FACE RECOGINITION BASED ON ATTENDANCE USING MACHINE LEARNING

A PROJECT REPORT

Submitted by

ALEX X	**211420118004**
LOGNATH L	**211420118032**
SATISH G	**211420118044**

in partial fulfillment for the award the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER AND COMMUNICATION ENGINEERING

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

MARCH 2024

BONAFIDE CERTIFICATE

Certified that this project report **"FACE RECOGNITION BASED ON ATTENDANCE USING MACHINE LEARNING"** is the bonafide work of **ALEX X (211420118004), LOGNATH L (211420118032), SATISH G (211420118044)** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr. B.ANNI PRINCY M.E., Ph.D.,	Dr. DHANALAKSHMI M.E,Ph.D.,
HEAD OF THE DEPARTMENT	**SUPERVISOR**
PROFESSOR	ASSOCIATE PROFESSOR,
COMPUTER AND COMMUNICATION ENGINEERING,	COMPUTER AND COMMUNICATION ENGINEERING,
PANIMALAR ENGINEERING COLLEGE,	PANIMALAR ENGINEERING COLLEGE,
NAZARATHPETTAI,	NAZARATHPETTAI,
POONAMALLEE,	POONAMALLEE,
CHENNAI- 600123.	CHENNAI- 600123.

Certified that the above candidate(s) was/ were examined in the End Semester

Mini Project Viva-Voce Examination held on.....

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We would like to extend our heartfelt and sincere thanks to our Directors Tmt. **C.VIJAYARAJESWARI, Dr. C. SAKTHIKUMAR, M.E., Ph.D**., and **Dr.SARANYASREE SAKTHIKUMAR B.E., M.B.A., Ph.D**., for providing us with the necessary facilities for completion of this project.

We also express our gratitude to our Principal **Dr.K.MANI, M.E., Ph.D.,** for his timely concern and encouragement provided to us throughout the course.

We thank the HOD of CCE Department, **Dr. B.ANNI PRINCY, M.E.,Ph.D**., Professor, for the support extended throughout the project.

We would like to thank our supervisor **Dr DHANALAKSHMI, M.E,Ph.D.,** Associate Professor and all the faculty members of the Department of CCE for their advice and suggestions for the successful completion of the project.

ALEX X

LOGNATH L

SATISH G

ABSTRACT

Automatic face recognition (AFR) technologies have made many improvements in the changing world. Smart Attendance using Real-Time Face Recognition is a real-world solution which comes with day to day activities of handling student attendance system. Face recognition-based attendance system is a process of recognizing the students face for taking attendance by using face biometrics based on high - definition monitor video and other information technology. In my face recognition project, a computer system will be able to find and recognize human faces fast and precisely in images or videos that are being captured through a surveillance camera. Numerous algorithms and techniques have been developed for improving the performance of face recognition but the concept to be implemented here is Machine learning. It helps in conversion ofthe frames of the video into images so that the face of the student can be easily recognized for their attendance so that the attendance database can be easily reflected automatically.Upon successful recognition, attendance is automatically recorded, eliminating the need for manual intervention and maintaining a precise digital log of attendance records. Additionally, the system provides real-time reporting and notifications, empowering administrators with valuable insights into attendance patterns and trends. Through its amalgamation of cutting-edge technologies, the FRBAS offers unparalleled accuracy, efficiency, security, and adaptability in attendance management across diverse organizational settings.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 AN OVERVIEW OF PROJECT

According to the previous attendance management system, the accuracy of the data collected is the biggest issue. This is because the attendance might not be recorded personally by the original person, in another word, the attendance of a particular person can be taken by a third party without therealization of the institution which violates the accuracy of the data. For example,student A is lazy toattend a particular class, so student B helped him/her to sign for the attendance which in fact student A didn't attend the class, but the system overlooked this matter due to no enforcement practiced. Supposing the institutionestablish enforcement, it might need to waste a lot of human resource and time which in turn will not be practical at all. Thus, all the recorded attendance in the previous system is not reliable for analysis usage. The second problem of the previous system is where it is too time consuming. Assuming the time taken for astudent to sign his/her attendance on a 3-4 paged name list is approximately 1 minute. In 1 hour, only approximately 60 students can sign their attendance whichis obviously inefficient and time consuming. The third issue is with the accessibility of that information by the legitimate concerned party. For an example, most of the parents are very concerned to track their child's actual whereabouts to ensure their kid really attend the classes in college/school. However in the previous system, there are no ways for the parents to access suchinformation.

1.2 SCOPE OF THE PROJECT

The technology aims in imparting a tremendous knowledge oriented technical innovations these days. Machine learning is one among the interesting domain that enables the machine to train itself by providing some datasets as input and provides an appropriate output during testing by applying different learning algorithms. Nowadays Attendance is considered as an important factor for both the student as well as the teacher of an educational organization. With the advancement of the machine learning technology the machine automatically detects theattendance performance of the students and maintains a record of thosecollected data. This project was carried out to show how a Local Binary Pattern Histogram (LBPH) face recognizer could be used for takingattendance of students.LBPH facial recognizer is a pre-trained facial recognition classifier. If enough dataset are available on the face that is needed to be identified, LBPH can perform facial recognition with high accuracy. Face Recognition Student AttendanceSystem is a desktop application that identifies and verifies student's identities withthe help of a digital image. Once the recognized face matches with the stored image, the attendance is completed and marked in the database for the student. This system will provide an alternative and easier way of taking attendance

CHAPTER 2

LITERATURE SURVEY

1.TITLE: Face Recognition using Eigenfaces

AUTHORS: Matthew Turk and Alex Pentland.

DESCRIPTION: The paper proposes a technique for face recognition based on principal component analysis (PCA), which is used to extract "eigenfaces" from a dataset of face images. The approach transforms face images into a smallset of characteristic feature images, called “eigenfaces”, which are the principal components of the initial training set of face images.These eigenfacescan be usedto represent new face images in a low-dimensional space, allowingfor efficient and accurate recognition. The authors evaluate their approach on a dataset of face images and demonstrate that it can achieve high levels of recognition accuracy with a relatively small number of eigenfaces.

LIMITATIONS:

The method may struggle with variations in pose, illumination, and expression, and may require careful preprocessing of face images to ensure that they are aligned and normalized. The Eigenfaces algorithm may produce false positivesor false negatives where an individual is either incorrectly identified or not recognized at all. This can lead to errors in attendance tracking, which can be problematic in educational or workplace settings.

2.TITLE: Robust Face Recognition via Sparse Representation.

AUTHORS: John Wright, Allen Yang, Arvind Ganesh, Shankar Sastry, andYi Ma

DESCRIPTION: The paper proposes a robust face recognition method based on sparse representation. The proposed method models each face image as a sparse linear combination of training faces and uses a 11-norm minimization algorithm to find the sparsest representation. The coefficients of the sparse linearcombination are obtained through optimization, where the sparsity constraint is imposed on the coefficients to ensure a unique representation of each face image.To recognize a new face image, the authors solve the optimization problem for the sparse coefficients using a computationally efficient algorithm, and then classify the image based on the nearest neighbor in the sparse coefficient space.The proposed method was evaluated on multiple benchmark datasets and achieved state-of-the-art performance in face recognition under various challenging conditions, including variations in lighting, pose, and expression.

LIMITATIONS:

The proposed method may be sensitive to occlusion and may require significant computational resources due to the use of sparse representation. Sparse representation based algorithms may be vulnerable to adversarial attacks, wherean attacker manipulates an input image to cause misclassification by the algorithm.

3.TITLE: A Multi-Task Learning Framework for Facial Expression Recognition and Face Recognition

AUTHORS: Xiaojiang Peng, Xingyu Sun, Jiahao Wei, and Shaoning Pang

DESCRIPTION: The paper proposes a multi-task learning framework for jointly performing facial expression recognition and face recognition. Theauthors observe that facial expressions and identity are closely related, and thattraining a system to perform both tasks simultaneously can improveperformance on both tasks. The proposed framework uses a shared convolutional neural network (CNN) architecture for feature extraction, followed by separate branches for expression recognition and face recognition. The authors evaluate the proposed framework on multiple benchmark datasets and show that it outperforms several state-of-the-art methods for both tasks.

LIMITATIONS:

The proposed method may be sensitive to variations in pose and lighting,and may require significant computational resources due to the use of machine learning techniques. Multi-task learning requires a more complex model architecture than single-task learning, which can increase the risk of overfittingand make the model more difficult to interpret and optimize. Multi- task learningcan make it more difficult to understand how the model is making its predictions, as the model may learn to combine features in non- intuitive ways to perform both tasks simultaneously.

4.TITLE: Improved Face Recognition via Deep Metric Learning and Adversarial Learning.

AUTHORS: Hongguang Zhang, Wei Zhang, and Shaohua Kevin Zhou.

DESCRIPTION: The paper proposes a novel method for improving face recognition performance using deep metric learning and adversarial learning. The authors observe that traditional face recognition methods may struggle with variations in pose, illumination, and expression, and that deep metric learning and adversarial learning can be used to address these challenges. The proposedmethod learns a discriminative embedding space using a deep neural network and adversarial training, which is robust to variations in pose, illumination, and expression. The authors evaluate the proposed method on multiple benchmark face recognition datasets and show that it outperforms several state-of-the-art methods for face recognition.

LIMITATIONS:

The proposed method may require significant computational resources and may be vulnerable to adversarial attacks. The performance of the proposed method is highly dependent on the quality and quantity of the data availablefor training the deep neural networks. If the data is limited or biased the performance of the method may be suboptimal.

5.TITLE: Deep Face Recognition

AUTHORS: Rajesh, Maneesha, Shaik Hafeez, Hari Krishna.

DESCRIPTION: The paper proposes a machine learning-based approach to face recognition, which achieves state-of-the-art performance on several benchmark face recognition datasets. The authors observe that traditional face recognition methods may struggle with variations in pose, illumination, and expression, andthat machine learning can be used to address these challenges. The proposed method learns a hierarchical representation of face images using a deep convolutional neural network (CNN), which is trained on a large-scale dataset of face images. The authors evaluate the proposed method on several benchmark face recognition datasets, including the LFW dataset, and show that it outperforms several state-of-the-art methods for face recognition.

LIMITATIONS:

The training of deep convolutional neural networks for face recognitioncan require significant computational resources and may be difficult to implement without access to specialized hardware. The use of deep face recognition models raises privacy concerns, particularly when the models are used for surveillance or identification purposes. Deep face recognition modelscan be vulnerable to adversarial attacks. Deep face recognition technology is still not 100% accurate

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

The existing attendance systems mentioned in the text are the portable fingerprint device system and the RFID-based system. In the portable fingerprint device system, students need to pre-configure their fingerprints on the device and record their fingerprint on the device during lecture hours or before to mark their attendance. This system may distract students during lectures, and it also requires prior configuration of fingerprints. In the RFID-based system, students need to carry a Radio Frequency Identity Card and place the ID on the card reader to recordtheir presence. However, there is a possibility of fraudulent access where studentsmay use other students' ID to mark their attendance.

3.1.1 PROBLEM DEFINITION

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

3.2 PROPOSED SYSTEM

The task of the proposed system is to capture the face of each student and to store it in the database for their attendance. The face of the student needs to be captured in such a manner that all the feature of the students' face needs to be detected. There is noneed for the teacher to manually take attendance in the class because the system records a video and through further processing steps the face is being recognized and the attendance database is updated. This system is developed using python idle vision. Originally developed by Intel, it was later supported by Willow Garage then Itsees (which was later acquired by Intel). The library is cross- platform and freeforuse under the open-source BSD license. Python is dynamically typed and garbage-collected. It supports multiple programming including structures object oriented and functional programming. Python is often described as a batteries included language due to its comprehensive standard library.

3.2.1 ADVANTAGES

Automated Attendance System (AAS) is a process to automatically estimate the presence orthe absence of the student in the classroom by using face recognition technology.It is also possible to recognize whether the student is sleeping or awake during the lecture andit can also be implemented in the exam sessions toensure the presence of the student.The presence of the students can be determined by capturing theirfaces on to a high-definition monitor live web camera streaming service.It becomes highly reliable for the machine to understand the presence ofall the students inthe classroom.

CHAPTER 4

REQUIREMENTSPECIFICATIONS

4.1 INTRODUCTION

One of the innovative dimensions of this study was the usage of feature selection algorithms to choose best features that improve the classification accuracy as well as reduce the execution time of the diagnosis system. The fundamental cause for getting ready this system is to provide a standard perception into the evaluation and necessities of the present device or state of affairs and for figuring out the runningtraits of the device. This Document performs a important position in the improvementlifestyles cycle (SDLC) because it describes the entire requirement of the device. It issupposed to be used via way of means of the builders and may be theprimary for the duration of checking out phase. Any modifications made to the necessities in the destiny will should undergo formal alternate approval process.

Functional Requirements:

Input: The major inputs are student id and name , their respective photos should be taken in real time, then it will be trained using LBPG algorithm and store it in a yml file. After that we track images in real time in a camera .

Output: The major outputs of the system are in excel sheet where attendance of the students are recorded and saved. This information can be also sent through mail to respective HOD or class in-charges.

4.2 HARDWARE AND SOFTWARE SPECIFICATION

4.2.1 HARDWAREREQUIREMENTS

- Processor - i3, i5, i7
- Speed - 1.60 GHz
- RAM - 4 GB,8 GB
- Hard Disk - 260 GB

4.2.2 SOFTWARE REQUIREMENTS

- Operating System - Windows 7/8/10
- Language - Python
- Tool - PyCharm

PYTHON

Python could be an interpreted, high-level and general programing language. Python is a style philosophy emphasizes code readability with its notable use of great indentation. Its language constructs and object-oriented approach aim to assist programmers write clear, logical code for tiny and large-scale projects. Python is commonly represented as A battery enclosed language because of its comprehensive commonplace library. Python is a multi-paradigm programming language, Object-oriented programming and structured programming language are absolutely supported and plenty of its options support practical programming and aspect- oriented programming (including by metaprogramming and metaobjects (magic methods)). several different paradigms are supported via extensions, together with design by contract and logic programming. Python uses dynamic writing and a mix of reference numeration and a cycle-detecting dustman for

memory management. It also options dynamic name resolution (late binding), that binds technique and variable names throughout program execution.

Python is an interpreter, object-oriented, high-level programing language with dynamic semantics. Its high-level inbuilt information structures, combined with dynamic typing and dynamic binding build it terribly engaging for fast Application Development, still as to be used as a scripting or glue language to attach existing partstogether. Python' simple, straightforward to find out syntax emphasizes readability and so reduces the cost of program maintenance. Python supports modules and packages,which inspires program modularity and code reuse. The Python interpreter and also the intensive commonplace library are on the market in supply or binary type at no cost for all major platforms and may be freely distributed. Often, programmers fall soft on with Python thanks to the magnified productivity it provides. Since there's nocompilation step, the edit-test-debug cycle is implausibly fast. Debugging Python programs is easy, a bug or dangerous input can ne'er cause a segmentation fault. Instead, once the interpreter discovers an error, it raises an exception. once the programdoesn't catch the exception, the interpreter prints a stack trace. A supply level computerprogramme permits examination of native and world variables, analysis of discretionalexpressions, setting breakpoints, stepping through the code a line at a time, and then on. The rightger is written in Python itself, testifying to Python' introverted power. Onthe opposite hand, usually the fastest thanks to debug a program is to feature a numberof print statements to the source, the quick edit-test-debug cycle makes this easy approach terribly effective.

Python is a dynamic programming language which supports several differentprogramming paradigms as follows:

Procedural programming

Object oriented programming

Functional programming

Standard: Python byte code is executed in the Python interpreter (similar to Java) platform independent code

Extremely versatile language Website development, data analysis,server maintenance, numerical analysis,

Syntax is clear, easy to read and learn (almost pseudo code)

Common language

Intuitive object-oriented programming

Full modularity, hierarchical packages

Comprehensive standard library for many tasks

Big community

Simply extendable via C/C++, wrapping of C/C++ libraries

Focus: Programming speed

INTRODUCTION TO PYTHON

Guido van Rossum began functioning on Python within the late 1980s, as a successor to the fundamental principle programming language, and initial dischargedit in 1991 as Python 0.9.0. Python two.0 was released in 2000 and introduced new features, akin to list comprehensions and a trash collection system victimization reference count and was discontinued with version 2.7.18 in 2020. Python 3.0 was released in 2008 and was a serious revision of the language that's notfully backward- compatible and far Python 2 code doesn't run unqualified on Python

3. Python 2.0 was released on sixteen Gregorian calendar month 2000, with several major new options, together with a cycle-detecting refuse collector and support for Unicode.

Python three.0 was discharged on 3 Gregorian calendar month two008. it had beena serious revision of the language that's not fully backward-compatible. several of its major features were backported to Python 2.6.x and 2.7.x version series. Releases of Python 3 embody the 2to3 utility, that automates (at least partially) the interpretation of Python 2 code to Python 3. Python 2.7's end-of-life date was at the start set at 2015then delayed to 2020 out of concern that an outsized body of existing code couldn't simply be forward-ported to Python 3. No a lot of security patches or different enhancements are going to be discharged for it. With Python 2's end-of-life, solely Python 3.6.x and later are supported. Python 3.9.2 and 3.8.8 were speeded up as all versions of Python (including 2.7) had security issues, resulting in potential remote code execution and internet cache poisoning.

DESIGN PHILOSOPHY AND FEATURES

The standard library has 2 modules (itertools and functools) that implement practical tools borrowed from Haskell and normal ML. instead of having all of its practicality designed into its core, Python was designed to be extremely extensile (with modules). This compact modularity has created it significantly standard as a method of adding programmable interfaces to existing applications. Python attempts for a simpler, less-cluttered syntax and synchronic linguistics whereasgiving developers a alternative in their writing methodology. Python developers striveto avoid premature optimization, and reject patches to non-critical elements of the CPython reference implementation that may provide marginal will increase in speed at the price of clarity. once speed is important, a Python technologist will move time-critical functions to extension modules written in languages reminiscent of C, or use PyPy, a just-in-time compiler. Cython is additionally available, that interprets a Pythonscript into C and makes direct C-level API calls into the Python interpreter.

SYNTAX AND SEMANTICS, INDENTATION

Python is supposed to be a simply clear language. Its format is visually uncluttered, and it usually uses English keywords wherever different languages use punctuation. not like several other languages, it doesn't use crisp brackets to delimit blocks, and semicolons once statements are allowed however are rarely, if ever, used.it's fewer syntactical exceptions and special cases than C or Pascal. Python uses whitespace indentation, instead of curly brackets or keywords, to delimitblocks.

TYPING

Python makes use of duck typing and has typed gadgets however untyped variable names. Type constraints aren't checked at assemble time; rather, operations on an item may also fail, signifying that the given item isn't always of a appropriate type. Despite being dynamically-typed, Python is strongly-typed, forbidding operations that aren't well-defined (for example, including a range of to a string) in preference to silently trying to make feel of them. Python permits programmers to outline their very own sorts the use of training, which can be most usually used for item-orientated programming. New times of training are builtthrough calling the class (for example, SpamClass () or EggsClass ()), and the trainingare times of the metaclass type (itself an example of itself), permittingmetaprogramming and reflection.

LIBRARIES

Python’s giant standard library, usually cited united of its greatest strengths, provides tools suited to several tasks. For Internet-facing applications, many standard formats and protocols corresponding to MIME and hypertext transfer protocol are supported. It includes modules for making graphical user

interfaces, connecting to relative databases, generating pseudorandom numbers, arithmetic with arbitrary-precision decimals, manipulating regular expressions, and unit testing. Some components of the quality library are lined by specifications (for example, the net Server entranceway Interface (WSGI) implementation wsgiref follows life 333), however most modules are not. they're given by their code, internaldocumentation, and check suites. However, as a result of most of the quality libraryis cross-platform Python code, solely a number of modules want sterilization or redaction for variant implementations. As of March 2021, the Python Package Index(PyPI), the official repository for third-party Python software, contains over 290,000packages with a large vary of functionality, including: Automation

- Data analytics
- Databases
- Documentation
- Graphical user interfaces
- Image processing
- Machine learning
- Mobile App
- Multimedia
- Computer Networking
- Scientific computing
- System administration
- Test frameworks

- Text processing
- Web frameworks
- Web scraping

DEVELOPMENT ENVIRONMENTS

Most Python implementations (including CPython) embody a read–eval– print loop (REPL), allowing them to perform as a instruction interpreter that the userenters statements consecutive and receives results immediately. different shells, as well as IDLE and IPython, add further skills like improved auto- completion, session state retention and syntax highlighting. still as normal desktop integrated development environments, there are internet browser-based IDEs; Sage science (intended for developing science and math-related Python programs); PythonAnywhere, a browser-based IDE and hosting environment; and cover IDE, anadvertisement Python IDE action scientific computing.

ANACONDA NAVIGATOR - JUPYTER NOTEBOOK

Anaconda may be a distribution of the Python associate degreed R programming languages for scientific computing (data science, machine learning applications, large-scale knowledge processing, prophetical analytics, etc.), that aimsto change package management and deployment. The distribution includes data- science packages appropriate for Windows, Linux, and macOS. it's developed and maintained by boa, Inc., that was based by Peter Wang and Travis Oliphant in 2012.As a boa, Inc. product, it is additionally referred to as boa Distribution or boa Individual Edition, whereas different merchandise from the corporate are boa Team

Edition and Anaconda Enterprise Edition, each of which aren't free. The subsequent applications are on the market by default in Navigator:

- JupyterLab
- Jupyter Notebook
- QtConsole
- Spyder
- Glue
- Orange
- Visual Studio Code

Jupyter Notebook (formerly IPython Notebooks) may be a net-based interactive machine setting for making Jupyter notebook documents. The "notebook" term will informally build respect to many alternative entities, chiefly the Jupyter web application, Jupyter Python web server, or Jupyter document format counting on context. A Jupyter Notebook document is a JSON document, following a versioned schema, containing an ordered list of input/output cells which might contain code, text (using Markdown), mathematics, plots and made media, sometimes ending with the ".ipynb" extension. A Jupyter Notebook will be born- again to variety of open commonplace output formats (HTML, presentation slides, LaTeX, PDF, Restructured Text, Markdown, Python) through "Download As" within the net interface, via the nbconvert library or "jupyter nbconvert" command interfaceduring a shell. To change visualisation of Jupyter notebook documents on the web, the nbconvert library is provided as a service through NbViewer which can take a URL to any publically on the market notebook document, convert it to markup language on the fly and show it to the user.

Jupyter Notebook provides a browser-based REPL engineered upon variety ofcommon ASCII text file libraries:

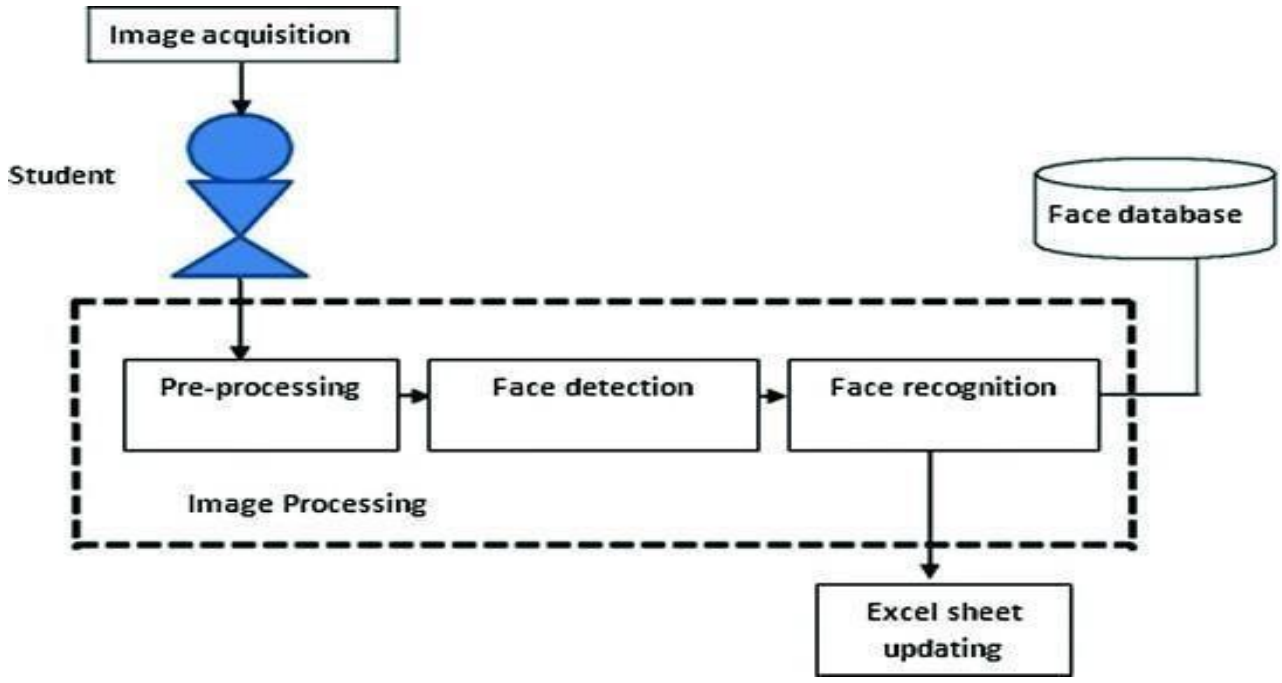
- IPython
- ØMQ (ZeroMQ)
- Tornado (web server)
- jQuery
- Bootstrap (front-end framework)
- MathJax

Jupyter Notebook will hook up with many kernels to permit programming inseveral languages. By default, Jupyter Notebook ships with the IPython kernel. As of the 2.3 unleash (October 2014), it absolutely was forty nine Jupyter-compatible kernels for several programming languages, together with Python, R, Julia and Haskell. The Notebook interface was additional to IPython within the 0.12 unleash (December 2011), renamed to Jupyter notebook in 2015 (IPython 4.0 – Jupyter 1.0).

CHAPTER 5

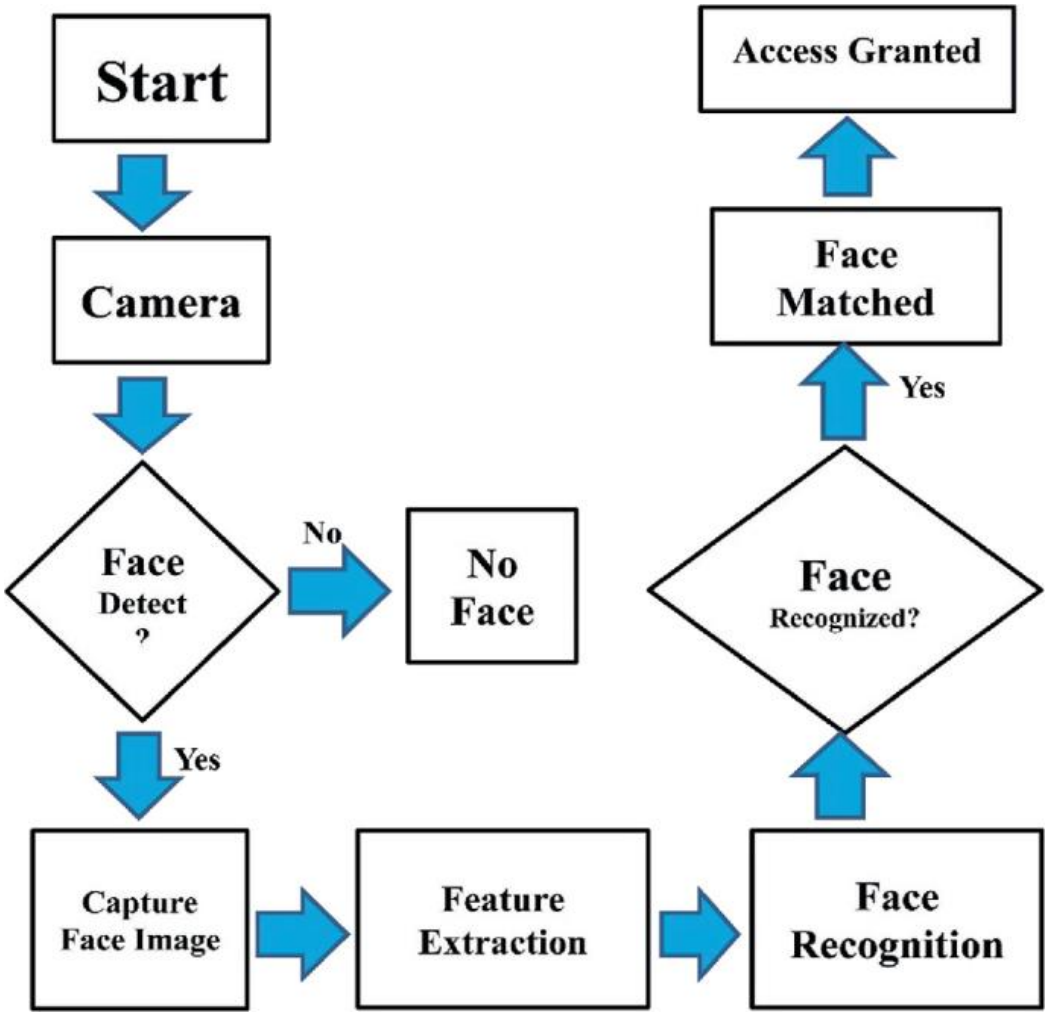
SYSTEM DESIGN

5.1 ARCHITECTURE DIAGRAM



EXPLANATION:

The architecture diagram for the Facial Recognition-Based Attendance System showcases a multi-stage process. Initially, input images or video streams undergo face detection using algorithms like Haar cascades or deep learning models. Extracted faces then undergo feature extraction, where techniques like Principal Component Analysis (PCA) condense facial features into a compact representation. Machine learning models like Support Vector Machines (SVM) or convolutional neural networks (CNNs) handle recognition, matching features with pre-registered identities. Successful matches trigger attendance recording, maintaining digital logs. Real-time reporting and notifications offer insights.



WORKING ARCHITECTURE OF HAAR CASCADE ALGORITHM

EXPLAINATION

The Haar Cascade algorithm operates by utilizing rectangular features at different scales and locations within an image. These features are weighted and summed to determine whether a region contains a face or not, making it a fundamental technique for face detection in computer vision applications.

5.2 UML DIAGRAMS

5.2.1 USECASE DIAGRAM

A Use case Diagram is used to present a graphical overview of the functionality provided by a system in terms of actors, their goals and any dependencies between those use cases. A Use Case describes a sequence of actions that provided something of unmeasurable value to an actor and is drawn as a horizontal ellipse. An actor is a person, organization or external system that plays a role in one or more interaction.use case diagram can be shown in fig5.1

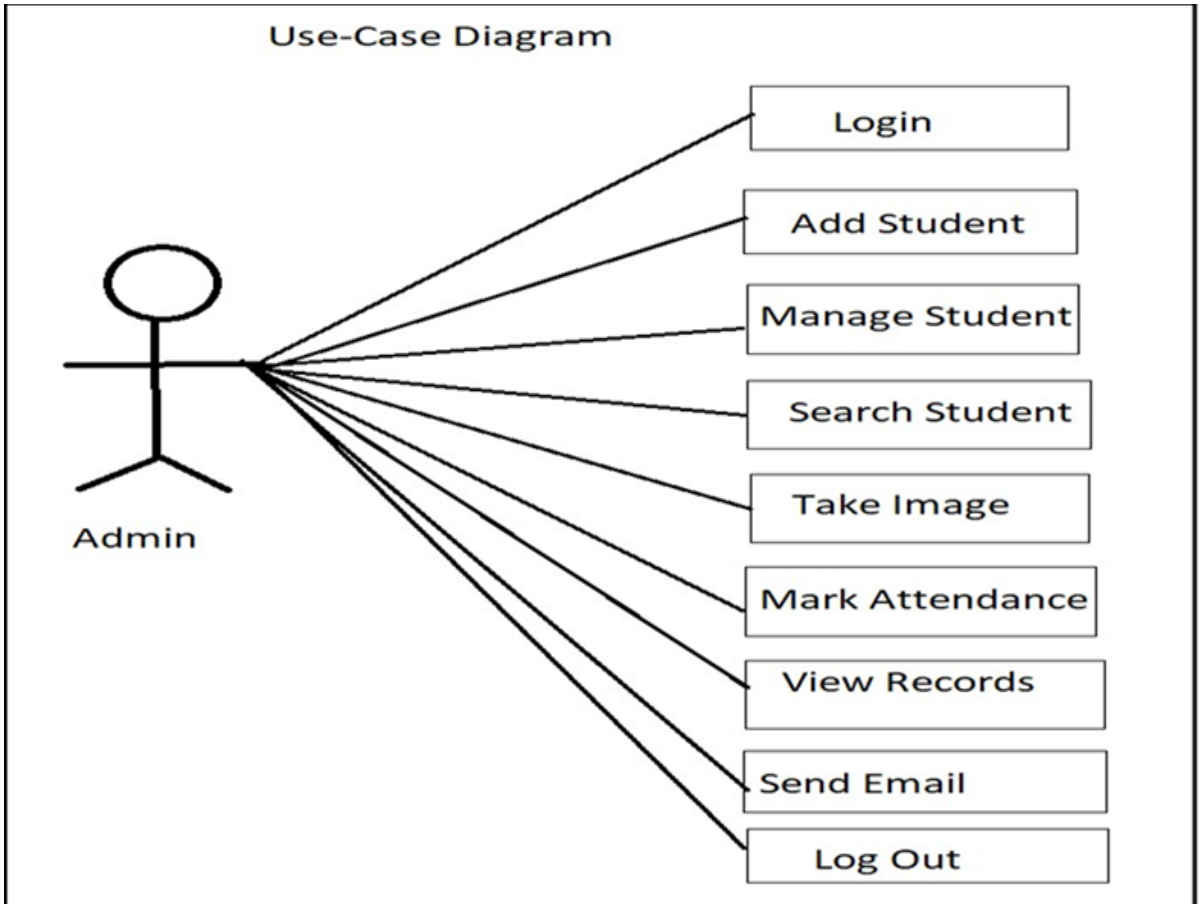


Fig 5.1 Use case Diagram

5.2.2 SEQUENCE DIAGRAM

A Sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of Message Sequence diagrams are sometimescalled event diagrams, event sceneries and timing diagram.sequence diagram can be shown in fig5.2

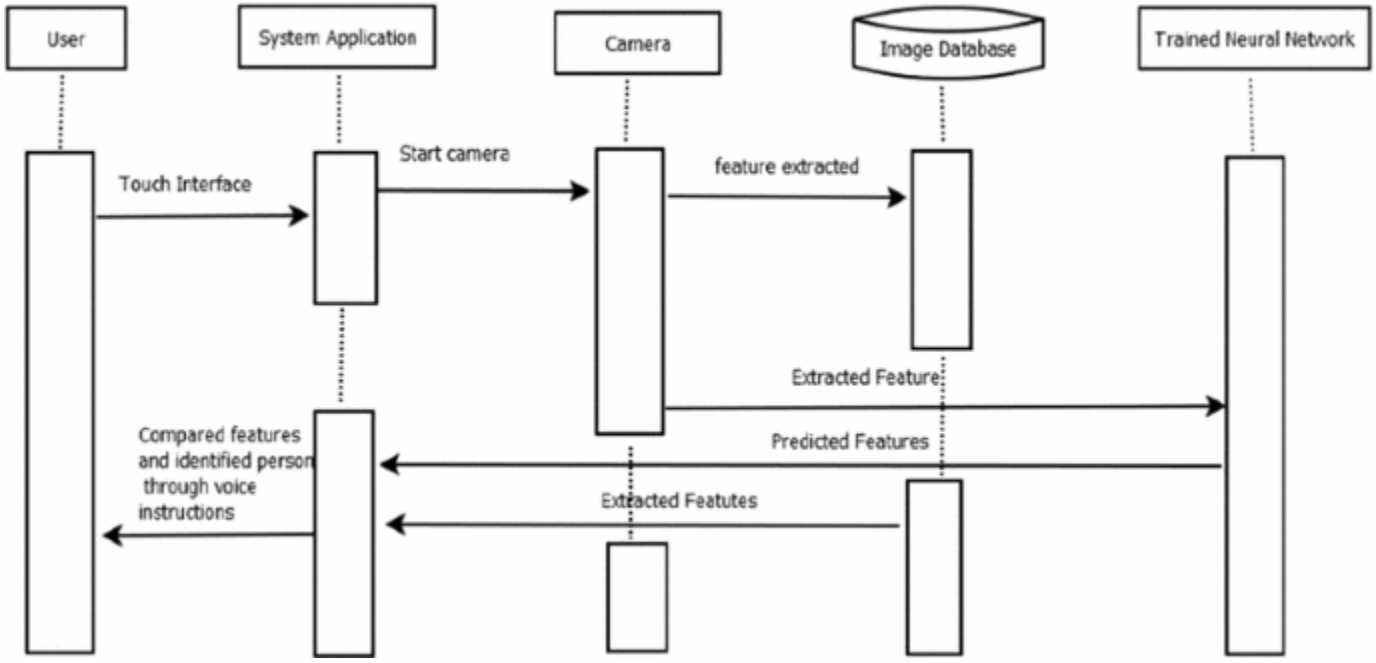


Fig 5.2 Sequence Diagram

5.3 CLASS DIAGRAM

A Class diagram in the Unified Modelling Language 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.class

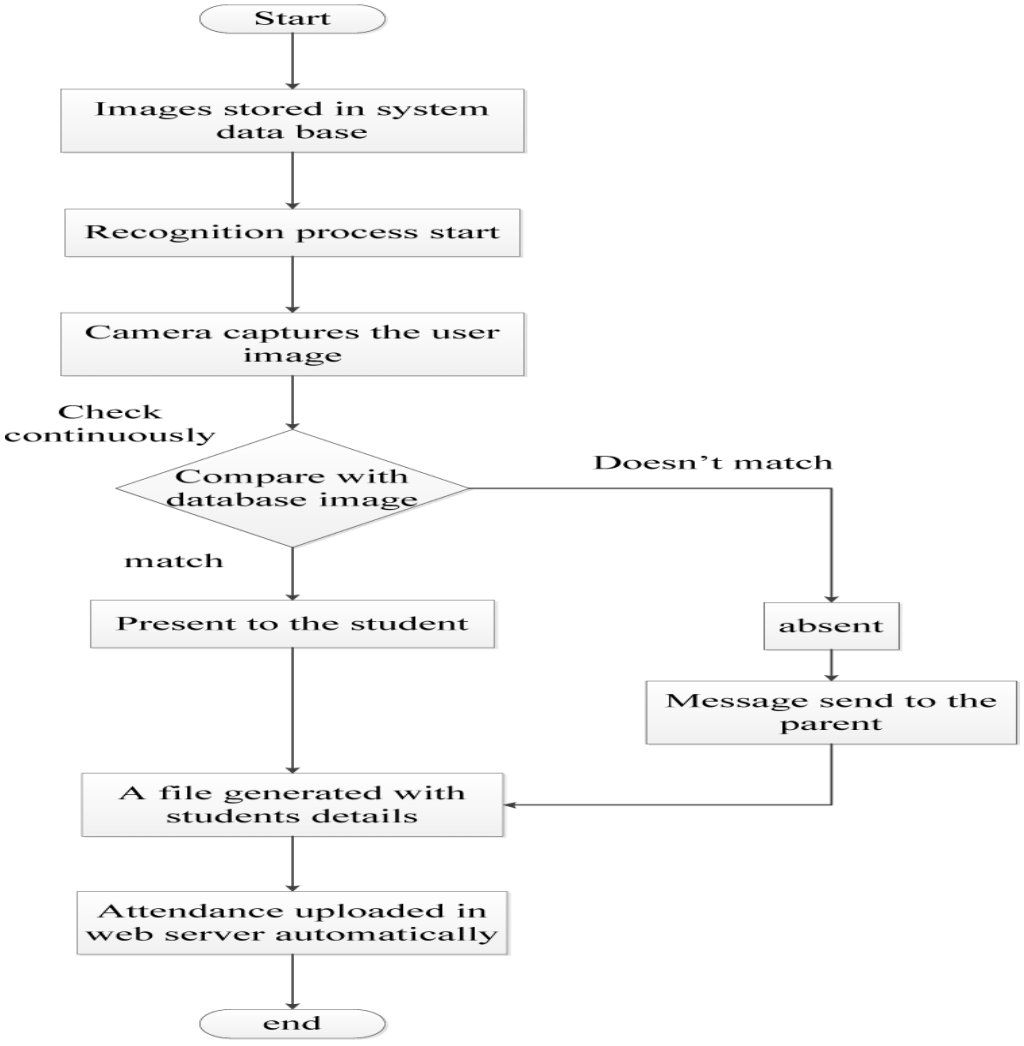
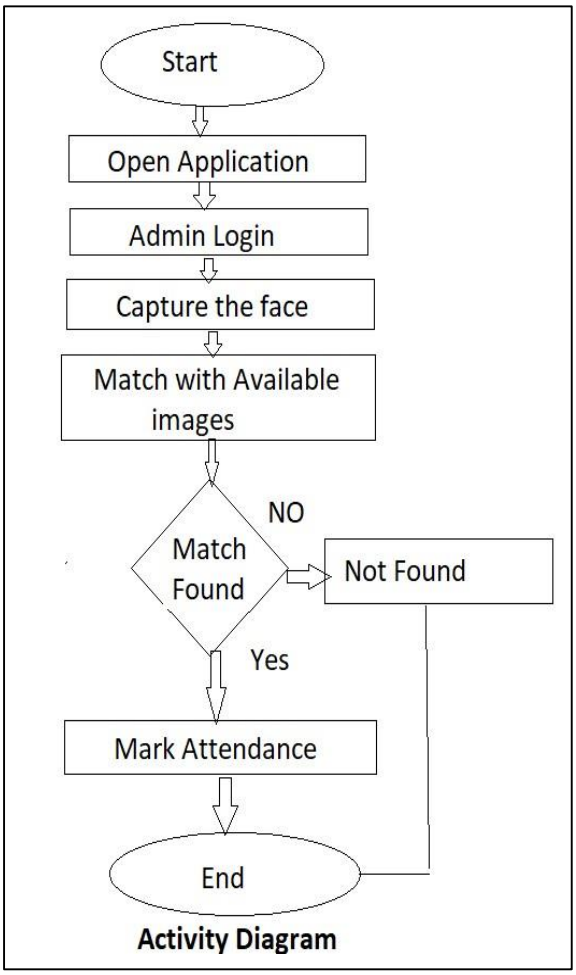


Fig 5.3 Class Diagram

5.3.1 ACTIVITY DIAGRAM

Activity diagram is a graphical representation of workflows of stepwise activities and actions with support for choice, iteration and concurrency. An activity diagram shows the over all flow of control.activity diagram can be shown in fig5.4



:

Fig 5.4 Activity Diagram

5.3.2 DATA FLOWDIAGRAM

A Data Flow Diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its aspects. It is a preliminary step used to create an overview of the system.DFD diagram can be shown in fig5.5

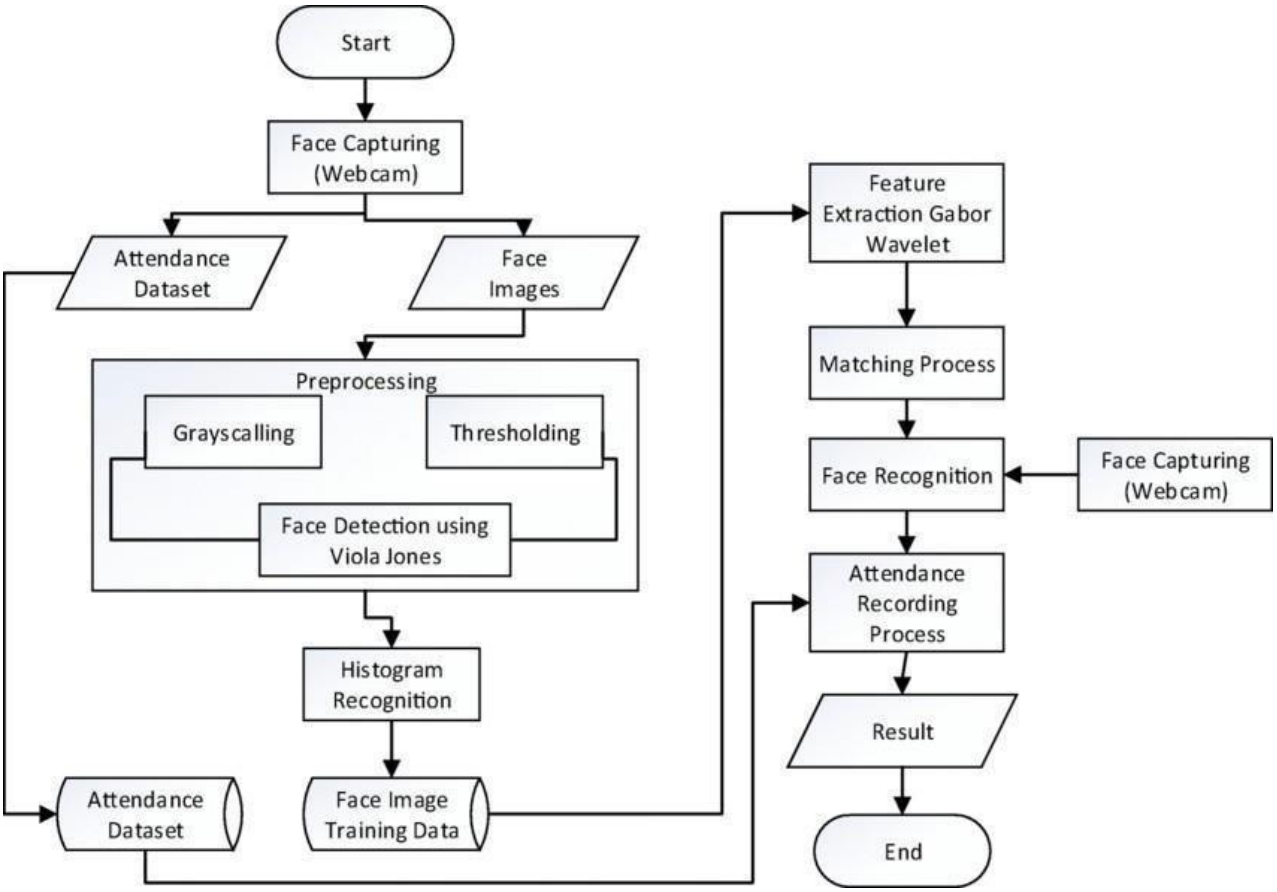


FIGURE 5.5

5.4 MODULES:

IMAGE CAPTURE AND IMAGE PROCESSING

HAAR CASCADE ALGORITHM

STUDENT ATTENDANCE

FACE RECOGNITION ALGORITHM AND THE LBPH

5.4.1 IMAGE CAPTURE AND IMAGE PROCESSING:

We need some HD camera in order to get results. We can capture the images from the video stream or by capturing each and every image from the webcam manually. Doing the frame capture from the stream of video will give us results in less time but we won't be able to capture the face properly in case we lose light or something and if the face is not captured properly. Digital image processing is the use of a digital computer to process digital images through an algorithm. As a subcategory or field of digital signal processing, digital image processing has many advantages over analog image processing. It allows a much wider range of algorithms to be applied to the input data and can avoid problems such as the build- up of noise and distortion during processing. Since images are defined over two dimensions (perhaps more) digital image processing may be modeled in the form of multidimensional systems. The generation and development of digital image processing are mainly affected by three factors: first, the development of computers; second, the development of mathematics (especially the creation and improvement of discrete mathematics theory); third, the demand for a wide range of applications in environment, agriculture, military, industry and medical science has increased.

5.4.2 HAAR CASCADE ALGORITHM:

Object Detection using Haar feature-based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones in their paper, “Rapid Object Detection using a Boosted Cascade of Simple. 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. Initially, the algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then we need to extract features from it. They are just like our convolutional kernel. Each feature is a single value obtained by subtracting sum of pixels under white rectangle from sum of pixels under black rectangle. Now all possible sizes and locations of each kernel is used to calculate plenty of features. For each feature calculation, we need to find sum of pixels under white and black rectangles. To solve this, they introduced the integral images. It simplifies calculation of sum of pixels, how large may be the number of pixels, to an operation involving just four pixels. It makes things super-fast. OpenCV comes with a trainer as well as detector. If you want to train your own classifier for any object like car, planes etc. you can use OpenCV to create one. OpenCV already contains many pre-trained classifiers for face, eyes, smile etc. Those XML files are stored in the folder. After that input image is loaded in grayscale mode This Algorithm mainly has following functionality.

5.4.3 STUDENT ATTENDANCE:

Using HAAR Algorithm it recognizes it’s a face and Take multiple images of student and trained with that face, training only done for one of each student. After that his profile has generated, so now we can track them they are in class or not using camera. If the student is there the system automatically

gives attendance to the particular student and saved his ID, name and Time. This helps staffs who are coming late or earlier. After updating in the attendance sheet. Send mail option is to send the attendance details to the particular staffs. This system helps to reduce the time and complexity of taking attendance manually.

5.4.4 FACE RECOGNITION ALGORITHM AND LBPH:

Image capture: The first step is to capture an image of the person's face using a camera or webcam. The image should be of good quality, with enough resolutionto capture the necessary facial features.

Face detection: The algorithm will then detect the face in the captured image usingthe Haar Cascade Classifier or a similar object detection algorithm. This involvesscanning the image for regions of interest that match the pre-trained model of facial features.

Feature extraction: Once the face has been detected, the algorithm will extract thenecessary features from the image using the LBPH algorithm or another suitable technique. This involves analyzing the texture of the face and generatinga uniquerepresentation of the person's facial features.

Face recognition: The system will then compare the extracted features with the database of previously registered faces to identify the individual. If the person's face is recognized, their attendance will be marked.

Database management: The system will store the data of the identified individuals,including their name and attendance record, in a database for future reference.

Attendance tracking: The system can be set up to generate reports on attendance data, including

CHAPTER 6

SOFTWARE TESTING

6.1 CODING

Once the design aspect of the system is finalized, the system enters into the coding and testing phase. The coding phase brings the actual system into action by converting the design of the system into the code in a given programming language. Therefore, a good coding style has to be taken whenever changes are required it easily screwed into the system.

6.2 CODING STANDARDS

Coding standards are guidelines to programming that focuses on the physical structure and appearance of the program. They make the code easier to read, understand and maintain. This phase of the system actually implements the blueprint developed during the design phase. The coding specification should be in such a way that any programmer must be able to understand the code and can bring about changes whenever felt necessary.

6.2.1 NAMING CONVENTIONS

Naming conventions of classes, data member, member functions, procedures etc., should be self-descriptive. One should even get the meaning and scope of the variable by its name. The conventions are adopted for easy understanding ofthe intended message by the user. So, it is customary to follow the conventions. Class names are problem domain equivalence and begin with capital letter and have mixed cases. Member function and data member name begins with a lowercase letter with each subsequent letters of the new words in uppercase and the rest of letters in

6.2.2 VALUE CONVENTIONS

Value conventions ensure values for variable at any point of time. This involves the following:

Proper default values for the variables.

Proper validation of values in the field.

Proper documentation of flag values.

6.2.3 SCRIPT WRITING AND COMMENTING STANDARD

Script writing is an art in which indentation is utmost important. Conditional and looping statements are to be properly aligned to facilitate easy understanding. Comments are included to minimize the number of surprises that could occur when going through the code.

6.3 TESTING

Testing is performed to identify errors. It is used for quality assurance. Testing is an integral part of the entire development and maintenance process. The goal of the testing during phase is to verify that the specification has been accurately and completely incorporated into the design, as well as to ensure the correctness of the design itself. For example, the design must not have any logic faults in the design is detected before coding commences, otherwise thecost of fixing the faults will be considerably higher as reflected.

Testing is one of the important steps in the software development phase. Testing checks for the errors, as a whole of the project testing involves the following test cases:

Static analysis is used to investigate the structural properties of the Source code.

Dynamic testing is used to investigate the behavior of the source code by executing the program on the test data.

6.4 TYPES OF TESTING

6.4.1 UNIT TESTING

Unit testing is conducted to verify the functional performance of each modular component of the software. Unit testing focuses on the smallest unit of the software design (i.e.), the module. The white-box testing techniques were heavily employed for unit testing.

6.4.2 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions testedare 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.

6.4.3 SYSTEM TESTING

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.

6.4.4 PERFORMANCE TESTING

The Performance test ensures that the output be produced within the time limits, and the time taken by the system for compiling, giving response to the users and request being send to the system for to retrieve the results.

6.4.5 INTEGRATION TESTING

Integration testing is a systematic technique for constructing the program structure, while at the same time conducting tests to uncover error associated with interfacing. The following are the types of Integration Testing: -

- Top-down Integration
- Bottom-up Integration

Top-down Integration

This method is an incremental approach to the construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the program module. The module subordinates to the main program module are incorporated into the structure in either a depth first or breathe first manner.

Bottom-up integration

This method begins the construction and testing with the modules at the lowest level in the program structure. Since the modules are integrated from the bottom up, processing required for modules subordinate to a given level is always available and the need for stubs is eliminated.

6.4.6 PROGRAM TESTING:

The logical and syntax errors have been pointed out by program testing. A syntax error is an error in a program statement that in violates one or more rules ofthe language in which it is written. An improperly defined field dimension or omitted keywords are common syntax error. These errors are shown through error messages generated by the computer. Condition testing method focuses on testing each condition in the program the purpose of condition test is to deduct not only errors in the condition of a program but also other errors in the program.

6.4.7 VALIDATION TESTING

At the culmination of integration testing, software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of software test-validation testing begins. Validation testing can be defined in many ways, but a simple definition is that validation succeeds when the software functions in manner that is reasonably expected by the customer. Software validation is achieved through a series of black box tests that demonstrateconformity with requirement. After validation test has been conducted, one of two conditions exists.

The function or performance characteristics confirm to specifications and are accepted

A validation from specification is uncovered and a deficiency created.

6.4.8 USER ACCEPTANCE TESTING

User acceptance of the system is key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keepingin touch with prospective system and user at the time of developing and making changes whenever required.

6.5 WHITEBOX AND BLACKBOX TESTING

WHITEBOX TESTING

This testing is also called as Glass box testing. In this testing, by knowing the specific functions that a product has been design to perform test can be conducted that demonstrate each function is fully operational at the same time searching for errors in each function. It is a test case design method that uses the control structure of the procedural design to derive test cases. Basis path testing isa white box testing.

Basis path testing:

Flow graph notation

Cyclometric complexity

Deriving testcases

Graph matrices Control

BLACKBOX TESTING

In this testing by knowing the internal operation of a product, test can be conducted to ensure that “all gears mesh”, that is the internal operation performs according to specification and all internal components have been adequately exercised. It fundamentally focuses on the functional requirements

Of the software.

The steps involved in black box test case design are:

Graph based testing methods

Equivalence partitioning

Boundary value analysis

Comparison testing

6.6 SOFTWARE TESTING STRATEGIES

A software testing strategy provides a road map for the software developer. Testing is a set activity that can be planned in advance and conducted systematically. For this reason, a template for software testing a set of steps into which we can placespecific test case design methods should be strategy should have thefollowing characteristics:

Testing begins at the module level and works "outward" toward the integration of the entire computer-based system.

Different testing techniques are appropriate at different points in time.

The developer of the software and an independent test group conducts testing.

Testing and Debugging are different activities but debugging must be accommodated in any testing strateg

CHAPTER 7

CONCLUSION & FUTURE WORK

CONCLUSION

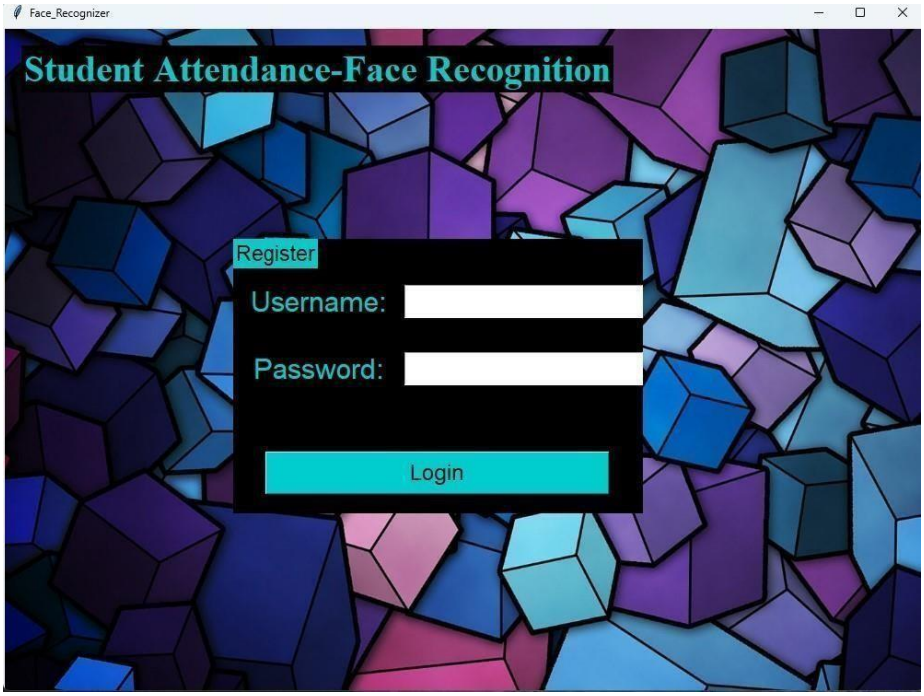
Capturing the images from camera or cc camera and applying techniques face detection and recognition can decrease the manual work from human and increasethe security safety, taking the decision from this recognition result. Based on thisface detection and recognition can used in implement so many application like automatic attendances system based on face recognition, worker attendances, security, safety, police application like finding thief in image that help to catchingthief. In this system we have implemented an attendance system for a lecture, section or laboratory by which lecturer or teaching assistant a record student's attendance. It saves time and effort, especially if it is a lecture with huge number of students. This attendance system shows the use of facial recognition techniquesfor the purpose of student attendance and for the further process this record of student can be used in exam related issues.

FUTURE SCOPE

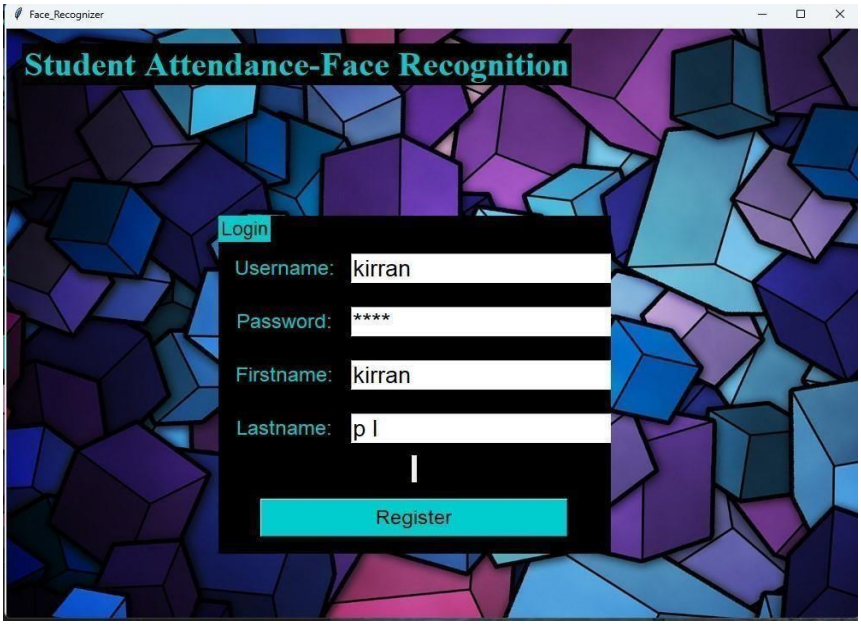
In the future work, we plan to shift the attendance checking scene into the virtual one in order to extend the on-site classroom attendance checking to the attendance checking in the online learning environment. We also hope to achievecontinuous non-disturbance attendance checking in order to be suitable for the applications of multiple learning scenarios.

APPENDIX B- SNAPSHOTS

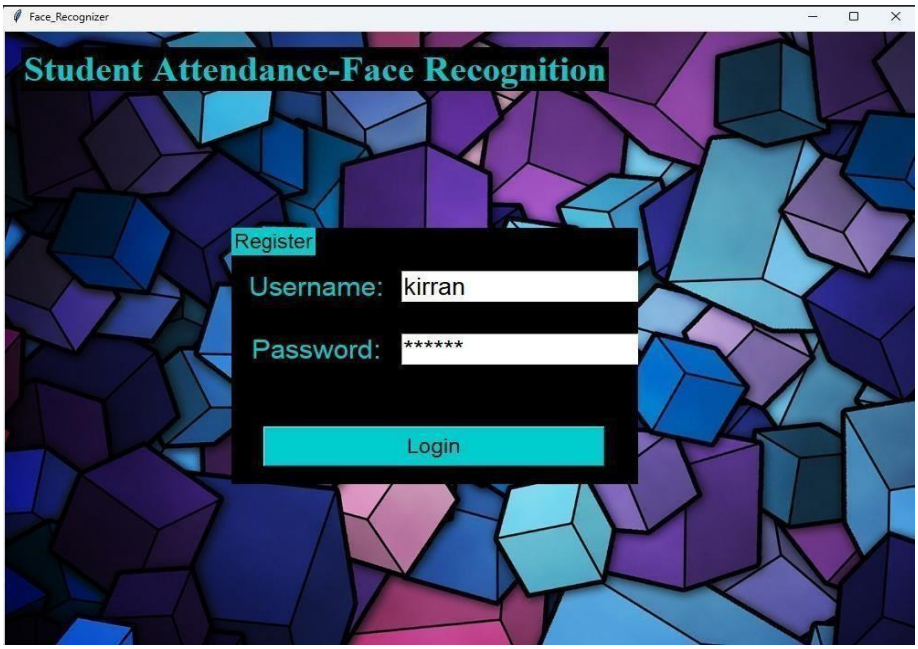
LOGIN PAGE:



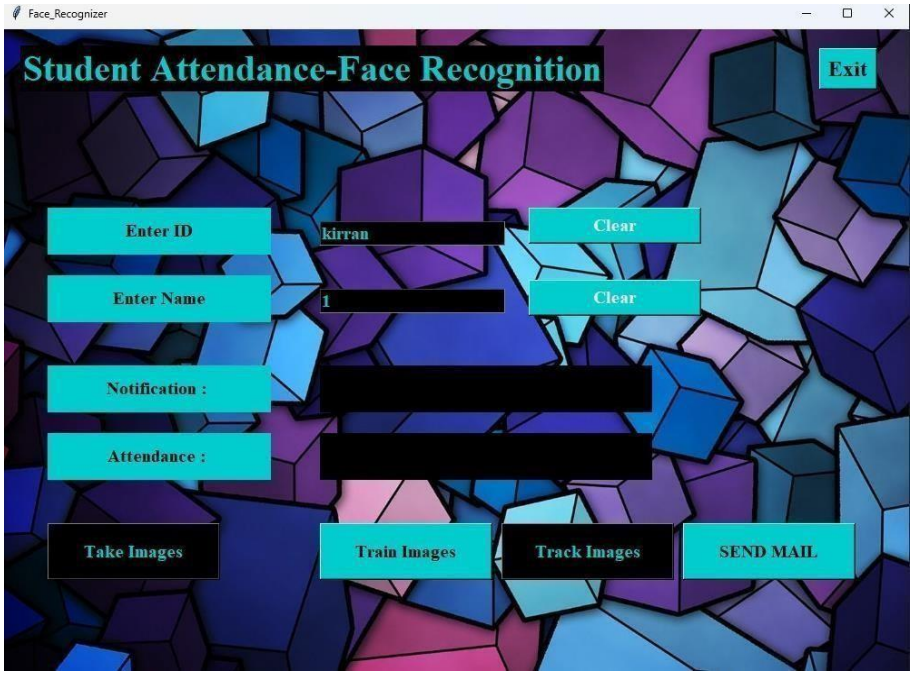
REGISTRATION PAGE:



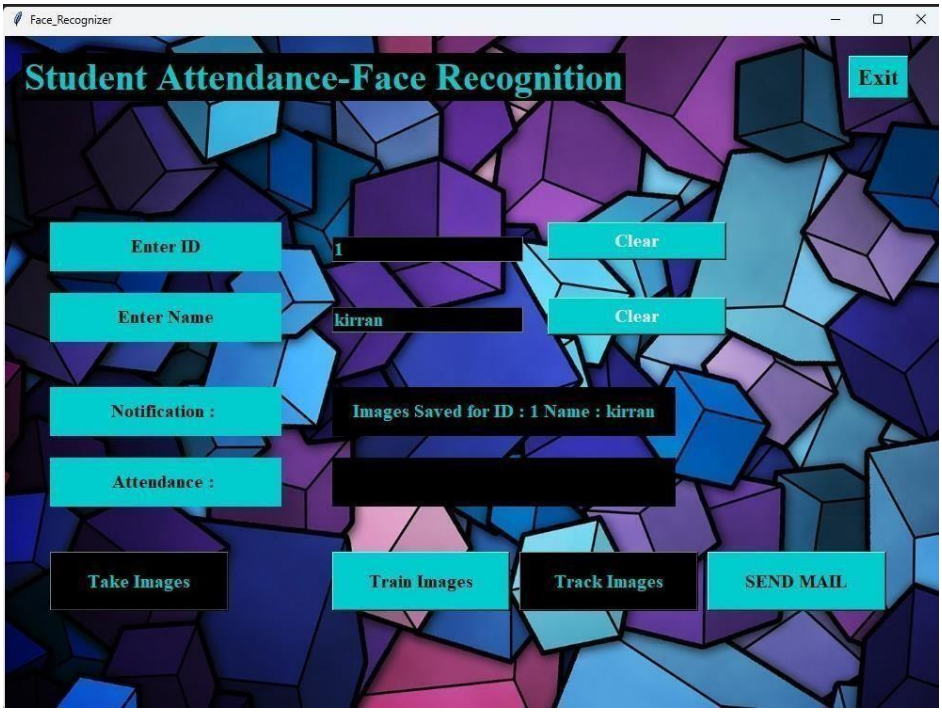
LOGING-IN:



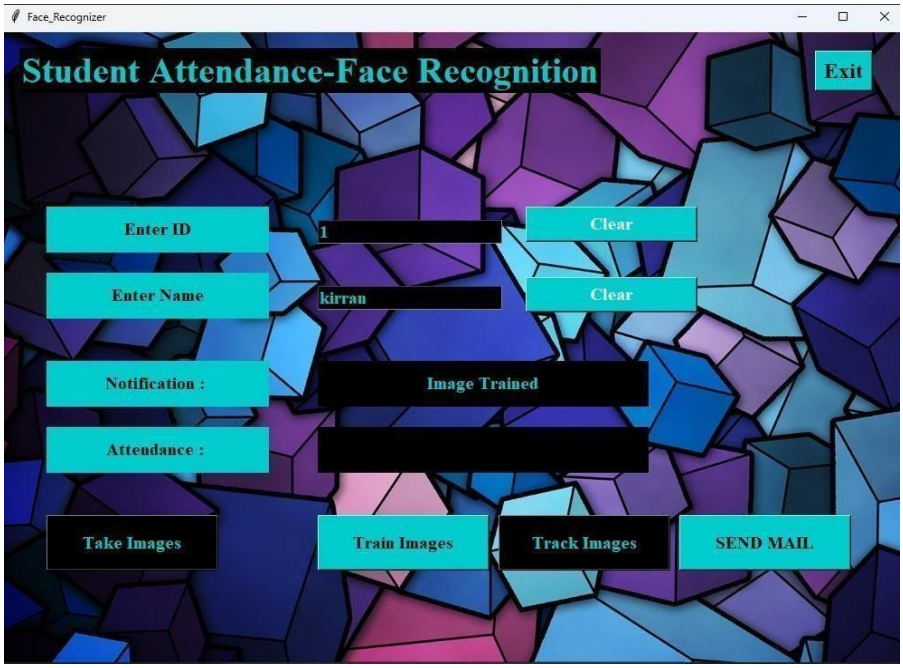
MAIN-PAGE:



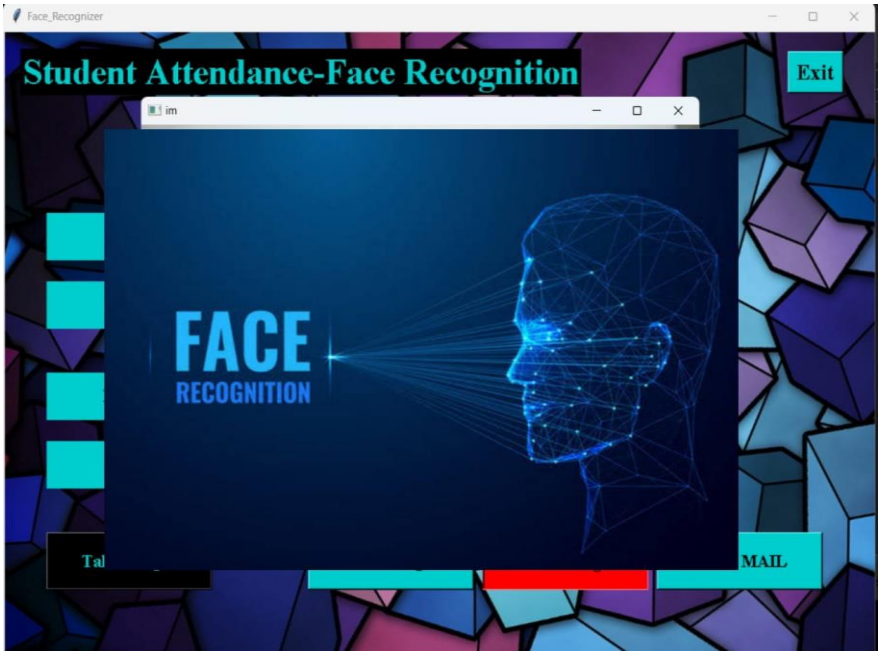
IMAGES OF THE STUDENT TAKEN:

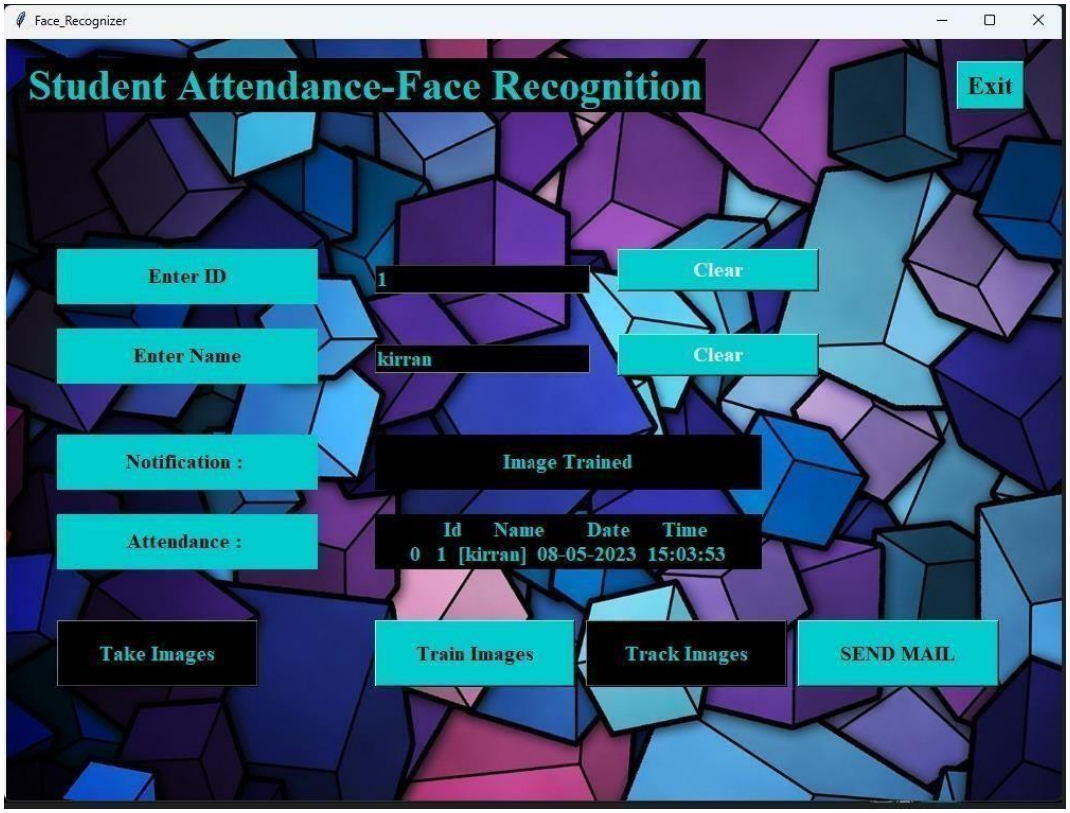


IMAGES OF STUDENTS TRAINED:



TRACKING THE FACES & MARKING ATTENDANCE:





AFTER TRACKING ATTENDENCE:

ANNEXURE

SOURCE 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 sendEmail import  sendEmail

window = tk.Tk()
#helv36 = tk.Font(family='Italic', size=36, weight='bold')
window.title("Face_Recogniser")

dialog_title = 'QUIT'
dialog_text = 'Are you sure?'
#answer = messagebox.askquestion(dialog_title, dialog_text)

window.geometry('1280x720')
window.configure(background='#eca4ed')

#window.attributes('-fullscreen',       True)

window.grid_rowconfigure(0, weight=1)
window.grid_columnconfigure(0, weight=1)

#path = "profile.jpg"
#Creates a Tkinter-compatible photo image, which can be used everywhere Tkinter.
#img =  ImageTk.PhotoImage(Image.open(path))

#The Label widget is a standard Tkinter widget used to display a text or image
```

the screen.

```
#panel = tk.Label(window, image = img)


#panel.pack(side = "left", fill = "y", expand = "no")

#cv_img = cv2.imread("img541.jpg")
#x, y, no_channels = cv_img.shape
#canvas = tk.Canvas(window, width = x, height =y)
#canvas.pack(side="left")
#photo = PIL.ImageTk.PhotoImage(image = PIL.Image.fromarray(cv_img))
# Add a PhotoImage to the Canvas
#canvas.create_image(0, 0, image=photo, anchor=tk.NW)

#msg = Message(window, text='Hello, world!')

# Font is a tuple of (font_family, size_in_points, style_modifier_string)



message = tk.Label(window, text="REC Attendance-Monitoring using face recognitition" ,bg="purple" ,fg="white" ,width=45 ,height=3,font=('times', 30, 'italic bold underline'))

message.place(x=150, y=20)

lbl = tk.Label(window, text="Enter ID",width=20 ,height=2 ,fg="red" ,bg="Pink" ,font=('times', 15, ' bold ') )
lbl.place(x=350, y=200)


txt = tk.Entry(window,width=20 ,bg="Pink" ,fg="red",font=('times', 15, ' bold '))
txt.place(x=650, y=215)


lbl2 = tk.Label(window, text="Enter Name",width=20 ,fg="red" ,bg="pink" ,height=2 ,font=('times', 15, ' bold '))
lbl2.place(x=350, y=275)
```

```
txt2 = tk.Entry(window,width=20 ,bg="pink" ,fg="red",font=('times', 15, ' bold ')
)
txt2.place(x=650, y=290)

lbl3 = tk.Label(window, text="Notification : ",width=20  ,fg="red"  ,bg="pink"
,height=2 ,font=('times', 15, ' bold underline '))
lbl3.place(x=350, y=375)

message = tk.Label(window, text="" ,bg="pink"  ,fg="red"  ,width=30  ,height=2,
activebackground = "yellow" ,font=('times', 15, ' bold '))
message.place(x=650, y=375)

lbl3 = tk.Label(window, text="Attendance : ",width=20  ,fg="red"  ,bg="pink"
,height=2 ,font=('times', 15, ' bold underline'))
lbl3.place(x=350, y=450)


message2 = tk.Label(window, text="" ,fg="red"   ,bg="pink",activeforeground =
"green",width=30 ,height=2 ,font=('times', 15, ' bold '))
message2.place(x=650, y=450)

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)
                #incrementing sample number
                sampleNum=sampleNum+1
                #saving the captured face in the dataset folder TrainingImage
                cv2.imwrite("TrainingImages\ "+name +"."+Id +'.'+ str(sampleNum) +
".jpg", gray[y:y+h,x:x+w])
                #display the frame
                cv2.imshow('frame',img)
            #wait for 100 miliseconds
            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.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)

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

def   getImagesAndLabels(path):
    #get the path of all the files in the folder
    imagePaths=[os.path.join(path,f) for f in os.listdir(path)]
    #print(imagePaths)

    #create empth face list
    faces=[]
    #create empty ID list
    Ids=[]
    #now looping through all the image paths and loading the Ids and the images
    for imagePath in imagePaths:
        #loading the image and converting it to gray scale
        pilImage=Image.open(imagePath).convert('L')
        #Now we are converting the PIL image into numpy array
        imageNp=np.array(pilImage,'uint8')
```

```
        #getting the Id from the image
        Id=int(os.path.split(imagePath)[-1].split(".")[1])
        # extract the face from the training image sample
        faces.append(imageNp)
        Ids.append(Id)
    return faces,Ids

def sendmail():
    sendEmail()
    return 0



def TrackImages():
    recognizer                                                                    =
cv2.face.LBPHFaceRecognizer_create()#cv2.createLBPHFaceRecognizer()
    recognizer.read("Trainner.yml")
    harcascadePath = "haarcascade_frontalface_default.xml"
    faceCascade = cv2.CascadeClassifier(harcascadePath)
    df=pd.read_csv("StudentDetails.csv")
    cam =  cv2.VideoCapture(0)
    font = cv2.FONT_HERSHEY_COMPLEX_SMALL
    col_names =  ['Id','Name','Date','Time']
    attendance = pd.DataFrame(columns = col_names)
    while True:
        ret,im =cam.read()
        gray = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)
        faces = faceCascade.detectMultiScale(gray, 1.3,5)
        for(x, y, w, h) in faces:
            cv2.rectangle(im,(x, y), (x + w, y + h), (255,0,0), 2)
            Id, conf = recognizer.predict(gray[y:y + h, x:x + w])
            #print (" Id:",Id)
            #print("Confidence:",conf)
            if conf < 50:
                ts = time.time()
                date = datetime.datetime.fromtimestamp(ts).strftime('%d-%m-%Y')
                #print(date)
                timeStamp                                                         =
```

```
datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
            aa=df.loc[df['ID'] == Id]['NAME'].values
            tt=str(Id)+"-"+aa
            attendance.loc[len(attendance)] = [Id, aa, date, timeStamp]
         else:
            Id='Unknown'
            #print ("ID:",Id)
            tt=str(Id)
         if(conf > 75):
            noOfFile=len(os.listdir("ImagesUnknown"))+1
            cv2.imwrite("ImagesUnknown\Image"+str(noOfFile) + ".jpg",
im[y:y+h,x:x+w])
         cv2.putText(im,str(tt),(x,y+h), font, 1,(255,255,255),2)
      attendance=attendance.drop_duplicates(subset=['Id'],keep='first')
      cv2.imshow('im',im)
      if (cv2.waitKey(1)==ord('q')):
         break
   ts = time.time()
   date = datetime.datetime.fromtimestamp(ts).strftime('%d-%m-%Y')
   Hour,Minute,Second=timeStamp.split(":")
   fileName="Attendance_"+date+"_"+Hour+"-"+Minute+"-"+Second+".csv"
   attendance.to_csv(fileName,index=False)
   cam.release()
   cv2.destroyAllWindows()
   res=attendance
   message2.configure(text= res)


clearButton = tk.Button(window, text="Clear", command=clear,fg="white"
,bg="purple" ,width=15 ,height=1 ,activebackground = "red" ,font=('times', 15, '
bold '))
clearButton.place(x=900, y=200)
clearButton2 = tk.Button(window, text="Clear", command=clear2 ,fg="white"
,bg="purple" ,width=15 ,height=1, activebackground = "Red" ,font=('times', 15, '
bold '))
clearButton2.place(x=900, y=280)
takeImg = tk.Button(window, text="Take Images", command=TakeImages
,fg="purple" ,bg="Orange" ,width=15 ,height=2, activebackground = "Red"
,font=('times', 15, 'bold '))
```

```
takeImg.place(x=350, y=550)
trainImg = tk.Button(window, text="Train Images", command=TrainImages ,fg="purple" ,bg="orange" ,width=15 ,height=2, activebackground = "Red" ,font=('times', 15, ' bold '))
trainImg.place(x=550, y=550)
trackImg = tk.Button(window, text="Track Images", command=TrackImages ,fg="purple" ,bg="orange" ,width=15 ,height=2, activebackground = "Red" ,font=('times', 15, ' bold '))
trackImg.place(x=750, y=550)
sendmailBtn = tk.Button(window, text="SEND MAIL", command=sendmail ,fg="white" ,bg="purple" ,width=15 ,height=2, activebackground = "Red" ,font=('times', 15, ' bold '))
sendmailBtn.place(x=950, y=550)


window.mainloop()
time.sleep(10)
```

SENDMAIL.PY:

```
import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
from email.mime.base import MIMEBase
from email import encoders
import os
def sendEmail():
    mail_content = '''Hello,
    This is a Attendance mail.
    In this mail we are sending student attendance excel file attachments.

    Thank You
    '''
    #The mail addresses and password
    sender_address ='triosmatlab@gmail.com'
    sender_pass ='trios@123'
    receiver_address ='diwa672@gmail.com'
```

```
    #Setup the MIME
    message = MIMEMultipart()
    message['From'] = sender_address
    message['To'] = receiver_address
    message['Subject'] = 'A Attendance Excel File. It has an attachment.'
    #The subject line
    #The body and the attachments for the mail
    message.attach(MIMEText(mail_content, 'plain'))
    attach_file_name = 'StudentDetails.csv'
    attach_file = open(attach_file_name, 'rb') # Open the file as binary mode
    payload = MIMEBase('application', 'octate-stream')
    payload.set_payload((attach_file).read())
    encoders.encode_base64(payload) #encode the attachment
    #add payload header with filename
    payload.add_header('Content-Decomposition', 'attachment',
filename=attach_file_name)
    message.attach(payload)
    #Create SMTP session for sending the mail
    session = smtplib.SMTP('smtp.gmail.com', 587) #use gmail with port
    session.starttls() #enable security
    session.login(sender_address, sender_pass) #login with mail_id and password
    text = message.as_string()
    session.sendmail(sender_address, receiver_address, text)
    session.quit()
    print('Mail Sent')
```

REFERENCES

•Murphy, K. P. (2012). "Machine Learning: A Probabilistic Perspective." MIT Press.

•Bishop, C. M. (1995). "Neural Networks for Pattern Recognition." Oxford University Press.

•Mitchell, T. M. (1997). "Machine Learning." McGraw-Hill.

•Shalev-Shwartz, S., & Ben-David, S. (2014). "Understanding Machine Learning: From Theory to Algorithms." Cambridge University Press.

•Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why Should I Trust You? Explaining the Predictions of Any Classifier." In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

•Sutton, R. S., & Barto, A. G. (2018). "Reinforcement Learning: An Introduction." MIT Press.

•LeCun, Y., Bengio, Y., & Hinton, G. (2015). "Deep Learning." Nature, 521(7553), 436-444.

•Chen, T., & Guestrin, C. (2016). "XGBoost: A Scalable Tree Boosting System." In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

•Bengio, Y., Simard, P., & Frasconi, P. (1994). "Learning Long-Term Dependencies with Gradient Descent is Difficult." IEEE Transactions on Neural Networks, 5(2), 157-166.

•He, K., Zhang, X., Ren, S., & Sun, J. (2016). "Deep Residual Learning for Image Recognition." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

•Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., & Anguelov, D. (2015). "Going Deeper with Convolutions." In Proceedings of the IEEE Conference on

Computer Vision and Pattern Recognition.
•Goodfellow, I., Bengio, Y., & Courville, A. (2016). "Deep Learning." MIT Press.
•Hastie, T., Tibshirani, R., & Friedman, J. (2009). "The Elements of Statistical Learning." Springer.
•Bishop, C. M. (2006). "Pattern Recognition and Machine Learning." Springer.
•Russell, S., & Norvig, P. (2010). "Artificial Intelligence: A Modern Approach." Prentice Hall.
•Kohavi, R., & Provost, F. (1998). "Glossary of Terms." Machine Learning, 30(2-3), 271-274.
•Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., ... & Hassabis, D. (2016). "Mastering the Game of Go with Deep Neural Networks and Tree Search." Nature, 529(7587), 484-489.
•Sutton, R. S., McAllester, D. A., Singh, S. P., & Mansour, Y. (1999). "Policy Gradient Methods for Reinforcement Learning with Function Approximation." In Advances in Neural Information Processing Systems.
•Cortes, C., & Vapnik, V. (1995). "Support-vector Networks." Machine Learning, 20(3), 273-297.
•Hochreiter, S., & Schmidhuber, J. (1997). "Long Short-Term Memory." Neural Computation, 9(8), 1735-1780.
•Caruana, R., Niculescu-Mizil, A., Crew, G., & Ksikes, A. (2004). "Ensemble Selection from Libraries of Models." In Proceedings of the Twenty-first International Conference on Machine Learning.
•Ruder, S. (2016). "An Overview of Gradient Descent Optimization Algorithms." arXiv preprint arXiv:1609.04747.
•Kingma, D. P., & Ba, J. (2014). "Adam: A Method for Stochastic Optimization." arXiv preprint arXiv:1412.6980.
•Simonyan, K., & Zisserman, A. (2014). "Very Deep Convolutional Networks for Large-Scale Image Recognition." arXiv preprint arXiv:1409.1556.
•Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). "Gradient-Based Learning