FACE RECOGNITION BASED ATTENDANCE SYSTEM

An industrial oriented mini project report

Submitted By

T. SWATHI (16W91A0557)
P.MANISH KUMAR (16W91A0544)
S. SRINITHA (16W91A0553)
S.RUTHWICK (16W91A0551)

Under the Esteemed Guidance of

Dr. Karthick Raghunath K M
Associate Professor,CSE
To
JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
In partial fulfilment of the requirements for award of degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE & ENGINEERING
2016– 2020

MALLA REDDY
INSTITUTE OF
ENGINEERING AND TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
MALLA REDDY INSTITUTE OF ENGINEERING AND TECHNOLOGY
(MRET-W9)
(Sponsored by Malla Reddy Educational society)
(Accredited by NBA, Permanent Affiliated to JNTU, Hyderabad)
Maisammaguda, Dhulapally post, Secunderabad-500014.

DECLARATION

We hereby declare that the project entitled "**FACE RECOGNITION BASED ATTENDANCE SYSTEM**" submitted to Malla Reddy Institute of Engineering and Technology (MRET-W9), affiliated to Jawaharlal Nehru Technological University Hyderabad (JNTUH) for the award of the degree of Bachelor of Technology in Computer Science & Engineering is a result of original industrial oriented project done by us.

It is further declared that the project report or any part thereof has not been previously submitted to any University or Institute for the award of degree or diploma.

T.SWATHI - (16W91A0557)

P.MANISH KUMAR - (16W91A0544)

S.SRINITHA - (16W91A0553)

S.RUTHWICK - (16W91A0551)

MALLA REDDY INSTITUTE OF ENGINEERING & TECHNOLOGY
(Sponsored by Malla Reddy Educational Society)
Accredited by NBA, Affiliated to JNTU, Hyderabad
Maisammaguda, Dhulapally (post via Hakimpet), Sec'Bad-500 014.
Phone: 040-65969674, Cell: 9348161223

Department of Computer Science and Engineering

PROJECT CERTIFICATE

This is to certify that this is the bonafide record of the project titled "**FACE RECOGNITION BASED ATTENDANCE SYSTEM**" is submitted by **T. Swathi (16W91A0557), P. Manish Kumar (16W91A0544), S. Srinitha (16W91A0553), S. Ruthwick (16W91A0551)** of B.Tech in the partial fulfilment of the requirements for the degree of **Bachelor of Technology in Computer Science and Engineering**, Dept. of Computer Science & Engineering and this has not been submitted for the award of any other degree of this institution.

HOD

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

First and foremost, I am grateful to the Principal **Dr. M. ASHOK**, for providing me with all the resources in the college to make my project a success. I thank him for his valuable suggestions at the time of seminars which encouraged me to give my best in the project.

I would like to express my gratitude to **Dr. ANANTHA RAMAN,** Head of the Department, Department of Computer Science and Engineering for his support and valuable suggestions during the dissertation work

I offer my sincere gratitude to my project - coordinator **Mr. M. GANESH KUMAR** and internal guide **Dr. KARTHICK RAGHUNATH KM** Associate Professor of Computer Science and Engineering department who has supported me throughout this project with their patience and valuable suggestions.

I would also like to thank all the supporting staff of the Dept. of CSE and all other departments who have been helpful directly or indirectly in making the project a success.

I am extremely grateful to my parents for their blessings and prayers for my completion of project that gave me strength to do my project.

T.SWATHI

P.MANISH KUMAR

S.SRINITHA

S.RUTHWICK

INDEX

1. INTRODUCTION

1.1 MOTIVATION:

In the recent years, face recognition technology has received significant attention as it has potential for a wide range of applications related to law enforcement as well as other enterprises. It is a technology capable of identifying or verifying a person from a digital image. This has emerged as an attractive solution to address many contemporary needs for identification and verification of identity claims.

Face is a typical multidimensional structure and needs good computational analysis for recognition. Also people have started to use image capturing devices never as before with the advent of smart phones and closed circuit television.

1.2 PROBLEM DEFINITION:

Face recognition based attendance system is an automated system which is developed to take attendance by recognizing faces. Face recognition based attendance system is a solution in many organizations. For example attendance in educational institutions is normally taken by a teacher and it is an important part of daily classroom evaluation. But it may appear that a teacher may miss someone or some students answer multiple times. This results in loss of data consistency. Face recognition based attendance system is a solution for this instead of a teacher manually taking the attendance.

1.3 OBJECTIVE OF THE PROJECT:

PURPOSE:

The main objective of this project is to offer system that simplify and automate the process of Recording and tracking student's attendance through face recognition technology. It is biometric technology to identify or verify a person from a digital image. Face recognition technology is widely used now-a-days. It involves determining if the image of the face of any given person matches with any of the images stored in the dataset. If the image is matched then the person is provided with the attendance. This project is aimed to provide a system that will make the attendance process faster and more precise.

1.4. LIMITATIONS OF THE PROJECT:

1. **Poor image quality:**

   Image quality affects how well face recognition algorithms work.

2. **Face angles can throw off face recognition's reliability:**

   The relative angle of the target's face influences the recognition score profoundly.

3. **Small image sizes make face recognition more difficult:**

   When a face-detection algorithm finds face in an image or in a still form from a video capture, the relative size of that face compared with the enrolled image size affects how well the face will be recognized.

4. **Data processing and Storage can limit face recognition technology:**

   Processing every frame of video is an enormous undertaking. So usually only a fraction is actually run through a recognition system.

1.5. ORGANIZATION OF DOCUMENTATION:

In this project documentation we have initially put the definition and objective of the project as well as the design of the project which is followed by the implementation and testing phases. Finally the project has been concluded successfully and also the future enhancements of the project were given in this documentation. All the screens of the system designed with the view to provide the user with easy operations in simple and efficient way, minimum key stroke possible instructions and important information is emphasized on the screen. Almost every screen is provided with no error and important message and selection facilitates. Each screen assigned to make it as much user friendly as possible by using interaction procedure.

2. LITERATURE SURVEY

2.1 INTRODUCTION

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, etc. The 3 techniques of face recognition in openCV library are:

1. Eigen faces algorithm
2. Fisher faces algorithm
3. Local Binary Pattern Histogram (LBPH) algorithm

1. EIGEN FACES ALGORITHM:

The main idea of eigen face is to get the features in mathematical sense instead of physical face feature by using the mathematical transform for recognition. There are two phases for face recognition using eigen faces. In this phase, a large group of individual faces is acted as the training set. These training images should be a good representation of all the faces that one might encounter. With the training images, a set of eigen-vectors is found by using Principal Component Analysis. A drawback is that it is very sensitive for lightening conditions and the position of head. Finding the eigen vectors and eigen values are time consuming.

2. FISHER FACES ALGORITHM:

Fisher face is similar to Eigen face but with improvement in better classification of different classes image. The fisher faces algorithm could have better accuracy in facial expression than Eigen face approach. But fisher face is more complex than eigen face in finding the projection

of face space. It depends upon the calculation of ratios which requires a lot of processing time. This algorithm also results in larger storage of the face and more processing time recognition.

3. LBPH ALGORITHM:

The proposed system uses this algorithm to recognize face. It is more better and faster than the other algorithms . It is also robust to lightening conditions.

The Local Binary Pattern Histogram(LBPH) algorithm is a simple solution on face recognition problem, which can recognize both front face and side face.

2.2 EXISTING SYSTEM:

In the existing system, the lecturer takes the attendance of the students during lecture time by calling each and every student or by passing the attendance sheet around the class. The lecturer then submits the student's attendance information through the online web-based attendance system and finally the students will get final attendance report using the system. During the manual system, the lecturer has many works to do especially when there are a large number of students in the classroom; like collecting, verifying, and managing students record. In reality, the manual system also takes more time for recording and calculating the average attendance of every student in the class. In addition to that, the automated system might give better benefits to the lecturer when we compare to the manual system.

2.3 DISADVANTAGES OF EXISTING SYSTEM:

- Attendance registers are prone to damage and being misplaced.
- It is very time consuming process.
- Inconsistency in data.

- Totally depends on the person assigned to maintain attendance.
- More complexity.

2.4 PROPOSED SYSTEM

The proposed system automates the manual attendance system using face recognition technology. The task of the proposed system is to capture the face of each student and to store them for their attendance. The face of the student needs to be captured in such a manner that all the features of the student's face needs to be detected. There is no need for the teacher to manually take attendance in the class because the system captures an image and through further processing steps the face is being recognized and the attendance is provided for the student.

In this project we address the issues which are encountered by a manual attendance system. We use Haar Cascade classifiers to detect faces and Local Binary Pattern Histogram(LBPH) algorithm to recognize the student's faces.

BLOCK DIAGRAM:

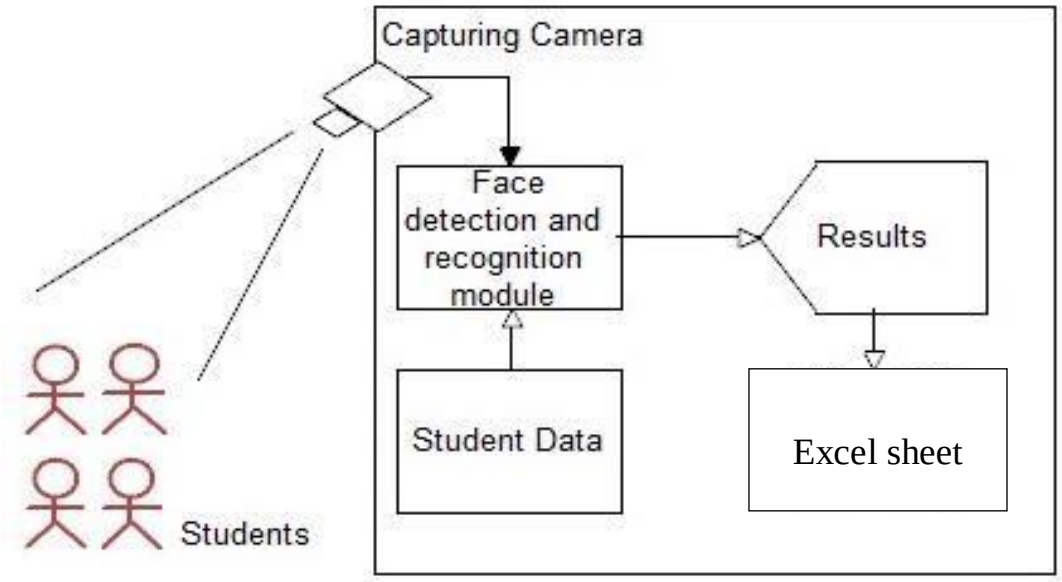


Fig. 2.1 Block diagram of proposed system

The block diagram describes the proposed system for Face Recognition based Classroom attendance system. The system requires a camera installed in the classroom at a position where it could capture all the students in the classroom and thus capture their images effectively. This image is processed to get the desired results. The working is explained in brief below:

- **Capturing Camera:**

Camera is installed in a classroom to capture the face of the student. The camera has to be placed such that it captures the face of all the students effectively. This camera has to be interfaced to computer system for further processing either through a wired or a wireless network. In our system we use the in-built camera of the laptop.

- **Processing:**

Facial recognition algorithm is applied on the captured image. The image is cropped and stored for processing. The module recognizes the images of the students face which have been stored manually with their ID codes in the dataset. The whole process requires the following steps:

a) Face detection and cropping : Initially we take facial image of the students. In our system we have taken 30 images each. This data is used later in the facial recognition algorithm. All the cropped image of the face is re sized to a 240 X 300 image.

The captured image of the classroom is initial scanned to detect faces. This is done using Haar Cascade classifier in Computer Vision library .This algorithm focuses more on speed and reliability. The detected faces are cropped and re-sized to a 240 X 300 image.

b) Training data set: In this process for each of the image the features are selected from the detected images and those features are converted into histograms using mathematical computations by LBPH algorithm.

c) Face recognition : Here, a student image is captured and again features are extracted from the image. Those features are converted into histograms and that histogram is compared with all other histograms which are created before. The face is recognized by comparing the distance between the histograms. The histogram with the closest distance is returned as a recognized image.

d) Attendance Recording: We use Excel spreadsheet to store the recorded attendance for easy-to-use output format, which is also the software which is familiar to majority of the institution staffs. This is done using Spreadsheet . If a student is recognized, the corresponding cell is updated with ‘yes’. Using the formatting in the Excel, we can effectively retrieve the information effectively.

ADVANTAGES OF PROPOSED SYSTEM:

- Reduces paperwork and saves time.
- Eliminates duplicate data entries.
- Easy attendance recording.
- Increased privacy and security where student cannot present himself or his friend while they are not.
- Reduces manual process errors and provides a reliable attendance system.

2.4 CONCLUSION

From the literature survey we can conclude that we have overcome the drawbacks of existing system and we introduce new technology which would be helpful to use . In proposed system to overcome that problem we create our project by providing sharing through e mail. This helps user to reduce time as well as work.

3. ANALYSIS

3.1 INTRODUCTION

The purpose of this SRS (software requirement specification) document is to mark and secure the required Uniform Resource locators (URL) and its functionalities for Intelligent Network Backup Tool. The SRS will define how our team and the client conceive the final product and the characteristics or functionality it must have.

Document Conventions:

The convention used in the size of fonts remains the same as for other documents in the project. The section headings have the largest font of 14, subheadings have a font size of 12(bold), and the text is on font 12. The priorities of the requirements are specified with the requirement statements.

Intended Audience and Reading Suggestions:

This document is intended for project developers, managers, users, testers and documentation writers. This document aims at discussing design and implementation constraints, dependencies, system features, external interface requirements and other non-functional requirements.

Identification of Needs:

The foremost and important necessity for a business firm or an organization is to know how they are performing in the market and parallel they need to know how to overcome their competitors in the market.

To do so we need to analysis our data based on all the available factors. The system requirements for the project to be accomplished are:

3.2 SOFTWARE REQUIREMENT SPECIFICATION

3.2.1 USER REQUIREMENTS

User can view the application and access the resources available in the application.

3.2.1 SOFTWARE REQUIREMENTS

1. Operating system : windows 10
2. Programing language : python
3. IDE : PyCharm
4. Libraries : OpenCV

3.3 HARDWARE REQUIREMENTS

1. Hard disk : 2GB (recommended 4GB) + 500MB for IDE
2. RAM : 3GB (recommended 8GB)
3. Integrated web camera

3.3 CONTENT DIAGRAM OF PROJECT

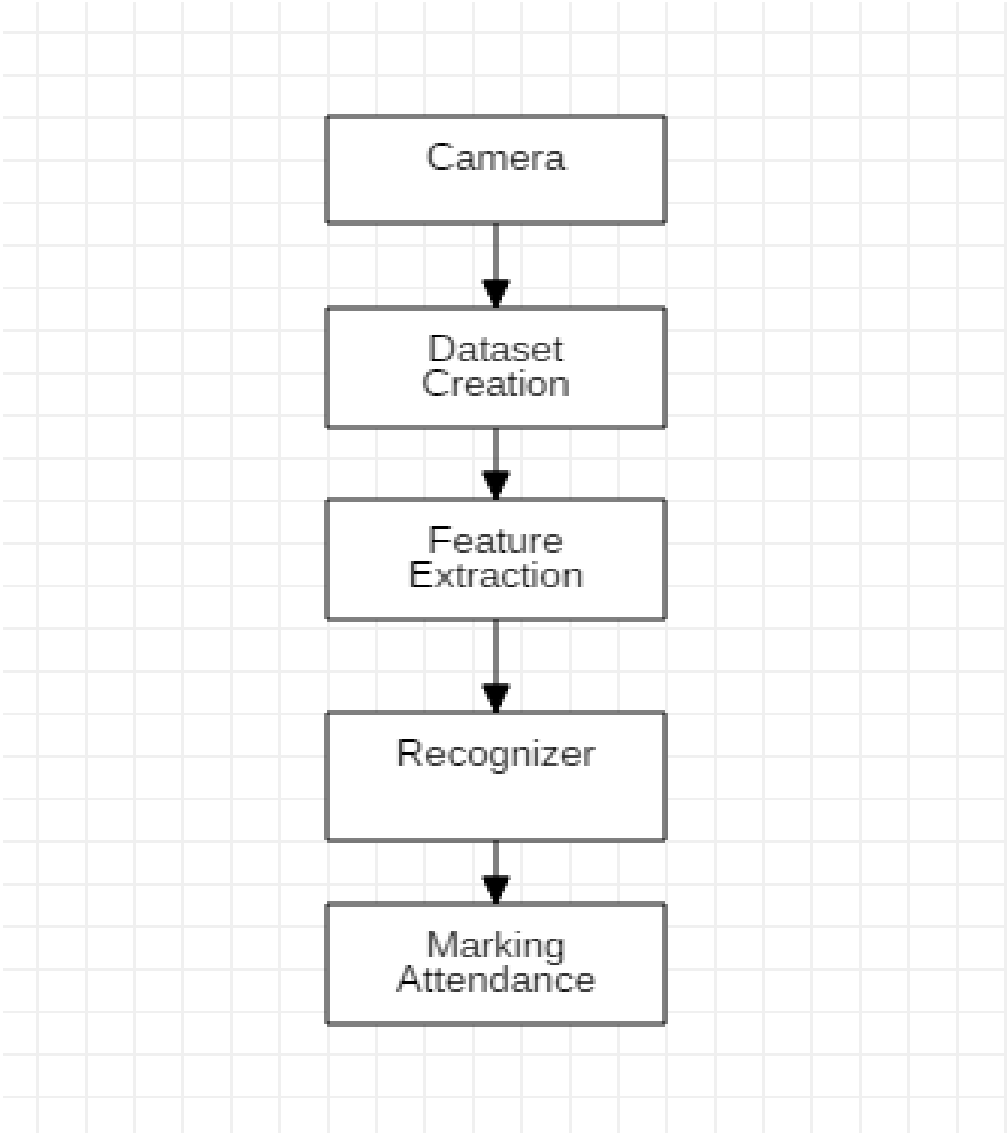


Fig: 3.1 Content Diagram

3.4 ALGORITHMS AND FLOWCHARTS

In this application we use Haar Cascade classifier to detect faces and LBPH algorithm to recognize faces.

1. Haar Cascade Classifier:

Face detection uses classifiers, which are algorithms that detects what is either a face(1) or not a face(0) in an image. It is a machine learning based approach where a cascade function is trained from a lot of positive(images of faces) and negative images(images without faces).

It is based on the Haar Wavelet technique to analyze pixels in the image into squares by function. This uses machine learning techniques to get a high degree of accuracy from what is called “training data”. This uses “integral image” concepts to compute the “features” detected. Haar Cascades use the Adaboost learning algorithm which selects a small number of important features from a large set to give an efficient result of classifiers**.**

Here we will deal with detection. OpenCV already contains many pre-trained classifiers for face, eyes, smile etc. Those XML files are stored in opencv/data/haarcascades/ folder.

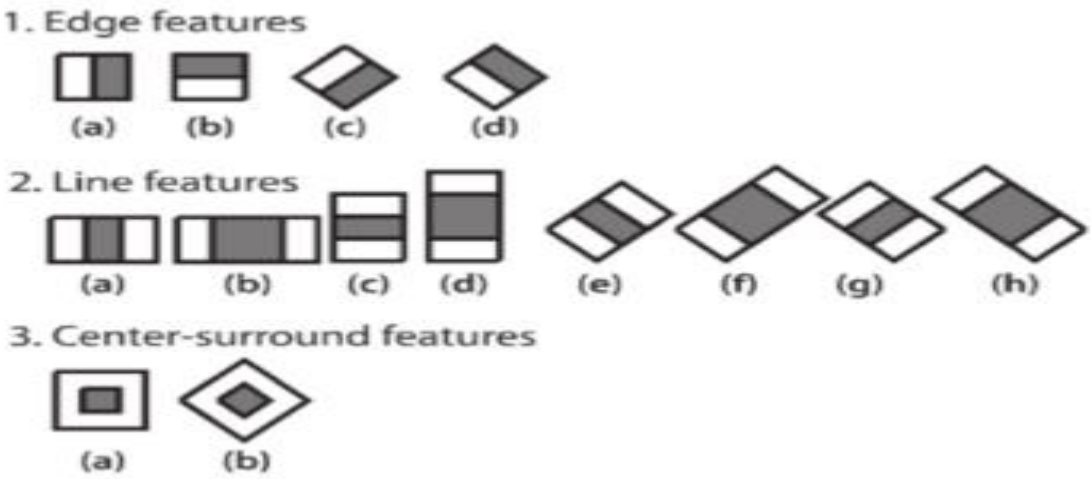


Fig. 3.2 Feature Extraction

In feature extraction, the algorithm uses training data to best identify features that it can consider a face.

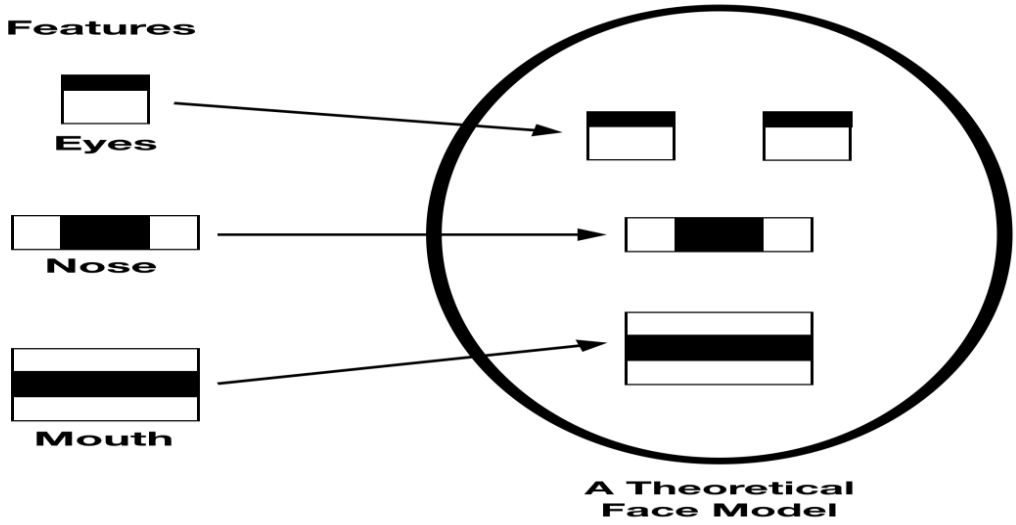


Fig.3.3 Face detection example

The training data used in this project is an XML file called:

haarcascade_frontalface_default.xml

2. Local Binary Pattern Histogram(LBPH) Algorithm:

It was first described in 1994 (LBP) and has since been found to be a powerful feature for texture classification. It has further been determined that when LBP is combined with histograms of oriented gradients (HOG) descriptor, it improves the detection performance considerably on some datasets.

Using the LBP combined with histograms we can represent the face images with a simple data vector.

Fig.3.4 General Face recognition structure

The above figure shows the input image that means captured by the camera that will detect the faces using face detection algorithm and convert the original image into a grayscale image for feature extraction. then we match the input image and current image by using verification and identification technique. we get good recognition result.

This algorithm is explained in the following steps:

Parameters: The LBPH algorithm uses 4 parameters:

- **Radius**: the radius is used to build the circular local binary pattern and represents the radius around the central pixel. It is usually set to 1.
- **Neighbors**: the number of sample points to build the circular local binary pattern. Keep in mind: the more sample points you include, the higher the computational cost. It is usually set to 8.
- **Grid X**: the number of cells in the horizontal direction. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector. It is usually set to 8.
- **Grid Y**: the number of cells in the vertical direction. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector. It is usually set to 8.

3. Training the Algorithm:

First, we need to train the algorithm. To do so, we need to use a dataset with the facial images of the people we want to recognize. We need to also set an ID (it may be a number or the name of the person) for each image, so the algorithm will use this information to recognize an input image and give you an output. Images of the same person must have the same ID. With the training set already constructed, let's see the LBPH computational steps.

4. Applying the LBP operation:

The first computational step of the LBPH is to create an intermediate image that describes the original image in a better way, by highlighting the facial characteristics.

To do so, the algorithm uses a concept of a sliding window, based on the parameters **radius** and **neighbors**.

The image below shows this procedure:

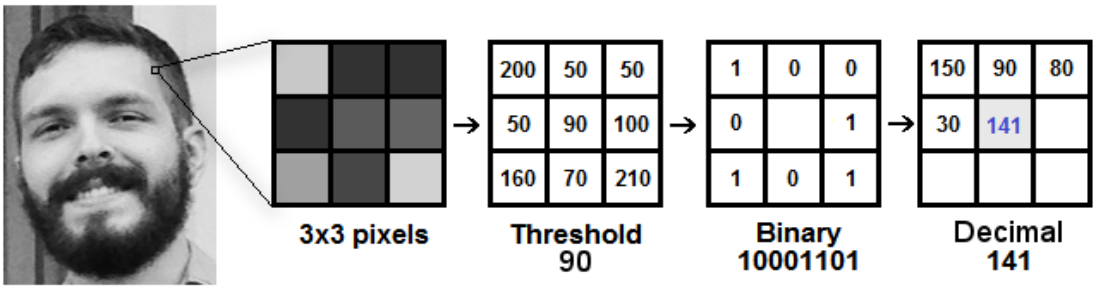


Fig. 3.5 LBPH operation

Based on the image above, let's break it into several small steps so we can understand it easily:

- Suppose we have a facial image in grayscale.
- We can get part of this image as a window of 3x3 pixels.
- It can also be represented as a 3x3 matrix containing the intensity of each pixel (0~255).
- Then, we need to take the central value of the matrix to be used as the threshold.
- This value will be used to define the new values from the 8 neighbors.
- For each neighbor of the central value (threshold), we set a new binary value. We set 1 for values equal or higher than the threshold and 0 for values lower than the threshold.

- Now, the matrix will contain only binary values (ignoring the central value). We need to concatenate each binary value from each position from the matrix line by line into a new binary value (e.g. 10001101). Note: some authors use other approaches to concatenate the binary values (e.g. clockwise direction), but the final result will be the same.

- Then, we convert this binary value to a decimal value and set it to the central value of the matrix, which is actually a pixel from the original image.

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

5. Extracting the Histograms:

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

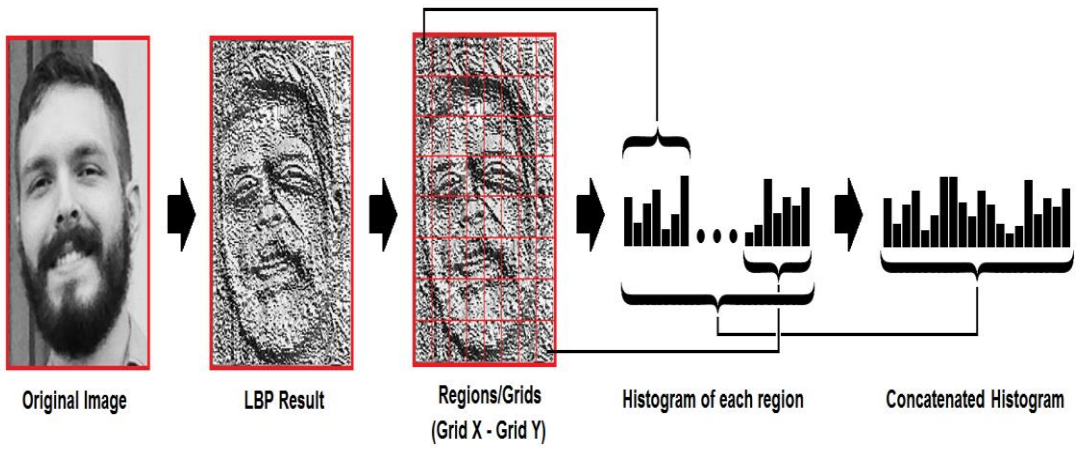


Fig.3.6Histogram creation

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

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

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

The LBPH algorithm is pretty much it.

6. Performing the face recognition:

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

- So to find the image that matches the input image we just need to compare two histograms and return the image with the closest histogram.
- We can use various approaches to compare the histograms (calculate the distance between two histograms), for example: **Euclidean distance**, **chi-square**, **absolute value**, etc. In this example, we can use the Euclidean distance (which is quite known) based on the following formula:

$$D = \sqrt{\sum_{i=1}^{n}(hist1_i - hist2_i)^2}$$

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

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

FLOWCHART:

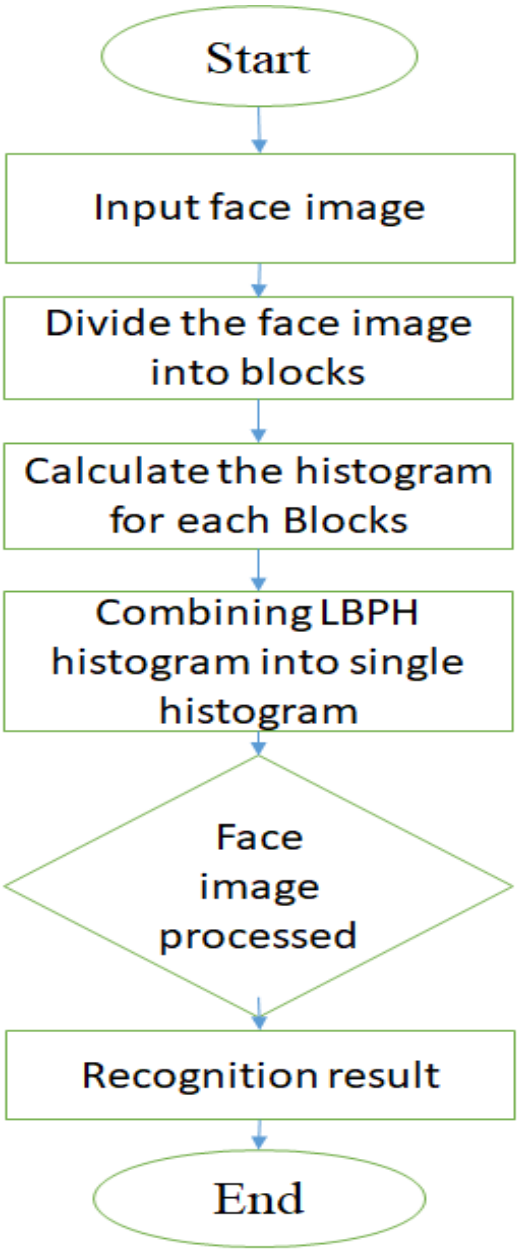


Fig .3.7 LBPH algorithm flowchart

4.1 CONCLUSION:

The analysis tells as the requirement specifications of the project. The functional requirements specify the functionality and functional requirements were as the software requirements tell the required software and supporting files to process the data. The hardware requirements tell about the hardware components required to run the software. Flowcharts describe the flow of the total process

4. DESIGN

4.1 INTRODUCTION

System Design is the process or art of defining the architecture components, modules, interfaces and data for a system to satisfy specified requirements. One should see as the applications of the systems

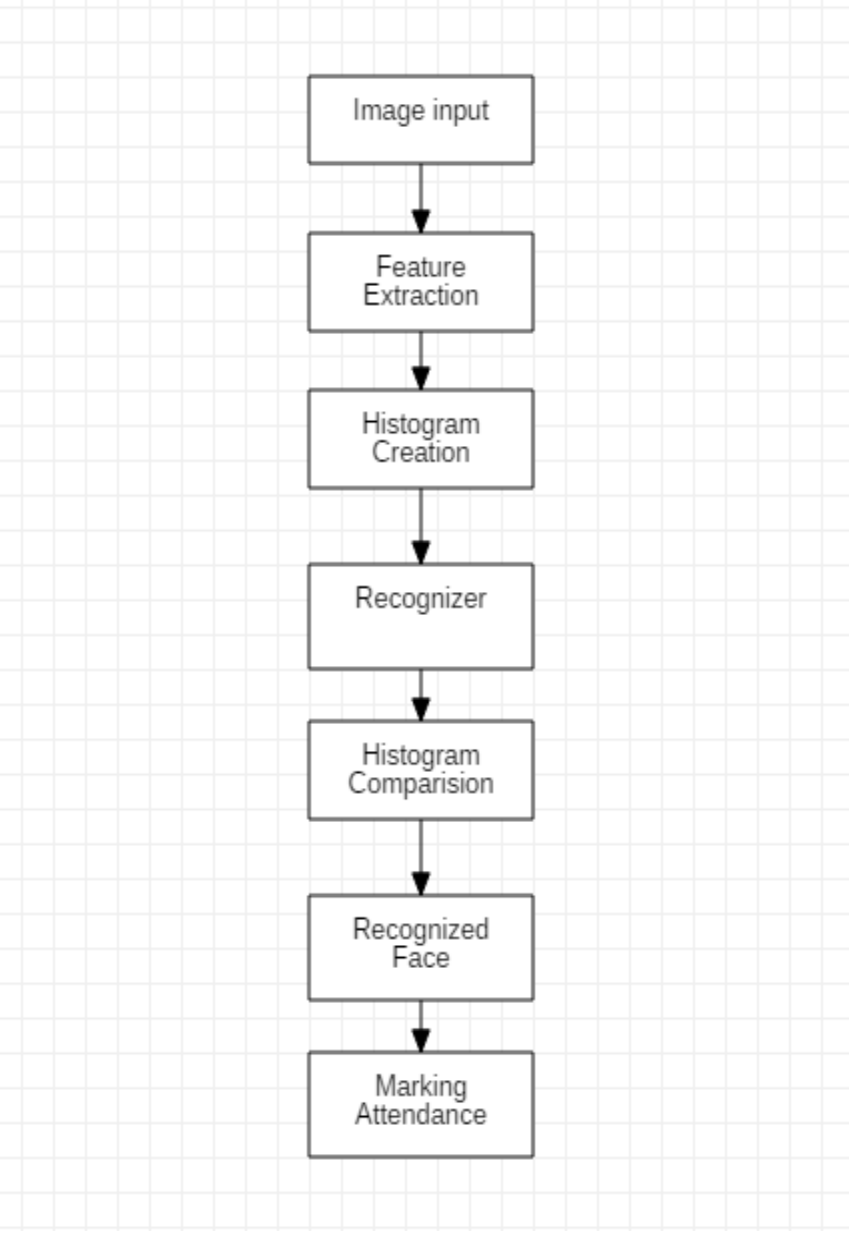


Fig: 4.1 System Design

4.2 UML DIAGRAMS:

Unified Modeling Language:

The Unified Modeling Language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic semantic and pragmatic rules.

A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows.

User Model View

i. This view represents the system from the user's perspective.

ii. The analysis representation describes a usage scenario from the end-users perspective.

Structural model view

i. In this model the data and functionality are arrived from inside the system.

ii. This model view models the static structures.

Behavioral Model View

It represents the dynamic of behavioral as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

Implementation Model View

In this the structural and behavioral as parts of the system are represented as they are to be built.

Environmental Model View

In this the structural and behavioral aspects of the environment in which the system

is to be implemented are UML is specifically constructed through two different domains they are:

- UML Analysis modeling, this focuses on the user model and structural model views of the system.
- UML design modeling, which focuses on the behavioral modeling, implementation modeling and environmental model views.
- UML is specifically constructed through two different domains they are:
- UML design modeling, which focuses on the behavioral modeling, implementation modeling and environmental model views.

Use case Diagrams represent the functionality of the system from a user's point of view. Use cases are used during requirements elicitation and analysis to represent the functionality of the system. Use cases focus on the behavior of the system from external point of view. Actors are external entities that interact with the system. Examples of actors include users like administrator, bank customer …etc., or another system like central database.

Use Case Diagram

A use case diagram at its simplest is a representation of a user's interaction with the system and depicting the specifications of the use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as well.

They provide the simplified and graphical representation of what the system must actually do. The purpose of use case diagram is to capture the dynamic aspect of a system. But this definition is too generic to describe the purpose. So we will look into some specific purpose which will distinguish it from other four diagrams.

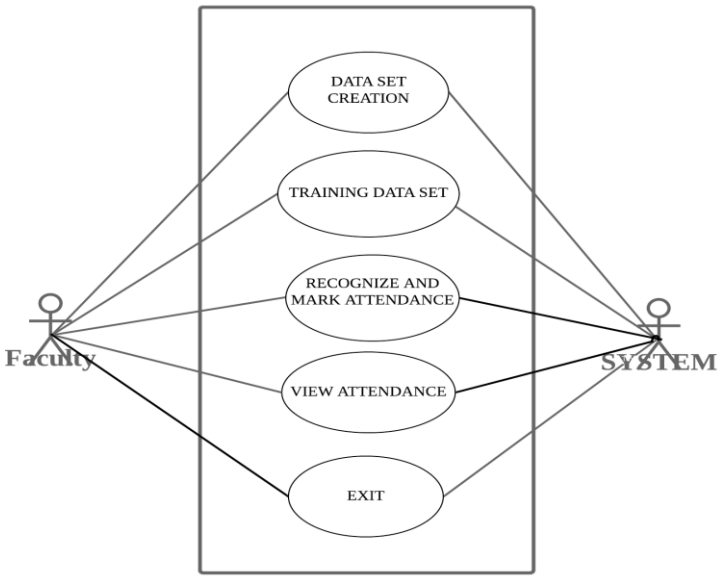


Fig.4.2 Use Case Diagram

Description:

1. **Actors:** Faculty, System
   - **Faculty:** Faculty is a person who maintains and monitors the system. He can perform all the actions which are included in the system.
   - **System:** Every action which is performed by the faculty is executed by the system.
2. **Use cases:** The actions that can be performed by the faculty are discussed below.
   - **Dataset Creation:** When this action is performed, immediately the camera starts capturing student images and they are stored in the system.
   - **Training Dataset:** In this action, the algorithm will be trained using the dataset of the images which are captured.
   - **Recognize and Mark Attendance:** In this action, the student's face is recognized and the attendance is marked in the Excel sheet.
   - **View Attendance:** In this action, the attendance is viewed by the faculty which is marked in the Excel sheet.
   - **Exit:** When this action is performed the application gets closed.

Sequence Diagram:

A Sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart.

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

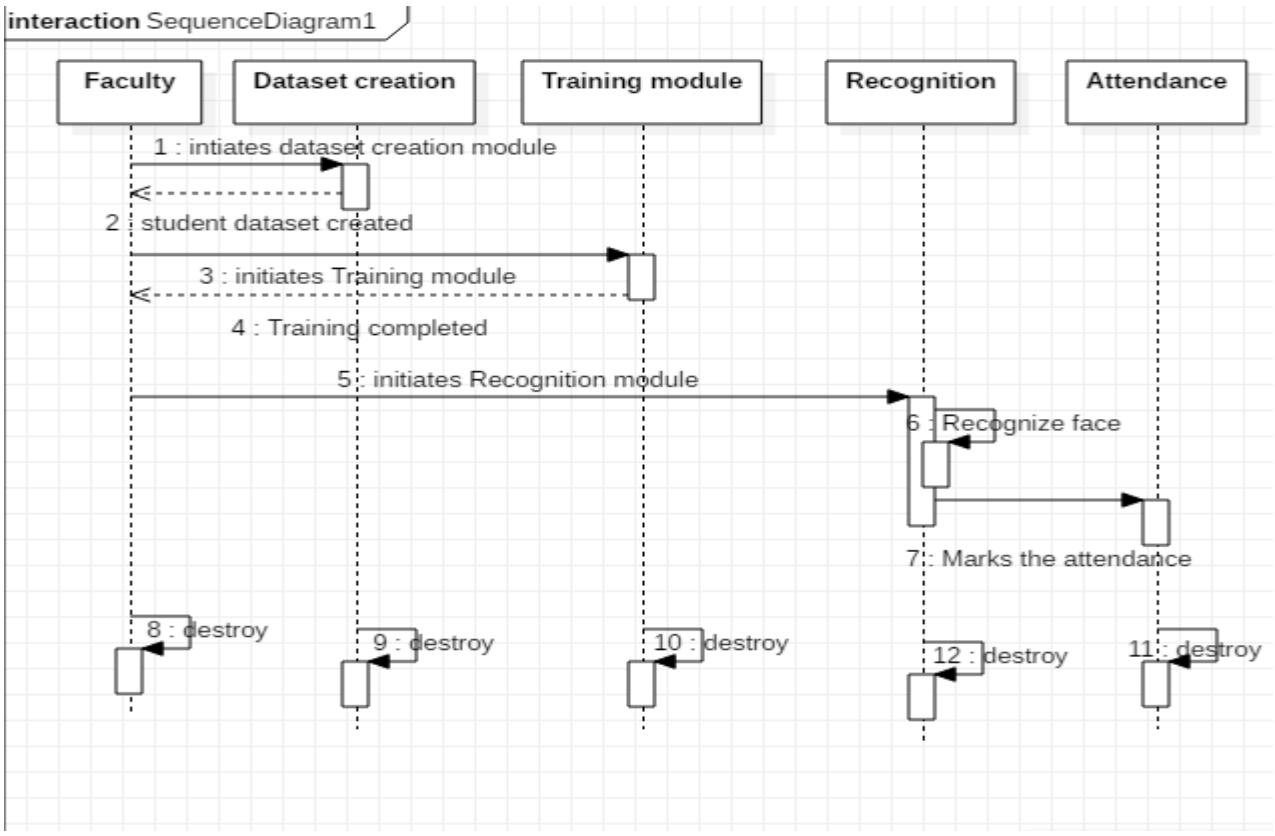


Fig.4.3 Sequence Diagram

Description:

1. Faculty initiates the dataset creation module.
2. Then the student dataset is created.
3. Faculty initiates the training module.
4. Then the algorithm is trained with the student dataset of images.
5. Faculty initiates the recognition module.
6. Then the algorithm predicts the student image as recognized image.
7. After recognition of student the attendance is then marked in the excel sheet

Class Diagram:

In software Engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects translating the models into programming code. Class diagrams can also be used for data modeling the classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed. The class diagram is the main building block of object oriented modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling.

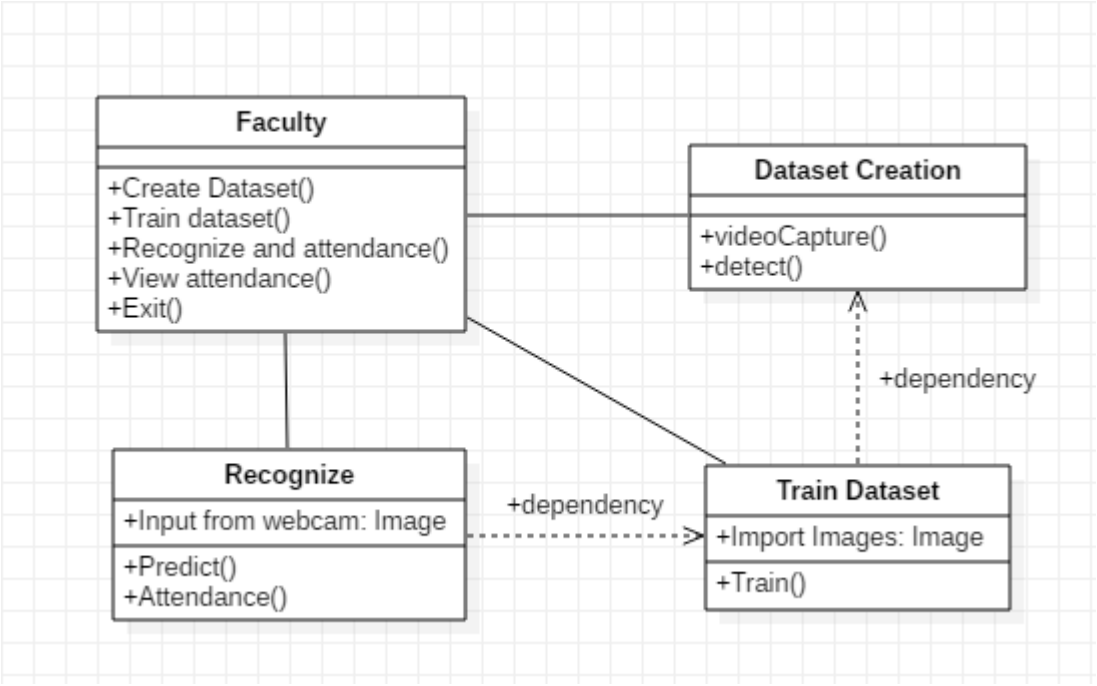


Fig.4.4 Class Diagram

Description

The above class diagram consists of 4 classes.

1. Faculty
2. Dataset Creation
3. Train Dataset
4. Recognize.

1. **Faculty:** The faculty class consists of operations that can be performed by the faculty.

   The operations are:

   - Create Dataset- The student dataset is created by this operation
   - Train Dataset - The algorithm is trained with student dataset by this operation.
   - Recognize and Attendance- The student is recognized and attendance is marked by this operation.
   - View Attendance- The attendance of the students is shown in the excel sheet by this operation.
   - Exit- The application gets closed by this operation.

2. **Dataset Creation:** This class consists of operations that can be used to create dataset of students. The operations are:

   - Video Capture- This operation makes the camera to record the video in order to take images of the students.
   - Detect- This operation detects the face of the student from the whole image.

3. **Train Dataset:** This class consists of attributes and operations which are required to train the algorithm.

   Attribute- images or dataset of students.

   Operation-Train

   In this operation the dataset is passed to this method and histograms are created

4. **Recognize:** This class consists of attributes and operations which are required to recognize the student's face.

   Attribute- It takes the input from the webcam to compare that input to the dataset of images.

   The operations are:

   - Predict- This operation predicts the student face as recognized face
   - Attendance- Once the student's face is recognized then this operation is performed

Activity Diagram

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flowis drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.

The basic purposes of activity diagrams are similar to other four diagrams. It captures the dynamic behavior of the system. Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part. It does not show any message flow from one activity to another. Activity diagram is sometimes considered as the flowchart. Although the diagrams look like a flowchart, they are not. It shows different flows such as parallel, branched, concurrent, and single.

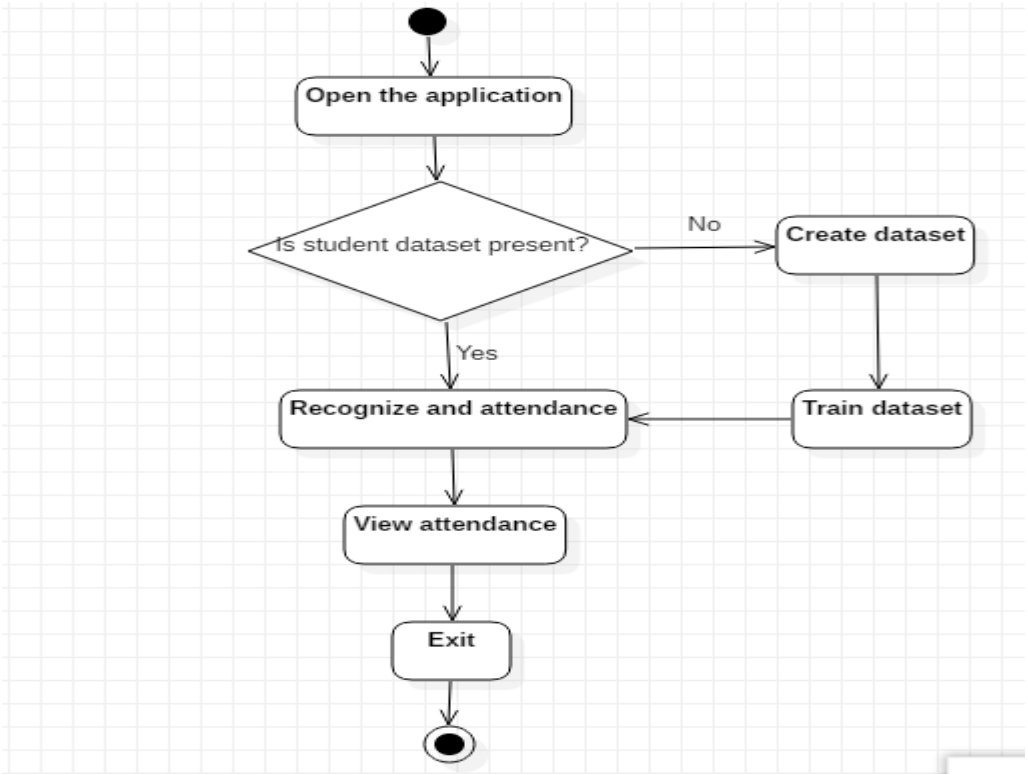


Fig.4.5 Activity Diagram

Description:

1. At first the faculty opens the application.
2. The faculty checks whether student's image is present in dataset.
3. If student image is not present, then the faculty creates the student dataset.
4. Then the algorithm is trained with student dataset and step 5 is performed.
5. If student dataset is present, then the faculty performs recognize and attendance activity.
6. The attendance is then marked if the student is recognized.
7. The attendance is then viewed by the faculty.

4.3 CONCLUSION:

By design content we can describe the required modules and different diagrams. Using diagrams what are the communications present and we can also understand the project easily. Modules help as in designing the project to fulfil the user requirements.

5. IMPLEMENTATION AND RESULT

5.1 INTRODUCTION

Implementation is the carrying out, execution, or practice of a plan, a method, or any design, idea, model, specifications, standard or policy for doing something. As such, implementation is the action that must follow any preliminary thinking in order for something to actually happen.

For an implementation process to be successful, many tasks between different departments need to be accomplished in sequence. Companies strive to use proven methodologies and enlist professional help to guide them through the implementation of a system but the failure of many implementation processes obtain stems from the lack of accurate planning in the beginning stage of project.

5.2 EXPLANATION OF KEY ELEMENTS:

1. PYTHON:

Python is an interpreted, high-level, general purpose programming language created by Guido Van Rossum and first released in 1991.

The features of Python language are:

- **Easy to Learn and Use**

Python is easy to learn and use. It is developer-friendly and high level programming language.

- **Expressive Language**

Python language is more expressive means that it is more understandable and readable.

- **Interpreted Language**

Python is an interpreted language i.e. interpreter executes the code line by line at a time. This makes debugging easy and thus suitable for beginners.

- **Cross-platform Language**

Python can run equally on different platforms such as Windows, Linux, Unix and Macintosh etc. So, we can say that Python is a portable language.

- **Free and Open Source**

Python language is freely available at official web address. The source-code is also available. Therefore it is open source.

- **Object-Oriented Language**

Python supports object oriented language and concepts of classes and objects come into existence.

- **Extensible**

It implies that other languages such as C/C++ can be used to compile the code and thus it can be used further in our python code.

- **Large Standard Library**

Python has a large and broad library and prvides rich set of module and functions for rapid application development.

- **GUI Programming Support**

Graphical user interfaces can be developed using Python.

- **Integrated**

It can be easily integrated with languages like C, C++, JAVA etc.

2. PYCHARM IDE:

PyCharm is an Integrated Development Environment **(**IDE**)** used in computer programming, specifically for the Python language. It is developed by the Czech company Jet Brains. It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCSes).

PyCharm Community Edition is totally free and open-source, available under the Apache 2.0 license. The feature set of this edition is limited to support pure Python coding, while the major functionality and complementary tooling is still there. Community Edition provides core Python language support with code completion, one-the-fly code analysis, refactoring, local debugger, test runner, virtualenv, version control integrations, etc.

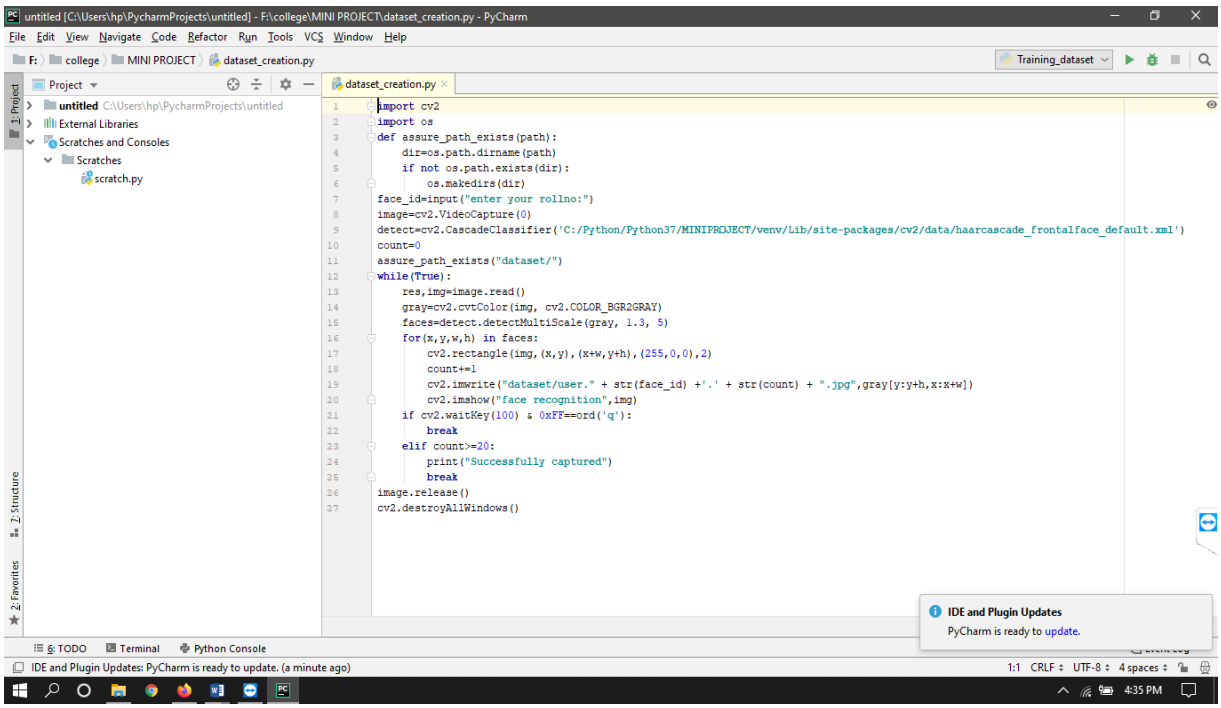


Fig.5.1 PyCharm IDE

3. OPEN CV:

OpenCV (Open source computer vision) is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage then Itseez (which was later acquired by Intel). The library is cross-platform and free for use under the open-source BSD license. OpenCV-Python is a library of Python bindings designed to solve computer vision problems.

Install openCV package through PIP which is a widely used package management system to install and manage software packages written in Python.

If you have Python version 3.4 or higher PIP is included by default.

OpenCV is written in C++ and its primary interface is in C++, but it still retains a less comprehensive though extensive older C interface.

Currently OpenCV supports a wide variety of programming languages like C++, Python, Java etc. and is available on different platforms including Windows, Linux, OS X, Android, iOS etc.

Install openCV by **pip install openCV-python** command in PyCharm IDE terminal

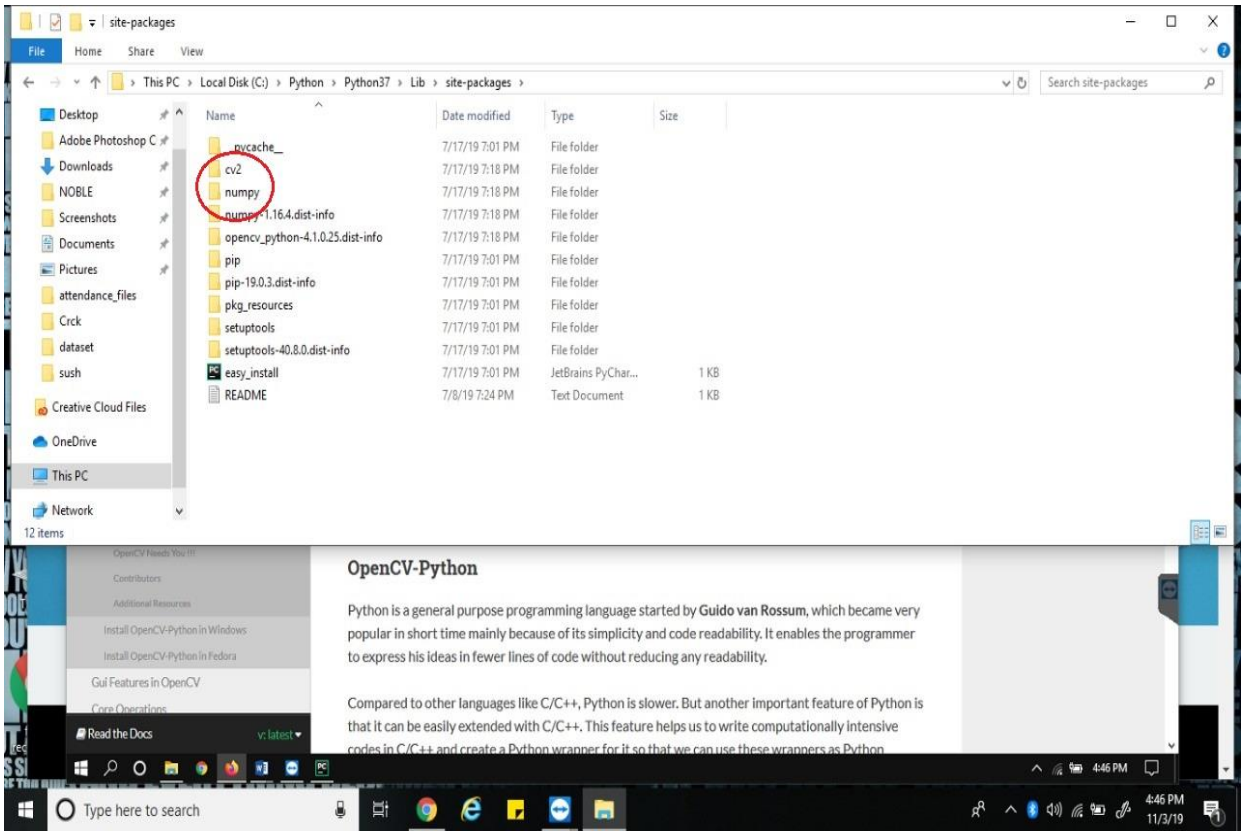


Fig. 5.2 OpenCV folder

4. NUMPY:

NumPy is Python package which stands for 'Numerical Python'. It is the core library for scientific computing. The Numpy is a multi-dimensional array used to store values of same data type. It is used to store complex mathematical data in the form of arrays for computations.

It contains among other things:

1. a powerful N-dimensional array object
2. sophisticated (broadcasting) functions
3. tools for integrating C/C++ and Fortran code
4. useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

Numpy data structures takes up less space. Numpy package is automatically installed with OpenCV package installation.

5. PIL (Python Imaging Library):

Python Imaging Library is a free library for the Python programming language that adds support for opening, manipulating, and saving many different file formats.

PIL offers several standard procedures for image manipulation. These include:

- per-pixel manipulations,
- masking and transparency handling,
- image filtering, such as blurring, contouring, smoothing, or edge finding,
- image enhancing, such as sharpening, adjusting brightness, contrast or color,
- adding text to images and much more.

XLWT is a library for developers to use to generate spreadsheet files compatible with Microsoft Excel versions 95 to 2003. The package itself is pure Python with no dependencies on modules or packages outside the standard Python distribution.

6. TKINTER:

Tkinter module is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

The Tk interface is located in a binary module named _Tkinter. This module contains the low-level interface to Tk, and should never be used directly by application programmers. It is usually a shared library (or DLL), but might in some cases be statically linked with the Python interpreter.

Creating a GUI application using Tkinter is an easy task.

1. Import the Tkinter module.

2. Create the GUI application main window.out

3. Add one or more widgets like button, checkbox etc. to the GUI application.

4. Enter the main event loop to take action against each event triggered by the user.

To install tkinter module the command **pip install tkinter** is used in PyCharm terminal

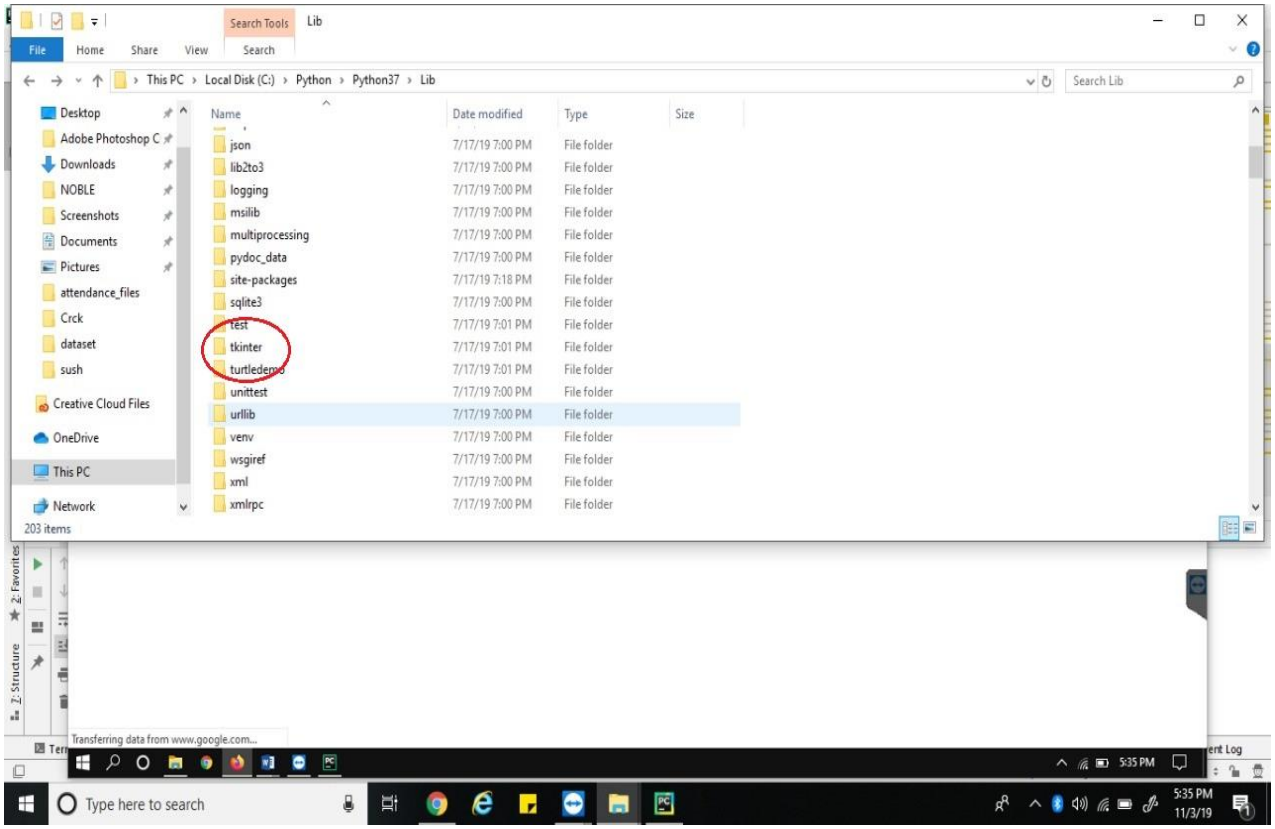


Fig. 5.3 showing tkinter library

5.4 METHOD OF IMPLEMENTATION :

The LBPH algorihtm is provided by the OpenCV library. We have used OpenCV which presents a Haar cascade classifier which is used for face detection. The Haar cascade classifier uses the LBPH algorithm.

Let see the implementation of face recognition in a step by step.

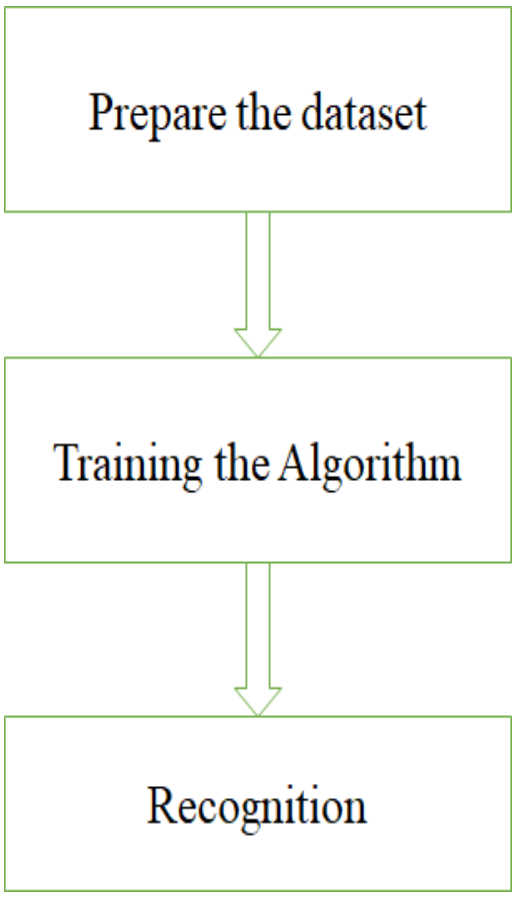


Fig .5.4 code implementation process

5.5 CODE IMPLEMENTATION

5.5.1 SOURCE CODE

1. Dataset_creation.py

```
import cv2
import os

def assure_path_exists(path):
    dir = os.path.dirname(path)
    if not os.path.exists(dir):
        os.makedirs(dir)

face_id = input('enter your id:')
# Start capturing video
vid_cam = cv2.VideoCapture(0)

# Detect object in video stream using Haarcascade Frontal Face
face_detector = cv2.CascadeClassifier('C:/Python/Python37/MINIPROJECT/venv/Lib/site-packages/cv2/data/haarcascade_frontalface_default.xml')

# Initialize sample face image
count = 0

assure_path_exists("dataset/")

# Start looping
while (True):

    # Capture video frame
    _, image_frame = vid_cam.read()

    # Convert frame to grayscale
```

```
    gray = cv2.cvtColor(image_frame, cv2.COLOR_BGR2GRAY)

    # Detect frames of different sizes, list of faces rectangles
    faces = face_detector.detectMultiScale(gray, 1.3, 5)
    # Loops for each faces
    for (x, y, w, h) in faces:
        # Crop the image frame into rectangle
        cv2.rectangle(image_frame, (x, y), (x + w, y + h), (255, 0, 0), 2)
        # Increment sample face image
        count += 1

        # Save the captured image into the datasets folder
        cv2.imwrite("dataset/User." + str(face_id) + '.' + str(count) + ".jpg", gray[y:y + h, x:x +
w])

        # Display the video frame, with bounded rectangle on the person's face
        cv2.imshow('frame', image_frame)

    # To stop taking video, press 'q' for at least 100ms
    if cv2.waitKey(100) & 0xFF == ord('q'):
        break

    # If image taken reach 100, stop taking video
    elif count >= 30:
        print("Successfully Captured")
        break

# Stop video
vid_cam.release()

# Close all started windows
cv2.destroyAllWindows()
```

2. training_dataset.py:

```
import os,cv2;
import numpy as np
from PIL import Image;
recognizer = cv2.face.LBPHFaceRecognizer_create()
detector= cv2.CascadeClassifier("haarcascade_frontalface_default.xml");
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
    faceSamples=[]
    #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=detector.detectMultiScale(imageNp)
        #If a face is there then append that in the list as well as Id of it
        for (x,y,w,h) in faces:
            faceSamples.append(imageNp[y:y+h,x:x+w])
            Ids.append(Id)
    return faceSamples,Ids
faces,Ids = getImagesAndLabels('dataSet')
s = recognizer.train(faces, np.array(Ids))
print("Successfully trained")
recognizer.write('trainer/trainer.yml')
```

3.recognizer.py:

```
import cv2, numpy as np;
import xlwrite
import time
import sys

start = time.time()
period = 8
face_cas = cv2.CascadeClassifier('C:/Python/Python37/MINIPROJECT/venv/Lib/site-packages/cv2/data/haarcascade_frontalface_default.xml')
cap = cv2.VideoCapture(0);
recognizer = cv2.face.LBPHFaceRecognizer_create();
recognizer.read('trainer/trainer.yml');
flag = 0;
id = 0;
filename = 'filename';
dict = {
    'item1': 1
}
# font = cv2.InitFont(cv2.cv.CV_FONT_HERSHEY_SIMPLEX, 5, 1, 0, 1, 1)
font = cv2.FONT_HERSHEY_SIMPLEX
while True:
    ret, img = cap.read();
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY);
    faces = face_cas.detectMultiScale(gray, 1.3, 7);
    for (x, y, w, h) in faces:
        roi_gray = gray[y:y + h, x:x + w]
        cv2.rectangle(img, (x, y), (x + w, y + h), (255, 0, 0), 2);
        id, conf = recognizer.predict(roi_gray)
        if (conf < 80):
            if((str(id)) not in dict):
                filename = xlwrite.output('attendance', 'class1',1, id, 'yes');
```

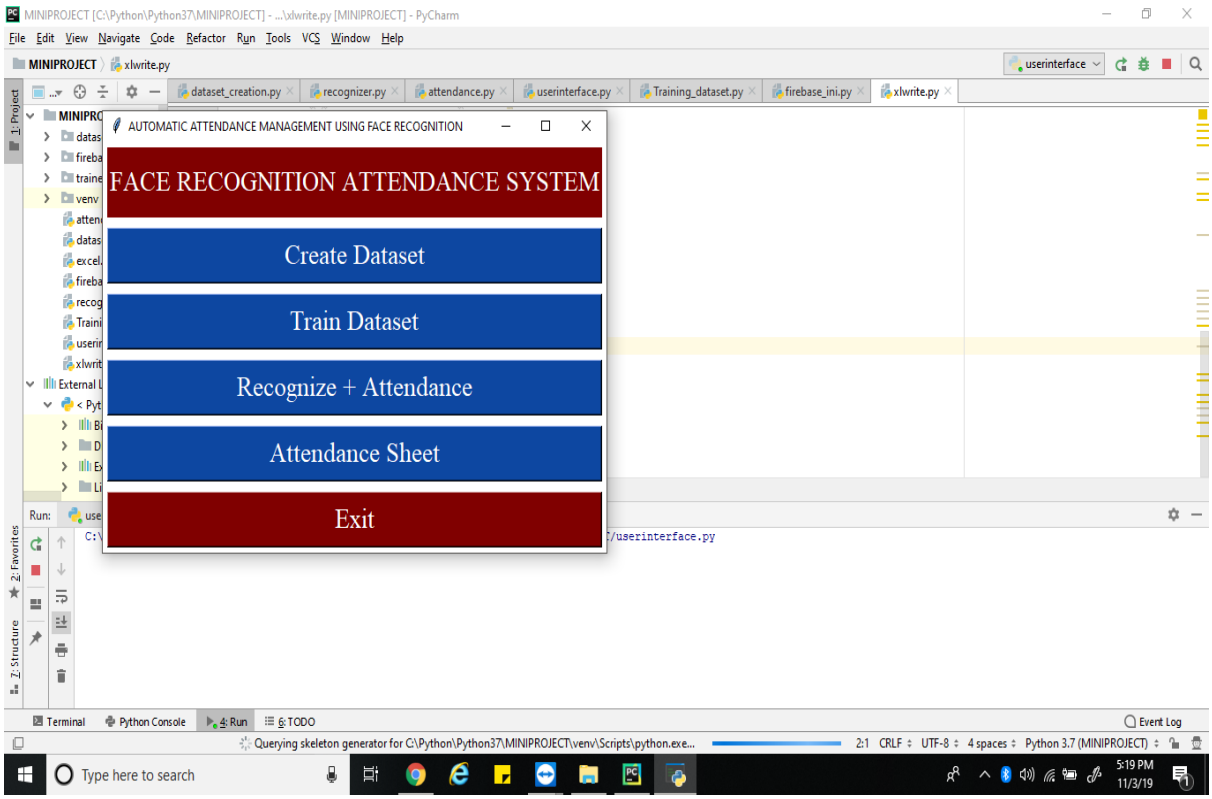
```
                dict[str(id)] = str(id);
            else:
            id = 'Unknown, can not recognize'
            flag = flag + 1
            break
        cv2.putText(img, str(id) + " " + str(conf), (x, y - 10), font, 0.55, (120, 255, 120), 1)
        # cv2.cv.PutText(cv2.cv.fromarray(img),str(id),(x,y+h),font,(0,0,255));
    cv2.imshow('frame', img);
    # cv2.imshow('gray',gray);
    if flag == 10:
        print("Transaction Blocked")
        break;
    if time.time() > start + period:
        break;
    if cv2.waitKey(100) & 0xFF == ord('q'):
        break;

cap.release();
cv2.destroyAllWindows();
```

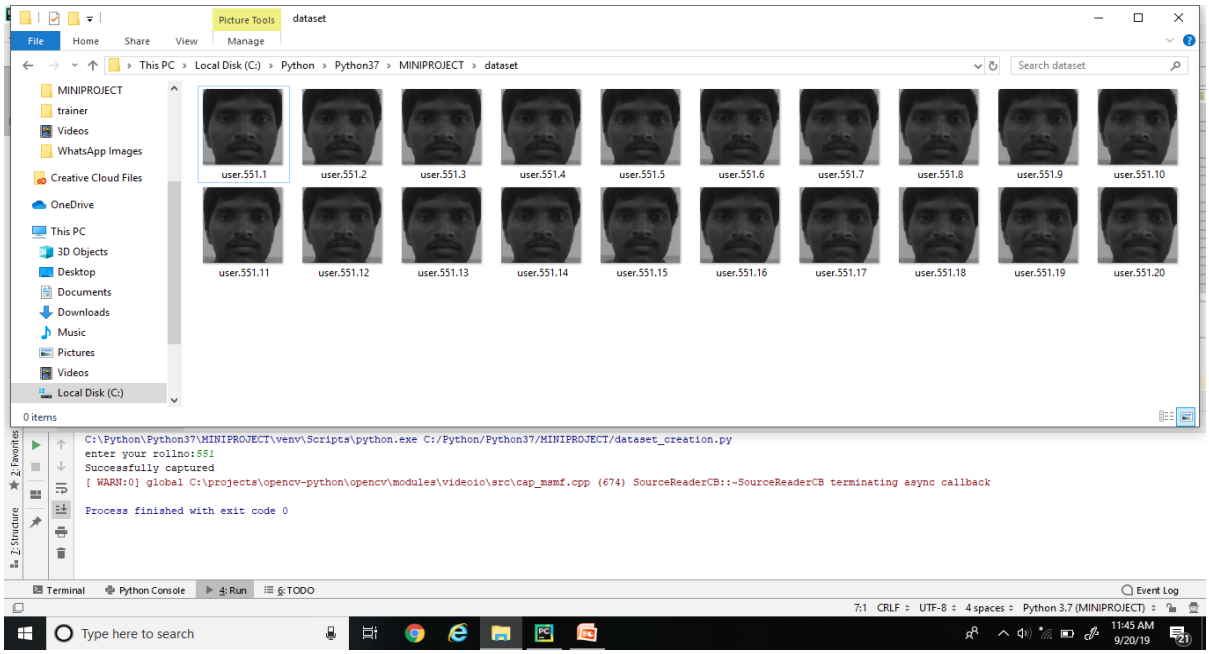
xlwrite.py:

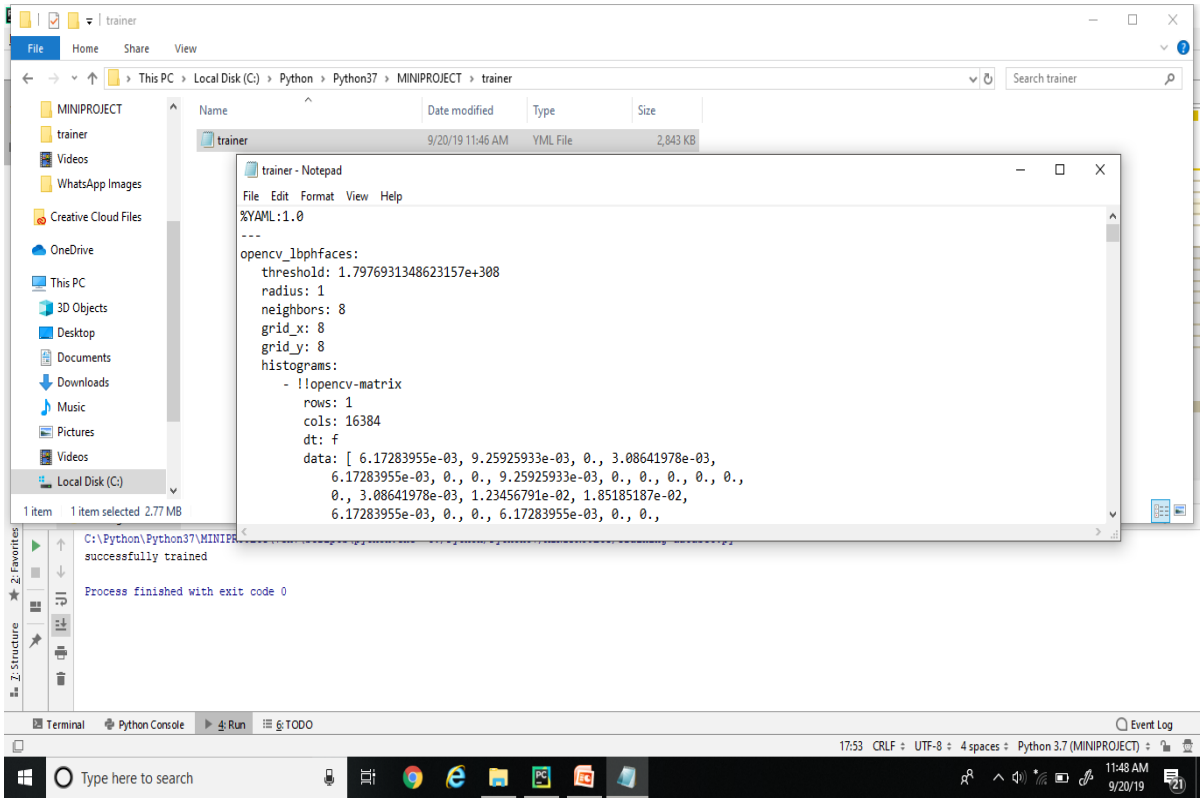
```
import xlwt;
from datetime import datetime;
from xlrd import open_workbook;
from xlwt import Workbook;
from xlutils.copy import copy
from pathlib import Path

def output(filename, sheet,num, name, present):
    my_file = Path('firebase/attendance_files/'+filename+str(datetime.now().date())+'.xls');
    if my_file.is_file():
        rb = open_workbook('firebase/attendance_files/' + filename +
str(datetime.now().date()) + '.xls');
        book = copy(rb);
```

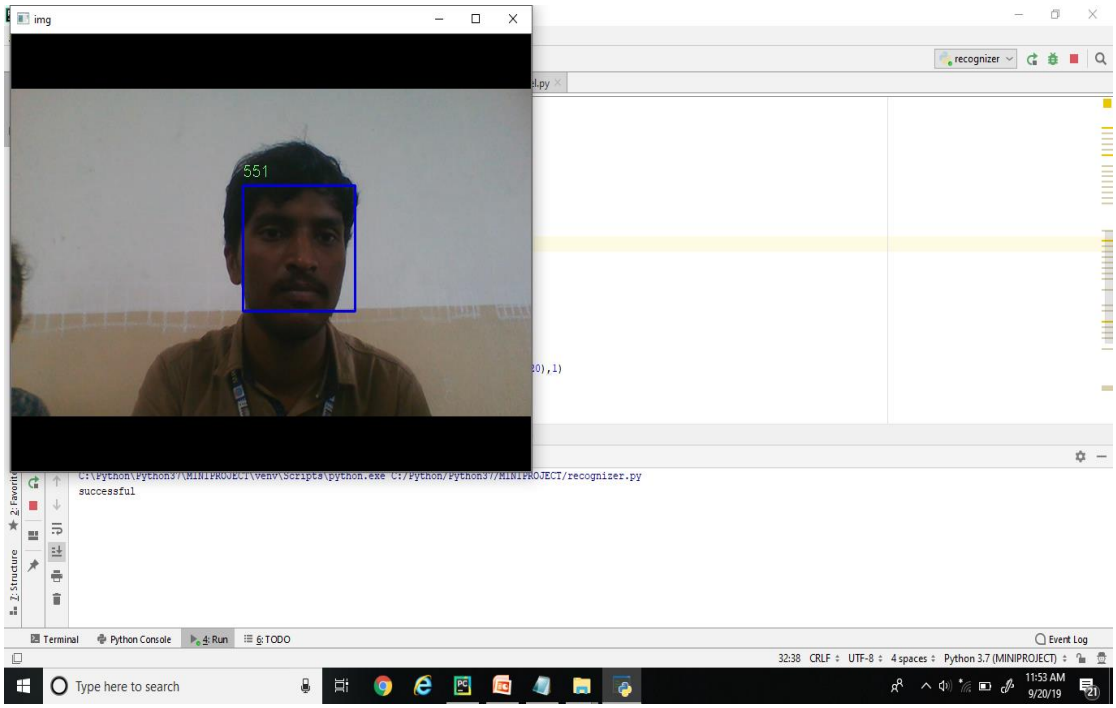
```
    sh = book.get_sheet(0)
    # file exists


else:
    book = xlwt.Workbook()
    sh = book.add_sheet(sheet)

style0 = xlwt.easyxf('font: name Times New Roman, color-index red, bold on',
                num_format_str='#,##0.00')
style1 = xlwt.easyxf(num_format_str='D-MMM-YY')

#variables = [x, y, z]
#x_desc = 'Display'
#y_desc = 'Dominance'
#z_desc = 'Test'
#desc = [x_desc, y_desc, z_desc]
sh.write(0,0,datetime.now().date(),style1);


col1_name = 'RollNo'
col2_name = 'Present'


sh.write(1,0,col1_name,style0);
sh.write(1, 1, col2_name,style0);

sh.write(num+1,0,name);
sh.write(num+1, 1, present);
#You may need to group the variables together
#for n, (v_desc, v) in enumerate(zip(desc, variables)):
fullname=filename+str(datetime.now().date())+'.xls';
book.save('firebase/attendance_files/'+fullname)
return fullname;
```

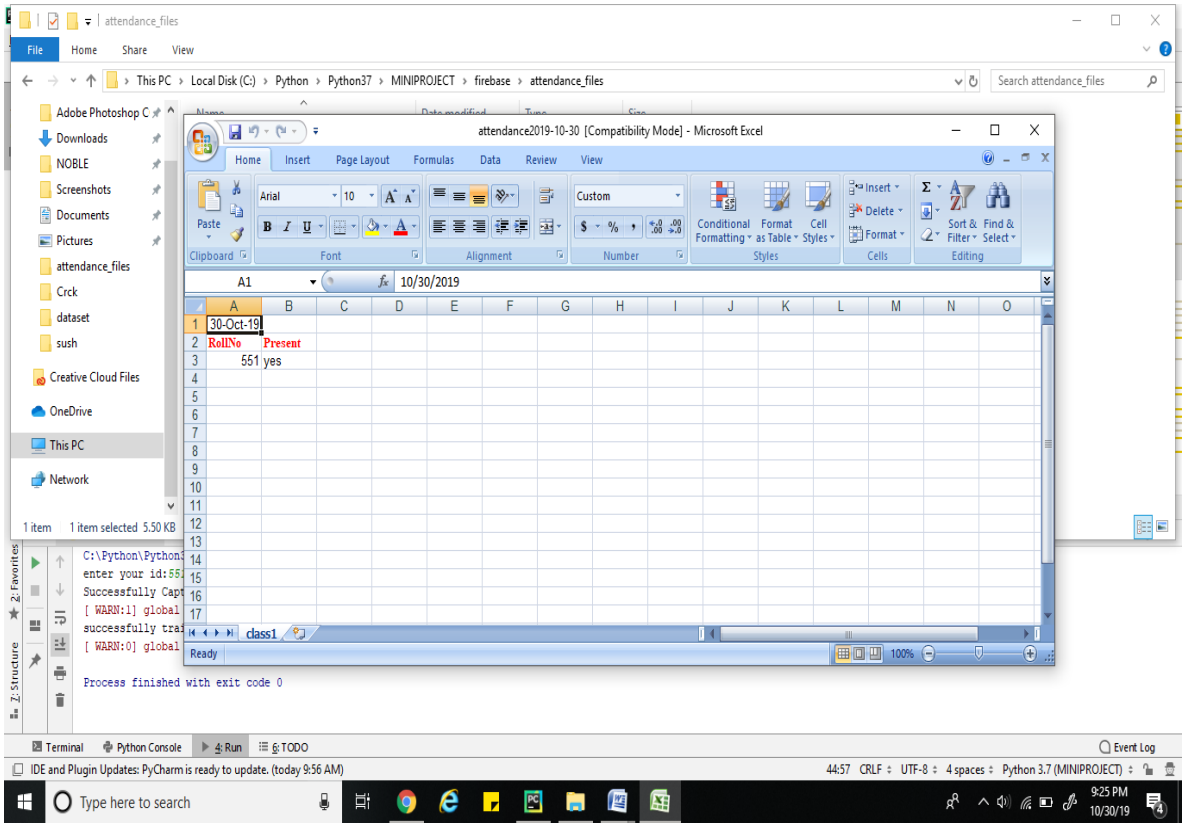
5.5.2 OUTPUT SCREENS




Picture Tools
dataset
File
Home
Share
View
Manage
This PC › Local Disk (C:) › Python › Python37 › MINIPROJECT › dataset
Search dataset
MINIPROJECT
trainer
Videos
WhatsApp Images
Creative Cloud Files
OneDrive
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
user.551.1
user.551.2
user.551.3
user.551.4
user.551.5
user.551.6
user.551.7
user.551.8
user.551.9
user.551.10
user.551.11
user.551.12
user.551.13
user.551.14
user.551.15
user.551.16
user.551.17
user.551.18
user.551.19
user.551.20
0 items
C:\Python\Python37\MINIPROJECT\venv\Scripts\python.exe C:/Python/Python37/MINIPROJECT/dataset_creation.py
enter your rollno:551
Successfully captured
[WARN:0] global C:\projects\opencv-python\opencv\modules\videoio\src\cap_msmf.cpp (674) SourceReaderCB::~SourceReaderCB terminating async callback
Process finished with exit code 0
Terminal
Python Console
4: Run
6: TODO
Event Log
7:1
CRLF
UTF-8
4 spaces
Python 3.7 (MINIPROJECT)
Type here to search
11:45 AM
9/20/19


trainer
File Home Share View
This PC › Local Disk (C:) › Python › Python37 › MINIPROJECT › trainer
Search trainer
MINIPROJECT
trainer
Videos
WhatsApp Images
Creative Cloud Files
OneDrive
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
Name
Date modified
Type
Size
trainer 9/20/19 11:46 AM YML File 2,843 KB
1 item 1 item selected 2.77 MB
trainer - Notepad
File Edit Format View Help
%YAML:1.0
---
opencv_lbphfaces:
threshold: 1.7976931348623157e+308
radius: 1
neighbors: 8
grid_x: 8
grid_y: 8
histograms:
- !!opencv-matrix
rows: 1
cols: 16384
dt: f
data: [6.17283955e-03, 9.25925933e-03, 0., 3.08641978e-03,
6.17283955e-03, 0., 0., 9.25925933e-03, 0., 0., 0., 0., 0.,
0., 3.08641978e-03, 1.23456791e-02, 1.85185187e-02,
6.17283955e-03, 0., 0., 6.17283955e-03, 0., 0.,
C:\Python\Python37\MINIPR
successfully trained
Process finished with exit code 0
2: Favorites
Z: Structure
Terminal
Python Console
4: Run
6: TODO
Event Log
17:53 CRLF UTF-8 4 spaces Python 3.7 (MINIPROJECT)
Type here to search
11:48 AM
9/20/19


img
551
recognizer
successful
Terminal
Python Console
4: Run
6: TODO
Event Log
Type here to search
11:53 AM
9/20/19


attendance_files
File Home Share View
This PC › Local Disk (C:) › Python › Python37 › MINIPROJECT › firebase › attendance_files
Search attendance_files
Adobe Photoshop C
Downloads
NOBLE
Screenshots
Documents
Pictures
attendance_files
Crck
dataset
sush
Creative Cloud Files
OneDrive
This PC
Network
1 item
1 item selected 5.50 KB
attendance2019-10-30 [Compatibility Mode] - Microsoft Excel
Home Insert Page Layout Formulas Data Review View
Paste
Clipboard
Font
Alignment
Number
Conditional Formatting
Format as Table
Cell Styles
Styles
Insert
Delete
Format
Cells
Sort & Filter
Find & Select
Editing
A1
10/30/2019
30-Oct-19
RollNo
Present
551 yes
class1
Ready
100%
C:\Python\Python3
enter your id:55
Successfully Capt
[WARN:1] global
successfully trai
[WARN:0] global
Process finished with exit code 0
Terminal
Python Console
4: Run
6: TODO
Event Log
IDE and Plugin Updates: PyCharm is ready to update. (today 9:56 AM)
44:57 CRLF UTF-8 4 spaces Python 3.7 (MINIPROJECT)
Type here to search
9:25 PM
10/30/19

5.6 CONCLUSION

At the time of the beginning of the development of this project we had kept certain goals in the mind, and it is great pleasure that the system is meeting most of its requirements. In this project report we have mentioned all the details of the system, which includes all the stages of the system.

The goals that are expected to achieve by the software are: Avoiding the errors by minimizing human interaction through user friendly screens to enter data and retrieve the information from the application. Portable and flexible for further enhancement.

6. TESTING AND VALIDATION

6.1 INTRODUCTION

Application Testing is the process used to help identify the correctness, completeness, security, and quality of developed user application. Testing is a process of technical investigation, performed on behalf of stakeholders, that is intended to reveal quality-related information about the product with respect to the context in which it is intended to operate. This includes, but is not limited to, the process of executing a program or application with the intent of finding errors. Quality is not an absolute; it is value to some person. With that in mind, testing can never completely establish the correctness of arbitrary computer software; testing furnishes a criticism or comparison that compares the state and behaviour of the product against a specification. An important point is that software testing should be distinguished from the separate discipline of Software Quality Assurance (SQA), which encompasses all business process areas, not just testing.

Testing levels

Unit testing tests the minimal software component and sub-component or modules by the programmers.

- Integration testing exposes defects in the interfaces and interaction between integrated components (modules).
- Functional testing tests the product according to programmable work. System testing tests an integrated system to verify/validate that it meets its requirements.
- Acceptance testing can be conducted by the client. It allows the end-user or customer or client to decide whether or not to accept the product. Acceptance testing may be performed after the testing and before the implementation phase. See also Development stage
- Beta testing comes after alpha testing. Versions of the software, known as beta versions, are released to a limited audience outside of the company. The software is released to groups of people so that further testing can ensure the product has few faults or bugs. Sometimes, beta versions are made available to the open public to increase the feedback field to a maximal number of future users.

Test cases, suites, scripts and scenarios:

A test case is a software testing document, which consists of event, action, input, output, expected result and actual result. Clinically defined (IEEE 829-1998) a test case is an input and an expected result. This can be as pragmatic as 'for condition x your derived result is y', whereas other test cases described in more detail the input scenario and what results might be expected. It can occasionally be a series of steps (but often steps are contained in a separate test procedure that can be exercised against multiple test cases, as a matter of economy) but with one expected result or expected outcome.

The optional fields are a test case ID, test step or order of execution number, related requirement(s), depth, test category, author, and check boxes for whether the test is automatable and has been automated. A test case should also contain a place for the actual result. These steps can be stored in a word processor document, spread sheet, database or other common repository. In a database system, you may also be able to see past test results and who generated the results and the system configuration used to generate those results. These past results would usually be stored in a separate table.

The term test script is the combination of a test case, test procedure and test data. Initially the term was derived from the by-product of work created by automated regression test tools. Today, test scripts can be manual, automated or a combination of both. The most common term for a collection of test cases is a test suite. The test suite often also contains more detailed instructions or goals for each collection of test cases. It definitely contains a section where the tester identifies the system configuration used during testing. A group of test cases may also contain prerequisite states or steps, and descriptions of the following tests.

6.2 TEST CASES

GUIDELINES FOR TEST CASES:

GUI Test Cases 1

- Total no of features that need to be check Look and Feel
- Look for Default values if at all any (date & Time, if at all any require) Look for spell check.

Test case name	Test case description	Expected value	Actual value	Result
User Operation Test	To check whether the user operations like buttons are working properly.	The buttons must on when it is pressed.	The result we get on checking.	True/False
Video Recording Test	To check whether the video is being recording properly.	The camera should be turned on immediately and the video must be recorded.	The result we get on checking.	True/False
Training Test	Check for the training samples given for the algorithm.	The training samples given to the algorithm should be correct.	The result we get on checking.	True/False
Recognition Test	To Check whether the algorithm recognizes person.	Displays the image with person id	The result we get on checking	True/False

6.3 CONCLUSION:

By testing the software we can know the errors in the software and we can modify the error occurring area. By doing testing process more than once we can know the rarely occurring errors

7. CONCLUSION

7.1 PROJECT CONCLUSION:

In this system we have implemented an attendance system foe a lecture, section or laboratory by which lecturers or teaching assistant can record student's attendance. It saves time and effort, especially if it is lecture with huge number of students. Automated attendance system has been envisioned for the purpose of reducing the drawback in the traditional (manual) system.This system cannot only merely help in the attendance system, but also improve the good will of an institution.

The entire project has been developed from the requirements to a complete system along side evaluation and testing. The system developed has achieved its aim and objectives. The client was happy with the overall performance of the system. However, though some challenges were encountered during implementation, they were addressed and implemented. Also, future work and strategies on how to improve the system are further in this section.

The face detection and recognition algorithm are studied thoroughly taking number of the test from different varying condition images. Attendance of the student are marked using the recognized face of every individual student and the data is stored in an attendance sheet. The attendance of every student marked automatically by recognizing their face with the face present in data set.

7.2 FUTURE ENHANCEMENT:

The system we have development has successfully, able to accomplish the task of marking the attendance in the classroom automatically and output is obtained in an excel sheet as desired in real – time. However, in order to develop a dedicated system which can be implemented in an educational institution, a very efficient algorithm which is insensitive to the lighting condition of the classroom has to be developed. Also a camera of optimum resolution has to be utilized in the system. Another important aspect where we can a work towards is to creating an online database of the attendance and the automatic update of the attendance. This can be done by creating a standalone module which can be installed in the classroom having access to internet, preferably a wireless system. This development can greatly improve the application of the project.

REFERENCE

REFFERED BOOKS

- Unified Modeling Language - James Rumbaugh, Grady Booch
- PYTHON Handbook - Wesly J.Chun

REFFERED SITES

- www.w3schools.com
- https://www.tutorialspoint.com
- https://en.wikipedia.org/wiki/Unified_Modeling_Language
- https://www.aitimejournal.com/@shanmugapriya.balamurugan/face-recognition-using-lbph-algorithm

REFERENCES

- Crawford, Mark. "Facial recognition progress report". SPIE Newsroom. Retrieved 2011-10-06.
- R. Kimmel and G. Sapiro (30 April 2003). "The Mathematics of Face Recognition". SIAM News. Archived from the original on 15 July 2007. Retrieved 2003-04-30.
- Bonsor, K. "How Facial Recognition Systems Work". Retrieved 2008-06-02
- Smith, Kelly. "Face Recognition" (PDF). Retrieved 2008-06-04.
- "What is Facial Recognition? - Definition from Techopedia". Techopedia.com. Retrieved 2018-08-27.

MRIET

B TECH 2019-20 BATCH SEMINAR REPORT

Name: _____ Father Name: _____

Roll No: _____ Contact No:_____

Project Title: _____

Company Name:_____ Contact No:_____

Final Records Submission Date: _____

Soft Copy ***Hard Copy*** ***HOD*** ***PRINCIPAL***

Verified By ***Verified By***