

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belgaum -590014, Karnataka State, India



A PROJECT REPORT ON

“IMPLEMENTING CCTV-BASED ATTENDANCE TAKING SUPPORT SYSTEM USING DEEP FACE RECOGNITION”

**BACHELOR OF ENGINEERING
IN
INFORMATION SCIENCE AND ENGINEERING**

Submitted by: -

MADHUSHREE L	1SJ17IS031
MEGHANA B R	1SJ17IS036
MOUNIKA S	1SJ17IS038
NAMITHA REDDY M	1SJ17IS041

UNDER THE GUIDANCE OF

Project Guide: -

CHANDRA SHEKAR J M
Assistant Professor
Dept. Of ISE
S J C Institute of Technology

Project Co-Ordinator: -

ARAVIND THEJAS CHANDRA
Associate Professor
Dept. of ISE
S J C Institute of Technology



S J C INSTITUTE OF TECHNOLOGY
Department of Information Science and Engineering
Chikkaballapur – 562101
2020-2021

S.J.C INSTITUTE OF TECHNOLOGY

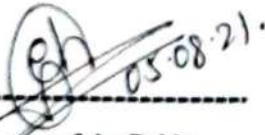
Information Science & Engineering Department

Chickballapur-562101



CERTIFICATE

This is to certify that the project work entitled "IMPLEMENTING CCTV-BASED ATTENDANCE TAKING SUPPORT SYSTEM USING DEEP FACE RECOGNITION" is a bonfide work carried out by MADHUSHREE L(1SJI7IS031), MEGHANA B R(1SJI7IS036), MOUNIKA S(1SJI7IS038), NAMITHA REDDY M(1SJI7IS041) in partial fulfillment for the award of **Bachelor of Engineering in Information science and Engineering in ^{Eighth} ~~seventh~~ semester** of the Visvesvaraya Technological University, Belagavi during the year 2020-2021. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report deposited in the department library. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the Bachelor of Engineering.


05.08.21

Signature of the Guide

Prof. CHANDRASHEKAR J M



Signature of the HOD

Prof. SATHEESH CHANDRA
REDDY





Signature of Principal



Dr. G T RAJU

External Examiners:

Name of the Examiners:

1. 
2. 

Signature with Date

ABSTRACT

Face recognition (FR) has received considerable attention in the field of security, especially in the use of closed-circuit television (CCTV) cameras in security monitoring. Although significant advances in the field of computer vision are made, advanced face recognition systems provide satisfactory performance only in controlled conditions. They deteriorate significantly in the face of real-world scenarios such as lighting conditions, motion blur, camera resolution, etc. This project shows how we design, implement, and conduct the empirical comparisons of machine learning open libraries in building attendance taking (AT) support systems using indoor security cameras called ATSS. Our system was deployed to record the appearances of 120 students in five classes. Our design allows for flexible system scaling, and it is not only usable for a school but a generic attendance system with CCTV.

ACKNOWLEDGEMENT

With Great pride we would like to convey our gratitude and appreciation to our alma-mater “**S.J.C Institute of Technology**” for giving us the required platform for the fulfillment of the Project on “IMPLEMENTING CCTV-BASED ATTENDANCE TAKING SUPPORT SYSTEM USING DEEP FACE RECOGNITION” as per the V.T.U requirements, for the eighth-semester Project.

We express our sincere thanks to **Dr. G T RAJU** , Principal of **S J C Institute of Technology**, Chikkaballapur for providing us with excellent infrastructure to complete the Project.

We express wholehearted gratitude to **Mr. SATHEESH CHANDRA REDDY** who is the respectable **HOD of Information Science and Engineering Department**. We wish to acknowledge her help in making my task easy by providing us with his valuable help and encouragement.

It is our pleasure to thank my guide **Mr. CHANDRA SHEKAR J M** ,Assistant Professor and co-ordinator **Mr. Aravind Thejas Chandra**, Associate Professor, Department of **Information Science and Engineering**, SJCIT for their guidance, encouragement and valuable suggestion from the beginning of the technical seminar till the completion without which this technical seminar would not have been accomplished. We are greatly indebted to them.

And last but not the least, we would be very pleased to express our heartfelt thanks to the teaching and non-teaching staff of the Department of Information Science and Engineering, SJCIT for their motivation and support. We also thank all those who extended their support and co-operation while bringing out this internship.

MADHUSHREE L	1SJ17IS031
MEGHANA B R	1SJ17IS036
MOUNIKA S	1SJ17IS038
NAMITHA REDDY M	1SJ17IS041

LIST OF FIGURES

Figure No	Name	Page No
5.1	System Architecture	13
5.2	Flow Chart	14
5.3	Use Case Diagram	15
5.4	Sequence Diagram	17
5.5	Data Flow Diagram (Level 0)	18
5.6	Data Flow Diagram (Level 1)	19
5.7	Data Flow Diagram (Level 2)	20
6.1	Flow Diagram	21
6.2	Different types of recognizing feature	23
6.3	Image which shows recognizing a particular feature	23
6.4	Cascade Classifier	24
6.5	Train Cascade Object Detector	25
6.6	Steps in testing	26
6.7	Sample images of detected face in different pose	31

LIST OF TABLES

Figure No	Name	Page No
7.1	Unit Test Case 1	41
7.2	Unit Test Case 2	42
7.3	Unit Test Case 3	42
7.4	Unit Test Case 4	43
7.5	Unit Test Case 5	43
7.6	Unit Test Case 6	43

CONTENTS

Chapter No	Chapter Name	Page No
CHAPTER 1	INTRODUCTION	1-3
	1.1 Introduction	
	1.2 Problem Identification	
	1.3 Problem Definition	
	1.4 Objectives	
CHAPTER 2	LITERATURE SURVEY	3-7
	2.1 Max-margin object detection .	
	2.2 Attendance management system using hybrid face recognition techniques.	
	2.3 FSRNet: End-to-End Learning Face Super-Resolution with Facial Priors .	
	2.4 Face Recognition-Based Automatic Classroom Attendance Management System	
CHAPTER 3	SYSTEM ANALYSIS	8
	3.1 Existing system	
	3.2 Proposed system	

6.1.1 HAAR Cascade Detection

6.1.2 LBPH Face recognition

6.2 Modules

6.2.1 Image Capturing

6.2.2 Training phase

6.2.3 Face recognition

6.3 Code

CHAPTER 7	TESTING	39-44
------------------	----------------	--------------

7.1 Testing

7.1.1 Unit testing

7.1.2 Integration testing

7.1.3 Functional testing

7.1.4 System testing

7.1.5 White box testing

7.1.6 Black box testing

7.1.7 Acceptance testing

7.2 Test cases

CHAPTER 8	CONCLUSION AND FUTURE ENHANCEMENT	45
------------------	--	-----------

BIBLIOGRAPHY	46-47
---------------------	--------------

APPENDIX A	SNAPSHOTS	48-52
-------------------	------------------	--------------

CHAPTER 1

INTRODUCTION

1.1 Introduction

Every day, the CCTV system operates to monitor the inside of a building for security. The system's resources allow developers to build computer vision-based applications to integrate with CCTV. Face recognition (FR) is an excellent biometric technique for identity authentication. It is possible to apply FR technology for automatic attendance taking at schools. There are several benefits from attendance considering using the existing camera system, such as save time and effort, provide striking evidence for quality assurance and human resource management tasks, avoid intermediary of infectious diseases. The existing attendance taking system that uses fingerprint recognition is facing several challenges due to large intra-class variability and substantial inter-class similarity mentioned by Dyre and Sumathi. Ngo et al. combined the data from the academic portal with different FR techniques for the task of taking attendance in the classroom. The result shows that their system works smoothly. However, the investment costs for procurement, camera installation at the school, and a large number of video processing are expensive. This research describes the solution to apply deep FR technology to perform AT via the existing CCTV system, which takes advantage of the available resources better and more suitable for different applications.

The attendance system, called ATSS, connects to the CCTV system. ATSS operates dependently on the CCTV system but does not leave any effect on the existing system. There are several components in the system: the camera, job master, job workers, central database, and user interface applications. The camera plays the role to record frames captured from the CCTV system for further processes. The job master constructs, schedules, and arranges tasks to job workers by using data leveraged from the academic portal. The processed data then stored in the central database are accessed by the user for reporting and manipulating data via the desktop application. The attendance data are also submitted to the academic portal. The system admin can configure all parameters. However, it allows the system to be more flexible in terms of processing; the system's response is possible to configure to fit the computational resources.

1.2 Problem Identification

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

1.3 Problem Definition

Every organization has adopted its own method for AMS. Some continue with the traditional method for taking attendance manually while some have adopted the biometric techniques. The traditional method makes it difficult to verify students one-by-one in a large classroom environment. Moreover, the manual labor involved in computing the attendance percentage becomes a major task. The Radio Frequency Identification (RFID) helps to identify a large number of crowds using radio waves. It has high efficiency and hands-free access control. But it is observed that it can be misused.

1.4 Objectives

- a To design an unobtrusive and automated attendance management system with a high accuracy of face detection and recognition algorithm for any pose variation face using a high definition camera without human intervention.

- b. To develop an attendance management system which identifies students' details and stores it in the database for the entire course for that particular instructor.
- c. To reduce the manual errors and to save the attendance taking time.

CHAPTER 2

LITERATURE SURVEY

1. **Title: Max-margin object detection Author: King, Davis Year: 2015**

Abstract: Most object detection methods operate by applying a binary classifier to sub-windows of an image, followed by a non-maximum suppression step where detections on overlapping sub-windows are removed. Since the number of possible sub-windows in even moderately sized image datasets is extremely large, the classifier is typically learned from only a subset of the windows. This avoids the computational difficulty of dealing with the entire set of sub-windows, however, as we will show in this paper, it leads to sub-optimal detector performance.

In particular, the main contribution of this paper is the introduction of a new method, Max-Margin Object Detection (MMOD), for learning to detect objects in images. This method does not perform any sub-sampling, but instead optimizes over all sub-windows. MMOD can be used to improve any object detection method which is linear in the learned parameters, such as HOG or bag-of-visual-word models. Using this approach we show substantial performance gains on three publicly available datasets. Strikingly, we show that a single rigid HOG filter can outperform a state-of-the-art deformable part model on the Face Detection Data Set and Benchmark when the HOG filter is learned via MMOD.

Advantages:

MMOD optimizes the overall accuracy of the entire detector, taking into account information which is typically ignored when training a detector.

Disadvantages:

It does not support more-complex scoring functions, possibly by using kernels.

2.Title: Attendance management system using hybrid face recognition techniques Author: Jayant, Nazare Kanchan, and Surekha Borra Year: 2016

Abstract: Attendance recording of a student in an academic organization plays a vital role in judging students performance. As manual labor involved in this process is time consuming, an automated Attendance Management System (AMS) based on face detection and face recognition techniques is proposed in this paper. The system employs modified Viola-Jones algorithm for face detection, and alignment- free partial face recognition algorithm for face recognition. After successful recognition of a student, the system automatically updates the attendance in the excel sheet. The proposed system improves the performance of existing attendance management systems by eliminating manual calling, marking and entry of attendance in institutional websites.

Advantages:

- i) Automatic tracking of the records of the students
- ii) Minimizing the manual labor and pressure on the lecturers for accurate marking of the attendance
- iii) Minimizing the time required for marking attendance and maximizing the time required for actual teaching process
- iv) Increase the efficiency of the overall system
- v) Improving the security

Disadvantages:

It requires huge space and huge training data.

3.FSRNet: End-to-End Learning Face Super-Resolution with Facial Priors

Author: Yu Chen¹, Ying Tai, Xiaoming Liu, Chunhua Shen, Jian Yang

Year: 2018

Abstract:

Face Super-Resolution (SR) is a domain-specific super resolution problem. The facial prior knowledge can be leveraged to better super-resolve face images. We present a novel deep end-to-end trainable Face Super-Resolution Network (FSRNet), which makes use of the geometry prior, i.e., facial landmark heat maps and parsing maps, to super resolve very low-resolution (LR) face images without well aligned requirement. Specifically, we first construct a coarse SR network to recover a coarse high-resolution (HR) image. Then, the coarse HR image is sent to two branches: a fine SR encoder and a prior information estimation network, which extracts the image features, and estimates landmark heat maps/parsing maps respectively. Both image features and prior information are sent to a fine SR decoder to recover the HR image. To generate realistic faces, we also propose the Face Super- Resolution Generative Adversarial Network (FSRGAN) to incorporate the adversarial loss into FSRNet. Further, we introduce two related tasks, face alignment and parsing, as the new evaluation metrics for face SR, which address the inconsistency of classic metrics w.r.t. visual perception.

Extensive experiments show that FSRNet and FSRGAN significantly outperforms state of the arts for very LR face SR, both quantitatively and qualitatively.

Advantages:

- i) It helps to improve the photometric recovery in terms of PSNR/SSIM.
- ii) provides a solution for accurate geometry estimation directly from very LR images.

Disadvantages:

It recognize a single face only.

4. Title: Face Recognition-Based Mobile Automatic Classroom Attendance Management System

Author: Samet, Refik, and Muhammed Tanriverdi Year: 2017

Abstract: Classroom attendance check is a contributing factor to student participation and the final success in the courses. Taking attendance by calling out names or passing around an attendance sheet are both time-consuming, and especially the latter is open to easy fraud. As an alternative, RFID, wireless, fingerprint, and iris and face recognition-based methods have been tested and developed for this purpose. Although these methods have some pros, high system installation costs are the main disadvantage. The present paper aims to propose a face recognition-based mobile automatic classroom attendance management system needing no extra equipment. To this end, a filtering system based on Euclidean distances calculated by three face recognition techniques, namely Eigenfaces, Fisherfaces and Local Binary Pattern, has been developed for facerecognition. The proposed system includes three different mobile applications for teachers, students, and parents to be installed on their smart phones to manage and perform the real-time attendance-taking process. The proposed system was tested among students at Ankara University, and the results obtained were very satisfactory.

Advantages:

System eliminates the cost for extra equipment, minimizes attendance-taking time, and allows users to access the data anytime and anywhere. Smart devices are very user friendly to perform classroom attendance monitoring. Teachers, students, and parents can use the application without any restrictions and in real-time. Since the internet connection speed has been steadily increasing, high quality, larger images can be sent to the server. With these technological developments, the accuracy rate of the proposed system will also be increased

Disadvantages:

Detection and recognition processes could be performed on smart devices once their the processor capacity is sufficiently increased.

CHAPTER 8

SYSTEM ANALYSIS

3.1 Existing System

Face recognition (FR) is an excellent biometric technique for identity authentication . It is possible to apply FR technology for automatic attendance taking at schools. There are several benefits from attendance considering using the existing camera system, such as save time and effort, provide striking evidence for quality assurance and human resource management tasks, avoid intermediary of infectious diseases. The existing attendance taking system that uses fingerprint recognition is facing several challenges due to large intra-class variability. The traditional method makes it difficult to verify students one-by-one in a large classroom environment. Moreover, the manual labor involved in computing the attendance percentage becomes a major task. The Radio Frequency Identification (RFID) helps to identify a large number of crowds using radio waves. It has high efficiency and hands-free access control. But it is observed that it can be misused.

3.2 Proposed System

In this project, we propose a new solution to make automatic attendance system using the camera. In this proposed system, the manual errors will be avoided. We can save the teaching time and we can avoid the waiting time of students. Finally it will provides the accurate attendance for the students. The automated attendance management system helps to put attendance of the student when the face is recognized using video surveillance . The existing attendance system has several limitations, so we propose the attendance system in such a way that it will remove the limitations of the existing system.

In our solution, by creating interface we generate the dataset. We trained the images using Haar Cascade and AdaBoost classifier. After completing training it will successfully detect and recognize faces and non faces. When stored images and compared images matched then attendance sheet get updated automatically with time and date. As it stored the entering time of every student it becomes easy for faculty member to keep track on time of student.

CHAPTER 9

SYSTEM REQUIREMENTS AND SPECIFICATIONS

4.1 HARDWARE REQUIREMENTS:

- System : Intel I5.
- Hard Disk : 120 GB.
- Monitor : 15'' LED
- Input Devices : Keyboard, Mouse
- Ram : 4 GB

4.1.1 CPU- INTEL CORE i5

Intel Core is a brand name that Intel uses for various mid-range to high-end consumer and business microprocessors. As of 2015 the current line up of Core processors included the Intel Core i7, Intel Core i5, and Intel Core i3. 5th generation Intel® Core™ i5 processors empower new innovations like Intel® Real Sense™ technology—bringing you features such as gesture control, 3D capture and edit, and innovative photo and video capabilities to your devices. Enjoy stunning visuals, built-in security, and an automatic burst of speed when you need it with Intel® Turbo Boost Technology 2.0.

4.1.2 RAM

When you load up an application on to your computer it loads into your available RAM memory. It is very quick type of memory. The more programs you load up, the more RAM is taken up. At the point where you have loaded up enough apps to take up all your free available physical RAM, your OS will create a swap-file on your hard drive. This file is used as a reserve for all additional apps you run. The trouble with that is that hard drives are a lot slower to read and write from than RAM memory is. Therefore, your computer will perform much slower at that point. Although new generation of SSD hard drives are much faster than

your traditional spinning drive, it is still best to have enough RAM available. If you are using Windows and want to want to know how much RAM you are using up, you can right click on task bar, then select start "Task Manager" and on the "performance" tab you will see a green bar indicating "Memory".

4.1.3 HARD DISK:

A hard disk drive (HDD), hard disk, hard drive or fixed disk is a data storage device used for storing and retrieving digital information using one or more rigid ("hard") rapidly rotating disks (platters) coated with magnetic material. The platters are paired with magnetic heads arranged on a moving actuator arm, which read and write data to the platter surfaces. Data is accessed in a random-access manner, meaning that individual blocks of data can be stored or retrieved in any order rather than sequentially. An HDD retains its data even when powered off.

4.2 SOFTWARE REQUIREMENTS:

- | | | |
|--------------------|---|-------------|
| ➤ Operating system | : | Windows 7. |
| ➤ Coding Language | : | PYTHON 3.6 |
| ➤ Tools | : | Python IDLE |

4.2.1 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 language.
- Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance.
- Python supports modules and packages, which encourages program modularity and code reuse.
- The Python interpreter and the extensive standard library are available as open source.

- Python is widely considered as the preferred language for teaching and learning ML (Machine Learning).

4.2.2 PYTHON IDLE :

- An IDE (or Integrated Development Environment) is a program dedicated to software development.
- IDEs integrate several tools specifically designed for software development. These tools usually include:
 - An editor designed to handle code (with, for example, syntax highlighting and auto-completion)
 - Build, execution, and debugging tools
 - Some form of source control
- When you install Python, IDLE is also installed by default.
- Its major features include the Python shell window(interactive interpreter), auto-completion, syntax highlighting, smart indentation, and a basic integrated debugger.
- We used python version 3.6.8

CHAPTER 5

SYSTEM DESIGN

5.1 Introduction to System Analysis

A system is an orderly group of interdependent components linked together according to a plan to achieve a specific objective. Its main characteristics are organization, interaction, interdependence, integration and a central objective. System analysis and design are the application of the system approach to problem solving generally using computers. To reconstruct a system the analyst must consider its elements output and inputs, processors, controls feedback and environment. Analysis is a detailed study of the various operations performed by a system and their relationships within and outside of the system. One aspect of analysis is defining the boundaries of the system and determining whether or not a candidate system should consider other related systems. During analysis data are collected on the available files decision points and transactions handled by the present system. This involves gathering information and using structured tools for analysis.

5.2 Feasibility Study

Feasibility is the determination of whether or not a project is worth doing. The process followed in making this determination is called feasibility Study. Depending on the results of the initial investigation, the survey is expanded to more detailed feasibility study. “FeasibilityStudy” is a test of the system proposal according to its workability, impact on the organization, ability to meet needs and effective use of the resources.

5.2.1 Economic Feasibility

This study is carried out to check the economic impact that the system will have on the market. The expenditure can be justified. Thus, the developed system must be in budget and should be effective and efficient in real time usage.

5.2.2 Operational Feasibility

This system reduces task execution times which are not identified by the existing system, making the users data completely safe.

5.2.3 Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have high demand on the available technical resources. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

5.3 System Architecture

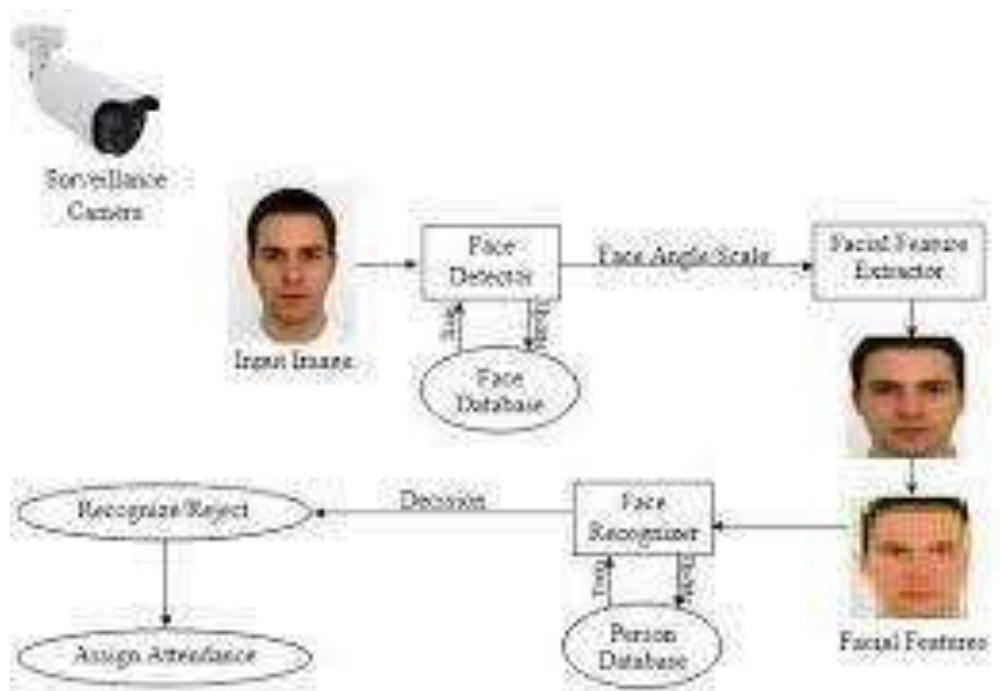


Fig 5.1 : System Architecture

5.4 Object Oriented Design

Object-oriented analysis and design (OOAD) is a technical approach for analysing and designing an application, system, or business by applying object-oriented programming, as well as using visual modelling throughout the software development process to guide stakeholder communication and product quality.

5.4.1 Flow Chart

A flowchart is a type of diagram that represents a workflow or process. A flowchart can also be defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving a task.

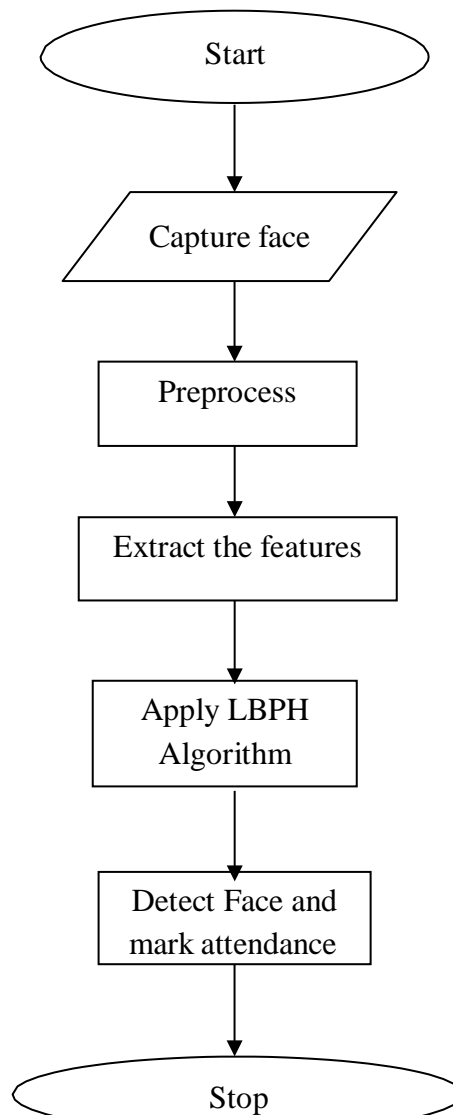


Fig 5.2 : Flow Chart

5.4.2 Use Case Diagram

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.

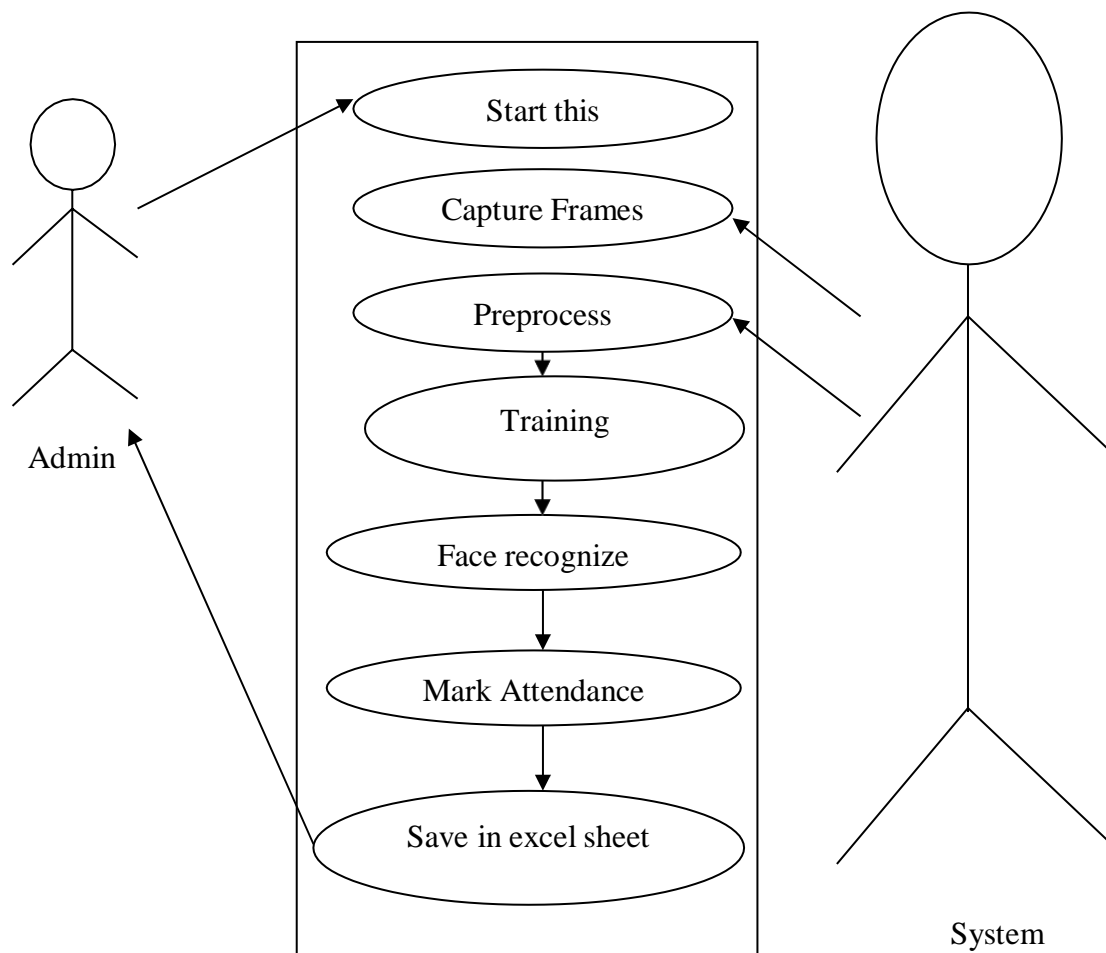


Fig 5.3 : Use Case Diagram

5.4.3 Sequence Diagram

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.

If the lifeline is that of an object, it demonstrates a role. Leaving the instance name blank can represent anonymous and unnamed instances.

Messages, written with horizontal arrows with the message name written above them, display interaction. Solid arrow heads represent synchronous calls, open arrow heads represent asynchronous messages, and dashed lines represent reply messages. If a caller sends a synchronous message, it must wait until the message is done, such as invoking a subroutine. If a caller sends an asynchronous message, it can continue processing and doesn't have to wait for a response. Asynchronous calls are present in multithreaded applications, event-driven applications and in message-oriented middleware. Activation boxes, or method-call boxes, are opaque rectangles drawn on top of lifelines to represent that processes are being performed in response to the message (Execution Specifications in UML).

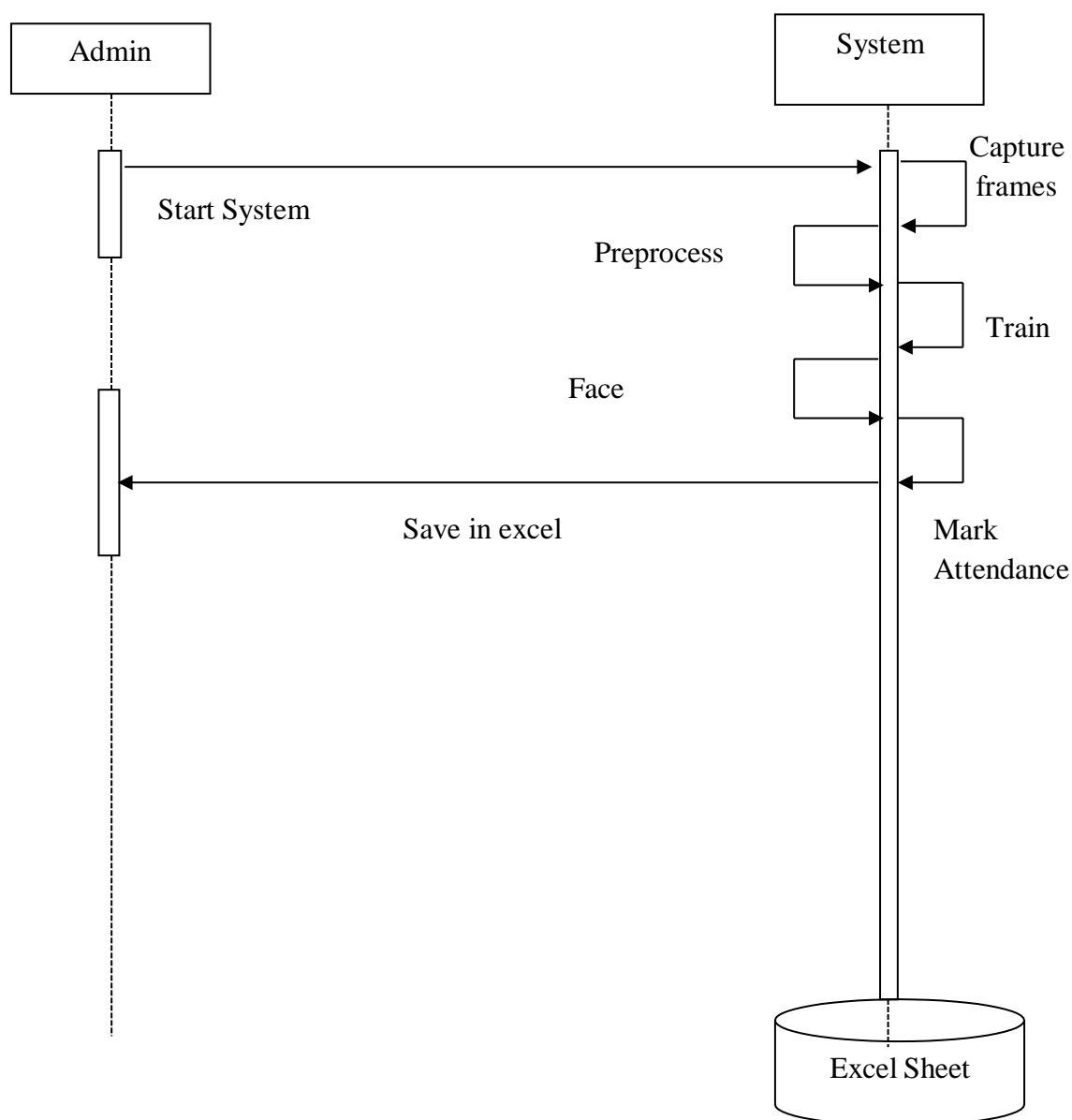


Fig 5.4 : Sequence Diagram

5.4.4 Data Flow Diagram

A data-flow diagram is a way of representing a flow of a data of a process or a system (usually an information system). The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow; there are no decision rules and no loops. Specific operations based on the data can be represented by a flowchart.

There are several notations for displaying data-flow diagrams. The notation presented above was described in 1979 by Tom DE Marco as part of Structured Analysis.

For each data flow, at least one of the endpoints (source and / or destination) must exist in a process. The refined representation of a process can be done in another data-flow diagram, which subdivides this process into sub-processes.

Level 0 :

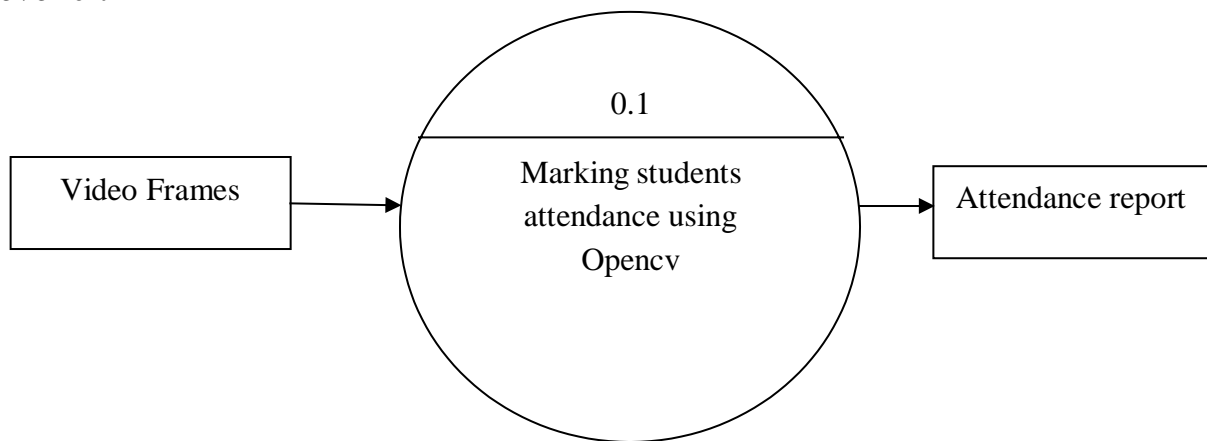


Fig 5.5 : Data Flow Diagram [Level 0]

Level 0 : describes the overall process of the project. We are using captures video frames as input. System will use the LBPH to detect face and it will generate attendance report in excel sheet .

Level 1 :

Level 1 : describes the first step of the project. We are using captures video frames as input. System will preprocess and extract the features.

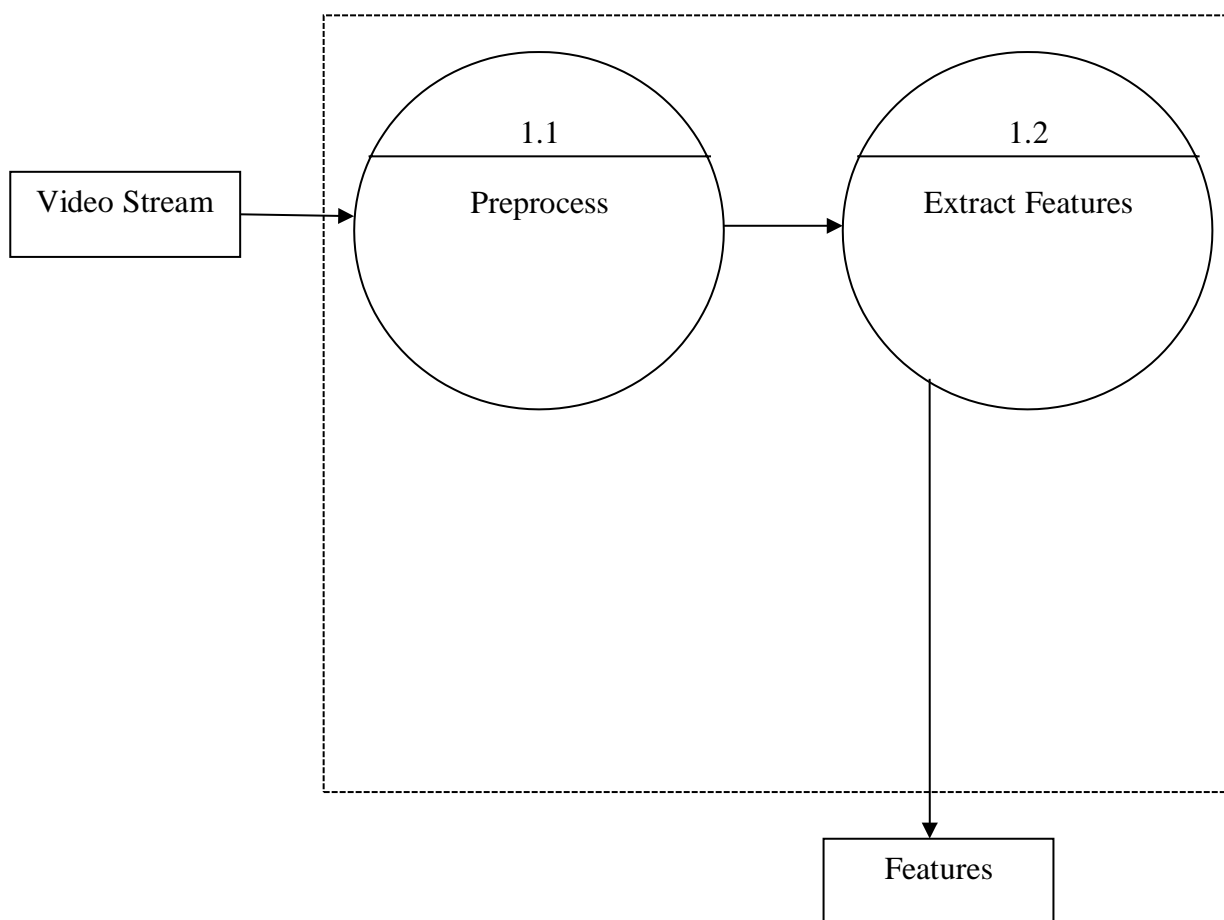


Fig 5.6 : Data Flow Diagram [Level 1]

Level 2 :

Level 2 : describes the final step of the project. We are using face features and trained data as input. System will use the LBPH algorithm to detect face and it will generate attendance report in excel sheet .

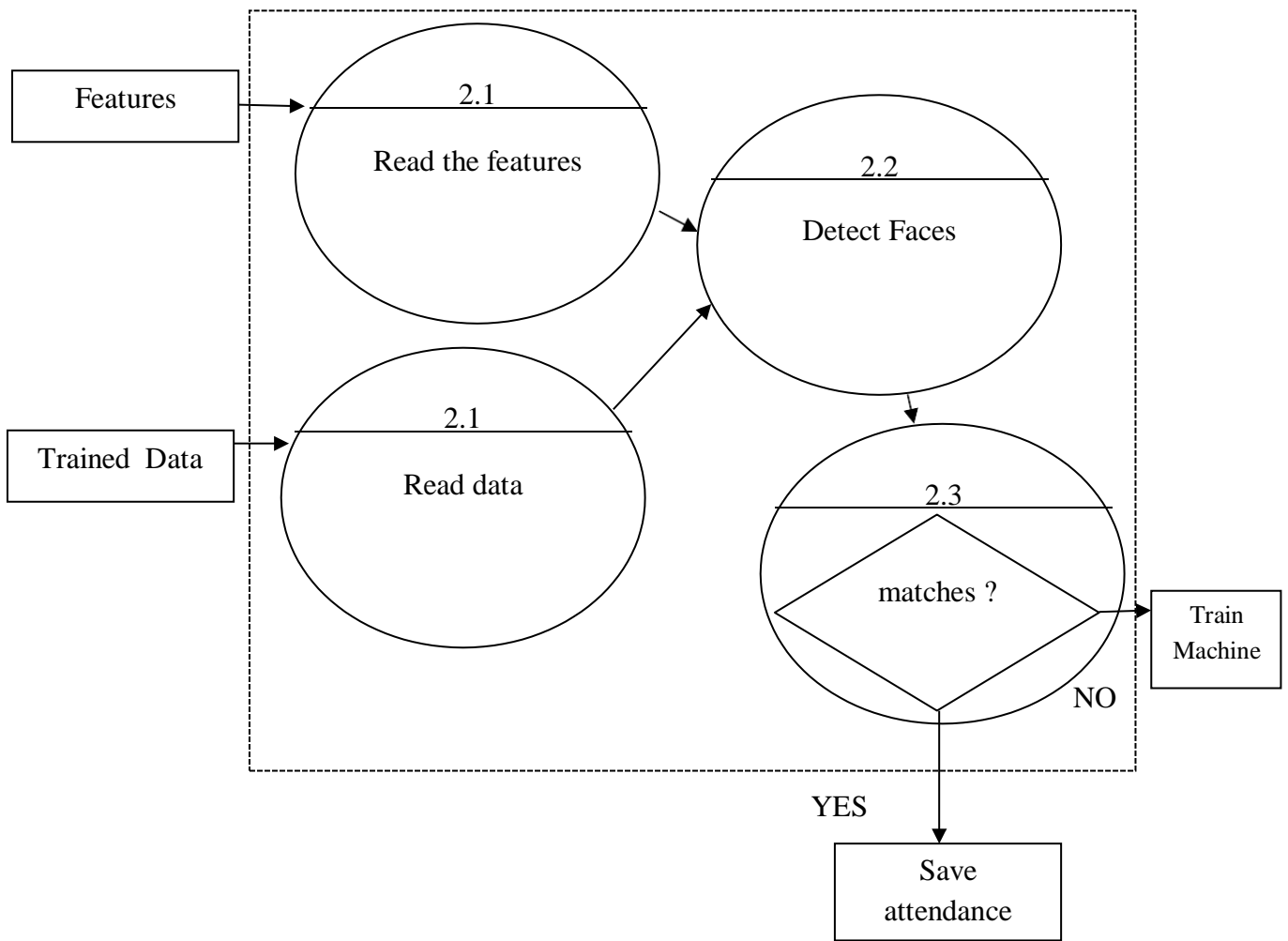


Fig 5.7 : Data Flow Diagram [Level 2]

CHAPTER 6

IMPLEMENTATION

6.1 Methodology

Methodology is the systematic, theoretical analysis of the methods applied to a field of study. It comprises the theoretical analysis of the body of methods and principles associated with a branch of knowledge. Typically, it encompasses concepts such as paradigm, theoretical model, phases and quantitative or qualitative techniques.

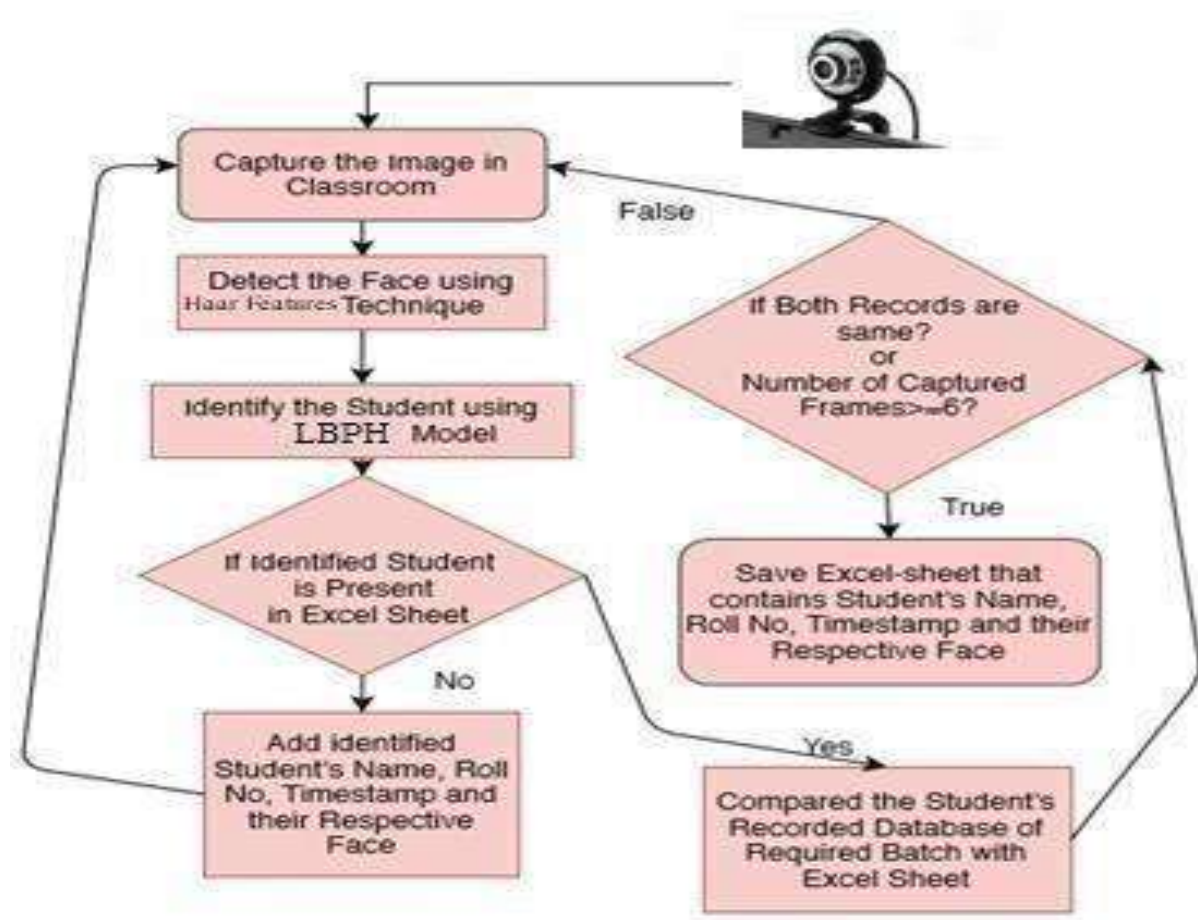


Fig 6.1 : Flow Diagram

The above mentioned fig represents the flow chart of the project. We are lively capturing the classroom and detect the face using Haar- Cascade technique and identify the student using LBPH If student is identified Then it will mark present in excel sheet if it is not identified We need to Register the student.

In this project we are using the two algorithms

1. Haar cascade Classifier
2. Local Binary Pattern Histogram(LBPH)

6.1.1 HAAR Cascade Face Detection:

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 images.

It is then used to detect objects in other images. The algorithm has four stages:

- 1 .Haar Feature Selection
- 2 .Creating Integral Images
3. Adaboost Training
4. Cascading Classifiers

It is well known for being able to detect faces and body parts in an image, but can be trained to identify almost any object.

Lets take face detection as an example. 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. A Haar feature considers adjacent rectangular regions at a specific location in a detection window, sums up the pixel intensities in each region and calculates the difference between these sums.

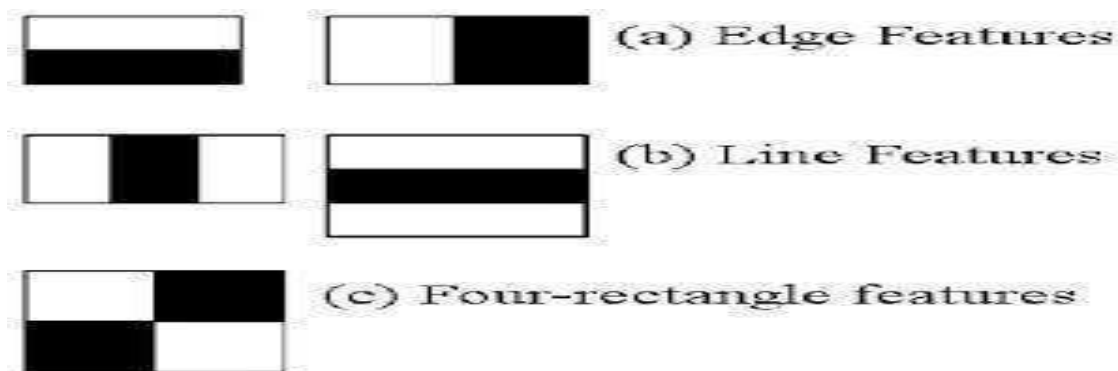


Fig 6.2 : Different types of recognizing feature

Integral Images are used to make this super fast.

But among all these features we calculated, most of them are irrelevant. For example, consider the image below. 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 irrelevant.



Fig 6.3 : Image which shows recognizing a particular feature

So how do we select the best features out of 160000+ features? This is accomplished using a concept called **Adaboost** which both selects the best features and trains the classifiers that use them. This algorithm constructs a “strong” classifier as a linear combination of weighted simple “weak” classifiers. The process is as follows.

During the detection phase, a window of the target size is moved over the input image, and for each subsection of the image and Haar features are calculated. You can see this in action in the video below. This difference is compared to a learned threshold that separates non-objects

from objects.

A large number of Haar features are necessary to describe an object with sufficient accuracy and are therefore organized into *cascade classifiers* to form a strong classifier.

Cascade Classifier

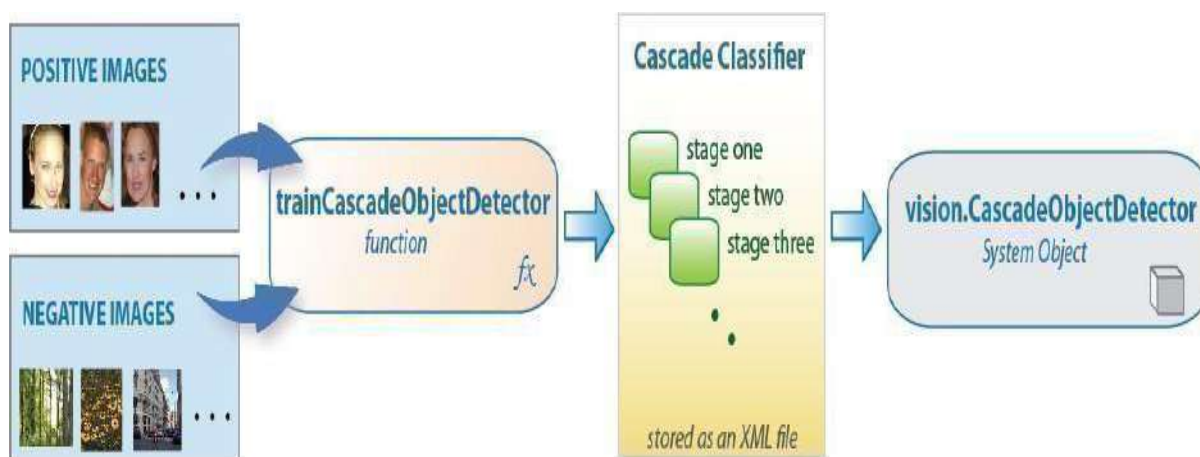


Fig 6.4 : Cascade Classifier

The cascade classifier consists of a collection of stages, where each stage is an ensemble of weaklearners. The weak learners are simple classifiers called *decision stumps*. Each stage is trained using a technique called boosting. *Boosting* provides the ability to train a highly accurate classifier by taking a weighted average of the decisions made by the weak learners.

Each stage of the classifier labels the region defined by the current location of the sliding window as either positive or negative. *Positive* indicates that an object was found and *negative* indicates no objects were found. If the label is negative, the classification of this region is complete, and the detector slides the window to the next location. If the label is positive, the classifier passes the region to the next stage. The detector reports an object found at the current window location when the final stage classifies the region as positive.

The stages are designed to reject negative samples as fast as possible. The assumption is that the vast majority of windows do not contain the object of interest. Conversely, true positives are rare and worth taking the time to verify.

- A *true positive* occurs when a positive sample is correctly classified.
- A *false positive* occurs when a negative sample is mistakenly classified as positive.
- A *false negative* occurs when a positive sample is mistakenly classified as negative.

Adding more stages reduces the overall false positive rate, but it also reduces the overall true positive rate.

Cascade classifier training requires a set of positive samples and a set of negative images. You must provide a set of positive images with regions of interest specified to be used as positive samples. You can use the Image Labeler to label objects of interest with bounding boxes. The Image Labeler outputs a table to use for positive samples. You also must provide a set of negative images from which the function generates negative samples automatically. To achieve acceptable detector accuracy, set the number of stages, feature type.

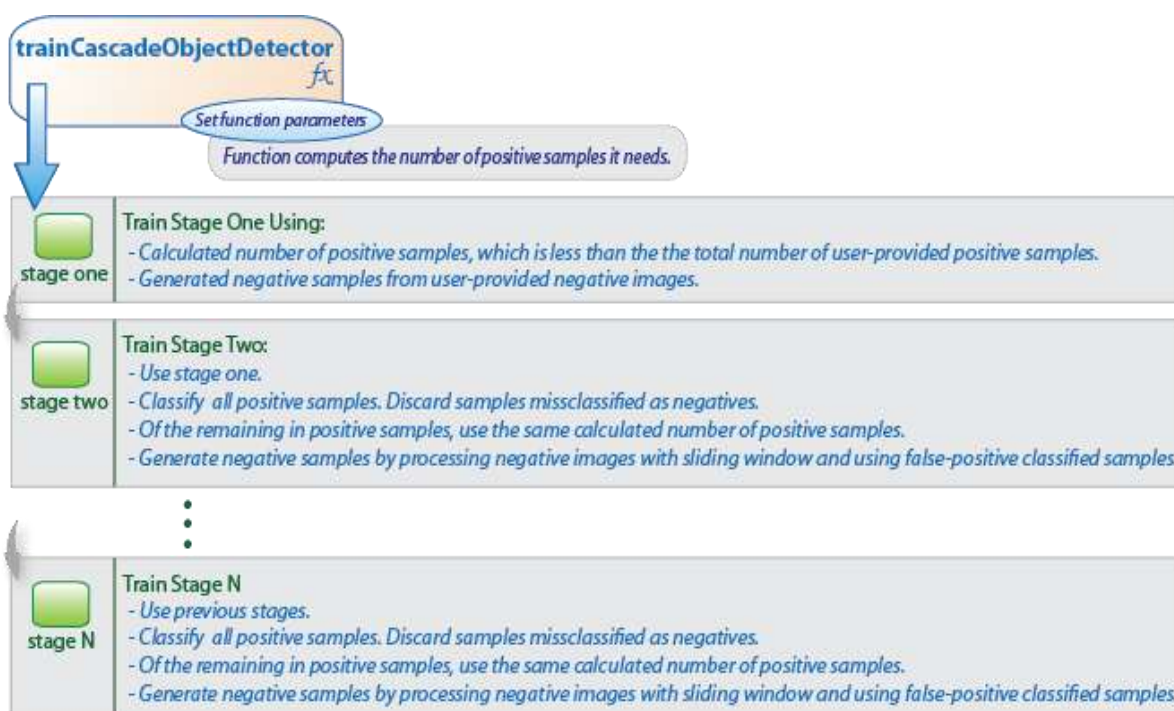


Fig 6.5 : Train Cascade Object detector

6.1.2 LBPH Face Reorganization:

Local Binary Patterns 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:

- creatingdataset

- faceacquisition
- feature extraction
- classification

The LBPH algorithm is a part of opencv.

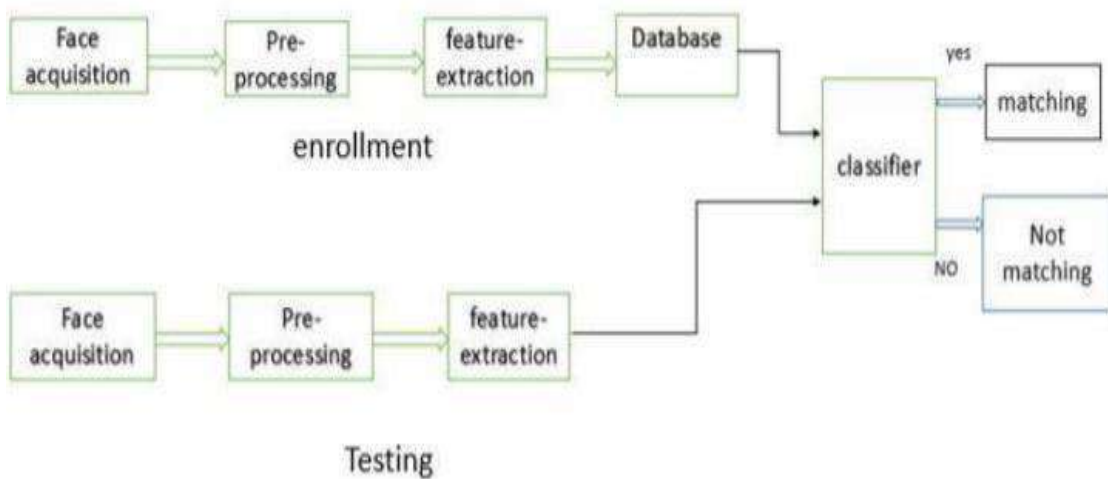


Fig 6.6 Steps in testing

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 .



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 neighborpixel.

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

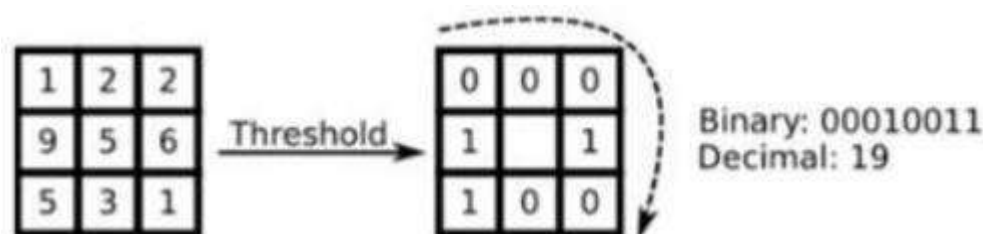
$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

If the value of neighbor is greater than or equal to the central value it is set as 1 otherwise it is set as 0.

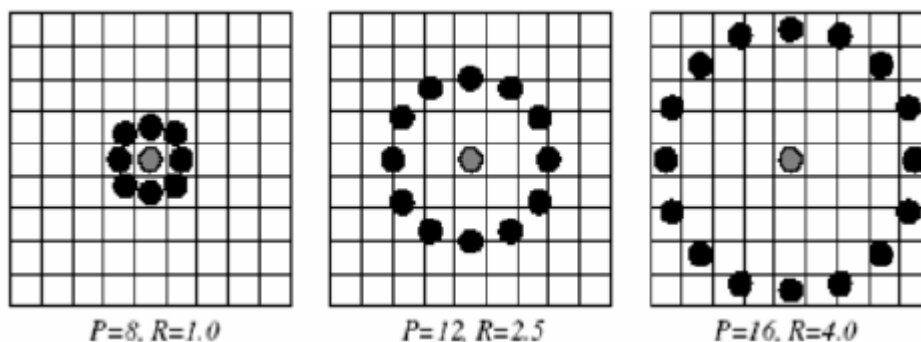
Thus, we obtain a total of 8 binary values from the 8 neighbors.

After combining these values we get a 8 bit binary number which is translated to decimal number for our convenience.

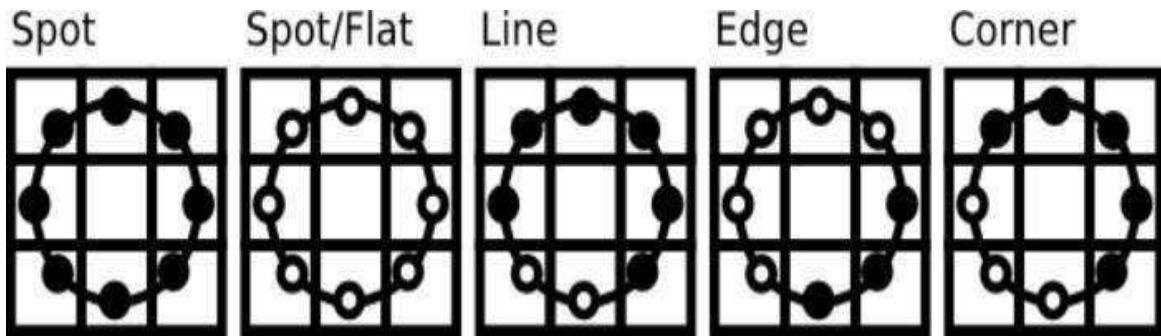
This decimal number is called the pixel LBP value and its range is 0-255.



Later it was noted that a fixed neighborhood 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 (X_c, Y_c) the position of the neighbor (X_p, Y_p) , p belonging to P can be calculated by:

$$\begin{aligned}
 x_p &= x_c + R \cos\left(\frac{2\pi p}{P}\right) \\
 y_p &= y_c - R \sin\left(\frac{2\pi p}{P}\right)
 \end{aligned}$$

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

If a points coordinate on the circle doesn't correspond to image coordinates, it get's interpolatedgenerally 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.

6.2 MODULES

- i) **Image Capturing**
- ii) **Training phase**
- iii) **Face recognition**

Module Descriptions:

6.2.1 Image Capturing:

Proposed System consists of a camera, placed in the classroom to capture all the students. From these captured image frames, using the opencv and system will detect the student's face in the captured image using Haar cascade face detection technique.

Pseudo Code:

Procedure Image Capture()

Input: Student face

Output: Captured Gray scale Images

Begin:

Step1: Start the camera

Step2: Read the frames

Step3: for each frames

Step3.1: Extract the face features like eyes, nose, jawline ,etc.

Step 3.2: Detect the face in each frame using Haar cascade technique.

Step 3.3: Capture face

Step 3.5: Save the face image in disk

End for

Step 4: return the saved message

End

6.2.2 Training phase

In the training phase we are applying the LBPH(local binary pattern Histogram) algorithm.

Pseudo Code:

Procedure Training Phase()

Input: Dataset(images and id,name)

Output: trained file(.yml file)

Begin:

Step1: Read the dataset

Step3: for each images in dataset

Step3.1: Extract the face features like eyes, nose, jawline ,etc.

Step 3.2: Extract the id and name of the student.

End for

Step 4: Train the extracted face features with the corresponding id and name using

LBPH.

Step 5: Save the trained file.

Step 5: return the trained message

End

6.2.3 Face recognition

We proposed face detection technique by incorporating Haar cascade classifier and LBPH techniques. This technique does not play out any sub-sampling, but it optimizes over all sub-

windows. This method is much accurate to detect all varied faces positioned frontal, tilted up/right/left/down and occluded faces with 99.69% accuracy. Following figure shows some samples of detected face using proposed method.

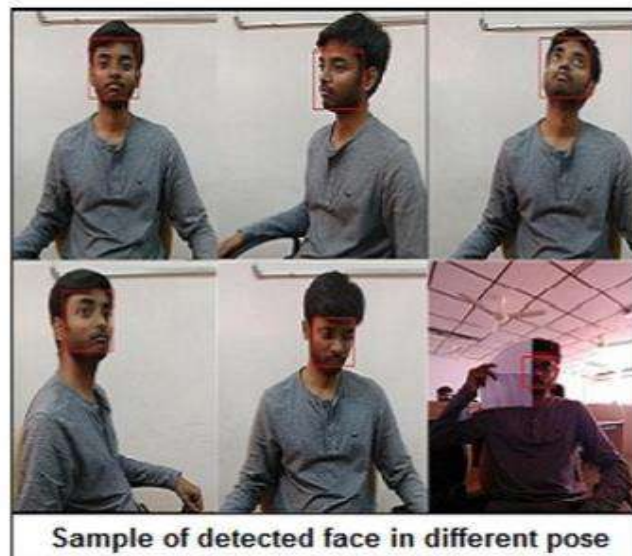


Fig 6.7 : Sample images of detected face in different pose

Pseudo Code:

Procedure Face recognition()

Input: live Video Stream

Output: Attendance marked in CSV file

Begin:

Step1: Read the image frame form the input stream

Step 2: Detect the face from the input image using Haar Cascade classifier

Step3: Extract the face features from the image using LBPH algorithm

Step3.1: Load Trained model.

Step 3.2: Match the Histogram values with the input image features with trained data.

Step 3.3: If matches >75 then

Step 3.3.1 Return the recognized student name and id

Else

Step 3.3.1 return no matches found

Step 4: Mark attendance for the recognized faces.

Step 5: Save the result in csv file.

Step 5: return the result message

End

6.3 CODE

```
import tkinter as tk

from tkinter import Message ,Text

import cv2,os

import shutil

import csv

import numpy as np

from PIL import Image, ImageTk

import pandas as pd

import datetime

import time

import tkinter.ttk as ttk

import tkinter.font as font
```

```
from PIL import ImageTk

window = tk.Tk()

window.title("Face_Recogniser")

dialog_title = 'QUIT'

dialog_text = 'Are you sure?'

window.geometry('1280x720')

WIDTH, HEIGHT = 1350, 800

img = ImageTk.PhotoImage(Image.open("p6.gif").resize((WIDTH, HEIGHT),
Image.ANTIALIAS))

x = tk.Label(image = img)

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

window.configure(background ='black')

window.attributes('-fullscreen', True)

window.grid_rowconfigure(0, weight=1)

window.grid_columnconfigure(0, weight=1)

message = tk.Label(window, text=" ATTENDANCE TAKING SUPPORT SYSTEM"
,width=35,height=2,font=('Helvetica', 20, ' bold '),compound="center")

message.place(x=480, y=30)

lbl = tk.Label(window, text="Enter ID",width=20 ,height=2 ,fg="white" ,bg =
"gray25",font=('calibri', 15, ' bold ') )

lbl.place(x=400, y=200)

txt = tk.Entry(window,width=20 ,bg="gray25" ,fg="white",font=('calibri', 15, ' bold '))
```

```
txt.place(x=700, y=215)

lbl2 = tk.Label(window, text="Enter Name",width=20 ,fg="white" ,bg="gray25"
,height=2 ,font=('calibri', 15, ' bold '))

lbl2.place(x=400, y=300)

txt2 = tk.Entry(window,width=20 ,bg="gray25" ,fg="white",font=('calibri', 15, ' bold '))

txt2.place(x=700, y=315)

lbl3 = tk.Label(window, text="Notification : ",width=20 ,fg="white" ,bg="gray25"
,height=2 ,font=('calibri', 15, ' bold '))

lbl3.place(x=400, y=400)

message = tk.Label(window, text="" ,bg="gray25" ,fg="white" ,width=30 ,height=2,
activebackground = "yellow" ,font=('calibri', 15, ' bold '))

message.place(x=700, y=400)

lbl3 = tk.Label(window, text="Attendance : ",width=20 ,fg="white" ,bg="gray25"
,height=2 ,font=('calibri', 15, ' bold '))

lbl3.place(x=400, y=650)

message2 = tk.Label(window, text="" ,fg="white" ,bg="gray25",activeforeground =
"green",width=30 ,height=2 ,font=('calibri', 15, ' bold '))

message2.place(x=700, y=650)

def clear():

    txt.delete(0, 'end')

    res = ""

    message.configure(text= res)
```

```
def clear2():  
  
    txt2.delete(0, 'end')  
  
    res = ""  
  
    message.configure(text= res)  
  
def is_number(s):  
  
    try:  
  
        float(s)  
  
        return True  
  
    except ValueError:  
  
        pass  
  
    try:  
  
        import unicodedata  
  
        unicodedata.numeric(s)  
  
        return True  
  
    except (TypeError, ValueError):  
  
        pass  
  
    return False  
  
def TakeImages():  
  
    Id=(txt.get())  
  
    name=(txt2.get())
```

```
if(is_number(Id) and name.isalpha()):

    cam = cv2.VideoCapture(0)

    harcascadePath = "haarcascade_frontalface_default.xml"

    detector=cv2.CascadeClassifier(harcascadePath)

    sampleNum=0

    while(True):

        ret, img = cam.read()

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

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

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

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

            sampleNum=sampleNum+1

            cv2.imwrite("TrainingImage\ "+name + "." +Id + '.' + str(sampleNum) + ".jpg",
gray[y:y+h,x:x+w])

            #display the frame

            cv2.imshow('frame',img)

            #wait for 100 milliseconds

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

                break

            # break if the sample number is morethan 100

            elif sampleNum>60:

                break

    cam.release()
```

```
cv2.destroyAllWindows()

res = "Images Saved for ID : " + Id + " Name : "+ name

row = [Id , name]

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

    writer = csv.writer(csvFile)

    writer.writerow(row)

csvFile.close()

message.configure(text= res)

else:

    if(is_number(Id)):

        res = "Enter Alphabetical Name"

        message.configure(text= res)

    if(name.isalpha()):

        res = "Enter Numeric Id"

        message.configure(text= res)
```

CHAPTER 7

TESTING

7.1 TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

7.1.1 Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

7.1.2 Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfactory, as shown by successful unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

7.1.3 Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

7.1.4 System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

7.1.5 White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

7.1.6 Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

7.1.7 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

7.2 TEST CASES

Table 7.1 : Unit Test case 1

Test Case#	UTC01
Test Name	User input format
Test Description	To test user input as face
Input	face
Expected Output	The file should be read by the program and display on the monitor
Actual Output	The file is read and display accordingly
Test Result	Success

Table 7.2 : Unit Test case 2

Test Case	UTC02
Test Name	User input format
Test Description	To test user input as face
Input	Not face
Expected Output	Screen should not appear wait for face
Actual Output	Idle state
Test Result	Success

Table 7.3 : Unit Test case 3

Test Case	UTC03
Test Name	Detection of face
Test Description	To test whether Detecting face or not?
Input	Face
Expected Output	It Should detect and mark face in rectangle box
Actual Output	detected and marked face in rectangle box
Test Result	Success

Table 7.4 : Unit Test case 4

Test Case	UTC04
Test Name	Feature Extraction
Test Description	To test whether extracting face features and id and name or not
Input	Id as int, name as string face as image
Expected Output	It extract the important features and train
Actual Output	Its extracted the important features and trained
Test Result	Success

Table 7.5 : Unit Test case 5

Test Case	UTC05
Test Name	Test case to determine correct predicted value of face
Test Description	To test whether a given statistical model predicts the student name based on the face input.
Input	A face image
Expected Output	The algorithm should predict the student name and mark attendance as per historical data collected
Actual Output	The predicted value by algorithm is closer to the specified value in the historical data.
Test Result	Success

Table 7.6 : Unit Test case 6

Test Case	UTC06
Test Name	Test case for importing valid python libraries
Test Description	To test whether an algorithm to implement congestion nodes works without sklearn and keras models
Input	Import all valid libraries sklearn, flask and keras libraries
Expected Output	An error should be thrown specifying “error importing libraries sklearn, flask and keras libraries”
Actual Output	An error is thrown
Test Result	Success

CHAPTER 8

CONCLUSION AND FUTURE ENHANCEMENT

In this approach, a face recognition based automated student attendance system is thoroughly described. The proposed approach provides a method to identify the individuals by comparing their input image obtained from recording video frame with respect to train image. This proposed approach able to detect and localize face from an input facial image, which is obtained from the recording video frame. Besides, it provides a method in pre- processing stage to enhance the image contrast and reduce the illumination effect. Extraction of features from the facial image is performed by applying both LBPH and Haar cascade. The algorithm designed to combine LBP and PCA able to stabilize the system by giving consistent results. The accuracy of this proposed approach is 100 % for high-quality images, 92.31 % for low- quality images and 95.76 % of Yale face database when two images per person are trained.

As a conclusion for analysis, the extraction of facial feature could be challenging especially in different lighting. In pre-processing stage, Contrast Limited Adaptive Histogram Equalization (CLAHE) able to reduce the illumination effect. CLAHE perform better compared to histogram equalization in terms of contrast improvement. Enhanced LBPH with larger radius size specifically, radius size two, perform better compared to original LBPH operator, with less affected by illumination and more consistent compared to other radius sizes.

BIBLIOGRAPHY

- [1] Minaee, S.; Abdolrashidi, A.; Su, H.; Bennamoun, M.; Zhang, D. Biometric recognition using deep learning: A survey. arXiv 2019, arXiv:1912.00271.
- [2] Son, N.T.; Chi, L.P.; Lam, P.T.; Van Dinh, T. Combination of facial recognition and interaction with academic portal in automatic attendance system. In Proceedings of the 2019 8th International Conference on Software and Computer Applications (ICSCA '19). ACM, New York, NY, USA, 19–21 February 2019; pp. 299–305. [CrossRef]
- [3] Shoba, D.; Sumathi, C.P. A Survey on Various Approaches to Fingerprint Matching for Personal Verification and Identification. International Journal of Computer Science & Engineering Survey (IJCSES), 7 August 2016; Volume 7, No.
- [4][CrossRef] 4. Balaban, S. Deep learning and face recognition: The state of the art. arXiv 2015, arXiv:1902.03524.
- [5] Yu, H.; Luo, Z.; Tang, Y. Transfer learning for face identification with deep face model. In Proceedings of the 2016 7th International Conference on Cloud Computing and Big Data (CCBD), Macau, China, 16–18 November 2016. [CrossRef]
- Wang, M.; Deng, W. Deep face recognition: A survey. arXiv 2019, arXiv:1804.06655v8.
- [6] Deng, J.; Guo, J.; Zafeiriou, S. Arcface: Additive angular margin loss for deep face recognition. arXiv 2018, arXiv:1801.07698.
- [7] Liu, W.; Wen, Y.; Yu, Z.; Li, M.; Raj, B.; Song, L. SphereFace: Deep hypersphere embedding for face recognition. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017. [CrossRef]
- [8] Schrof, F.; Kalenichenko, D.; Philbin, J. FaceNet: A unified embedding for face recognition and clustering. arXiv 2015, arXiv:1503.03832V3.
- [9] Wang, H.; Wang, Y.; Zhou, Z.; Ji, X.; Gong, D.; Zhou, J.; Liu, W. CosFace: Large margin cosine loss for deep face recognition. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2018.[CrossRef]

- [10] Ranjan, R.; Sankaranarayanan, S.; Bansal, A.; Bodla, N.; Chen, J.C.; Patel, V.M.; Castillo, C.D.; Chellappa, R. Deep learning for understanding faces: Machines may be just as good, or better, than humans. *IEEE Signal Process. Mag.* 2018, 35, 66–83. [CrossRef]
- [11] Zhang, K.; Zhang, Z.; Li, Z.; Qiao, Y. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Process. Lett.* 2016, 23, 1499–1503. [CrossRef]
- [11] Wang, H.; Li, Z.; Ji, X.; Wang, Y. Face R-CNN. *arXiv* 2017, arXiv:1706.01061.
- [12] Feng, Z.-H.; Kittler, J.; Awais, M.; Huber, P.; Wu, X.-J. Face detection, bounding box aggregation and pose estimation for robust facial landmark localisation in the wild. In *Proceedings of the 2017 IEEE Conference on Computer Vision and pattern Recognition Workshops, (CVPRW), Honolulu, HI, USA, 21–26 July 2017.* [CrossRef].

APPENDIX A

SNAPSHOTS

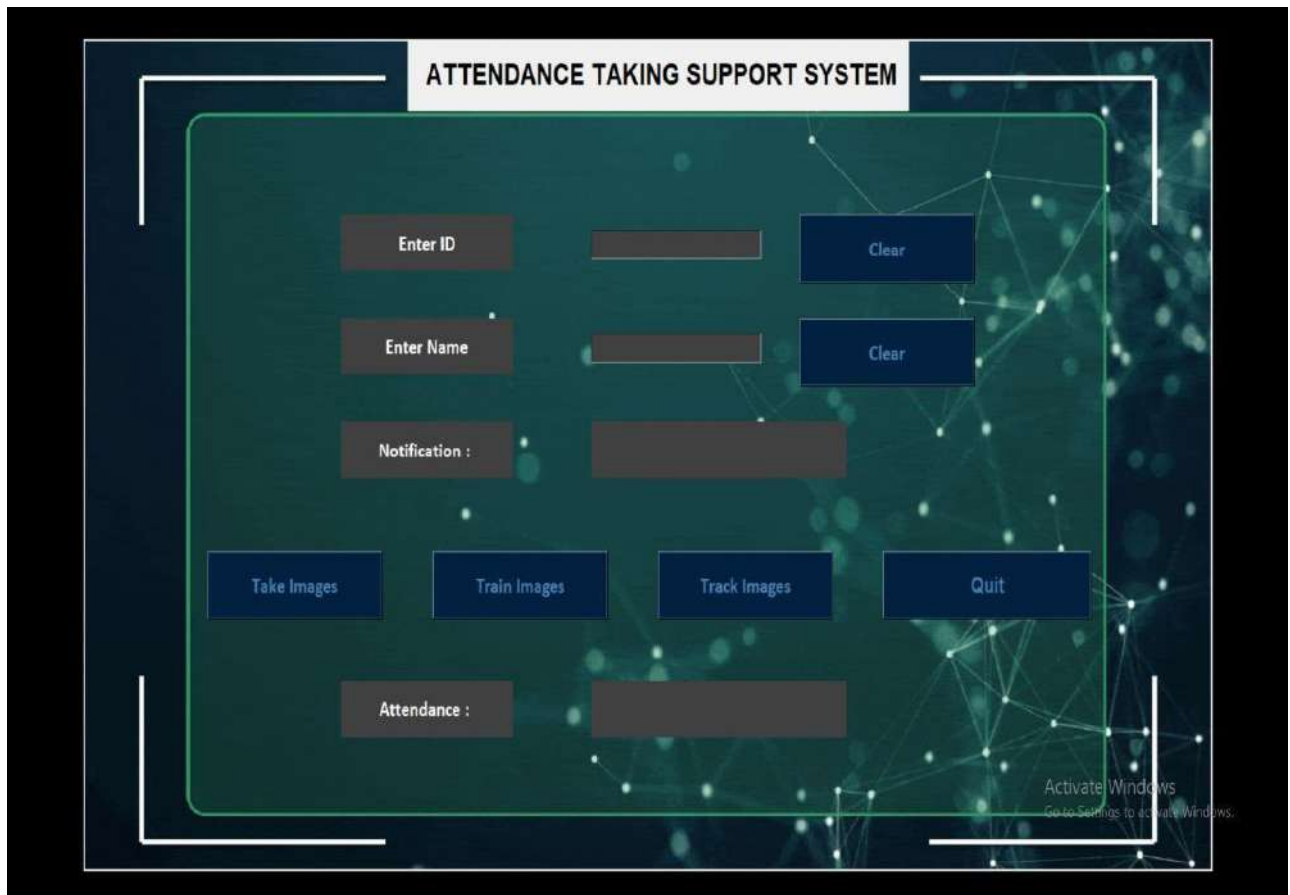


Fig 1 : Attendance taking support system

This is the front end where there is space for entering the student ID and Name . The ID is unique for every student and notification column which gets you informed what happen after you click each button. And we can see all the buttons like train images, take Images, Track images and quit which can be used for respective functions.

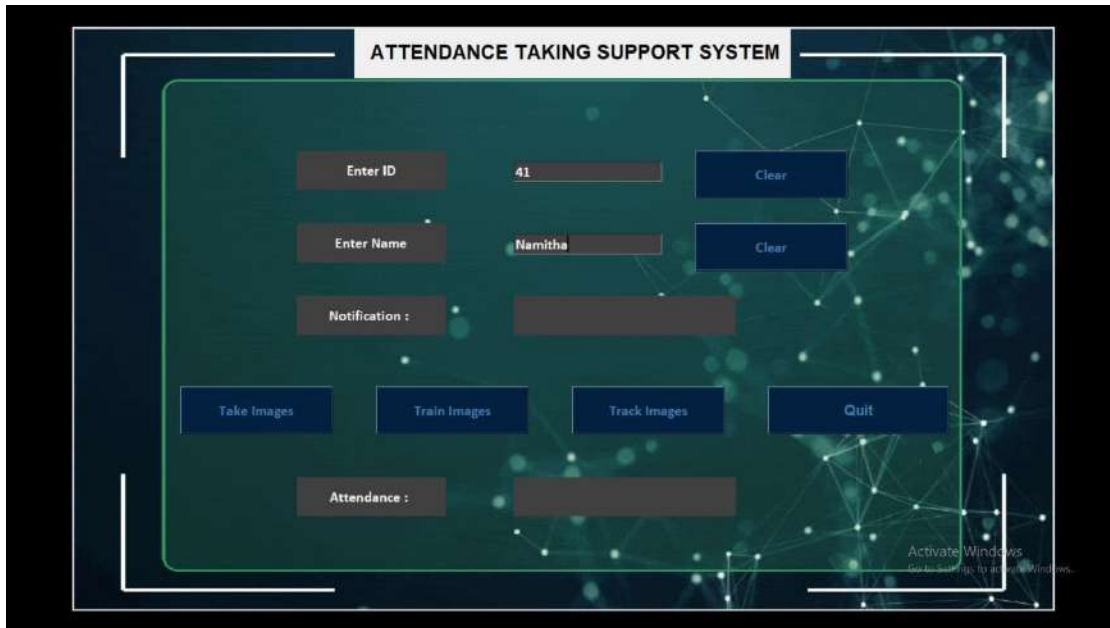


Fig 2 : Entering the student ID and Name

Here is the example where we have entered 41 as ID and Namitha as name.

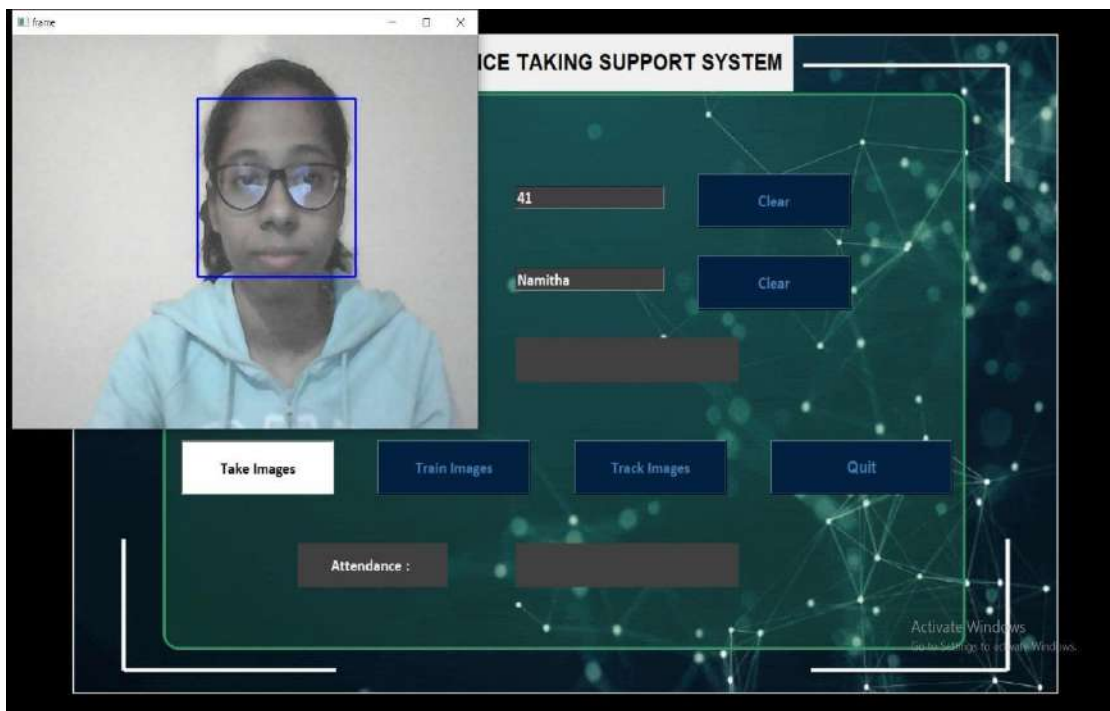


Fig 3 : Taking Images

Now when clicked on Take images button, the camera gets switched on and it will take 200 images per student and store it with their ID and Name.

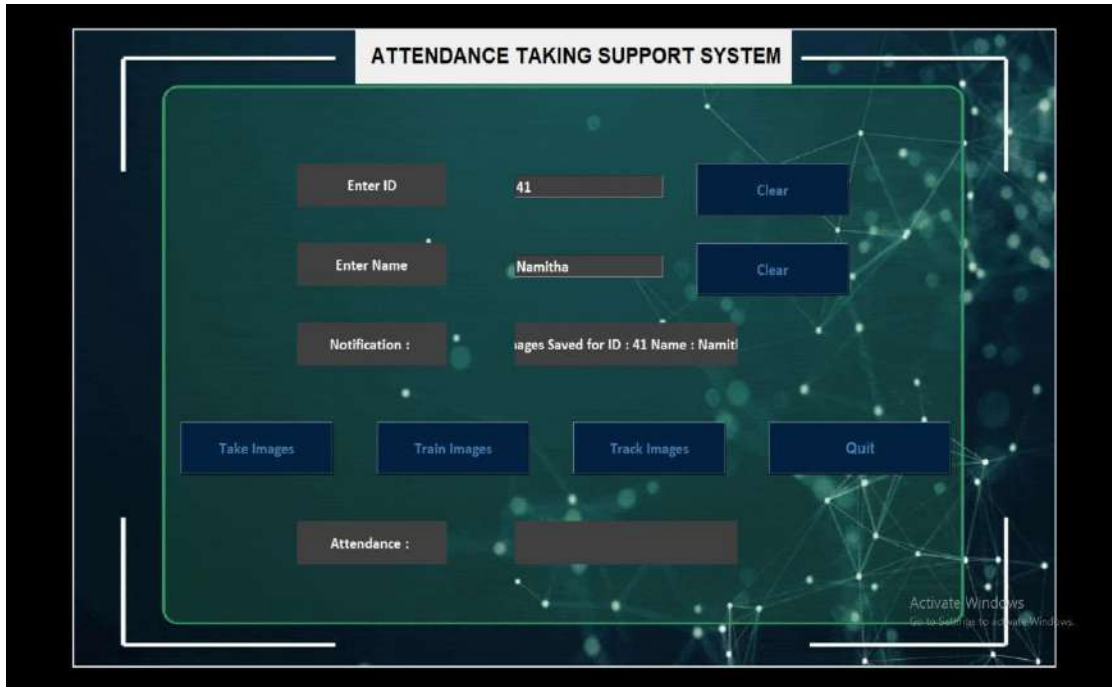


Fig 4 : Images saved with their respective ID and Name

Here in the notification column we can notice that the images are saved with their respective IDs and Name

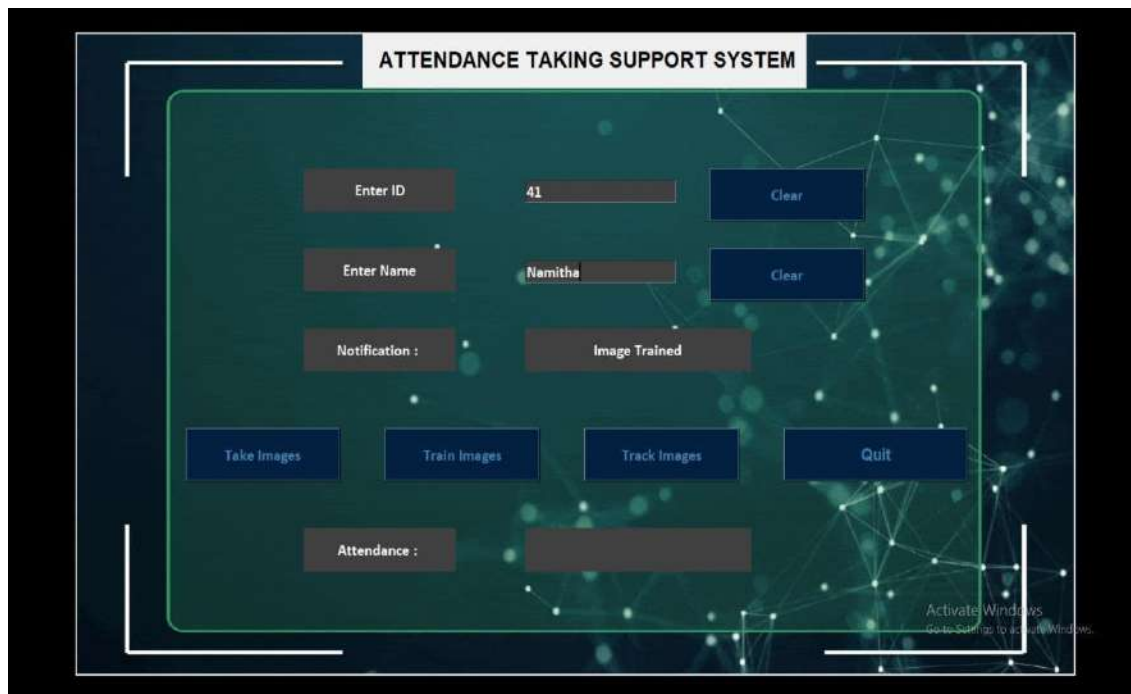


Fig 5 : Images are Trained

When clicked on Train images , the images will get trained, and Notification says “Image Trained”.

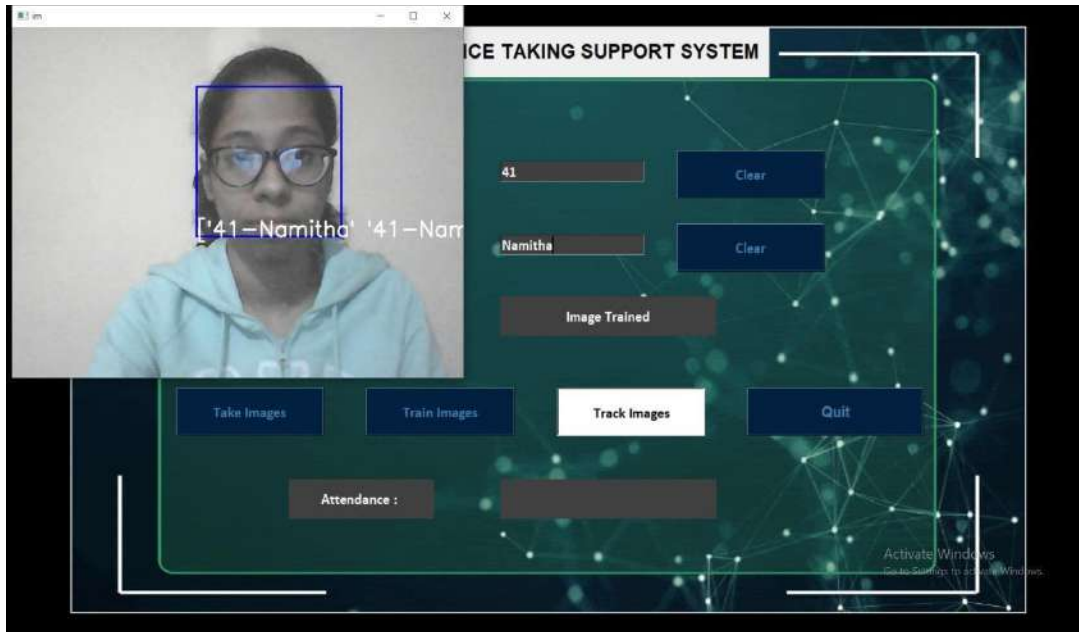


Fig 6 : Tracking the Image

Images can be tracked by clicking on Track images button, we can see here that the student is identified with their id and name.

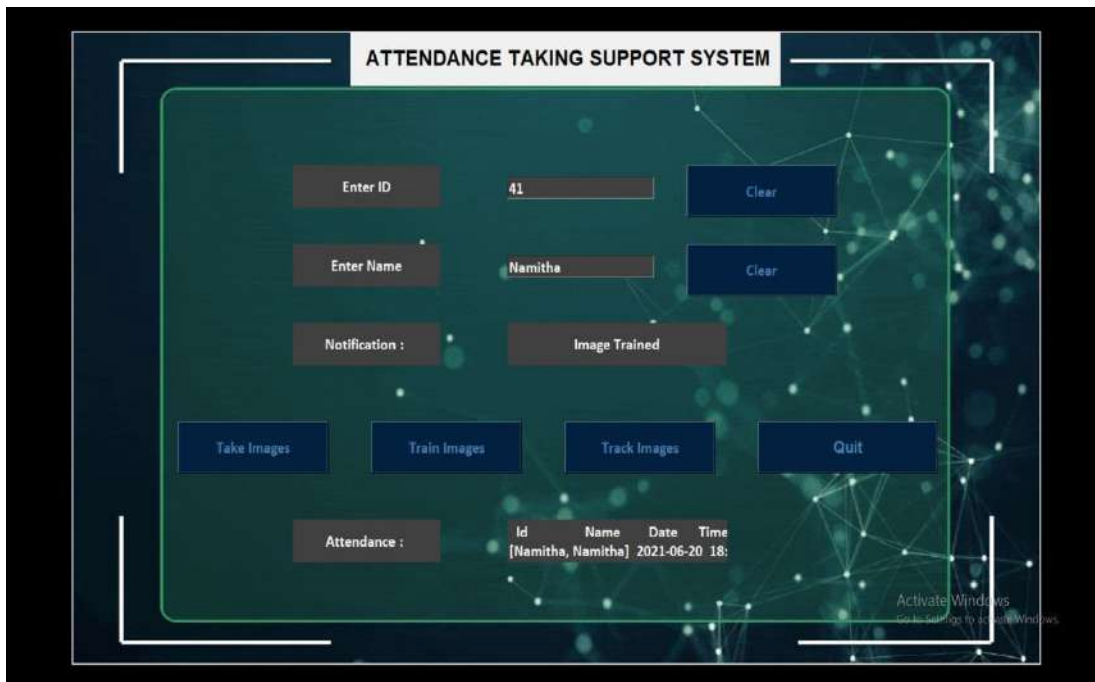


Fig 7 : Attendance is marked

Finally , the attendance of that person is marked and that is notified to us in the attendance section with date and time.

