haar Cascade Classifier

The project deployed on Haar Cascade classifier to find the positive and negative of the face and LBPH (Local binary pattern histogram) algorithm for face recognition by using python programming and OpenCV library.

5.1 Local Binary Pattern Histogram (LBPH):

A local binary pattern (LBP) is a type of visual descriptor used for classification in computer vision. LBP is the particular case of the Texture Spectrum model proposed in 1990. LBP was first described in 1994. It has since been found to be a powerful feature for texture classification; it has further been determined that when LBP is combined with the Histogram of oriented gradients (HOG) descriptor, it improves the detection performance considerably on some datasets. A comparison of several improvements of the original LBP in the field of background subtraction was made in 2015 by Silva et al. A full survey of the different versions of LBO can be found in Bouwmans. Python mahotas, an open source computer vision package which includes an implementation of LBPs. Open CV's cascade classifiers support LBPs as of version2. LBP Library is a collection of eleven local binary patterns (LBP) algorithms developed for background subtraction problem.

In the Local Binary Patterns Histogram algorithm (LBPH) for face recognition. It is based on local binary operator and is one of the best performing texture descriptors. The need for facial recognition systems is increasing day by day. They are being used in entrance control, surveillance system, Smartphone unlocking etc. In this project we will use LBPH to extract features from an input test image and match them with the faces in system's database. Local Binary Pattern Histogram algorithm was proposed in 2006. It is based on local binary operator. It is widely used in facial recognition due to its computational simplicity and discriminative power. The steps involved to achieve this are:

- Creating dataset
- Face acquisition
- Feature extraction
- Classification

The LBPH algorithm is a part of OpenCV.

Steps:

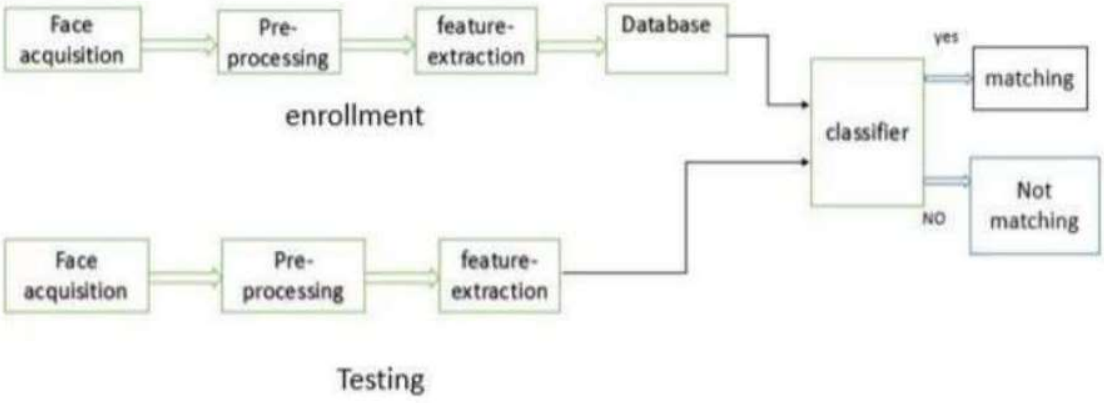


Fig 5.1: Process involved in LBPH

- Suppose we have an image having dimensions N x M.
- We divide it into regions of same height and width resulting in m x m dimension for every region.

- Local binary operator is used for every region. The LBP operator is defined in window of 3x3.

$$LBP(x_c, y_c) = \sum_{p=0}^{P-1} 2^p s(i_p - i_c)$$

here '(Xc,Yc)' is central pixel with intensity 'Ic'. And 'In' being the intensity of the the neighbour pixel

- Using median pixel value as threshold, it compares a pixel to its 8 closest pixels using this function.

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

- If the value of neighbour is greater than or equal to the central value it is set as 1 otherwise it is set as 0.
- Thus, we obtain a total of 8 binary values from the 8 neighbours.
- After combining these values, we get an 8 bit binary number which is translated to decimal number for our convenience.
- This decimal number is called the pixel LBP value and its range is 0-255.

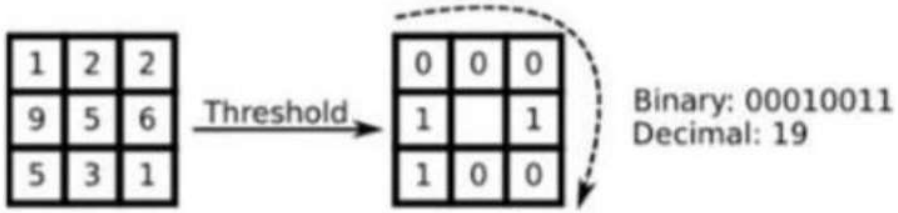
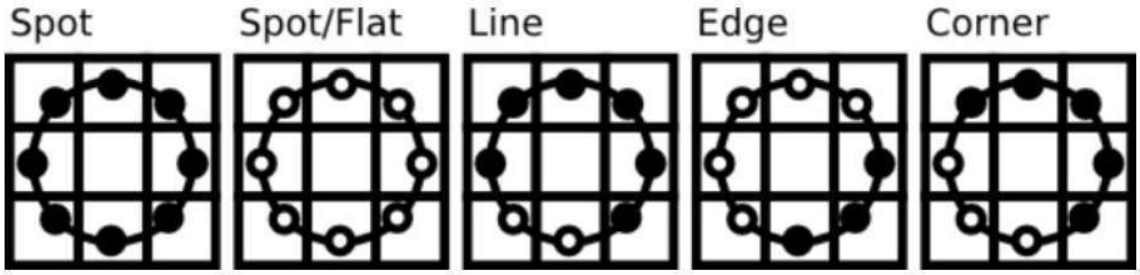


Fig 5.2: LBP pixel value

- Later it was noted that a fixed neighbourhood fails to encode details varying in scale. The algorithm was improved to use different number of radius and neighbors, now it was known as circular LBP.
- The idea here is to align an arbitrary number of neighbors on a circle with a variable radius. This way the following neighborhoods are captured:



- For a given point (Xc, Yc) the position of the neighbour (Xp, Yp), p belonging to P can be calculated by:

$$x_p = x_c + R\cos(\frac{2\pi p}{P})$$

$$y_p = y_c - R\sin(\frac{2\pi p}{P})$$

here R is radius of the circle and P is the number of sample points.

- If a point coordinate on the circle doesn't correspond to image coordinates, it gets interpolated generally by bilinear interpolation:

$$f(x,y) \approx \begin{bmatrix} 1-x & x \end{bmatrix} \begin{bmatrix} f(0,0) & f(0,1) \\ f(1,0) & f(1,1) \end{bmatrix} \begin{bmatrix} 1-y \\ y \end{bmatrix}$$

- The LBP operator is robust against monotonic gray scale transformations.

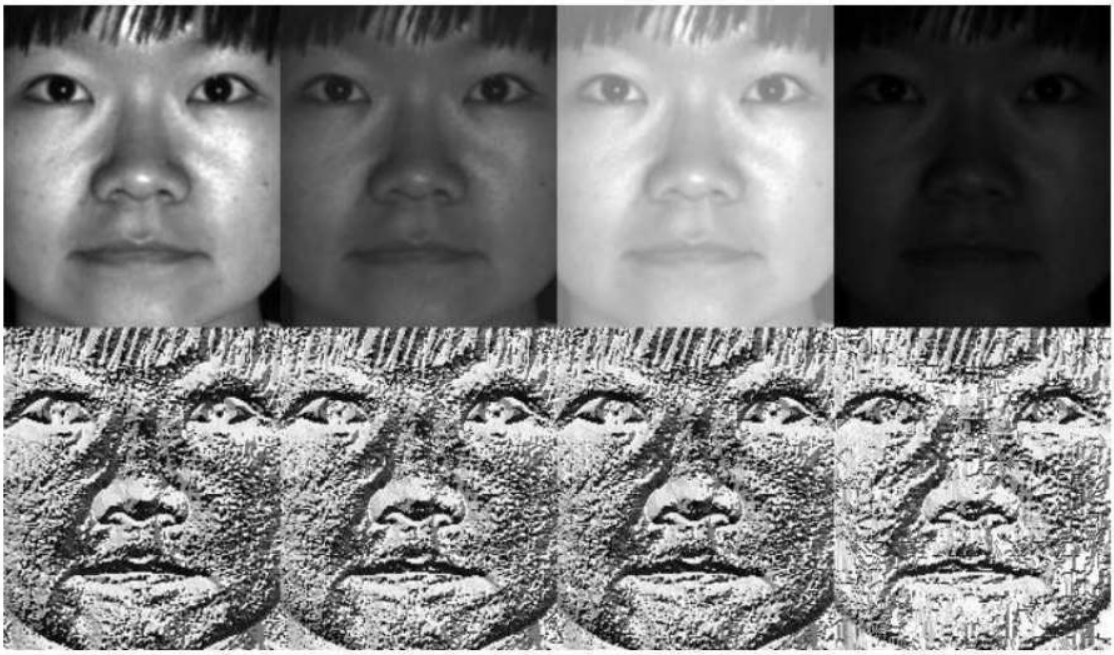


Fig 5.3: monotonic grey scale transformations

- After the generation of LBP value histogram of the region is created by counting the number of similar LBP values in the region.

- After creation of histogram for each region all the histograms are merged to form a single histogram and this is known as feature vector of the image.

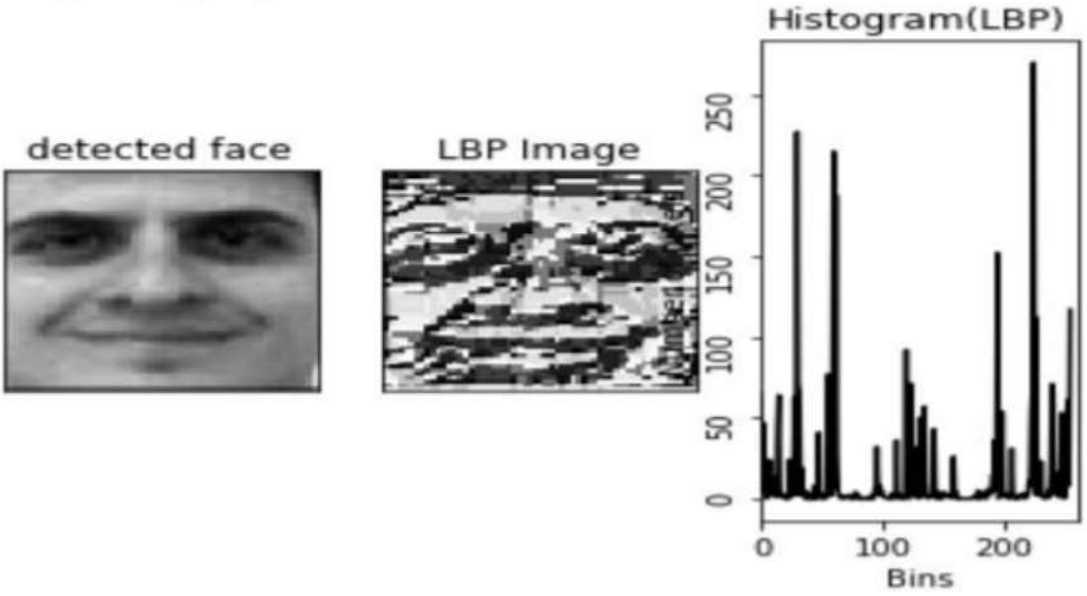


Fig 5.4: Feature vector of the image

- Now we compare the histograms of the test image and the images in the database and then we return the image with the closest histogram.
  (This can be done using many techniques like Euclidean distance, chi-square, absolute value etc)
- The Euclidean distance is calculated by comparing the test image features with features stored in the dataset. The minimum distance between test and original image gives the matching rate.

$$d(a,b) = \sqrt{\sum_{i=1}^{n} |a_i - b_i|^2}$$

- As an output we get an ID of the image from the database if the test image is recognised.

LBPH can recognise both side and front faces and it is not affected by illumination variations which means that it is more flexible.

Implementation:

The dataset can be created by taking images from webcam or from saved images. We will take many samples of a single person. A unique ID or a name is a given to a person in the database.

#import the necessary libraries

Import cv2

Import OS

- Creating LBPH model and training it with the prepared data
  model = cv. face. Create LBPH Face Recognizer ()
  model. train (faces, np. array (labels))
- Testing the trained model using a test image
  def predict_image (test_image)
  img = test_image. Copy ()
  face, bounding_ box = face_detection (img)
  label = model. Predict (face)
  label_text = database [label-1]
  print (label)
  print (label_text)
  (x, y, w, h) = bounding_box
  cv2.rectangle (img, (x, y), (x + w, y + h), (0,255, 0), 2)
  cv2.Font_HERSHEY_PLAIN, 1.5, (0, 255, 0), 2)
  Return img
  test1 = cv2. imread ("test/tom.jpg")
  predict1 =predict_image (test1)
  cv2.imshow ('Face Recognition', predict1)
  cv2.waitkey (0)
  cv2.destroyAllWindows ()

Advantages of LBPH Algorithm:

- It can represent local features in the images.
- LBPH can recognize both side and front faces.
- LBPH method will probably work better on our training and testing dataset.
- Raw intensity data are used directly for learning and recognition without any significant low -level or mid-level processing.
- Data compression is achieved by the low-dimensional subspace representation.
- Recognition is simple and efficient compared to other matching approaches.
- No knowledge of geometry and reflectance of faces are required.

5.2 Haar Cascade:

Haar Cascade is a machine learning object detection algorithm used to identify objects in an image or video and based on the concept of features proposed by Paul Viola and Michael Jones in their paper "Rapid Object Detection using a Boosted Cascade of simple Features in 2001. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative image. It is then used to detect objects in their images. Luckily, OpenCV offers predefined Haar Cascade algorithm, organized into categories depending on the images they have been trained on.

Now let's see how this algorithm concretely works. The idea of Haar Cascade is extracting features from images using a kind of 'filter', similar to the concept of the convolutional kernel. These filters are called Haar features.

Algorithm:

```
import numpy as np

    import cv2

    face_cascade = cv2. Cascade Classifier ("haarcascade_frontalface_default. Xml")

    eye_cascade =cv2.CascadeClassifier (1'haarcascade_eye.xml")

    img = cv2. imread ("image .jpg")

    gray =cv2. CvtColor (img, cv2.COLOR_BGR2GRAY)

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

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

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

    roi_gray = gray [y: y+h, x: x+w]

    roi_color = img[y: y+h, x: x=w]

    eyes = eye_cascade.detectMultiScale (roi_gray)

    for (ex, ey, ew, eh) in eyes:

    cv2.rectangle (roi_color, (ex, ey), (ex+ew, ey+eh), (0, 255, 0), 2)

    cv2 .imshow ('img', img)

    cv2.Waitkey (0)

    cv2.destroyAllWindows ()
```

Initially, the algorithm needs a lot of positive images of faces and negative images without faces to train the classifier. Then we need to extract features from it. First step is to collect the Haar features.

- Haar Feature Selection
- Creating Integral images
- Adaboost Training
- Cascading Classifiers

As you can see, our algorithm worked pretty well! If you explore the whole library of Haar Cascade algorithm, you will see that their specific models improve trained on different features of the human's physical aspect, hence you can your model by adding more features detection.

But among all these features we calculated, most of them are irrelevant. For example, consider the image. Top row shows two good features. The first feature selected seems to focus on the property that the region of the eyes is often darker than the region of the nose and cheeks. The second feature selected relies on the property that the eyes are darker than the bridge of the nose. But the same windows applying on cheeks or any other place is irrevalant. So how do we select the best features out of 160000+ features? It is achieved by Adaboost.

For this, we can apply each and every feature on all the training images. For each feature, it finds the best threshold which will classify the faces to positive and negative. But obviously, there will be errors rate misclassifications. We select the features with minimum error rate, which means they are the features that best classifies the face and non-face images.

Haar Cascade classifier is based on the Haar Wavelet technique to analyze pixels in the images into squares by function. This uses "integral images" concepts to compute the" features" detected. Haar Cascades use the Adaboost learning algorithm which selects a small number of important features from a large to give an efficient result of classifiers then use cascading techniques to detect face in an image. Haar Cascade classifier is based on Viola Jones detection algorithm which is trained in given some input faces and non-faces and training a classifier which identifies a face. Viola Jones face detection algorithm is trained and weights are stored in the disk. All we do is take the features from the file and apply to our image; it faces is present in the image we get the face location.

A Haar Cascade is basically a classifier which is used to detect the object for which it has been trained for, from the source. Better results are obtained by using high quality images and increasing the amount of stages for which the classifier is trained. So, face recognition can be made done easy representing by using haar cascade classifier algorithm.

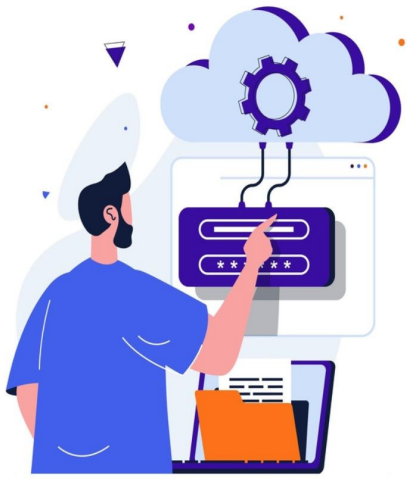
Chapter-7

SCREENSHOTS

login page

facendance

WE THINK FEATURES



Lets Get In

Username : pd365000@gmail.com

Password :

Login

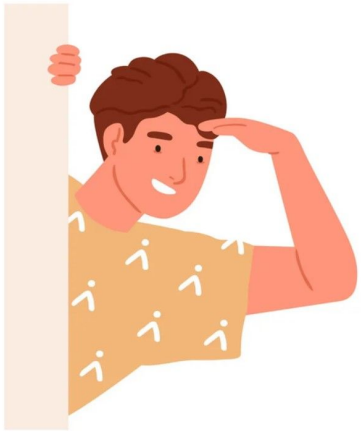
Register

Forgot Credentials ?

registration page

facendance

WE THINK FEATURES



Ready To Register ! Lets Go Then

First Name : Prince

Last Name : Dwivedi

Contact No. : 7440324634

Email Id : pd365000@outlook.co

Secret Phrase : Your Pet's Name

Answer : jojo

Password : pass_pass@123

Confirm Password : pass_pass@123

Agree to terms and conditions

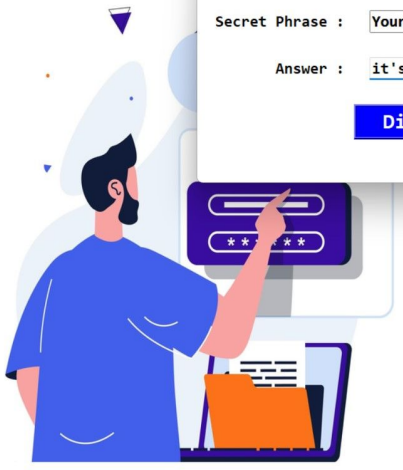
Register

Login

login page

facendance

WE THINK FEATURES



Forgot Password

Forgot Password

Secret Phrase : Your Secret

Answer : it's a secret

Dive In

Lets Get In

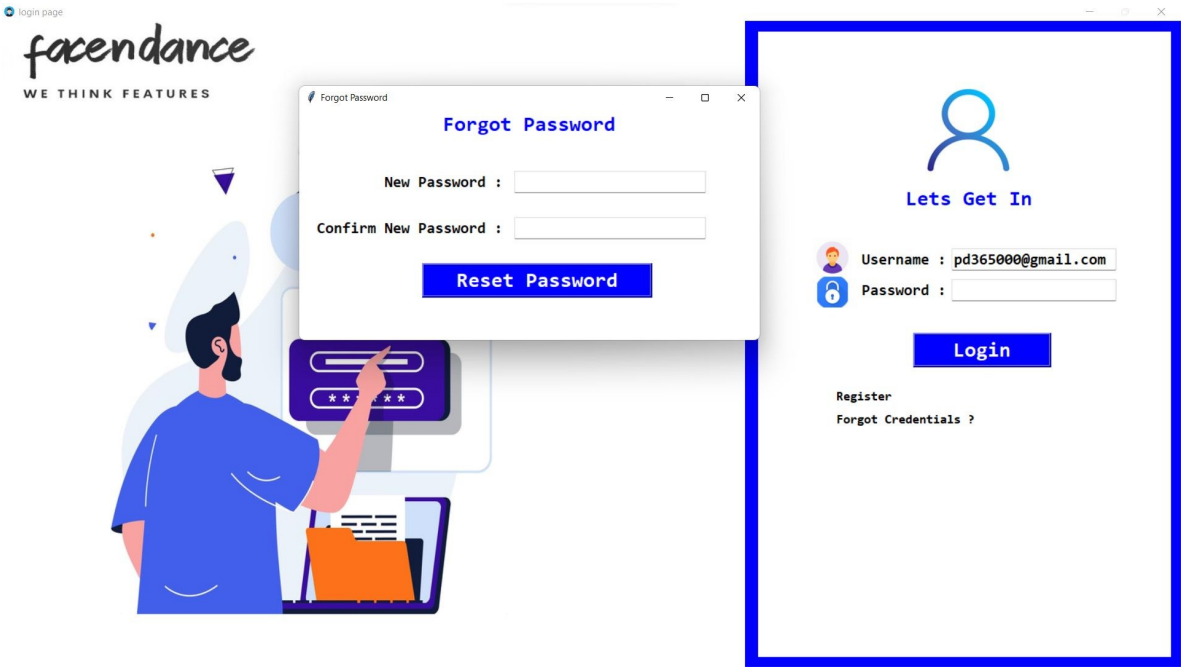
Username : pd365000@gmail.com

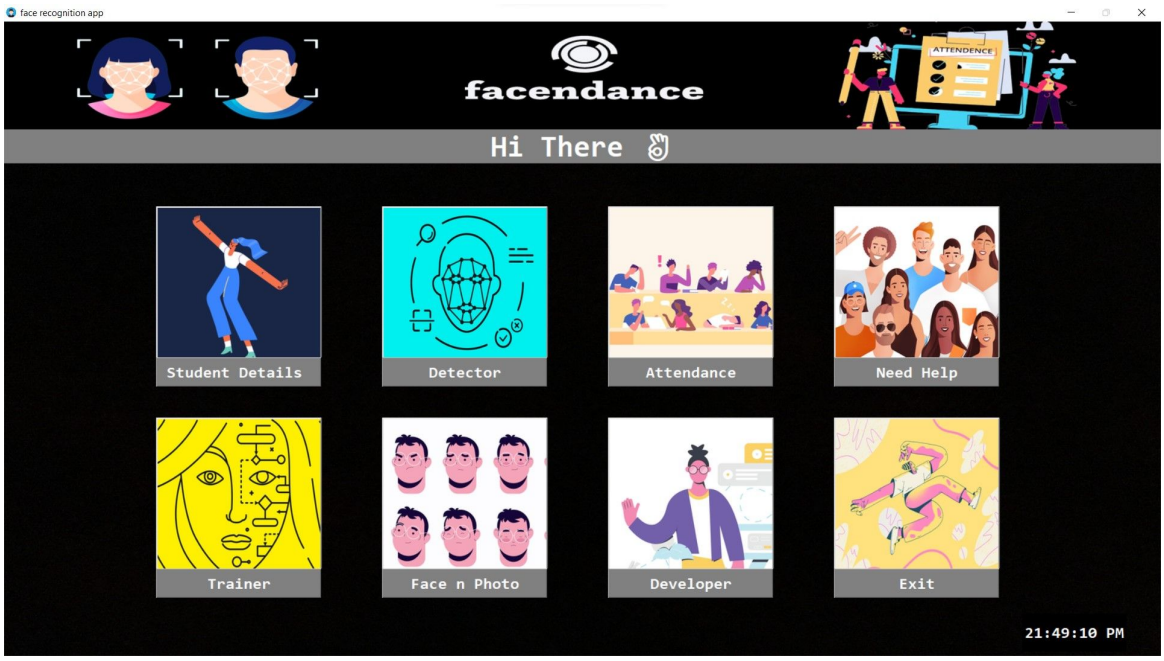
Password :

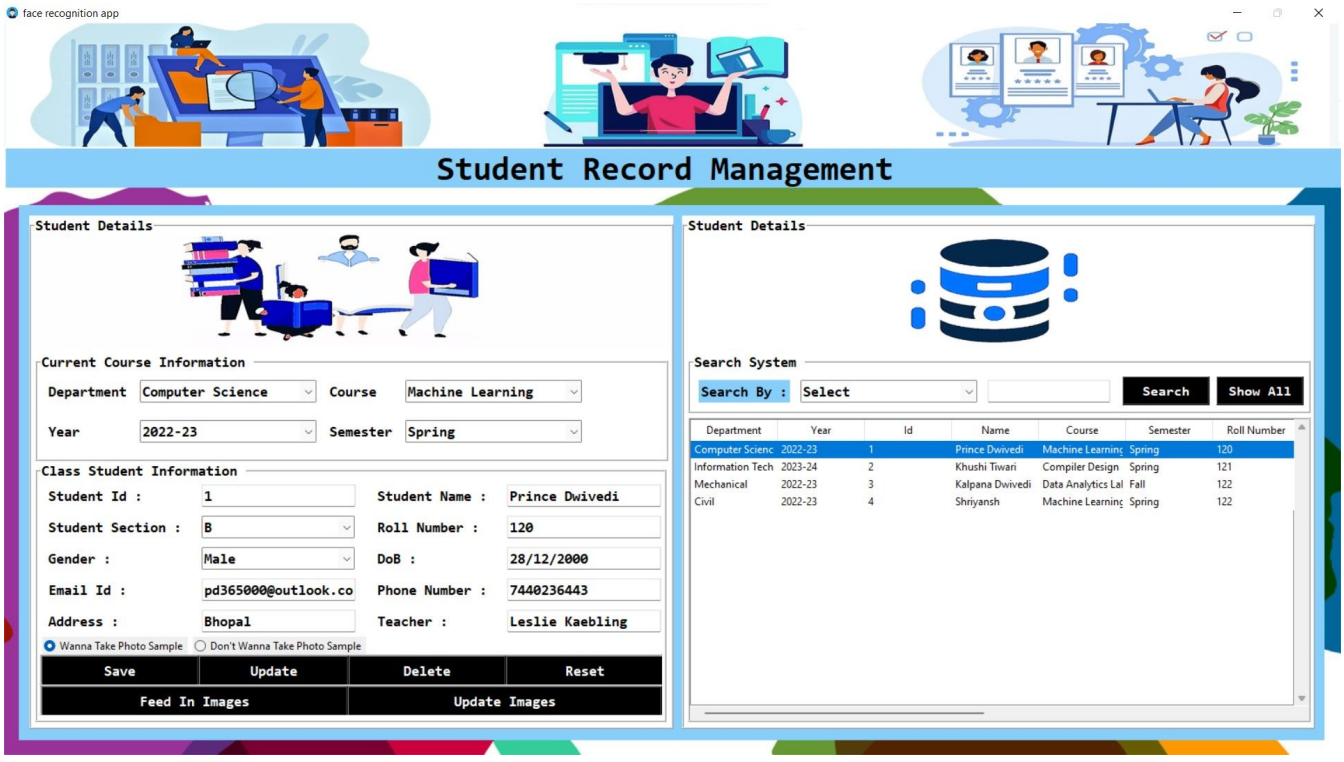
Login

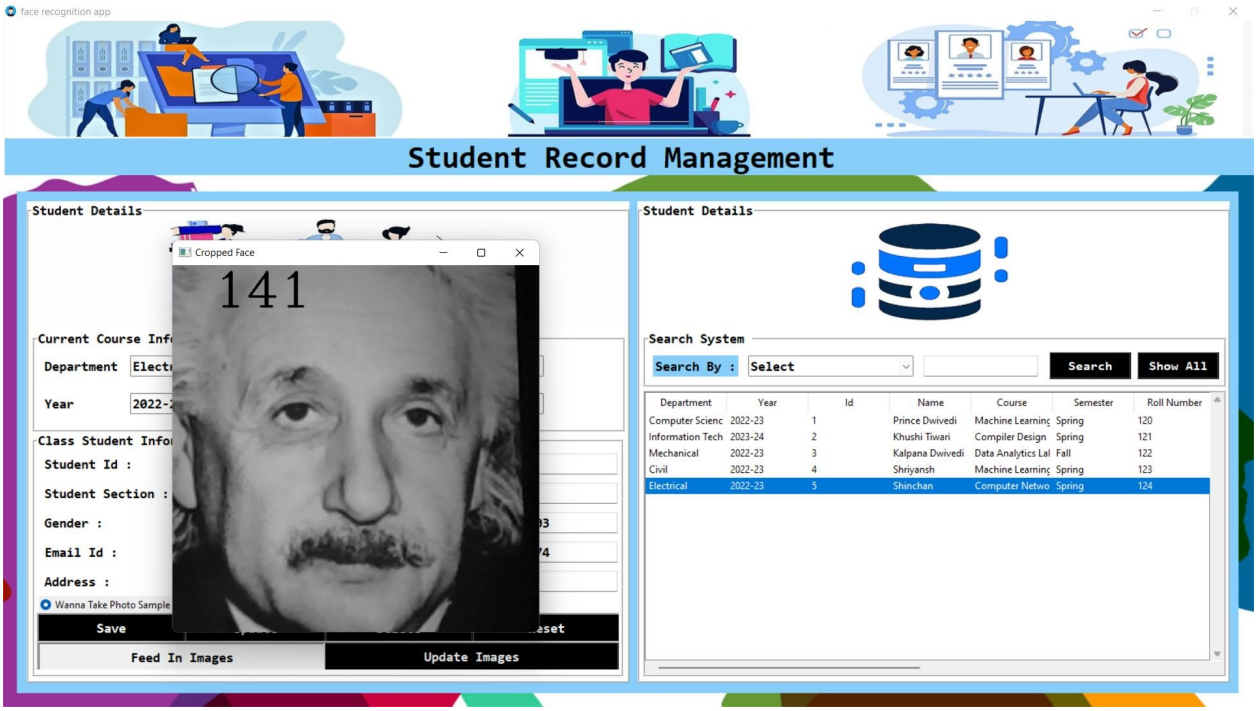
Register

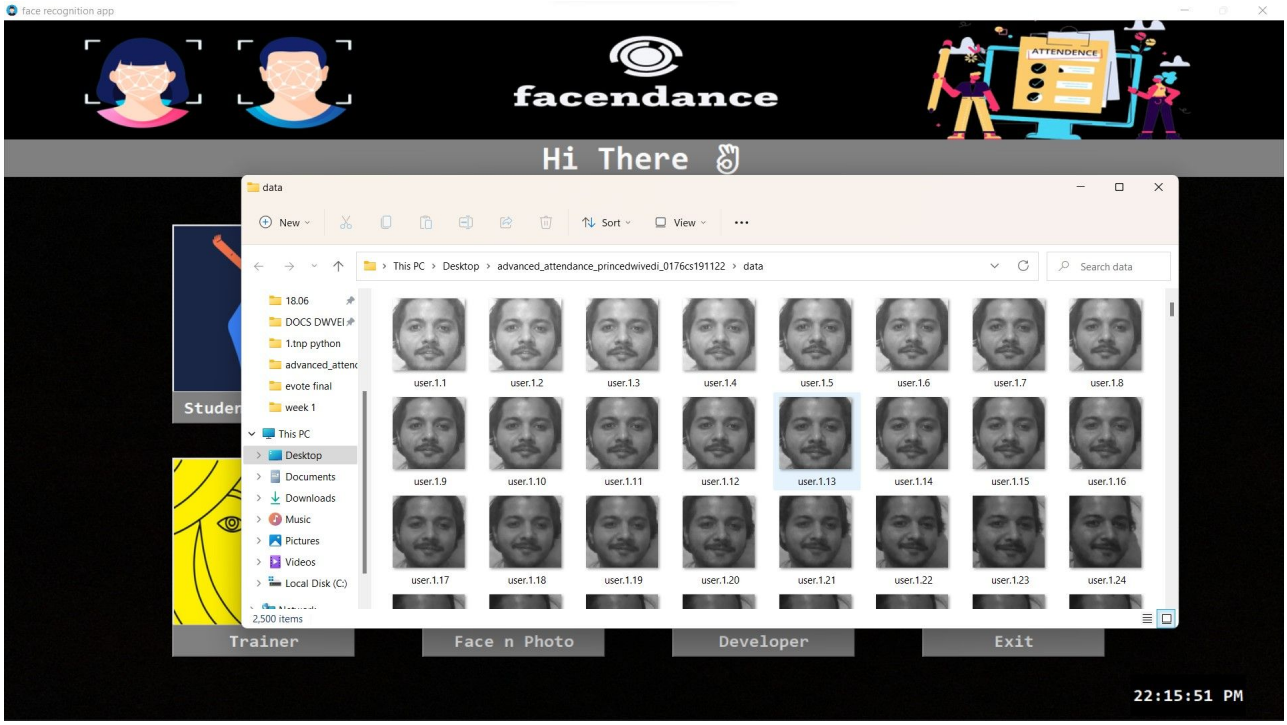
Forgot Credentials ?

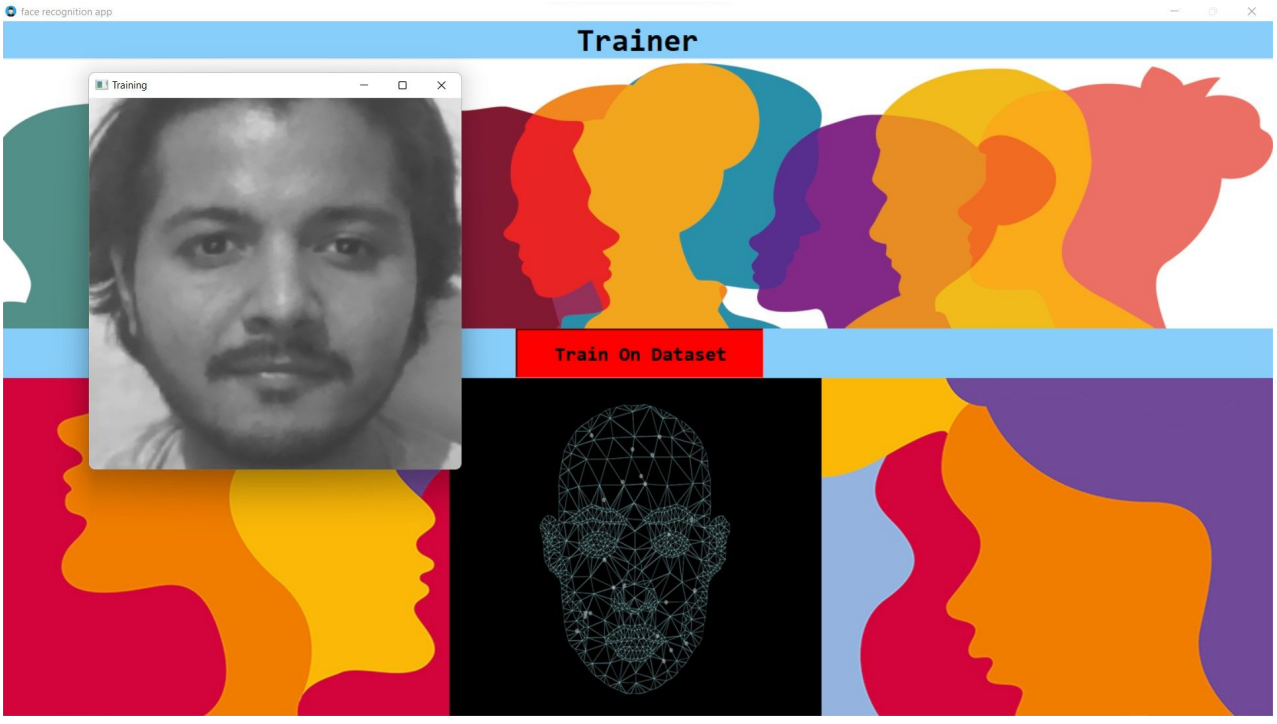

login page
facendance
WE THINK FEATURES
Forgot Password
Forgot Password
New Password :
Confirm New Password :
Reset Password
Lets Get In
Username : pd365000@gmail.com
Password :
Login
Register
Forgot Credentials ?


face recognition app
facendance
ATTENDENCE
Hi There
Student Details
Detector
Attendance
Need Help
Trainer
Face n Photo
Developer
Exit
21:49:10 PM


face recognition app
Student Record Management
Student Details
Current Course Information
Department Computer Science
Course Machine Learning
Year 2022-23
Semester Spring
Class Student Information
Student Id : 1
Student Name : Prince Dwivedi
Student Section : B
Roll Number : 120
Gender : Male
DoB : 28/12/2000
Email Id : pd365000@outlook.co
Phone Number : 7440236443
Address : Bhopal
Teacher : Leslie Kaebling
Wanna Take Photo Sample
Don't Wanna Take Photo Sample
Save
Update
Delete
Reset
Feed In Images
Update Images
Student Details
Search System
Search By : Select
Search
Show All
Department Year Id Name Course Semester Roll Number
Computer Scienc 2022-23 1 Prince Dwivedi Machine Learning Spring 120
Information Tech 2023-24 2 Khushi Tiwari Compiler Design Spring 121
Mechanical 2022-23 3 Kalpana Dwivedi Data Analytics Lal Fall 122
Civil 2022-23 4 Shriyansh Machine Learning Spring 122


face recognition app
Student Record Management
Student Details
Cropped Face
141
Current Course Info
Department
Year
Class Student Info
Student Id :
Student Section :
Gender :
Email Id :
Address :
Wanna Take Photo Sample
Save
Feed In Images
Update Images
Student Details
Search System
Search By :
Select
Search
Show All
Department Year Id Name Course Semester Roll Number
Computer Scienc 2022-23 1 Prince Dwivedi Machine Learning Spring 120
Information Tech 2023-24 2 Khushi Tiwari Compiler Design Spring 121
Mechanical 2022-23 3 Kalpana Dwivedi Data Analytics Lal Fall 122
Civil 2022-23 4 Shriyansh Machine Learning Spring 123
Electrical 2022-23 5 Shinchan Computer Netwo Spring 124

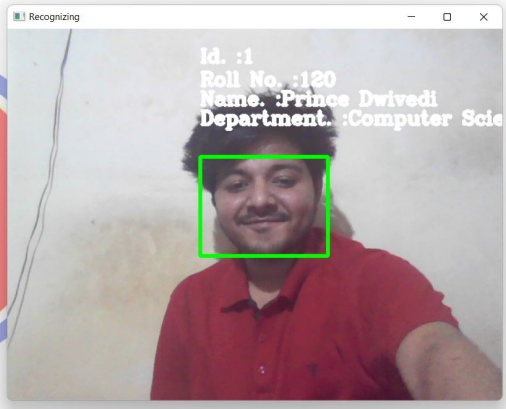

face recognition app
facendance
ATTENDENCE
Hi There
data
New
Sort
View
This PC › Desktop › advanced_attendance_princedwivedi_0176cs191122 › data
Search data
18.06
DOCS DWVEI
1.tnp python
advanced_attenc
evote final
week 1
This PC
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
2,500 items
user.1.1 user.1.2 user.1.3 user.1.4 user.1.5 user.1.6 user.1.7 user.1.8
user.1.9 user.1.10 user.1.11 user.1.12 user.1.13 user.1.14 user.1.15 user.1.16
user.1.17 user.1.18 user.1.19 user.1.20 user.1.21 user.1.22 user.1.23 user.1.24
Trainer
Face n Photo
Developer
Exit
22:15:51 PM


face recognition app
Trainer
Training
Train On Dataset

face recognition app

Recognition

Recognizing



Face Recognition

face recognition app

Facendance Management

Student Attendance Details

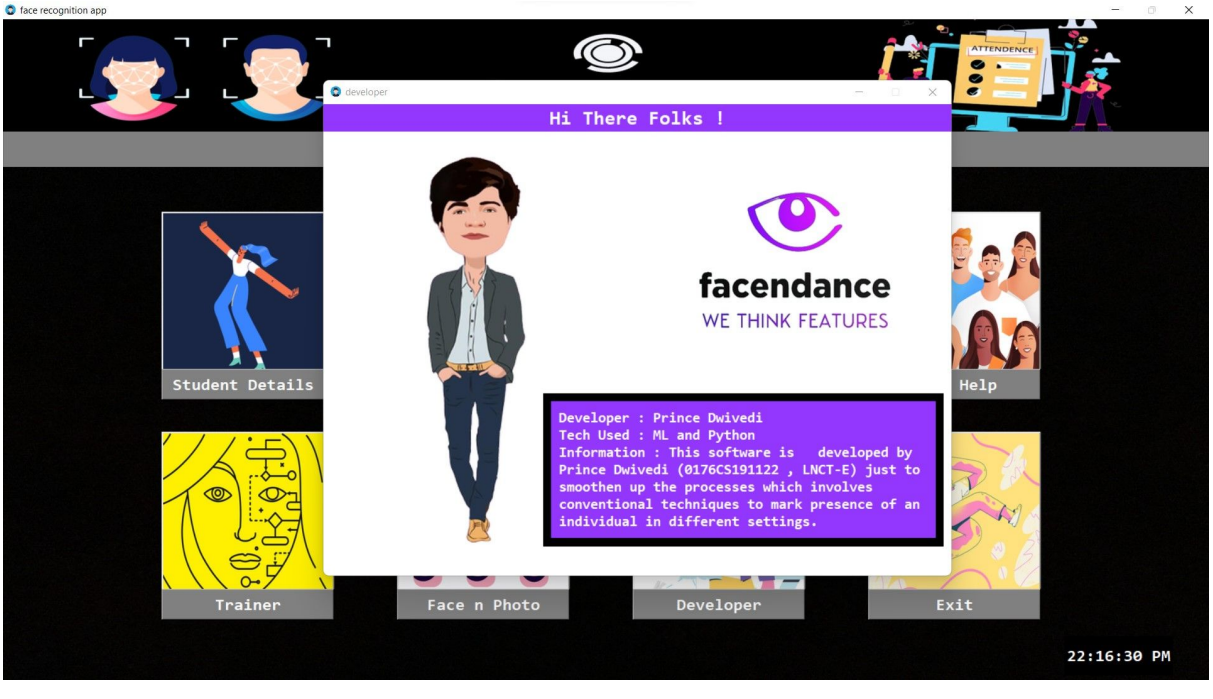
Attendance Id :	1	Roll No :	120
Name :	Prince Dwivedi	Department :	Computer Science
Date :	22:09:24	Date :	17/04/2022
Attendance Status :	Present		

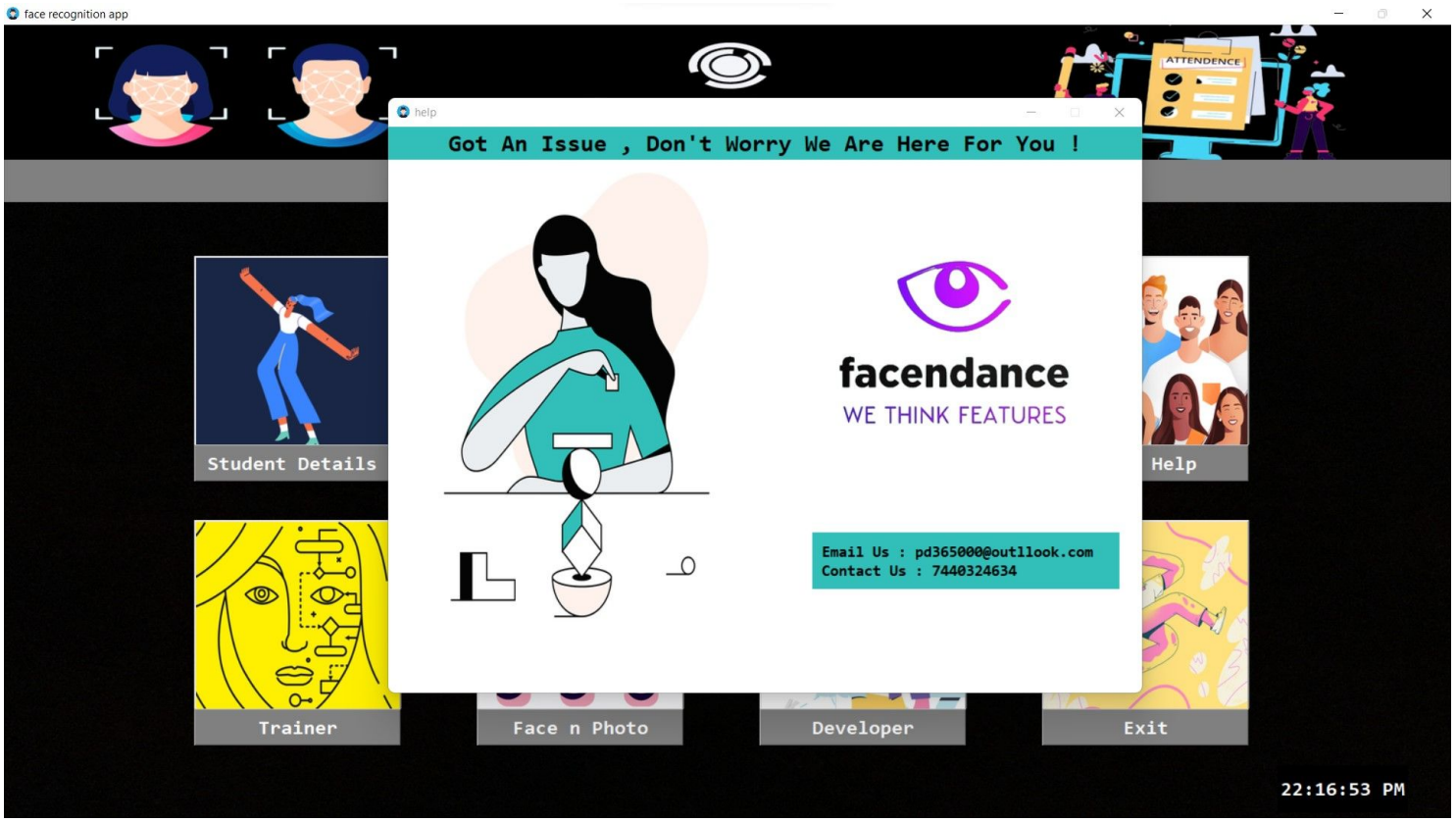
Import csv | Export csv | Update | Reset

Student Attendance Details

Attendance	Ro	Name	Department	Time	Date	Attendance Statu
1	12	Prince Dwivedi	Computer Scienc	22:09:24	17/04/2022	Present
2	12	khushi tiwari	Computer Scienc	18:38:11	21/04/2022	Present
3	12	Kalpana Dwivedi	Mechanical	22:39:02	21/04/2022	Present
4	12	Shriyansh	Civil	19:35:34	30/04/2022	Present
5	12	Shinchan	Electrical	22:14:18	03/05/2022	Present

face recognition app





C:/Users/ERA/Desktop/desktop/4.advanced_attendance_princedwivedi_0176cs191122 - Spyder (Python 3.9)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\ERA\Desktop\desktop\4.advanced_attendance_princedwivedi_0176cs191122\login.py

MAIN.py | train.py | recognition.py | Attendance.py | student.py | login.py | register.py | developer.py | helper.py

```
# -*- coding: utf-8 -*-
"""
Created on Tue Apr 19 16:18:46 2022

@author: ERA
"""

from tkinter import*
from tkinter import ttk  # cool toolkit
from PIL import Image, ImageTk  # for images  , cropping and stuff
from tkinter import messagebox  # for eeror info or warning
import mysql.connector
from tkvideo import tkvideo
from register import Register
from MAIN import Face_Recognition_System



class login_window:
    def __init__(self,root):
        self.root=root

        self.width = root.winfo_screenwidth()
        self.height = root.winfo_screenheight()
        self.root.geometry(str(self.width)+"x"+str(self.height)+"+0+0")
        self.root.wm_iconbitmap("face.ico")
        self.root.title("login page")
        self.root.config(background="white")

        self.root.state('zoomed')
        self.root.resizable(0,0)


        img = Image.open(
            r"images\logo.jpg")#x.jpg
        # (W,H) ,ANTIALIAS HIGH LEVEL TO LOW LEVEL IMAGE
        img = img.resize(
            (int(self.width/2-100-100), self.height-150), Image.ANTIALIAS)
        self.photoimg = ImageTk.PhotoImage(img)  # set to a variable

        #to set it on window with label
        # lable inside root , with images help
```

Usage

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

New to Spyder? Read our tutorial

Console 1/A

```
Python 3.9.7 (default, Sep 16 2021, 16:59:28) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.29.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/ERA/Desktop/desktop/
4.advanced_attendance_princedwivedi_0176cs191122/login.py', wdir='C:/
Users/ERA/Desktop/desktop/
4.advanced_attendance_princedwivedi_0176cs191122')
```

LSP Python: ready | conda: base (Python 3.9.7) | Line 1, Col 1 | UTF-8 | CRLF | RW | Mem 62%

Chapter-8
TESTING

8.TESTING

8.1 Introduction

Once source code has been generated, software must be tested to uncover (and correct) as many errors as possible before delivery to customer. Our goal is to design a series of test cases that have a high likelihood of finding errors. To uncover the errors software techniques are used. These techniques provide systematic guidance for designing test that

(1) Exercise the internal logic of software components and

(2) Exercise the input and output domains of the program to uncover errors in program function, behavior and performance

Steps:

Software is tested from two different perspectives:

(1) Internal program logic is exercised using —White box‖ test case design Techniques.

(2) Software requirements are exercised using —block box testcase

Design techniques. In both cases, the intent is to find the maximum number of errors with the Minimum amount of effort and time.

Testing should be made at every level of performing the task. By making testing we can know what our mistakes are. So, testing should be done and it is very primary aspect. There are different types of testing methods. These testing methods were divided into two types.

1. White box testing
2. Black box testing

1. White Box Testing

White box testing requires access to the source code. White box testing requires knowing what makes software secure or insecure, how to think like an attacker, and how to use different testing tools and techniques. The first step in white box testing is to comprehend and analyze source code, so knowing what makes software secure is a fundamental requirement. Second, to create tests that exploit software, a tester must think like an attacker. Third, to perform testing effectively, testers need to know the different tools and techniques available for white box testing.

In this testing only the output is checked for correctness. The logical flow of the data is not checked. In my project. I tested the source code, in that all independent paths have been executed. And all loops at their boundaries and within their operational.

2. Black Box Testing

Black box testing treats the software as a "black box", examining functionality without any knowledge of internal implementation. The tester is only aware of what the software is supposed to do, not how it does. Black box testing methods include: equivalence partitioning, boundary value analysis, all-pairs testing, state transition tables, decision table testing, fuzz testing, model-based testing, use case testing, exploratory testing and specification-based testing.

8.2 Testing Methodologies:

A strategy for software testing must accommodate low-level tests that are necessary to verify that a small source code segment has been correctly implemented as well as high-level tests that validate major system functions against customer requirements. A strategy must provide guidance for the practitioner and a set of milestones for the manager. Because the steps of the test strategy occur at a time when deadline pressure begins to rise, progress must be measurable and problems must surface as early as possible. Following testing techniques are well known and the same strategy is adopted during this project testing.

Unit testing:

Unit testing focuses verification effort on the smallest unit of software design- the software component or module. The unit test is white-box oriented. The unit testing implemented in every module of student attendance management System. by giving correct manual input to the system, the data are stored in database and retrieved. If you want required module to access input or get the output from the End user. any error will accrue the time will provide handler to show what type of error will accrued.

System testing:

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. Below we have described the two types of testing which have been taken for this project. it is to check all modules worked on input basis. If you want change any values or inputs will change all information. so specified input is must.

8.3 Test cases

Test case is an object for execution for other modules in the architecture does not represent any interaction by itself. A test case is a set of sequential steps to execute a test operating on a set of predefined inputs to produce certain expected outputs. There are two types of test cases:
- Manual and automated.

A manual test case is executed manually while an automated test case is executed using automation. In system testing, test data should cover the possible values of each parameter based on the requirements. Since testing every value is impractical, a few values should be chosen from each equivalence class. An equivalence class is a set of values that should all be treated the same. Ideally, test cases that check error conditions are written separately from the functional test cases and should have steps to verify the error messages and logs.

Realistically, if functional test cases are not yet written, it is ok for testers to check for error conditions when performing normal functional test cases. It should be clear which test data, if any is expected to trigger errors.

Test cases:

S.NO	TEST CASE DESCRIPTION	EXPECTED VALUE	ACTUAL VALUE	RESULT
1	Camera opens after clicking the take image button.	Camera is opened	Camera is opened	Pass
2	Capturing of image	After opening the camera, image is captured	After opening the camera, image is captured	Pass
3	Notification message showed in the notification bar after capturing the image	Notification is showed	Notification is showed	Pass
4	Check whether id and name are stored in Students Details file.	Details are stored in the CSV file (Students Details)	Details are stored in the CSV file (Students Details)	Pass
5	Checking of image samples are stored in Training image folder	Images are stored	Images are stored	Pass
6	check the images trained or not after clicking train image button	Images are trained and a notification message is displayed on the Notification bar	Images are trained and a notification message is displayed on the Notification bar	Pass
7	Check whether the camera opens after clicking the track image button	Camera is opened	Camera is opened	Pass

8	Check whether attendance stored in the attendance folder after quitting.	Attendance is stored	Attendance is stored	Pass
9	Detecting multiple faces at once	Multiple faces are detected	Multiple faces are detected	Pass
10	Update attendance for multiple people at once	Update attendance for all faces are detected	Attendance is updated for only some persons	Fail
11	Check whether the camera recognizes the detected faces or not	Name and id of the recognized person is displayed	Name and id of the recognized person is displayed	Pass
12	Check whether space and special character symbols are allowed while entering the name of the student	space is allowed but no special characters are allowed while entering the name	Either of the two are not allowed while entering the name	Fail

Table No:8.3

Chapter-9

CONCLUSION AND FUTURE SCOPE

9. CONCLUSION AND FUTURE SCOPE

9.1 Conclusion:

Thus, the aim of our project is to capture the images of the students, convert it into frames, relate it with the database to ensure their presence or absence, mark attendance to the particular student to maintain the record. The Automated face Recognition Attendance System helps in increasing the accuracy and speed ultimately achieve the high-precision real-time attendance to meet the need for automatic classroom evaluation. This system is designed to minimize the human effort for taking the attendance manually that take place in every college. As the attendance marking process is done without any human interference, which is the main scope in the system.

9.2 Future scope:

Besides, we can simplify the system and make more efficient by taking advantage of multiple face detections to mark attendance of all the visible faces in single attempt. This will be economical and more efficient use of face recognition for attendance marking. We also consider to develop an android application for this system in near future.

CHAPTER – 10

BIBLIOGRAPHY

10. BIBLIOGRAPHY

Weblinks:

1. https://towardsdatascience.com/computer-vision-detecting-objects-using-haar-cascade-classifier-4585472829a9

2. https://iq.opengenus.org/lbph-algorithm-for-face-recognition/

3. https://github.com/ashishdubey10/Face-Recognition-Based-Attendance-System/blob/master/haarcascade_frontalface_default.xml

4. https://www.edureka.co/blog/python-opencv-tutorial/

5. https://www.edureka.co/blog/tkinter-tutorial/