A

Major Project Report

On

FACE RECOGNITION ATTENDANCE SYSTEM USING PYTHON

Submitted in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology

By

G. Sanjay Deep
(20EG105414)

V. Hemanth
(20EG105450)

D. Nithin Reddy
(20EG105408)

Under the guidance of

Mr. E. Radha Krishnaiah

Assistant Professor

Department of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
ANURAG UNIVERSITY
VENTAKAPUR (V), GHATKESAR (M), MEDCHAL (D), T.S - 500088
TELANGANA
(2023-2024)

DECLARATION

We hereby declare that the report entitled **"Face Recognition Attendance System using Python"** submitted to the **Anurag University** in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology (B.Tech)** in **Computer Science and Engineering** is a record of an original work done by us under the guidance of **Mr. E. Radha Krishnaiah, Assistant Professor** and this report has not been submitted to any other university for the award of any other degree or diploma.

Place: Anurag University, Hyderabad
Date:

G. Sanjay Deep
(20EG105414)

V. Hemanth
(20EG105450)

D. Nithin Reddy
(20EG105408)

CERTIFICATE

This is to certify that the project report entitled **"Face Recognition Attendance System using Python"** being submitted by **Mr. G. Sanjay Deep** bearing the Hall Ticket number **20EG105414, Mr. V. Hemanth** bearing the Hall Ticket number **20EG105450**, **Mr. D. Nithin Reddy** bearing the Hall Ticket number **20EG105408** respectively in partial fulfilment of the requirements for the award of the degree of the **Bachelor of Technology** in **Computer Science and Engineering** to **Anurag University** is a record of bonafide work carried out by them under my guidance and supervision from 2023 to 2024.

The results presented in this report have been verified and found to be satisfactory. The results embodied in this report have not been submitted to any other University for the award of any other degree or diploma.

Signature of The Supervisor
Mr. E. Radha Krishnaiah
Assistant Professor
Department of CSE

Signature Dean
Dr. G. Vishnu Murthy
Dean, CSE

External Examiner

ACKNOWLEDGMENT

We would like to express our sincere thanks and deep sense of gratitude to project supervisor **Mr. E. Radha Krishnaiah, Assistant Professor, Department of Computer Science and Engineering, Anurag University** for his constant encouragement and inspiring guidance without which this project could not have been completed. His critical reviews and constructive comments improved our grasp of the subject and steered to the fruitful completion of the work. His patience, guidance and encouragement made this project possible.

We would like to acknowledge our sincere gratitude for the support extended by **Dr. G. VISHNU MURTHY**, Dean, Department of Computer Science and Engineering, Anurag University. We also express deep sense of gratitude to **Dr. V. V. S. S. S. BALARAM**, Academic coordinator. **Dr. PALLAM RAVI**, Project Coordinator and project review committee members, whose research expertise and commitment to the highest standards continuously motivated us during the crucial stages of our project work.

We would like to express our special thanks to **Dr. V. VIJAYA KUMAR, Dean School of Engineering, Anurag University**, for his encouragement and timely support in our B. Tech program.

G. Sanjay Deep
(20EG105414)

V. Hemanth
(20EG105450)

D. Nithin Reddy
(20EG105408)

ABSTRACT

The main purpose of this project is to build a face recognition-based attendance monitoring system for educational institution to enhance and upgrade the current attendance system into more efficient and effective as compared to before. The current old system has a lot of ambiguity that caused inaccurate and inefficient attendance taking. Many problems arise when the authority is unable to enforce the regulation that exist in the old system. The technology working behind will be the face recognition system. The human face is one of the natural traits that can uniquely identify an individual. Therefore, it is used to trace identity as the possibilities for a face to deviate or being duplicated is low. In this project, face databases will be created to pump data into the recognizer algorithm. Then, during the attendance taking session, faces will be compared against the database to seek for identity. When an individual is identified, its attendance will be taken down automatically saving necessary information into a excel sheet. At the end of the day, the excel sheet containing attendance information regarding all individuals are mailed to the respective faculty.

Keywords - Facial Recognition, Attendance System, Automation,
OpenCV (Cv2), Computer Vision, Image Processing, Biometrics,
Datetime Module, Group Setting, Student Attendance, Real-time Recognition, Identification, Verification

TABLE OF CONTENT

List of Figures

List of Tables

List of Abbreviations

Abbreviations	Full Form
NFC	Near Field Communication
DNN	Deep Neural Networks
RFID	Radio Frequency Identification
LDA	Linear Discriminant Analysis
Open CV	Open-Source Computer Vision Library
LBPH	Local Binary Patterns Histograms
UML	Unified Modelling Language
DFD	Data Flow Diagram
HCC	Haar Cascades Classifier
IDE	Integrated Development Environment
FPS	Frames Per Second

1. Introduction

Face recognition attendance systems are a popular and emerging technology that is being used in a variety of industries, including education, healthcare, and security. These systems offer a number of advantages over traditional attendance systems, such as fingerprint scanners and ID cards. Face recognition systems are more contactless, which can help to reduce the spread of germs. They are also more difficult to tamper with, as it is difficult to forge a face. This is a project about Facial Recognition-Based Attendance System for Educational Institutions. In this chapter, the problem and motivation, research objectives, project scope, project contributions and the background information of the project will be discussed in detail.

1.1 Problem Statement and Motivation

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 the realization of the institution which violates the accuracy of the data. For example, student A is lazy to attend 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 institution establish an 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 a student 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 which is obviously inefficient and time consuming. The third issue is with the accessibility of those 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 such information. Therefore, evolution is needed to be done to the previous system to

improve efficiency, data accuracy and provides accessibility to the information for those legitimate party.

1.2 Research Objectives

In order to solve the drawbacks of the previous system stated in 1.1, the existing system will need to evolve. The proposed system will reduce the paperwork where attendance will no longer involve any manual recording. The new system will also reduce the total time needed to do attendance recording. The new system will acquire individual attendance by means of facial recognition to secure data accuracy of the attendance.

The following are objectives of the project:

- To develop a portable Smart Attendance System which is handy and self-powered.
- To ensure the speed of the attendance recording process is faster than the previous system which can go as fast as approximately 3 second for each student.
- Have enough memory space to store the database.

1.3 Project Scope and Direction

The main intention of this project is to solve the issues encountered in the old attendance system while reproducing a brand new innovative smart system that can provide convenience to the institution. In this project, an application will be developed which is capable of recognising the identity of each individuals and eventually record down the data into a database system. Apart from that, an excel sheet is created which shows the students attendance and is directly mailed to the respected faculty.

1.4 Problem Definition

Develop a Face Recognition Attendance System to automate the process of tracking attendance in educational institutions or workplaces. The system should accurately identify individuals based on their facial features captured through images or live video feeds, and record their attendance accordingly. The goal is to streamline the attendance-taking process, eliminate manual entry errors, and enhance security measures.

Key Objectives:

1.**Accurate Identification**: Implement a robust face recognition algorithm capable of accurately identifying individuals from a database of enrolled faces, even under varying lighting conditions, angles, and facial expressions.

2.**Real-Time Recognition**: Enable real-time face recognition to efficiently track attendance as individuals enter or exit designated areas, such as classrooms or office premises.

3.**Database Management**: Develop a database to store and manage enrolled individuals' facial data securely, ensuring data integrity and privacy compliance.

4.**User-Friendly Interface**: Design an intuitive user interface for administrators to enroll individuals, monitor attendance records, and manage system settings easily.

5.**Integration**: Provide integration capabilities with existing attendance management systems or databases, enabling seamless data synchronization and reporting.

6.**Performance Optimization**: Optimize the system for speed and efficiency to handle large volumes of face recognition requests efficiently without compromising accuracy.

7.**Security Measures**: Implement security measures to prevent unauthorized access to the system and ensure the protection of sensitive attendance data.

8.**Scalability**: Design the system to be scalable, capable of accommodating growth in the number of users and transactions over time.

9.**Adaptability:** Ensure the system can adapt to different environments and scenarios, such as variations in camera quality, crowd density, and environmental conditions.

10.**Compliance**: Adhere to relevant legal and regulatory requirements concerning data privacy and security, such as GDPR, CCPA, or other applicable standards.

11.**Performance Evaluation**: Establish metrics and procedures for evaluating the system's performance regularly, including accuracy, speed, reliability, and user satisfaction.

12.**Documentation and Training**: Provide comprehensive documentation and training resources for administrators and end-users to facilitate system deployment, usage, and troubleshooting.

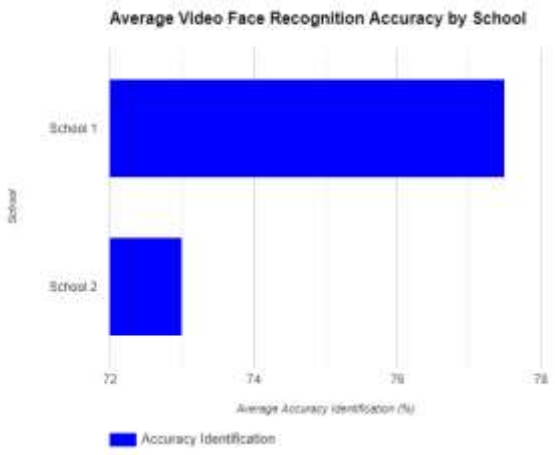
1.5. Problem Illustration

In our investigation of the effectiveness of face recognition attendance systems, we focused on two colleges, A and B, situated within a province. Our study involved conducting experiments within these colleges to assess the accuracy of the system's performance. We selected a sample of 150 college students who regularly need to use attendance cards.

Table 1.5.1. Face Recognition problem in Existing Approach

Video Face Recognition accuracy	School 1		School 2	
	Test group	Control group	Test group	Control group
Accuracy Identification	75	80	68	78
Video Blur	30	5	20	10

1.5.1. School 1 vs School 2 Accuracy testing



2. Literature Survey

2.1 Attendance System Using NFC Technology with Embedded Camera on Mobile Device

According to research journal "Attendance System Using NFC (Near Field Communication) Technology with Embedded Camera on Mobile Device" (Bhise, Khichi, Korde,Lokare, 2015). The attendance system is improved by using NFCtechnology and mobile application. According to the research paper, each student is given a NFC tag that has a unique ID during their enrolment into the college. Attendance of each class will then be taken by touching or moving these tags on the lecturer mobile phone. The embedded camera on the phone will then capture the student's face to send all the data to the college server to do validation and verification. The advantages of this method is where the NFC is simple to use, and the speed of connection establishment is very high. It indeed speeds up the attendance taking process a lot. However, this system couldn't automatically spot the violation when the NFC tag is not personally tagged by the original owner. Apart from that, the convenience of the system which uses the mobile phone as the NFC reader was actually an inconvenience to the lecturer. Imagine if the lecturer had forgotten to bring their mobile phones to work, what would be the backup procedure for the attendance to be recorded? Moreover, most of the lecturer will not likely to prefer their personal smart phones to be used in this way due to privacy matter. Hence, unique information about the student like biometrics or face recognition, which is guanine for a student should be used in replacement of the NFC tag. This will ensure attendance to be taken originally by the actual student.

2.2 Face Recognition Based Attendance Marking System

The second research journals "Face Recognition Based Attendance Marking System" (SenthamilSelvi, Chitrakala, Antony Jenitha, 2014) is based on the identification of face recognition to solve the previous attendance system's issues. This system uses camera to capture the images of the employee to do face detection and recognition. The captured image is compared one by one with the face database to search for the worker's face where attendance will be marked when a result is found in the face database. The main advantage of this system is where attendance is marked on the

server which is highly secure where no one can mark the attendance of other. Moreover, in this proposed system, the face detection algorithm is improved by using the skin classification technique to increase the accuracy of the detection process. Although more efforts are invested in the accuracy of the face detection algorithm, the system is yet not portable. This system requires a standalone computer which will need a constant power supply that makes it not portable. This type of system is only suitable for marking staff's attendance as they only need to report their presence once a day, unlike students which require to report their attendance at every class on a

particular day, it will be inconvenient if the attendance marking system is not portable. Thus, to solve this issue, the whole attendance management system can be developed on an portable module so that it can be work just by executing the python program.

2.3 Fingerprint Based Attendance System Using Microcontroller and LabView

The third research journal "Fingerprint Based Attendance System Using Microcontroller and LabView" (Kumar Yadav, Singh, Pujari, Mishra, 2015) proposed a solution of using fingerprint to mark the attendance. This system is using 2 microcontrollers to deal with the fingerprint recognition process. Firstly, the fingerprint pattern will be obtained through a fingerprint sensor, then the information will be transmitted to microcontroller 1. Next microcontroller 1 will pass the information to microcontroller 2 to do the checking with the database that resides in it. After finding a student's match, the details are sent to the PC through serial communication to be displayed. This design is good as it accelerates development while maintaining design flexibility and simplifies testing. But again, this system is attached to a PC which make it not portable. Other than that, the database information cannot be accessible easily. Meaning that, for the parents whom are interested in knowing their child's attendance cannot easily or conveniently access the information. Therefore, to provide accessibility of the student's information to the legitimate concerned party, the information can be uploaded to a web server for easy access. While the authentication for the appropriate access can be enforced through a login screen.

2.4 RFID based Student Attendance System

According to the fourth research journal "RFID based Student Attendance System" (Hussain, Dugar, Deka, Hannan, 2014), the proposed solution is almost similar to the

first research journal where RFID technology is used to improve the older attendance system. In this system, a tag and a reader is again used as a method of tracking the attendance of the students. The difference between the first journals with this is where attendance's information can be accessed through a web portal. It provides more convenient for information retrieval. Again, this system is imperfect in the sense that, firstly, it is not portable, as the RFID reader can only work when it is connected to a PC. Secondly, the RFID tag is not a guanine information that can uniquely identify a student, thus, resulting in the inaccuracy of the collected attendance information.

2.5 Face recognition with DNNs

In 2017, Marko Arsenovic proposed the method for face recognition tasks combining various modern approaches and the state of art crafts in deep learning. The method was divided into several important stages including obtaining the training dataset and augmentation, preparing images and training DNNs and last was integration into existing system to test the proposed method. This method was tested in IT company where five employees were volunteered in this research. The dataset included photographs of them and this dataset was used for training DNN. The employees took several different positions while being photographed. In order to make this approach applicable for production usage it is very important to capture small photographs of every employee at the site. It was possible to achieve high accuracy using DNN on larger datasets. The augmentation process was splitted in two stages. The first was noising and second was blurring the image at different levels. While developing face recognition model included several different steps: face detection, image preprocessing, generating face embeddings and classification. The last step was integrating with existing system. It was determined that 95% of accuracy can be achieved by using this approach. These results are enabling further research for purpose of obtaining even higher accuracy and making this solution production ready. In 2014,

2.6 Face recognition

The approach to implement and develop face recognition algorithm provided by OpenCV. The main goal was to get the best facial recognition algorithm and implement it as the main case study. Principal Component Analysis was developed

to overcome expensive computation and amounts of storage of older face recognition methods such as correlation methods. This research will compare the performance of two algorithms Eigenface and Fisherface based on Receiver Operating Characteristic curve. There will be two main features created the first is Collect Face Data and second is Attendance Recognition. Collect Face Data is used to collect face images of multiple students who are going to use this application and allows preprocessing the face images to provide better performance and results. Attendance Recognition is used for training the face recognized against the training set. It also helps users to set the countdown timer where students must present their face before the countdown ends to fill the attendance. Based on the results it was found that Eigenface algorithm to be implemented in Attendance System application

2.7. Human Face with Mask Detection and Recognition

Face recognition and detection has vital role in the technical field especially in the field of security In this paper we see how to detect the face with mask and detected face is recognized and marked the attendance smartly. Face with mask detection and recognition is now very important in current pandemic situation. In pandemic to continue the service ,education with stress free and easily this approach is very useful.

Table 2.1. Comparison of Existing Methods

Sl.No	Author(s)	Strategies	Advantages	Disadvantages
1	Hoa Yang and Xiaofeng han	LDA method	1.High detection Speed. 2.High Accuracy.	1. Long Training Time. 2.Limited Head Pose. 3.Not able to detect dark faces.
2	Shivangi Awasthi, Shubhangi Awasthi	Local Binary Pattern Histogram	1.Simple computation. 2.High tolerance against the monotonic illumination changes.	1.Only used for binary and grey images. 2.Overall performance is inaccurate compared to Viola Jones Algorithm.
3	S.Sveleba , I. Katerynchuk , I Karpa	Ada Boost Algorithm	Need not to have any prior knowledge about face structure.	The result highly depends on the training data and affected by weak classifiers.
4	R.Nandhini, N. Duraimuruga n , and S. P.Chokkalinga m	SMQT Features and SNOW Classifier Method	1.Capable to deal with lighting problem in object detection. 2.Efficient in computation.	The region contain very similar to grey value regions will be misidentified as face.
5	H.K.Nguyen, M.T. Chew	Neural-Network	High accuracy only if large size of image was trained.	1. Detection process is slow and computation is complex.

3. Proposed Method

Face Recognition algorithm with OpenCV

The proposed methodology involves utilizing OpenCV in Python for face recognition. Initially, pre-trained face detection models are loaded, followed by capturing video frames for real-time processing. Using the Haar cascade classifier, faces are detected within each frame. Optionally, a pre-trained face recognition model can be incorporated for further analysis. During the recognition loop, detected faces are marked with rectangles, and if applicable, predicted labels and confidence scores are displayed. This comprehensive approach facilitates efficient real-time face recognition directly from the camera feed.

3.1.1 Face Recognition Algorithm Overview

Face recognition is a technology that can identify the face of an individual whose image is stored in a dataset. Although other identification methods may be more accurate, facial recognition has been an important focus of research because it is easy to implement, convenient, and non-obtrusive.

A face recognition algorithm is a basic component of a face detection and recognition system. Face recognition algorithms typically perform the following main tasks:

- Detect faces in images, videos or live streams
- Compute a mathematical model of the face image
- Compare the model derived from a face to an image in a training set or database
- Evaluate the comparison to see whether the face shows the required individual

The Face Recognition process encompasses the following stages:

Steps :-

Step 1: Import the OpenCV Package

Step 2: Read the Image

Step 3: Convert the Image to Grayscale

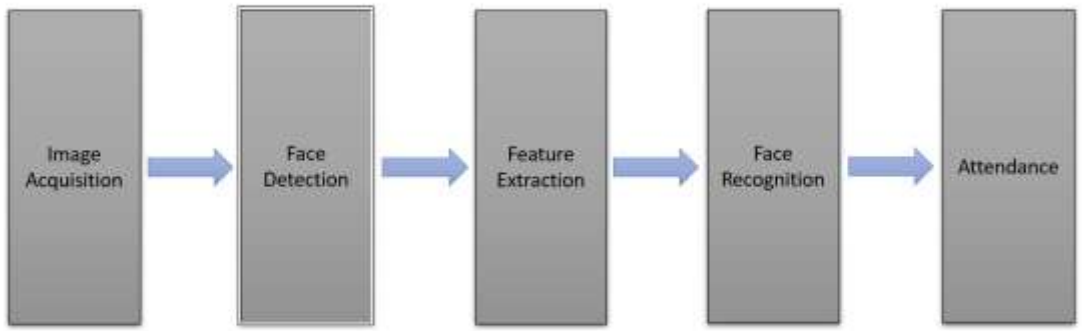
Step 4: Load the Classifier

Step 5: Perform the Face Detection

Step 6: Drawing a Bounding Box

Step 7: Displaying the Image

Figure 3.1.1. Overview of Face Recognition algorithm



3.1.2 Prepare and upload training data:

Set Up Environment: Make sure you have Python installed along with OpenCV library.

You can install OpenCV using pip:

- **pip install opencv-python**

Collect Training Data: You need images of people's faces for training the recognition model. You can either collect images from the internet or capture them using a camera.

Organize Data: Organize the images into folders where each folder represents a different person. The folder names can be the names or IDs of the individuals.

Face Detection: Use OpenCV's built-in face detection capabilities to detect faces in the images. You can use the Haar cascades or a more advanced deep learning-based face detector.

Face Alignment (Optional): Align detected faces for better recognition results. You can use techniques like face landmark detection to achieve this.

Train the Model: Use a machine learning algorithm like Eigenfaces, Fisherfaces, or Local Binary Patterns Histograms (LBPH) to train the recognition model using the prepared data.

Save the Model: Once the model is trained, save it to a file so that you can later load it for recognition tasks.

Attendance System Implementation: Implement the attendance system using the trained model. This involves capturing images of faces from a camera, recognizing them using the trained model, and updating attendance records accordingly.

Upload Training Data: Depending on your system's architecture, you might need to upload the training data (images and labels) to a server or a database for centralized access.

3.1.3 Create a Face Recognition Model:

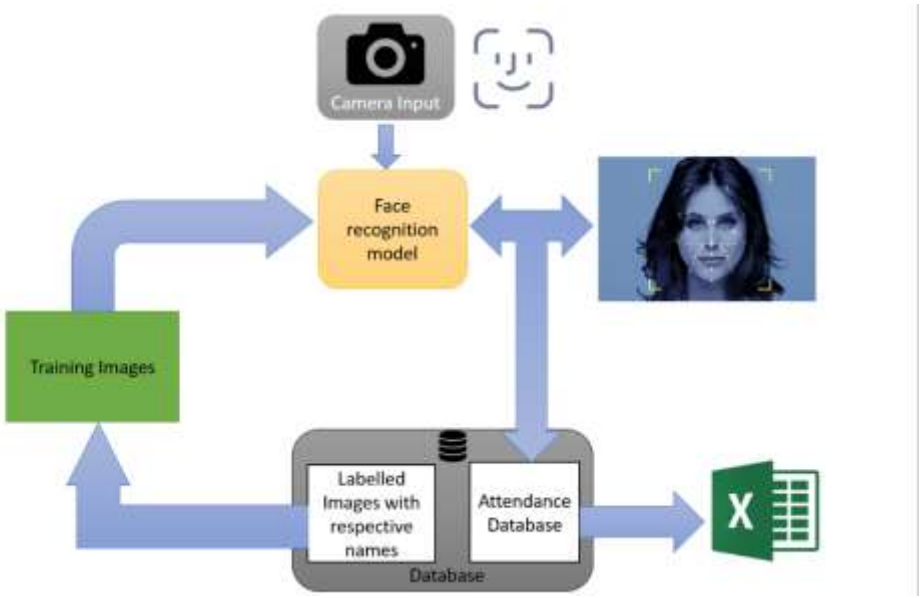
In this script:

- We load a pre-trained face detector using Haar Cascade.
- We load a pre-trained face recognition model (you should replace "face_recognizer.yml" with your trained model).
- We define a dictionary to map label IDs to human-readable labels (names of persons).
- We capture frames from the camera and detect faces in each frame.
- For each detected face, we recognize the face using the pre-trained model.
- If the confidence is below a certain threshold, we display the recognized label;

otherwise, we label it as "Unknown".

- The loop continues until the user presses 'q'.

Figure 3.1.3 Concept Tree of Face Recognition System



The overall algorithm overview:

Here's an overall algorithm for a face recognition attendance system using OpenCV:

Initialize System: Set up the necessary environment, including installing required libraries and loading pre-trained models.

Database Setup: Set up a database to store information about individuals, including their names and unique identifiers.

Face Enrollment:

Capture images of individuals to enroll in the system.

Detect faces in the captured images using a pre-trained face detection model.

Preprocess the detected face regions (resize, grayscale conversion).

Extract features from the preprocessed face regions (e.g., using Local Binary Patterns or Histogram of Oriented Gradients).

Store the extracted features along with corresponding identifiers in the database.

Attendance Recording:

- Continuously capture video frames from a camera.
- Detect faces in the captured frames using the same pre-trained face detection model.
- Preprocess the detected face regions.
- Extract features from the preprocessed face regions.
- Compare the extracted features with the features stored in the database.
- If a match is found, record the attendance for the corresponding individual.
- Optionally, display the recognized faces in real-time with their names or identifiers.

Attendance Logging:

Store attendance records in a log file or database, including timestamps and the names or identifiers of individuals present.

User Interface (Optional):

Develop a user interface for interacting with the system, allowing administrators to view attendance records, enroll new individuals, and configure system settings.

Reports and Analysis (Optional):

Generate attendance reports for specific time periods or individuals.

Analyze attendance data to identify trends and patterns.

Deployment:

Deploy the system in the desired environment (e.g., classrooms, offices).

Test the system extensively to ensure reliability and accuracy.

Maintenance and Updates:

Regularly update the system with new faces or improvements to the recognition algorithm.

Monitor system performance and address any issues that arise.

3.2 SYSTEM DESIGN

System design is transition from a user-oriented document to programmers or data base personnel. The design is a solution, how to approach to the creation of a new system. This is composed of several steps. It provides the understanding and procedural details necessary for implementing the system recommended in the feasibility study. Designing goes through logical and physical stages of development, logical design reviews the present physical system, prepare input and output specification, details of implementation plan and prepare a logical design walkthrough.

SOFTWARE DESIGN:

In designing the software following principles are followed:

Modularity and partitioning: Software is designed such that, each system should consist of hierarchy of modules and serve to partition into separate function.

Coupling: Modules should have little dependence on other modules of a system.

Cohesion: Modules should carry out in a single processing function.

Shared use: Avoid duplication by allowing a single module be called by other that need the function it provides.

3.2.1 System Architecture

Architecture diagram is a diagram of a system, in which the principal parts or functions are represented by blocks connected by lines that show the relationships of the blocks. The block diagram is typically used for a higher level, less detailed description aimed more at understanding the overall concepts and less at understanding the details of implementation.

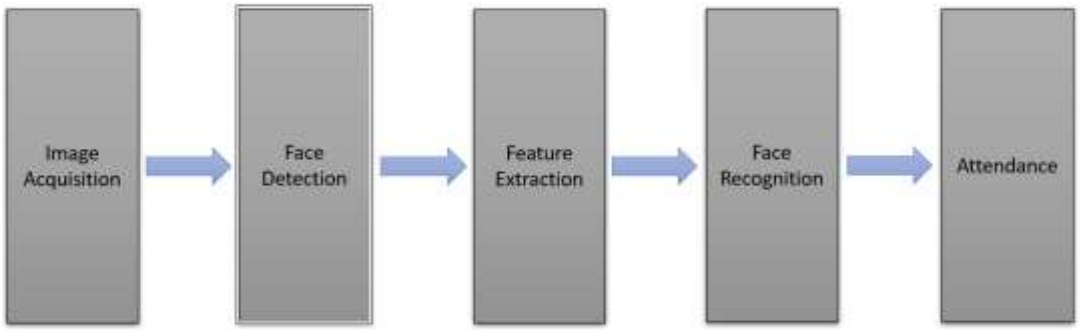


Figure 3.2.1 System Architecture

3.2.2 Data Flow Diagram

1.The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.

2.The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.

3.DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.

4.DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

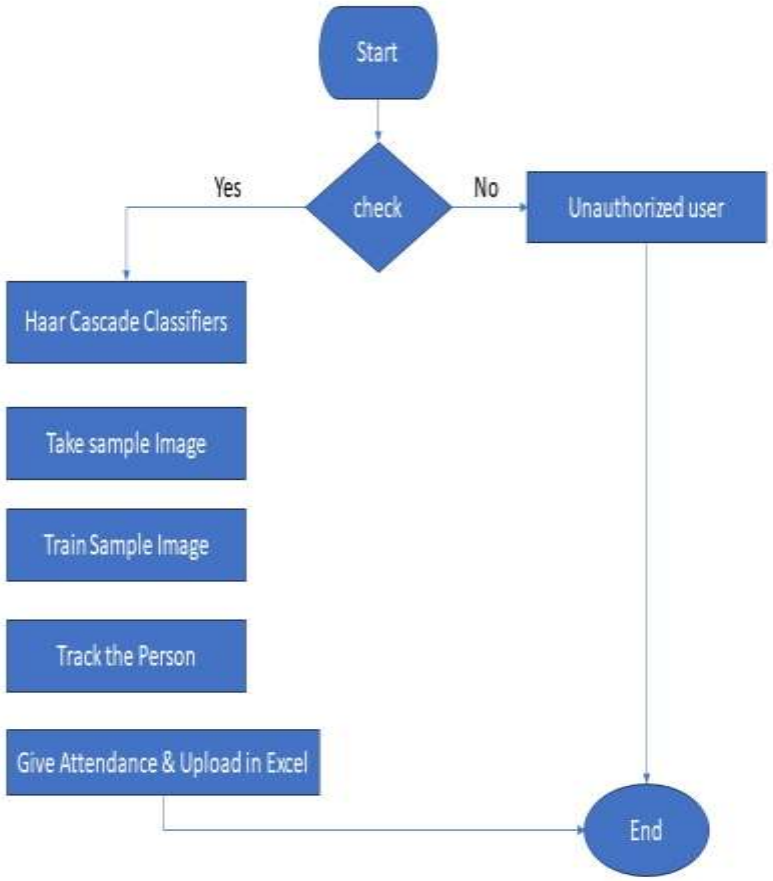


Figure 3.2.2 Data Flow Diagram

INPUT DESIGN:

Input Design plays a vital role in the life cycle of software development, it requires very careful attention of developers. The input design is to feed data to the application as accurate as possible. So, inputs are supposed to be designed effectively so that the errors occurring while feeding are minimized. According to Software Engineering Concepts, the input forms or screens are designed to provide to have a validation control over the input limit, range and other related validations. Input design is the process of converting the user created input into a computer-based format. The goal of the input design is to make the data entry logical and free from errors.

Validations are required for each data entered. Whenever a user enters an erroneous data, error message is displayed and the user can move on to the subsequent pages after completing all the entries in the current page.

OUTPUT DESIGN:

The Output from the computer is required to mainly create an efficient method of communication within the company primarily among the project leader and his team members, in other words, the administrator and the clients.

The output of VPN is the system which allows the project leader to manage his clients in terms of creating new clients and assigning new projects to them, maintaining a record of the project validity and providing folder level access to each client on the user side depending on the projects allotted to him. After completion of a project, a new project may be assigned to the client. User authentication procedures are maintained at the initial stages itself.

Advantages:

Face Detection: This category involves the detection of faces within images or video frames. OpenCV provides various methods for face detection, such as Haar cascades, HOG (Histogram of Oriented Gradients) detectors, or deep learning-based approaches.

Face Recognition: Once faces are detected, the system needs to recognize and identify them. Face recognition involves comparing the detected faces with a database of known individuals' faces to determine their identity. This typically involves encoding facial features into high-dimensional vectors and comparing them using similarity metrics like Euclidean distance or cosine similarity.

Database Management: This category involves managing the database of known individuals' faces. It includes functionalities such as adding new faces to the database,

removing or updating existing entries, and maintaining the integrity and security of the database.

Attendance Tracking: This category involves tracking attendance based on the recognized faces. Once a face is recognized, the system records the attendance of the corresponding individual. This data may be stored in a database or exported to other systems for further processing.

User Interface: A user interface is essential for interacting with the attendance system. This includes functionalities such as displaying live video feeds, showing attendance status, providing options for adding or updating faces in the database, and generating reports.

Performance Optimization: Optimizing the performance of the face recognition system is crucial, especially for real-time applications. This involves techniques such as multi-threading, batch processing, and hardware acceleration (e.g., GPU processing) to ensure efficient and responsive operation.

Security and Privacy: Ensuring the security and privacy of individuals' facial data is paramount. This category includes implementing encryption and access control mechanisms to protect the facial data stored in the system and comply with relevant privacy regulations.

Error Handling and Logging: Proper error handling and logging mechanisms are essential for diagnosing and troubleshooting issues that may arise during system operation. This category involves logging events, errors, and exceptions and providing feedback to users when errors occur.

Data-driven insights: Attendance data collected through the face recognition system can be used for analytics and reporting purposes. By analyzing attendance patterns over time, organizations can gain valuable insights into employee or student behavior, attendance trends, and overall performance.

3.3 Classifiers of Face Recognition

3.3.1 Face Recognition

Data Collection:

You'll need a collection of images containing faces of the individuals you want your classifier to recognize. Ensure these images capture variations in pose, lighting, and expression for better accuracy.

Preprocessing:

Images might need resizing to a standard dimension for the classifier to process efficiently.

Techniques like normalization or grayscale conversion might be applied to improve consistency.

Face Detection:

You'll use an algorithm to identify and isolate the facial regions within each image. Popular libraries like OpenCV or libraries specifically designed for face recognition like face_recognition can help with this step.

Feature Extraction:

This is where the magic happens! We extract a numerical representation of the facial features, essentially creating a unique "faceprint" for each individual. Deep learning models like Convolutional Neural Networks (CNNs) are highly effective for this task.

Training:

The extracted facial encodings and corresponding identities (names) are fed into the classifier for training. The classifier learns to associate specific features with particular individuals.

Recognition:

Once trained, you can use the classifier on new images. Faces are detected and encoded, then compared to the encodings stored from the training data. The classifier outputs the name of the recognized person (or "unknown" if no match is found) based on the closest match.

Here are some popular tools and libraries that can simplify this process:

face_recognition (Python library): Provides functions for face detection, encoding, and recognition with a user-friendly interface.

OpenCV (computer vision library): Offers functionalities for image processing, face detection, and machine learning

Additional Considerations:

Computational Resources: Training a face recognition classifier can be computationally expensive, especially with deep learning models. Consider using cloud platforms or GPUs if dealing with large datasets.

Privacy Concerns: Ensure you have consent to use individuals' faces for training data.

3.3.2 Haar cascades

Smart Attendance using Haar cascades in OpenCV Object Location utilizing Haar highlight based course classifiers is a compelling item discovery strategy that uses a machine learning based approach where a course capacity is prepared from a considerable measure of positive and negative pictures. It is then used to recognize protests in different pictures.

Initially, the calculation needs a considerable measure of positive (pictures of autos) and negative (pictures without autos) to prepare the classifier. At that point, we have to concentrate highlights from it. For this, hear highlights appeared in beneath picture are utilized. They are much the same as our convolutional part. Each component is a solitary esteem acquired by subtracting total of pixels under white rectangle from aggregate of pixels under dark rectangle.

Now every single conceivable size and areas of every part is utilized to ascertain a lot of components. (Simply envision what amount of calculation it needs? Indeed, even a 24x24 window comes about more than 160000 components). For each component computation, we have to discover whole of pixels under white and dark rectangles. To tackle this, they presented the necessary pictures.

Now, we apply each component on all the preparation pictures. For each component, it finds the best limit which will characterize the countenances to positive and negative. Be that as it may, clearly, there will be blunders or misclassifications. We select the elements with least mistake rate, which implies they are the elements that best orders the auto and non-auto pictures.

So now you take a picture. Take each 24x24 window. Apply 6000 elements to it. Check on the off chance that it is auto or not.

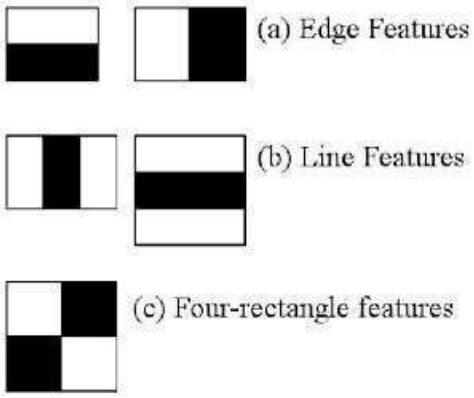


Figure 3.3.2 Haar cascades

Face detection

Once a face is detected, using the cascadeClassifier() function on the haar cascade developed.

Now the time is started which was initialized to 0.

Using the ratio in the image for each cm travelled by the detected image and real-time, the actual face is detected.

In this chapter, we will discuss and analyze about the developing process of face

recognition attendance system including software requirement specification (SRS) and comparison between existing and proposed system. The functional and non-functional requirements are included in SRS part to provide complete description and overview of system requirement before the developing process is carried out.

3.2.9 SYSTEM ANALYSIS

3.2.9.1 HARDWARE AND SOFTWARE REQUIREMENTS

HARDWARE REQUIREMENTS

- Camera Module with good (4) mega pixels
- Power Supply Cable
- 4 GB RAM
- Dual core processor

SOFTWARE REQUIRMENTS

- Technology/Language : Python
- Data Base : SQLite
- Operating System : Windows
- IDE : PyCharm

4. Implementation

4.1. FUNCTIONALITY:

Image Acquisition:

- Capture images either through cameras or by uploading existing images.
- Ensure proper lighting and image quality for accurate face recognition.
- Support different image formats and resolutions.

Face Detection:

- Detect faces within acquired images or video streams.
- Handle varying lighting conditions, angles, and facial orientations.
- Use algorithms like Haar cascades, deep learning-based methods (e.g., CNNs), or a combination of both for robust face detection.

Face Extraction:

- Extract the detected face regions from the acquired images.
- Normalize face images for consistent features and dimensions.
- Handle issues such as multiple faces in a single image or partial occlusions.

Face Recognition:

- Compare the extracted faces with the enrolled faces in the system.
- Utilize facial feature descriptors such as Eigenfaces, Fisherfaces, LBPH (Local Binary Patterns Histograms), or deep learning-based embeddings (e.g., FaceNet, VGGFace) for recognition.
- Implement algorithms for similarity matching (e.g., Euclidean distance, cosine similarity) to determine identity.
- Update face recognition models periodically to improve accuracy and adapt to variations in appearance.

Attendance:

- Log attendance based on recognized faces.
- Record timestamps indicating when attendance is marked.
- Provide real-time or batch processing options for attendance tracking.

- Store attendance records securely in a database with relevant metadata (e.g., user ID, time, date).

4.2 Features of a Face Based Attendance System

Real-Time Face Detection

As the title states, one of the top features of a biometric face attendance system is that it can recognize faces from any live video feed and that too from awkward angles or even in low-light conditions. Also, it is able to quickly identify and process the detected faces from hundreds of employees in the database. In addition, AI-based face recognition tools are effective in detecting multiple faces in a single field of camera view and matching them against existing face images stored in databases.

Identification of a Live Person

A biometric face attendance system has the ability to identify and distinguish between a real person and a photograph. This feature is critical in preventing any kind of security breach or unauthorized persons from illegally accessing your backend systems.

Ability to Track Attendance Offline

One of the best things about a biometric face attendance system is that it does not require continuous internet connectivity in order to record the attendance of the employees. This feature is unavailable in most of the biometric face attendance systems.

Ease of Use

Having advanced and multiple features is of no use if the system is not user-friendly. User-friendly biometric face attendance systems need to have features such as a centralized dashboard to display the attendance-related data, add and delete employees, easy installation, report generation, etc.

Seamless & Effective Integration with Payroll and HR Systems

This attendance system can be easily integrated with HR software and payroll systems. This not only helps in reducing administrative costs of the HR department but also in improving overall work productivity.

Support for Multiple Platforms

The biometric face attendance system works well on multiple platforms such as smartphones, laptops, tablets, and many more. This shows that the biometric face attendance system supports multiple platforms including Android and Apple iOS. Similarly, it also supports operating systems such as Windows, Linux, and macOS.

Support & Maintenance

Most biometric face attendance systems are easy to install and use. They require comparatively less maintenance. However, no system is error or maintenance-free. Purchasing a biometric face attendance system from a reliable and professional company will not only guarantee on-time maintenance but also save time.

4.3. Attributes:

Face Detection:

Accurate face detection is crucial for identifying individuals within images or video frames.

The system should utilize robust algorithms capable of detecting faces under various conditions, including different lighting, orientations, and occlusions.

It should handle multiple faces in a scene efficiently and accurately.

Face Recognition:

Face recognition involves identifying individuals by comparing detected faces with a database of known individuals.

The system should use sophisticated algorithms to extract and compare facial features accurately.

It should be capable of handling variations in facial expressions, poses, and appearances.

Attendance Tracking:

The system should keep track of attendance records for individuals, recording their arrival times and dates.

It should maintain a reliable and secure database of attendance records for further analysis and reporting.

Database Management:

Efficient management of a database containing facial features or encodings of known individuals is essential.

The system should allow for easy addition, removal, and updating of individuals' information in the database.

It should ensure data integrity and security to protect individuals' privacy.

Real-time Processing:

For real-time applications, the system should process video streams efficiently to detect and recognize faces in real-time.

It should be optimized for speed and accuracy to handle video input at high frame rates.

Accuracy and Reliability:

The system should provide high accuracy in face detection and recognition to minimize false positives and false negatives.

It should be robust against variations in environmental conditions, such as lighting changes, shadows, and background clutter.

The system should maintain consistent performance across different scenarios and demographics.

Scalability:

As the number of individuals and attendance records grows, the system should scale efficiently to handle the increased workload.

It should be designed to support large databases without compromising performance or reliability.

Security:

The system should ensure the security of facial data and attendance records, employing encryption and access controls to prevent unauthorized access.

It should comply with data protection regulations and standards to safeguard individuals' privacy rights.

User Interface:

A user-friendly interface is essential for administrators to manage the system effectively.

It should provide intuitive controls for adding, removing, and editing individuals' information in the database.

The interface should also offer features for generating attendance reports, monitoring system status, and configuring settings.

Integration:

The system should be compatible with other systems or platforms commonly used in attendance management, such as timekeeping systems or access control systems.

It should provide APIs or interfaces for seamless integration with third-party software and services.

Alerts and Notifications:

Optionally, the system may include features for sending alerts or notifications to administrators for attendance-related events, such as late arrivals or absentees.

It should support configurable rules and triggers for generating alerts based on predefined criteria.

4.4. Experimental Screenshot

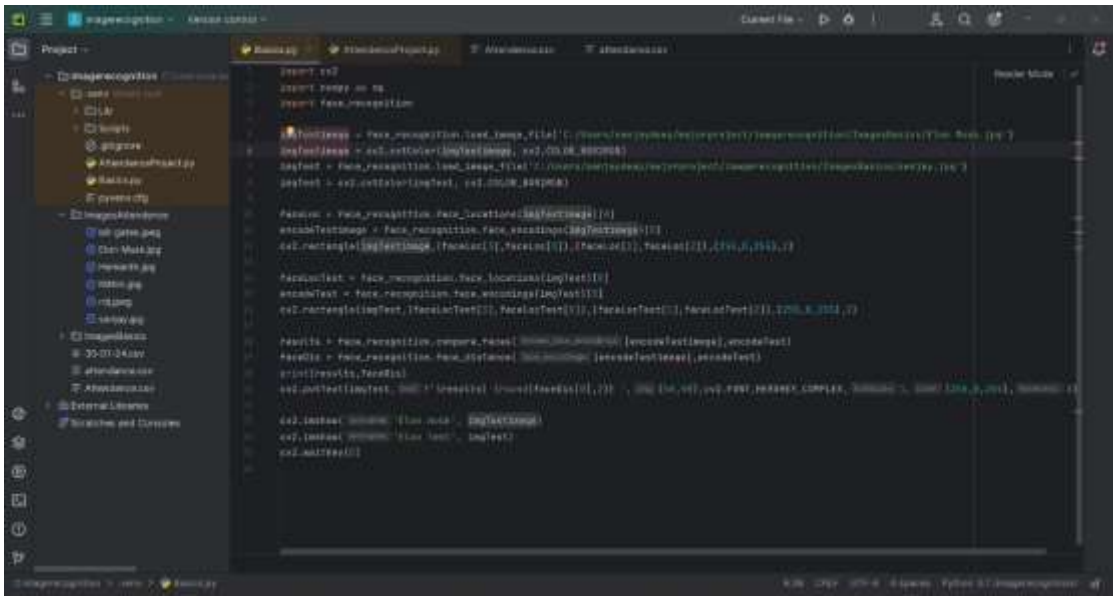


Figure 4.4.1. Testing Data.py

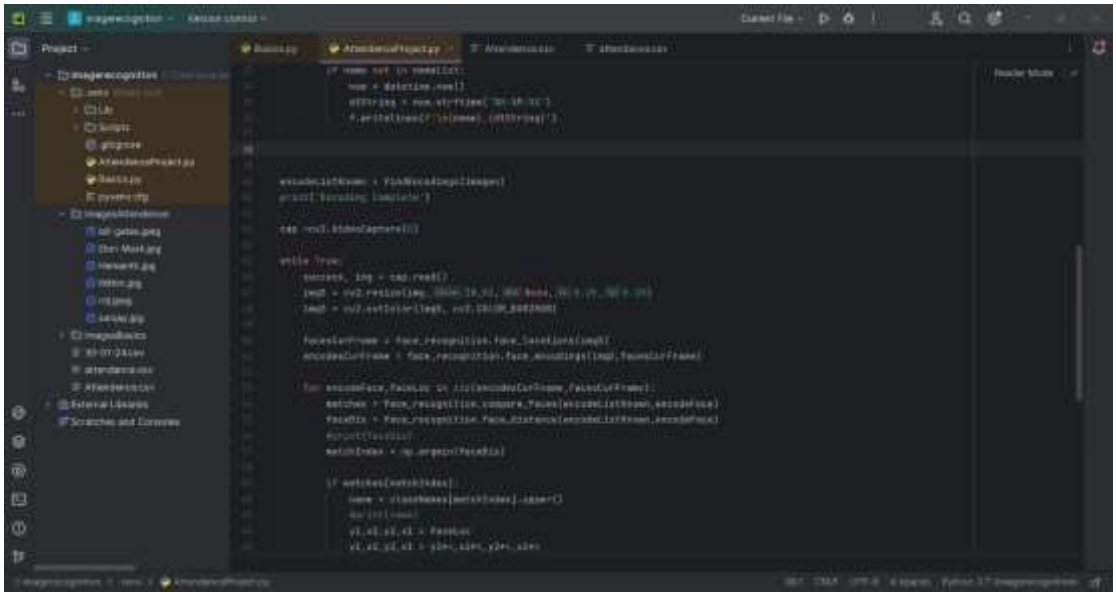


Figure 4.4.2. Attendance.py

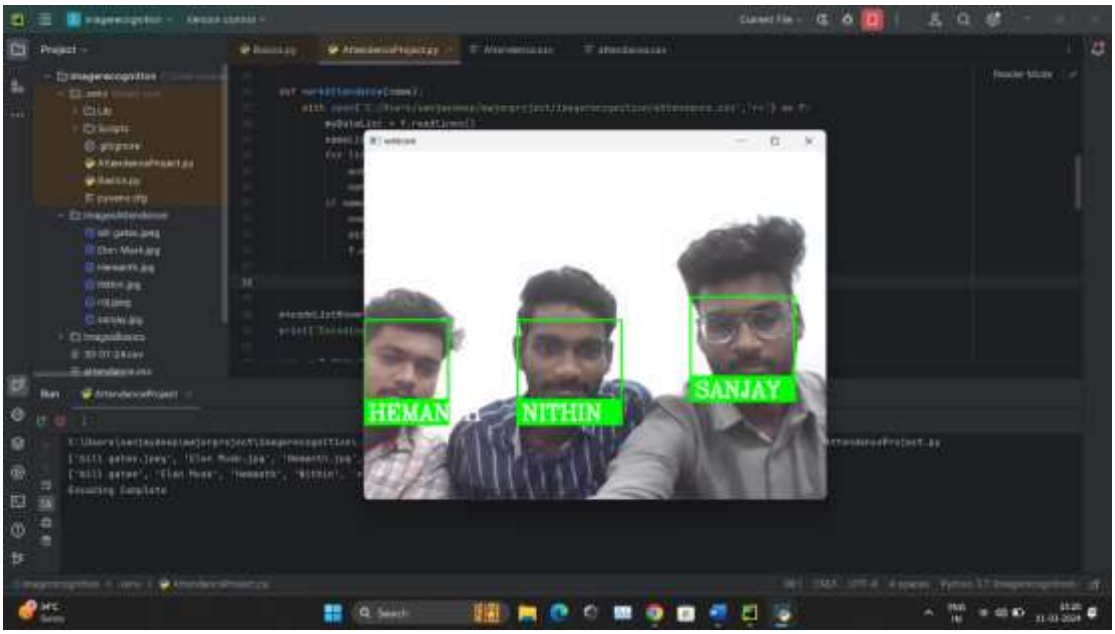


Figure 4.4.4 Attendance Interface

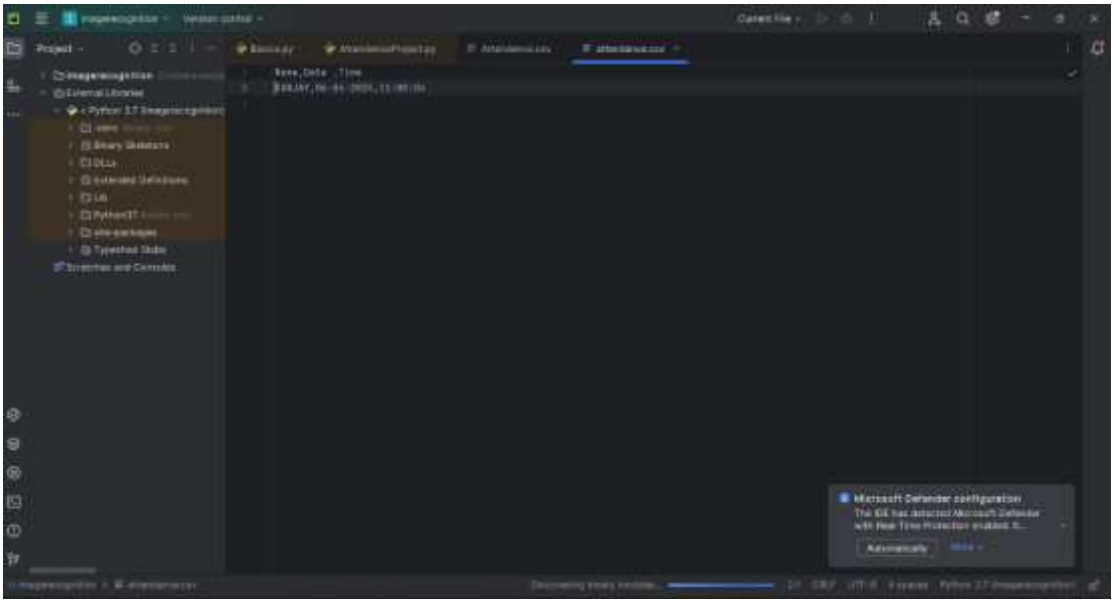


Figure 4.4.3 Attendance.csv

4.5. Dataset

The Face Recognition Attendance System Dataset is a collection of face images along with corresponding names of individuals intended for the development and evaluation of face recognition algorithms specifically tailored for attendance tracking systems. The dataset encompasses a diverse range of individuals in terms of age, gender, ethnicity, and facial characteristics to ensure robustness and inclusivity in the face recognition model.

Face Images:
The dataset includes a comprehensive set of high-resolution face images captured under various lighting conditions, facial expressions, and angles.
Images are stored in commonly used formats such as JPEG or PNG.
Each image is associated with a unique identifier corresponding to the individual's name.

Names:
The dataset includes a list of individuals' names, each linked to one or more face images.
Names are provided in text format and can include first names, last names, or full names depending on availability.
The names serve as ground truth labels for training and evaluating the face recognition model.

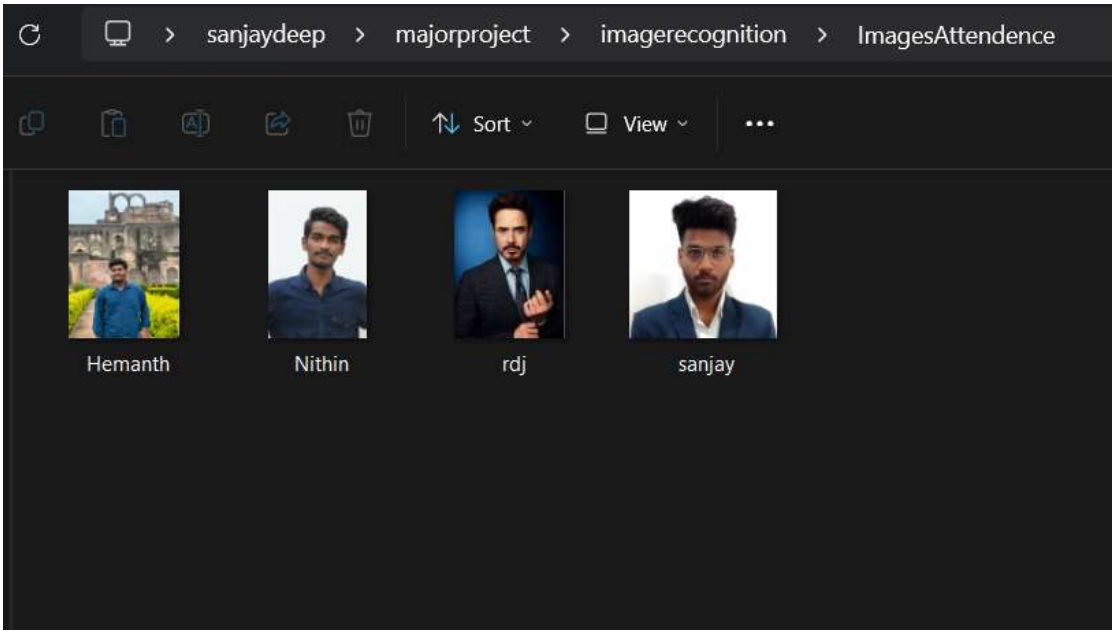
Usage:
The Face Recognition Attendance System Dataset can be utilized for various purposes, including:
Training and evaluating face recognition algorithms for attendance tracking systems in educational institutions, workplaces, or other organizational settings.
Research and development of facial recognition technologies aimed at enhancing security and access control systems.
Academic research in the field of computer vision, machine learning, and artificial intelligence.

Ethical Considerations:

While using the dataset for research or development purposes, it is imperative to uphold ethical standards and privacy considerations. Proper consent should be obtained from individuals whose images are included in the dataset, ensuring compliance with data protection regulations and guidelines. Moreover, efforts should be made to anonymize and protect sensitive information to prevent misuse or unauthorized access to personal data.

Conclusion:

The Face Recognition Attendance System Dataset provides a valuable resource for researchers, developers, and practitioners working in the field of facial recognition technology. By leveraging this dataset, advancements can be made towards the development of robust, accurate, and ethically sound face recognition systems that facilitate efficient attendance tracking and enhance security measures in various domains.



4.5.1. Dataset

5. Experimental Setup

Used PyCharm and MS Excel to develop this Face Recognition Attendance System.

5.1. PyCharm

PyCharm is an integrated development environment (IDE) specifically designed for Python development. It is developed by JetBrains, known for creating powerful IDEs for various programming languages. PyCharm provides a comprehensive set of tools and features to streamline Python development, making it a popular choice among developers, particularly those working on projects of varying scales and complexities.

Key Features of PyCharm:

Code Editor:

PyCharm offers a sophisticated code editor with syntax highlighting, code completion, and intelligent code analysis features. It provides support for various Python versions and integrates seamlessly with virtual environments.

Code Navigation and Refactoring:

The IDE enables developers to navigate through their codebase efficiently with features like Go to Definition, Find Usages, and Quick Documentation. Additionally, PyCharm offers powerful refactoring tools to improve code maintainability and readability.

Integrated Debugger:

PyCharm includes a built-in debugger that allows developers to debug Python code interactively. It supports breakpoints, watches, and expression evaluation, helping identify and fix bugs more effectively.

Version Control Integration:

PyCharm integrates with popular version control systems such as Git, SVN, Mercurial, and Perforce. Developers can perform version control operations directly within the IDE, including committing changes, viewing diffs, and managing branches.

Testing Support:

PyCharm provides extensive support for writing and running tests, including unit tests, integration tests, and doctest. It integrates with testing frameworks like pytest, unittest, and doctest, offering features such as test runners, test coverage analysis, and test result visualization.

Intelligent Code Assistance:

The IDE offers intelligent code assistance features, including code completion, code inspections, and intention actions. PyCharm suggests code improvements, detects potential errors, and offers quick-fix solutions to enhance developer productivity.

Web Development Tools:

In addition to Python development, PyCharm includes features for web development using technologies like HTML, CSS, JavaScript, and popular web frameworks such as Django and Flask. It provides HTML/CSS/JS code completion, debugging support, and integration with front-end development tools.

Extensibility:

PyCharm supports customization and extensibility through plugins and integrations with third-party tools. Developers can enhance the IDE's functionality by installing plugins from the JetBrains Marketplace or developing custom plugins using the PyCharm SDK.

Cross-Platform Support:

PyCharm is available for multiple platforms, including Windows, macOS, and Linux, ensuring a consistent development experience across different operating systems.

5.2 Setup PyCharm

Setting up PyCharm involves several steps to ensure that you have the IDE installed and configured correctly for your Python development environment. Here's a step-by-step guide to setting up PyCharm:

Download PyCharm:

Visit the JetBrains website (https://www.jetbrains.com/pycharm/download/) and

download the appropriate version of PyCharm for your operating system (Windows, macOS, or Linux).

Install PyCharm:

Once the download is complete, locate the downloaded installation file and run it. Follow the on-screen instructions to install PyCharm on your system. Accept the license agreement, choose the installation location, and complete the installation process.

Launch PyCharm:

After installation, launch PyCharm from the Start menu (Windows) or the Applications folder (macOS). On Linux, you can launch PyCharm from the terminal or using the application launcher, depending on your distribution.

Configure PyCharm:

Upon launching PyCharm for the first time, you will be prompted to customize your IDE settings. You can choose the UI theme, keymap, and other preferences according to your preferences. Once you've configured the initial settings, click "Next" to proceed.

Set Up a New Project:

PyCharm will ask you to create a new project or open an existing one. If you're starting a new project, select the location where you want to create the project and choose the interpreter. PyCharm allows you to use system interpreters or create virtual environments for your projects.

Configure Python Interpreter:

If you're using a virtual environment or a specific Python interpreter, you need to configure it in PyCharm. Go to "File" > "Settings" (or "PyCharm" > "Preferences" on macOS) and navigate to "Project" > "Python Interpreter." Click on the gear icon and select "Add." Choose the interpreter from the list or add a new interpreter by specifying the path to the Python executable.

Install Required Packages:

If your project requires additional Python packages, you can install them using PyCharm's built-in package manager. Open the terminal within PyCharm (View > Tool

Windows > Terminal) and use pip or conda commands to install the required packages.

Explore Features:

Take some time to explore PyCharm's features and familiarize yourself with the IDE. Experiment with the code editor, debugger, version control integration, and other tools to maximize your productivity.

Learn Keyboard Shortcuts:

PyCharm offers numerous keyboard shortcuts to streamline your workflow. Learning these shortcuts can significantly improve your coding efficiency. You can find a list of default keyboard shortcuts in the PyCharm documentation or customize them according to your preferences.

Update PyCharm (Optional):

Periodically check for updates to PyCharm to ensure that you have the latest features and bug fixes. PyCharm provides automatic updates, but you can also manually check for updates by going to "Help" > "Check for Updates."

5.3. MS Excel

Excel is a spreadsheet program from Microsoft and a component of its Office product group for business applications. Microsoft Excel enables users to format, organize and calculate data in a spreadsheet.

By organizing data using software like Excel, data analysts and other users can make information easier to view as data is added or changed. Excel contains a large number of boxes called cells that are ordered in rows and columns. Data is placed in these cells.

Excel is a part of the Microsoft Office and Office 365 suites and is compatible with other applications in the Office suite. The spreadsheet software is available for Windows, macOS, Android and iOS platforms.

5.4. Libraries Used:

OpenCV:

OpenCV (Open Source Computer Vision Library) is a widely used open-source library for computer vision and image processing tasks. It provides various functions and algorithms for tasks such as image manipulation, object detection, and video analysis. In the context of face recognition attendance systems, OpenCV is often utilized for tasks like face detection, image preprocessing, and displaying results.

cv2:

The "cv2" module is a Python binding for OpenCV, providing an interface to access OpenCV functions within Python code. It allows developers to utilize OpenCV functionalities seamlessly in Python-based projects, including face recognition and attendance systems.

dlib:

Dlib is a modern C++ toolkit containing machine learning algorithms and tools for building complex software in C++ to solve real-world problems. It is renowned for its implementation of facial landmark detection and facial recognition algorithms, which are vital components in face recognition systems. Dlib offers robust and efficient solutions for face detection, facial feature extraction, and facial recognition tasks.

face_recognition:

Face_recognition is a Python library built on top of dlib and OpenCV, providing a high-level interface for face recognition tasks. It simplifies the process of face detection, facial feature extraction, and recognition, making it easier for developers to integrate facial recognition capabilities into their applications. The library offers simple APIs for training face recognition models and recognizing faces in images or videos.

datetime:

The datetime module in Python provides classes for manipulating dates and times. In the context of attendance systems, the datetime module is often used for timestamping

attendance records, calculating durations, or scheduling tasks related to attendance tracking.

numpy:

Numpy is a fundamental package for scientific computing in Python, providing support for multidimensional arrays, mathematical functions, and operations. It is extensively used in image processing and data manipulation tasks within face recognition systems. Numpy arrays are commonly employed to represent images and perform various matrix operations required during image processing.

os:

The os module in Python provides a portable way to interact with the operating system. It is utilized in face recognition systems for tasks such as file and directory manipulation, accessing system paths, or managing files containing face images and attendance records.

Bounding Box:

In computer vision and image processing, a bounding box is a rectangular frame drawn around an object or region of interest in an image. It is commonly used to localize and delineate objects detected within images, including faces in face recognition systems. Bounding boxes provide spatial information about the location and extent of detected objects, facilitating further analysis and processing. Libraries like OpenCV and dlib offer functions for drawing bounding boxes around detected faces, enabling accurate localization and recognition of individuals in images or video streams.

These libraries play pivotal roles in the development of face recognition attendance systems, providing essential functionalities for image processing, face detection, recognition, and attendance tracking. Integrating these libraries allows developers to build robust and efficient systems tailored to their specific requirements.

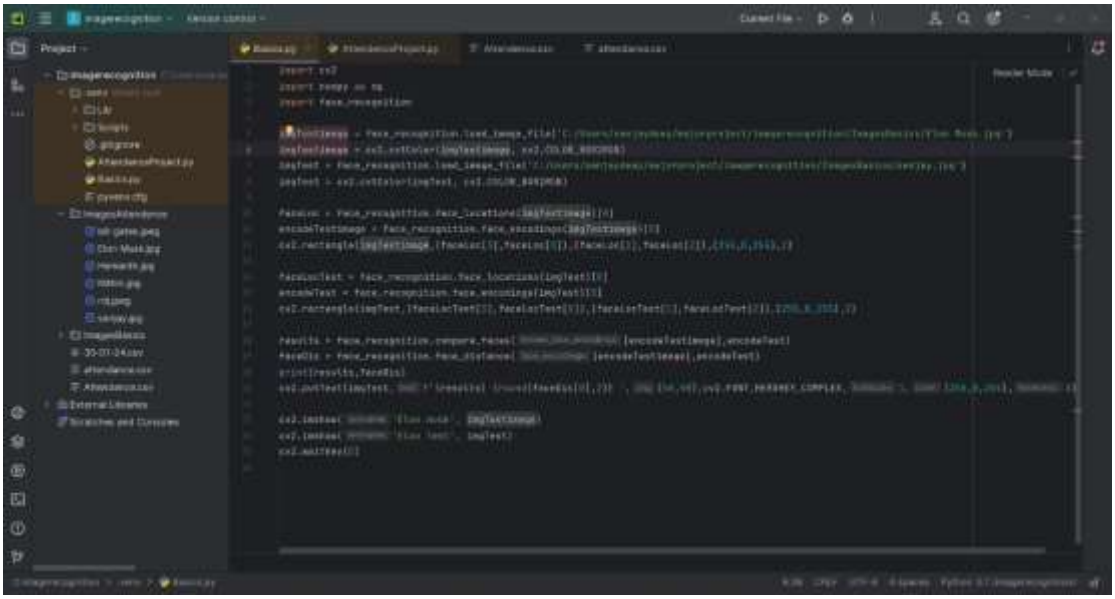


Figure. 5.4.1. Coding environment screenshot

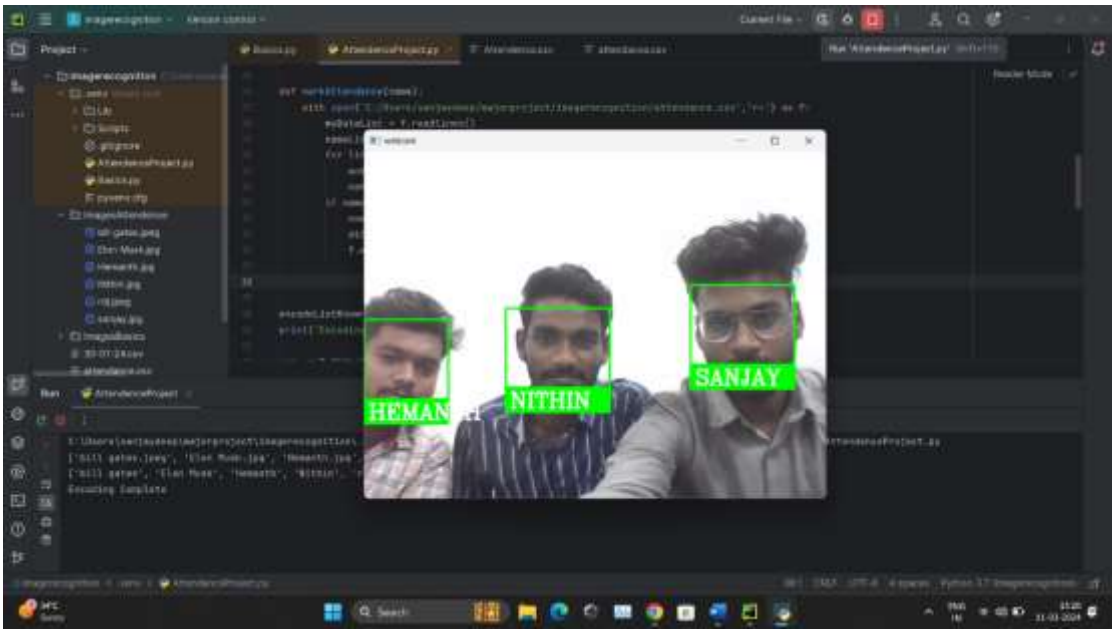


Figure. 5.4.2. User Interface

5.5. Parameters

Accuracy:

The ratio of correctly identified faces to the total number of faces in the dataset. It is commonly expressed as a percentage.

$$\text{Accuracy} = \frac{\text{(Total Number of faces)}}{\text{(Number of correctly identified Face)}} \times 100$$

Frame Processing Time (t_frame):

The time taken to process a single frame or image containing faces.

Frames Per Second (FPS):

The number of frames processed in one second. It is the reciprocal of the frame processing time.

$$FPS = 1 / t\_frame$$

Recall:

$$\text{Recall} = \frac{\text{True Positives + False Negatives}}{\text{True Positives}}$$

6. Discussion of Results

In the initial stage of setting up our face recognition attendance system, it's crucial to register individuals into the database by providing their names and corresponding registered numbers for storage. Subsequently, we proceed to capture images of individuals using a webcam, such as a Logitech webcam. After selecting the desired camera, we initialize the camera to start capturing images, which are then displayed within designated axes.

These captured images are automatically saved in folders created based on the registered numbers entered earlier. Additionally, the data pertaining to each individual is stored in the database for future reference. Moving forward, we utilize the webcam to capture an image, which is then subjected to face recognition algorithms to determine the identity of the individual by cross-referencing against the stored database.

This process ensures seamless integration between image capture, database management, and real-time face recognition functionalities within our attendance system.

Parameters Focused –

Parameter	Previous methods	Proposed method
Accuracy	Matrix Tools	Sklearn
Recall	Excel	NumPy - sklearn
Fps	FPS Monitoring system	OpenCV with Python

Table 6.1. Parameters

Accuracy Testing -

Video Face recognition accuracy	Test Group A	Control Group A	Test Group B	Control Group B
Accuracy identification (%)	82	94	83	96

Table 6.2. Accuracy

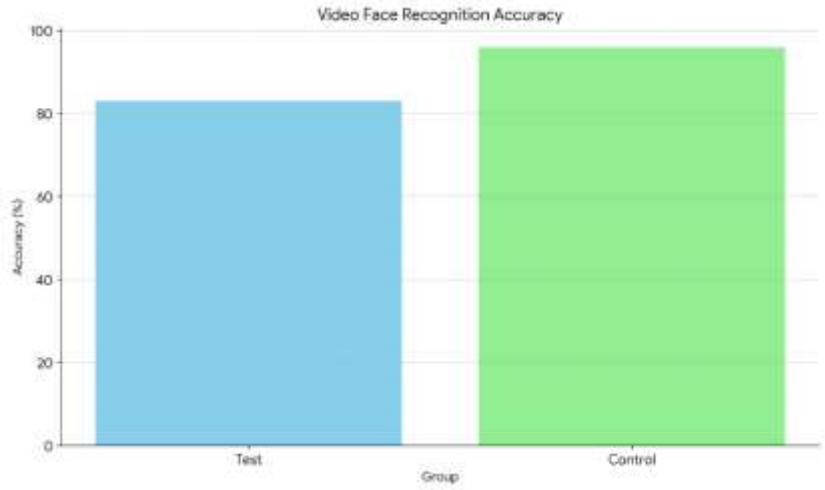


Figure 6.2 Accuracy

According to the survey conducted students who used the application found to be nearly satisfies by the performance and the answers generated by the proposed method.

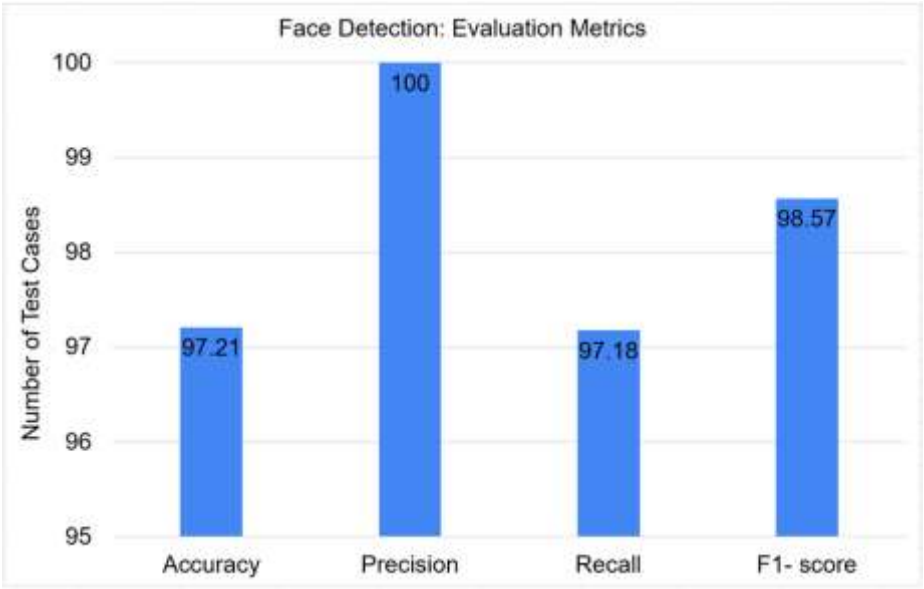


Figure 6.3 Evaluation Graphs

Table 6.4. Face Orientations

Face Orientations	Detection Rate	Recognition Rate
0 degree	98.7%	95%
18 degree	98.7%	78%
54 degree	98.7%	0%
72 degree	0%	0%
90 degree	0%	0%

The face orientations percentages for all 5 models are illustrated in Fig.6.4

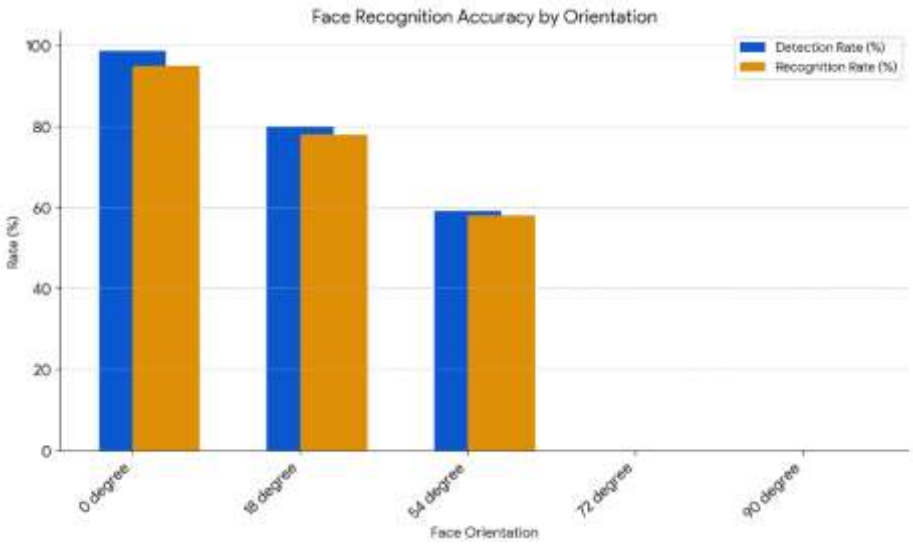


Figure. 6.4. Face Orientations

The comparison of FPS under GPU processing are illustrated in Fig.6.5

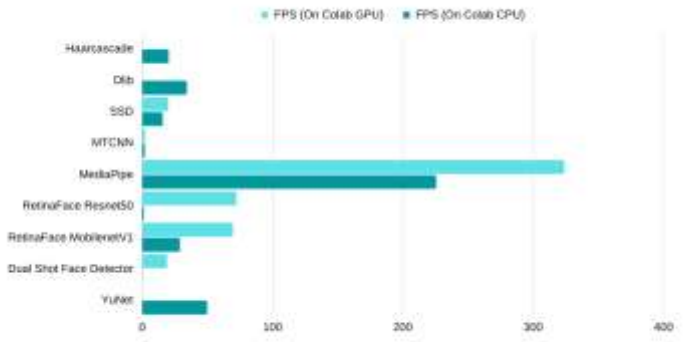


Figure 6.5. Frames Per second

GPU-based configurations offer superior performance in terms of Frames Per Second for a face recognition attendance system.

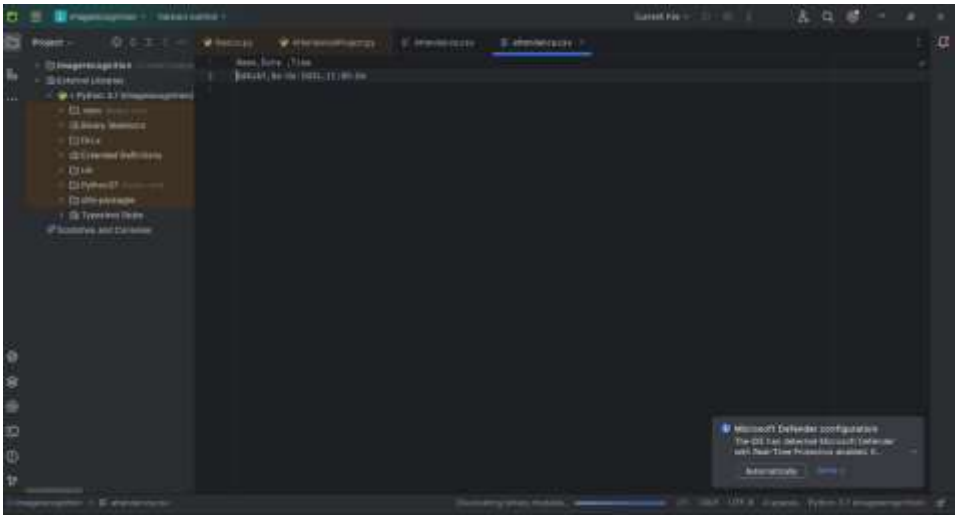


Figure 6.6 Face Recognized and noted

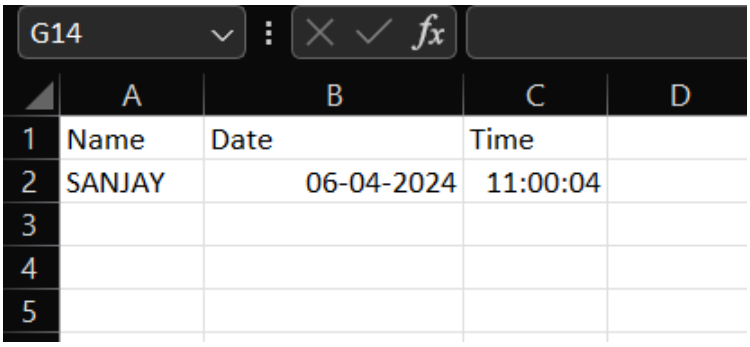


Figure 6.7 Attendance in MS Excel

7. Conclusion

In conclusion, the integration of facial recognition technology into the attendance monitoring system has effectively addressed the shortcomings of traditional attendance-taking methods and face recognition techniques, providing institutions with a more accurate and efficient solution. By leveraging technology to overcome these flaws, we have not only ensured precise attendance tracking but also significantly reduced reliance on manual processes, thereby saving resources and minimizing human intervention.

The implementation of facial recognition has not only streamlined the attendance process but also eliminated the potential for errors and inaccuracies associated with manual data entry. Moreover, the successful establishment of a robust face database and the seamless functioning of the face recognition system underscore the feasibility and effectiveness of this approach. Additionally, compared with previous facial recognition methods, we have increased accuracy, precision, and efficiency, significantly reducing the time-taking process.

Furthermore, the utilization of SSD storage has facilitated the storage of face data, ensuring that the system can accommodate a large volume of information without compromising performance. This demonstrates the adaptability of modern storage solutions to the demands of data-intensive applications like facial recognition.

In essence, the successful deployment of the face recognition attendance system marks a significant milestone in modernizing attendance management practices, offering institutions a reliable, accurate, and efficient solution to their attendance monitoring needs.

8. Future Enhancements

For future enhancements, the face recognition attendance system can explore avenues such as real-time processing for instant tracking, multi-factor authentication integrating additional biometric modalities, and mobile integration for convenient attendance marking via smartphones. Transitioning to cloud-based storage offers scalability and accessibility while ensuring data security. Automated reporting functionalities can generate insights for informed decision-making, while behavioural analytics add layers of security by detecting anomalies.

Continuous learning techniques can update the model's accuracy, and user feedback mechanisms foster improvement. Integration with other systems enhances interoperability, while privacy features ensure compliance with regulations. Customization options cater to specific institutional needs, and AI-based attendance prediction enables proactive planning. These enhancements collectively elevate the system's functionality, usability, and efficiency, meeting evolving demands effectively.

9. Reference

[1] Hao Yang and Xiaofeng Han “Face Recognition Attendance system” May 2022, Volume 8, IEEE paper J. Comput. Appl., vol. 133, no. 12, pp. 20–24, Jan. 2022.

[2] C. Stoll, R. Palluel-Germain, R. Caldara, J. Lao, M. W. G. Dye, F. Aptel, and O. Pascalis, ‘‘Face recognition is shaped by the use of sign language,’’ J. Deaf Stud. Deaf Educ., vol. 23, no. 1, pp. 1–9, 2020.

[3] W. Deng, J. Hu, and J. Guo, ‘‘Face recognition via collaborative representation: Its discriminant nature and superposed representation,’’ IEEE Trans. Pattern Anal. Mach. Intell., vol. 40, no. 10, pp. 2513–2521, Oct. 2018.

[4] T. Pei, L. Zhang, B. Wang, F. Li, and Z. Zhang, ‘‘Decision pyramid classifier for face recognition under complex variations using single sample per person,’’ Pattern Recognit., vol. 64, pp. 305–313, Apr. 2017.

[5] H. Shi, X. Wang, D. Yi, Z. Lei, X. Zhu, and S. Z. Li, ‘‘Cross-modality face recognition via heterogeneous joint Bayesian,’’ IEEE Signal Process. Lett., vol. 24, no. 1, pp. 81–85, Jan. 2017.

[6] C. Ding and D. Tao, ‘‘Trunk-branch ensemble convolutional neural networks for video-based face recognition,’’ IEEE Trans. Pattern Anal. Mach. Intell., vol. 40, no. 4, pp. 1002–1014, Apr. 2018.

[7] V. B. Nemirovskiy, A. K. Stoyanov, and D. S. Goremykina, ‘‘Face recognition based on the proximity measure clustering,’’ Inst. Cybern. Tomsk Polytech. Univ., vol. 40, no. 5, pp. 740–745, 2016.

[8] K. Taniya, M. Nidhi, and T. Nandini, ‘‘Automated human resource and attendance management system based on real time face recognition,’’

IJSRSET, vol. 16, no. 4, pp. 847–853, 2016.

[9] D. Wu, Y. Tang, G. Lin, and H. Hu, ‘‘Roboust face recognition based on significance local directional pattern and deep learning,’’ J. Optoelectron. Laser, vol. 27, no. 6, pp. 655–661, 2016.

[10] Y. Sun, J. Zhao, and Y. Hu, ‘‘Supervised sparsity preserving projections for face recognition,’’ Proc. SPIE, vol. 8009, no. 4, pp. 357– 366, 2017.

[11] L. Best-Rowden and A. K. Jain, ‘‘Longitudinal study of automatic face recognition,’’ IEEE Trans. Pattern Anal. Mach. Intell., vol. 40, no. 1, pp. 148–162, Jan. 2018.

[12] T. Valentine, M. B. Lewis, and P. J. Hills, ‘‘Face-space: A unifying concept in face recognition research,’’ Quart. J. Experim. Psychol., vol. 69, no. 10, pp. 1996–2019, Oct. 2016.

[13] Y. Duan, J. Lu, J. Feng, and J. Zhou, ‘‘Context-aware local binary feature learning for face recognition,’’ IEEE Trans. Pattern Anal. Mach. Intell., vol. 40, no. 5, pp. 1139–1153, May 2018.

[14] S. Chakraborty, S. K. Singh, and P. Chakraborty, ‘‘Local gradient hexa pattern: A descriptor for face recognition and retrieval,’’ IEEE Trans. Circuits Syst. Video Technol., vol. 28, no. 1, pp. 171–180, Jan. 2018.

[15] Q. Feng, C. Yuan, J.-S. Pan, J.-F. Yang, Y.-T. Chou, Y. Zhou, and W. Li, ‘‘Superimposed sparse parameter classifiers for face recognition,’’ IEEE Trans. Cybern., vol. 47, no. 2, pp. 378–390, Feb. 2017. [16] S. P. Mudunuri and S. Biswas, ‘‘Low resolution face recognition across variations in pose and illumination,’’ IEEE Trans. Pattern Anal. Mach. Intell., vol. 38, no. 5, pp. 1034–1040, May 2016.

[17] Y. Li, W. Song, and C. Cheng, ‘‘Attendance system of face recognition

based on raspberry pi%'' Microcontrollers Embedded Syst. Appl., vol. 16, no. 11, pp. 28–30, 34, 2016.

[18] S. Bharadwaj, H. S. Bhatt, M. Vatsa, and R. Singh, ''Domain specific learn-

ing for newborn face recognition,'' IEEE Trans. Inf. Forensics Security, vol. 11, no. 7, pp. 1630–1641, Jul. 2016.

[19] X. Zhu, H. Liu, Z. Lei, H. Shi, F. Yang, D. Yi, G. Qi, and S. Z. Li, ''Large-

scale bisample learning on ID versus spot face recognition,'' Int. J. Comput.

Vis., vol. 127, nos. 6–7, pp. 684–700, Jun. 2019.

[20] H. Nguyen, W. Yang, B. Sheng, and C. Sun, ''Discriminative low-rank

dictionary learning for face recognition,'' Neurocomputing, vol. 173, pp. 541–551, Jan. 2016.

[21] H.-K. Ji, Q.-S. Sun, Z.-X. Ji, Y.-H. Yuan, and G.-Q. Zhang, ''Collaborative

probabilistic labels for face recognition from single sample per person,''

Pattern Recognit., vol. 62, pp. 125–134, Feb. 2017.

[22] L. Wang and S. Chen, ''Joint representation classification for collective

face recognition,'' Pattern Recognit., vol. 63, pp. 182–192, Mar. 2017.

[23] K. Solanki and P. Pittalia, ''Review of face recognition techniques,'' *Int. J. Comput. Appl.*, vol. 133, no. 12, pp. 20–24, Jan. 2016.

[24] Borkar N. R. &Kuwelkar S. "Real-time implementation of face recognition system" 2017 International Conference on Computing Methodologies and Communication (ICCMC)

[25].Goyal K, Agarwal K & Kumar R "Face detection and tracking: Using OpenCV" 2017 International Conference of Electronics, Communication and Aerospace Technology (ICECA)

[26].Kwen Yan, Shaohui Huang, Yaoxin Song, Wei Lung, Neng Fan “Face Recognition Based on Convolution Neural Network” 2017
36th Chinese Control Conference.
[27].Musab Coskun, AsyegulUcar, OzalYidirim, Yakup Demir “Face Recognition Based on Convolution Neural Network” 2017
International Conference on Modern Electrical and Energy Systems: MEES 2017
[28].Vinita Bhandiwad, Bhanu Tekwani “Face Recognition and Detection using Neural Networks” International Conference on Trends
in Electronics and Informatics, ICEI 2017
[29]. E. Varadhanrajan, R. Dharani, S. Jeevitha and B. Kavinmathi “Automatic Attendance Management System using Face Detection”
2016 Online International Conference on Green Engineering and Technologies (IC-GET)
[30].Karen Simonyan and Andrew Zisserman “VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE
RECOGNITION” ICLR 2015
[31].UmmeAiman, Virendra P. Vishwakarma “FACE RECOGNITION USING MODIFIED DEEP LEARNING NEURAL
NETWORK” IEEE-40222
[32].KhemPuthea, Rudy Hartanto, RisnauriHidayat “A Review Paper on Attendance Marking System based on Face Recognition” 2017
2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE)
[33].Marko Arsenovic, SedjanSladojevic, Andras Andrela and Darko Stefanovic “FaceTime- Deep Learning Based Face Recognition
Attendance System” IEEE 15th International Symposium on Intelligent Systems and Informatics. September14-16, 2017
[34] Pang, Y. Zhang, L. LI, M. Liu, Z. Liu, Z. Liu, Z. Liu, Z. Liu, Z. Liu, Z (2004). A Face Recognition Method Based on Gabor-LDA. In Computer Science Lecture Notes. Germany's Springer-Verlag. This is issue 3331.

[35] A. Amine, S. Ghouzali, and M. Rziza (2006). Face Detection Using Skin Color Information in Still Color Images. The Second International Symposium on Communications, Control, and Signal Processing's Proceedings. [14] Y.N. Chae, J.N.

Chung, and H.S. Yang (2008). Efficient Face Detection Using Color Filtering In IEEE's 19th International Conference on Pattern Recognition Proceedings. 1–4 pages [36] J. Brand and J. Mason, J. Brand and J. Mason, J. Mason, J. Mason, J. Mason, J. Mason, J (2000). A Comparison of Three Methods for Detecting Human Skin at the Pixel Level. Volume 1, page 1056–1059. In Proceedings of the 54 International Conference on Pattern Recognition, Vol. 1, page 1056–1059.