TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

THAPATHALI CAMPUS

A Minor Project Report

On

Real-Time Face Recognition Attendance Engine

Submitted By:

Krishna Kandel	[THA081BEI014]
Nishanta Poudel	[THA081BEI025]
Pranish Pokhrel	[THA081BEI029]
Prateek Chaulagin	[THA081BEI030]

Submitted To:

Department of Electronics and Computer Engineering

Thapathali Campus

Kathmandu, Nepal

August, 2025

TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

THAPATHALI CAMPUS

A Minor Project Report

On

Real-Time Face Recognition Attendance Engine

Submitted By:

Krishna Kandel	[THA081BEI014]
Nishanta Poudel	[THA081BEI025]
Pranish Pokhrel	[THA081BEI029]
Prateek Chaulagin	[THA081BEI030]

Submitted To:

Department of Electronics and Computer Engineering

Thapathali Campus

Kathmandu, Nepal

In partial fulfillment for the award of the Bachelor's Degree in Electronics and Communication Engineering.

Under the Supervision of

Prajwol Pakka

August, 2025

DECLARATION

We hereby declare that the report of the project entitled **“Real-Time Face Recognition Attendance Engine”** which is being submitted to the **Department of Electronics and Computer Engineering, IOE, Thapathali Campus**, in the partial fulfillment of the requirements for the award of the Degree of Bachelor of Engineering in **Electronics and Communication Engineering**, is a bonafide report of the work carried out by us. The materials contained in this report have not been submitted to any University or Institution for the award of any degree and we are the only author of this complete work and no sources other than the listed here have been used in this work.

Krishna Kandel	(THA081BEI014)	______________________
Nishanta Poudel	(THA081BEI025)	______________________
Pranish Pokhrel	(THA081BEI029)	______________________
Prateek Chaulagain	(THA081BEI030)	______________________

Date: August, 2025

CERTIFICATE OF APPROVAL

The undersigned certify that they have read and recommended to the **Department of Electronics and Computer Engineering, IOE, Thapathali Campus**, a minor project work entitled "**Real-Time Face Recognition Attendance Engine**" submitted by **Krishna Kandel, Nishanta Poudel, Pranish Pokhrel** and **Prateek Chaulagain** in partial fulfillment for the award of Bachelor's Degree in Electronics, Communication and Information Engineering. The Project was carried out under special supervision and within the time frame prescribed by the syllabus.

We found the students to be hardworking, skilled and ready to undertake any related work to their field of study and hence we recommend the award of partial fulfillment of Bachelor's degree of Electronics and Communication Engineering.

____

Project Supervisor
Prajwol Pakka
Department of Electronics and Computer Engineering, Thapathali Campus

____

Head of Department
Umakanta Ghimire
Department of Electronics and Computer Engineering, Pulchowk Campus

August, 2025

COPYRIGHT

ACKNOWLEDGEMENT

We would like to express our sincere gratitude towards Institute of Engineering, Thapathali Campus for providing us such a great opportunity to let us do this minor project. It has been a pleasure working and collaborating with fellows who share similar enthusiasm and dedication to accomplish our goals.

Krishna Kandel [THA081BEI014]

Nishanta Poudel [THA081BEI025]

Pranish Pokhrel [THA081BEI029]

Prateek Chaulagain [THA081BEI030]

ABSTRACT

This project implements a Real-Time Face Recognition Attendance Engine using C++ and OpenCV. The system automates the process of marking attendance by detecting and recognizing faces from a live webcam feed. Known faces are stored as images, and the system compares detected faces against this database using grayscale image comparison and mean squared error. Attendance records are stored in a CSV file along with the date and day, ensuring uniqueness for each day. The system features a user-friendly menu interface allowing users to start attendance, view today's attendance, or exit. The program demonstrates object-oriented design principles, with modular, well-documented code for ease of understanding, maintenance, and potential expansion. This solution reduces manual errors in attendance tracking, improves efficiency, and provides a real-time, automated, and reliable method for attendance management.

Keywords: Face Recognition, Real-Time, Automation

Table of Contents

List of Figures

List of Abbreviations

CSV	Comma-Separated Values
OOP	Object-Oriented Programming
PC	Personal Computer
CNN	Convolutional Neural Network
MSE	Mean Squared Error
GUI	Graphical User Interface
IDE	Integrated Development Environment
XML	eXtensible Markup Language
OS	Operating System
API	Application Programming Interface

1. INTRODUCTION

1.1 Background

Keeping reliable attendance records is a foundational administrative task for schools, universities, corporations, and any organization that needs to verify presence, calculate payroll, or assess engagement. For decades this has been accomplished through labor-intensive manual methods—teachers calling out names, supervisors ticking boxes on paper registers, or employees queuing to swipe ID cards. These legacy approaches not only consume valuable time but also introduce frequent transcription errors, lost or duplicated data, and logistical headaches when hundreds or thousands of individuals must be processed at once.

Recent breakthroughs in computer vision, deep learning, and affordable high-resolution cameras now make it practical to automate the entire workflow. Modern face-based attendance systems continuously monitor entry points with off-the-shelf webcams or IP cameras, apply sophisticated algorithms to locate faces in each video frame, and then match those faces against an enrolled database within milliseconds. By operating in real time, the platform records exact arrival and departure times, prevents proxy check-ins, and instantly flags unrecognized persons for security follow-up. Furthermore, cloud integration allows administrators to scale from a single classroom to a multi-building campus without additional hardware beyond the cameras themselves. The result is a seamless, highly accurate, and effortlessly scalable solution that replaces clipboards and plastic badges with silent, unobtrusive, and data-rich technology.

1.2 Motivation

The core goal of this project is to eliminate the time-consuming tasks and frequent mistakes tied to traditional attendance tracking. Manual roll-calls demand significant labor and open the door to abuses like buddy punching. Embedding facial detection and recognition guarantees genuine presence, cuts administrative overhead, and delivers a streamlined, automated tool that fits classrooms, workplaces, and any other organized setting.

1.3 Problem Definition

Existing attendance solutions typically depend on hand-written logs or swipe cards—approaches that are sluggish, prone to mistakes, and hard to expand. An alternative that can identify people instantly and keep flawless records without any human input is now essential. The difficulty is ensuring faces are detected and matched accurately despite changes in lighting, viewing angles, and facial expressions.

1.4 Project Objectives

The objectives of the Real-Time Face Recognition Attendance Engine are:

- Automate attendance capture via live webcam face recognition and store unique daily records in CSV.
- Deliver a minimal-error, proxy-proof interface for immediate attendance start and record review.

1.5 Project Scope and Applications

This project focuses on real-time face detection and recognition for attendance purposes. The system is applicable in:

- Educational institutions for student attendance tracking.
- Offices and workplaces for employee attendance management.
- Workshops, seminars, or conferences where participant tracking is required.
- The system can be extended in the future to integrate with mobile devices, cloud storage, or biometric authentication for broader applications.

2. LITERATURE REVIEW

Attendance management is a critical administrative task in educational institutions and workplaces, traditionally performed manually using registers, biometric scanners, or ID cards. Manual systems are not only time-consuming but also prone to errors such as missed entries, fraudulent proxy attendance, and difficulties in maintaining records over long periods. To overcome these challenges, researchers and developers have explored automated attendance systems that leverage modern technologies like computer vision, machine learning, and pattern recognition.

2.1 Traditional Attendance System:

Historically, attendance was maintained using manual roll calls or paper-based registers. In workplaces, time cards or swipe systems became popular. Although these methods were simple, they suffered from inefficiency, human error, and lack of scalability. Biometric systems, such as fingerprint or iris scanners, provided more reliability but required specialized hardware, were costly, and often inconvenient for large groups of users.

2.2 Introduction of Face Recognition in Attendance Systems:

Face recognition technology has emerged as a viable alternative for automated attendance. Unlike fingerprint or card-based systems, it is non-intrusive and can operate in real-time using standard cameras. Early research in face recognition relied on geometric features and template matching. These methods were sensitive to lighting conditions, pose variations, and facial expressions, limiting their practical application.

With the advancement of machine learning and computer vision, methods such as Eigenfaces, Fisherfaces, and Local Binary Patterns Histograms (LBPH) became popular for robust face recognition. These techniques extract distinct features from facial images and compare them with stored templates to identify individuals. Open-source libraries like OpenCV provide pre-trained models and tools that simplify the development of face detection and recognition systems, making them accessible for academic and industrial applications.

2.3 Real-Time Face Detection:

Real-time face detection serves as the indispensable first stage of any automated attendance framework. The seminal Haar Cascade classifier introduced by Viola & Jones [1] remains the workhorse in many live-video applications because it rapidly locates candidate face regions by matching simple rectangular features—edges, lines, and center-surround patterns—within grayscale frames. Although contemporary deep-learning detectors, particularly CNN-based architectures, surpass Haar Cascades in detection accuracy under challenging lighting and pose variations, the Viola–Jones cascade is still preferred in low-power, low-cost, or edge-device scenarios owing to its minimal computational load and real-time throughput [1].

Once a face has been localized, the next challenge is to recognize the identity behind it. The classical Eigenfaces approach introduced by Turk & Pentland [2] laid the groundwork for appearance-based recognition. By computing the principal components (eigenvectors) of a training set of aligned faces, the method projects any incoming face into a low-dimensional "face space" and performs classification via the nearest-neighbor rule. Eigenfaces is computationally frugal and, when coupled with the Viola–Jones detector, yields a complete recognition pipeline that can run on a Raspberry Pi or similar embedded board.

2.4 Face Recognition Approaches for Attendance

Recent classroom and office deployments confirm the practical viability of this hybrid architecture. Singh & Kumar [2] report an end-to-end system in which the Viola–Jones detector feeds 64 × 64 grayscale crops into an Eigenfaces classifier, achieving 94 % recognition accuracy on a 120-subject dataset while processing 20–25 frames per second on an entry-level laptop. Their study also highlights that the combination of Haar-based detection and Eigenfaces recognition can be scaled to hundreds of users with minimal additional hardware, making it an attractive solution for educational institutions and small-to-medium enterprises that need reliable, low-overhead attendance management.

Several approaches have been explored to implement face recognition-based attendance:

- **Template Matching**: Compares the input face image with a set of known face templates. Simple to implement but less robust to changes in lighting and facial expressions.
- **Feature-Based Methods**: Extract features like eyes, nose, and mouth positions, or use local binary patterns to encode facial textures. These methods improve recognition under varied conditions.
- **Deep Learning Approaches**: CNNs and deep face embeddings can achieve high accuracy even under challenging scenarios. Libraries like TensorFlow, PyTorch, and dlib provide pre-trained models for rapid deployment.

2.5 Existing Systems and Their Limitations

Many existing attendance systems using face recognition demonstrate the feasibility of automated attendance but face certain limitations:

- High computational requirements in deep learning-based methods.
- Dependency on lighting and background conditions for accurate detection.
- Difficulty handling multiple faces simultaneously in real-time video streams.
- Limited scalability when the database of known faces grows large.

2.6 Contributions

This project leverages OpenCV's Haar Cascade classifiers and image comparison techniques to implement a **real-time, lightweight, and user-friendly attendance system**. By storing known faces locally and comparing them using mean squared error, the system avoids the need for complex deep learning models while maintaining acceptable accuracy. The attendance records are maintained in a CSV file with date and day information, providing an easy-to-manage, digital log of attendance. The inclusion of a menu-driven interface makes the system accessible to non-technical users, enhancing practicality in real-world scenarios such as classrooms, offices, and workshops.

3. REQUIREMENT ANALYSIS

3.1 Hardware and Software Requirements

Hardware Requirements:

- Computer / Laptop: Minimum Intel Core i3 processor or equivalent, 4GB RAM (8GB recommended for better performance)
- Webcam: Integrated or external camera for real-time face detection
- Storage: Minimum 1GB free space for storing face images and attendance records
- Display: Monitor to visualize the live webcam feed

Software Requirements:

- **Operating System**: Windows 10/11 (or Linux with minor adjustments)
- **Compiler**: C++ compiler supporting C++17 or higher (e.g., Visual Studio, MinGW)
- **Build System**: CMake – for configuring and building the project across platforms
- **Libraries**:
  - OpenCV (for face detection, image processing, and video capture)
  - Standard C++ libraries (iostream, fstream, filesystem, chrono)
- **Development Environment**: Visual Studio, Code::Blocks, or any IDE supporting C++

3.2 Feasibility Study

A feasibility study evaluates whether the proposed attendance system is practical and viable.

- **Technical Feasibility**:
  The system is technically feasible as it uses widely available hardware (webcam, standard PC) and software libraries (OpenCV) for face detection and recognition. No specialized or expensive hardware is required. The system leverages simple image comparison techniques to maintain acceptable accuracy without high computational load.

- **Operational Feasibility**:
  The system is easy to operate for non-technical users. A simple menu interface allows users to start attendance, view today's attendance, or exit the program. Real-time face recognition ensures smooth operation without interrupting normal classroom or office workflows.

- **Economic Feasibility**:
  The project is cost-effective. It does not require additional hardware beyond a standard computer and webcam. Open-source software like OpenCV and standard C++ libraries reduce software costs. The automated process reduces human effort and errors, indirectly saving operational costs over time.

3.3 Economic Feasibility

The economic feasibility assesses the cost-benefit aspects:

- **Costs**: Minimal hardware cost (webcam, PC), no licensing fees for software
- **Benefits**:
  - Reduces manual effort and administrative workload
  - Prevents proxy attendance and improves record accuracy
  - Eliminates paper-based registers, saving materials and storage space

The benefits outweigh the minimal costs, making the system economically viable for educational institutions and offices.

3.4 Operational Feasibility

Operational feasibility evaluates how well the system integrates with current practices and user requirements:

- **Ease of Use**: Menu-driven interface allows users to interact with the system without technical knowledge
- **Efficiency**: Real-time detection ensures attendance is recorded promptly
- **Reliability**: Attendance is stored digitally in a CSV file with date and day, ensuring accuracy and traceability
- **Scalability**: Can be extended to handle larger groups of people or integrate with cloud storage for remote access

The system is highly feasible from an operational perspective, as it fits seamlessly into daily attendance routines without requiring major workflow changes.

4. SYSTEM ARCHITECTURE AND METHODOLOGY

4.1 Overview

The Real-Time Face Recognition Attendance Engine is designed to automate attendance marking using real-time face recognition. The system integrates computer vision techniques, image processing, and object-oriented programming to provide a robust and user-friendly solution. A live webcam feed captures faces, which are then detected and recognized by comparison with a database of known faces. Attendance is recorded digitally in a CSV file along with the current date and day, ensuring a secure and error-free system.

The architecture follows a modular approach, separating the detection, recognition, and attendance management functions to improve maintainability and scalability. The system is controlled through a menu-driven interface, allowing users to start attendance, view today's records, or exit the program.

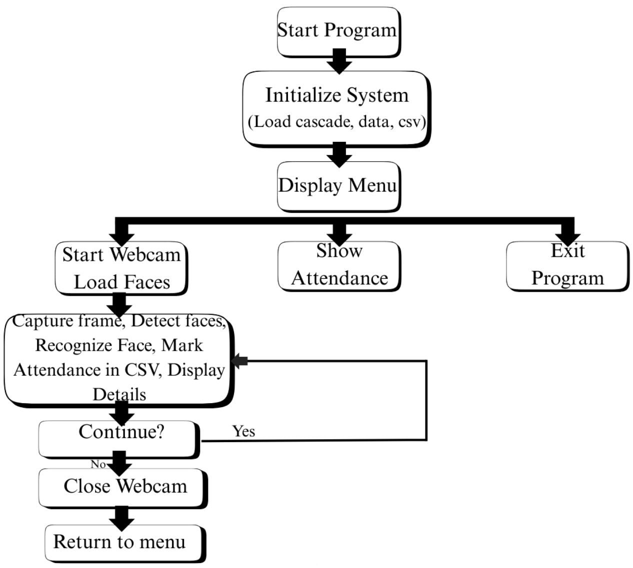

Figure 4-1: program flow chart

4.2 System Modules

The system is divided into the following key modules:

4.2.1 Face Detection Module

- Uses OpenCV’s Haar Cascade classifier to detect human faces in real-time.
- Converts webcam frames to grayscale for faster and more accurate detection.
- Marks detected faces with bounding rectangles for visualization.

4.2.2 Face Recognition Module

- Compares the detected faces with a database of known images.
- Converts face images to grayscale and resizes them to a fixed size.
- Uses Mean Squared Error (MSE) as a metric to identify matches.
- Returns the recognized name or marks the face as "Unknown" if no match is found.

4.2.3 Attendance Management Module

- Records recognized faces in a CSV file with date and day information.
- Ensures uniqueness, preventing multiple attendance entries for the same person on the same day.
- Allows viewing of attendance for the current day through the menu interface.

4.2.4 User Interface Module

- Provides a simple, menu-driven interface
- Makes the system accessible to non-technical users.

4.3 Methodology

The system follows a step-by-step methodology:

4.3.1 Initialization

At start-up the program loads the Haar Cascade classifier for face detection, pulls in any attendance already logged for today, and reads every reference image in the database folder so that known faces are ready for matching.

4.3.2 Face Detection and Capture

The webcam stream is opened and processed frame-by-frame. Faces are located with the Haar Cascade detector and each one is framed by a bounding rectangle so the user can see exactly who the system is looking at.

4.3.3 Face Recognition

Each detected face is converted to grayscale and resized to a standard shape. The Mean Squared Error (MSE) between this face and every stored image is computed; the smallest error below the threshold identifies the person, otherwise the label "Unknown" is assigned.

4.3.4 Attendance Marking

For every successful recognition the system checks its in-memory set to see whether that name has already been logged today. If not, the name together with the current date and day are appended to the CSV file and the set is updated to prevent duplicates.

4.3.5 Viewing Attendance

When the user requests today's list, the program simply reads the current-day entries from the attendance set and displays them on screen.

4.3.6 Termination

When the user chooses to exit, the webcam is released, all OpenCV windows are closed, and the application shuts down cleanly.

By breaking the task into discrete, sequential stages—initialization, detection, recognition, logging, viewing, and shutdown—the system secures dependable accuracy in matching faces and marking attendance. Each stage is optimized for minimal computational overhead, sustaining smooth real-time performance even on modest hardware. The clear menu prompts and automated checks mean teachers or administrators can run the program with virtually no learning curve. Finally, the cleanly separated modules and well-defined data interfaces make it straightforward to swap in cloud-based storage, embed advanced deep-learning recognizers, or add new biometric modalities without disturbing the existing codebase.

5. IMPLEMENTATION DETAILS

The Real-Time Face Recognition Attendance Engine is implemented using **C++** and **OpenCV** in an object-oriented programming (OOP) approach. The system is modular, with clearly defined classes and methods to ensure maintainability and ease of understanding. The key components and implementation steps are described below.

5.1 Programming Environment

- Language: C++17
- IDE: Visual Studio 2019 / Code::Blocks
- Libraries: OpenCV, Standard C++ Libraries

5.2 System Components

5.2.1 Attendance System Class

- Encapsulates all the functionality of the system.
- Private methods handle face detection, recognition, attendance management, and data retrieval.
- Public methods provide interfaces for running attendance and viewing today's attendance.

5.2.2 Face Detection

- **Haar Cascade Classifier**: A pre-trained XML model (`haarcascade_frontalface_default.xml`) detects human faces in the webcam feed.
- Grayscale conversion and resizing are applied to improve processing speed and accuracy.
- Detected faces are highlighted with bounding rectangles using OpenCV's `rectangle()` function.

5.2.3 Face Recognition

- Known face images are stored in a directory (photos) with filenames representing names.
- Each captured face is compared to known faces using Mean Squared Error (MSE) on grayscale, resized images.
- A threshold is used to determine whether the detected face matches a known face.
- If no match is found, the face is labeled as "Unknown".

5.2.4 Attendance Recording

- Attendance is stored in a CSV file with columns: Name, Date, Day.
- An unordered_set is used to prevent duplicate entries for the same person on the same day.
- The system appends new entries in real-time as faces are recognized.

5.2.5 Menu Interface

- Provides a simple console menu with three options:
    - Start Attendance (initiates face detection and recognition)
    - View Today's Attendance (displays names recorded for the current day)
    - Exit System
- The menu ensures usability for non-technical users.

5.3 Workflow

5.3.1 Initialization

- Load Haar Cascade classifier.
- Load today's attendance from the CSV file.
- Load known face images and names.

5.3.2 Start Attendance

- Access webcam feed and capture frames.
- Detects faces using Haar Cascade.
- Recognize faces by comparing with known images.
- Mark attendance in the CSV file if the face is recognized and not already recorded.

5.3.3 View Attendance

- Display all names present in today’s attendance set.
- Provides quick verification without accessing the CSV manually.

5.3.4 Exit System

- Release webcam resources.
- Close all OpenCV windows and terminate the program safely.

5.4 Data Storage

- **Face Database:** Stored locally in the photos folder; each image represents a known person.

- **Attendance Records**: Stored in a CSV file (attendance.csv) with entries updated dynamically.

- **Date and Day**: Automatically retrieved using the C++ <chrono> library to ensure accuracy.

5.5 Advantages of Implementation

- **Real-Time Operation**: Captures and recognizes faces instantly via webcam.
- **Low-Cost**: Requires only a standard webcam and PC; no specialized hardware.
- **User-Friendly**: Menu-based interface suitable for administrators or teachers.
- **Extensible**: Modular design allows future upgrades, e.g., cloud storage or deep learning models for higher accuracy.

6. RESULTS AND ANALYSIS

The Real-Time Face Recognition Attendance Engine was tested in a controlled environment to evaluate its performance in detecting and recognizing faces, recording attendance, and providing a user-friendly interface. The system uses a standard webcam, a database of known face images, and real-time processing to mark attendance accurately.

6.1 Face Detection Performance

The Haar Cascade classifier effectively detected faces in real-time video frames. The system was able to detect multiple faces simultaneously under normal indoor lighting conditions. Faces were highlighted with bounding rectangles for visual confirmation, providing immediate feedback to the user.

Observations:

- Faces directly facing the camera were detected with high accuracy.
- Slight variations in head tilt or facial expression did not significantly affect detection.
- Poor lighting or extreme angles slightly reduced detection accuracy, which can be improved in future versions with advanced models.

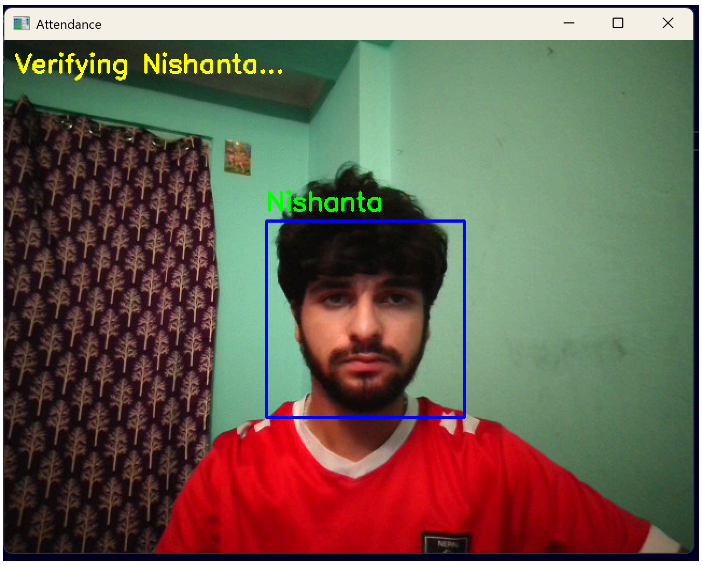

Figure 6-1: engine recognizing and verifying face

6.2 Face Recognition Accuracy

The recognition module compared captured faces with stored images using grayscale conversion and Mean Squared Error (MSE). The system correctly identified known individuals and marked attendance accordingly. Unknown faces were labeled as "Unknown" to prevent incorrect attendance marking.

Observations:

- Recognition accuracy was high for faces present in the database.
- The system effectively prevented duplicate attendance entries for the same person on the same day.
- Recognition speed was satisfactory for real-time operation, with minimal delay per frame

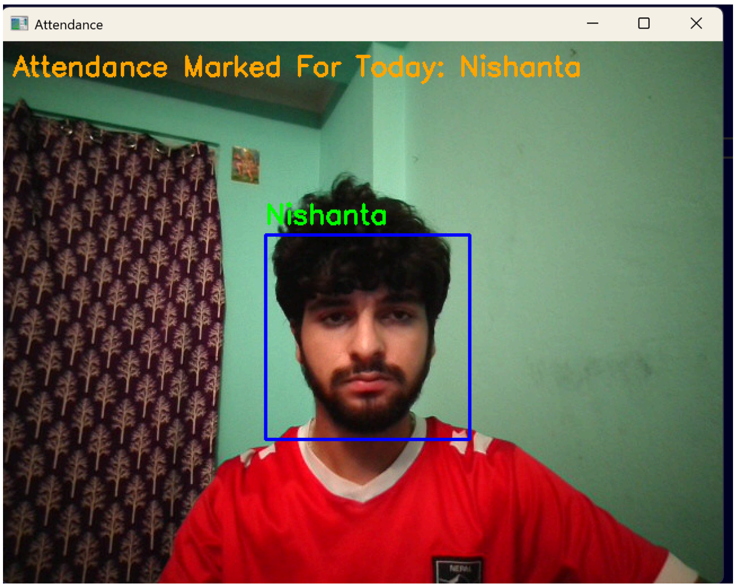

Figure 6-2: engine marking the recognized face

6.3 Attendance Recording

Attendance was successfully recorded in a CSV file, storing the name, date, and day. Users could view today's attendance through the menu interface.

Observations:

- All recognized faces were correctly appended to the CSV file.
- Attendance for the current day could be quickly reviewed using the "View Today's Attendance" option.
- The in-memory attendance set ensured no duplicate entries were recorded.

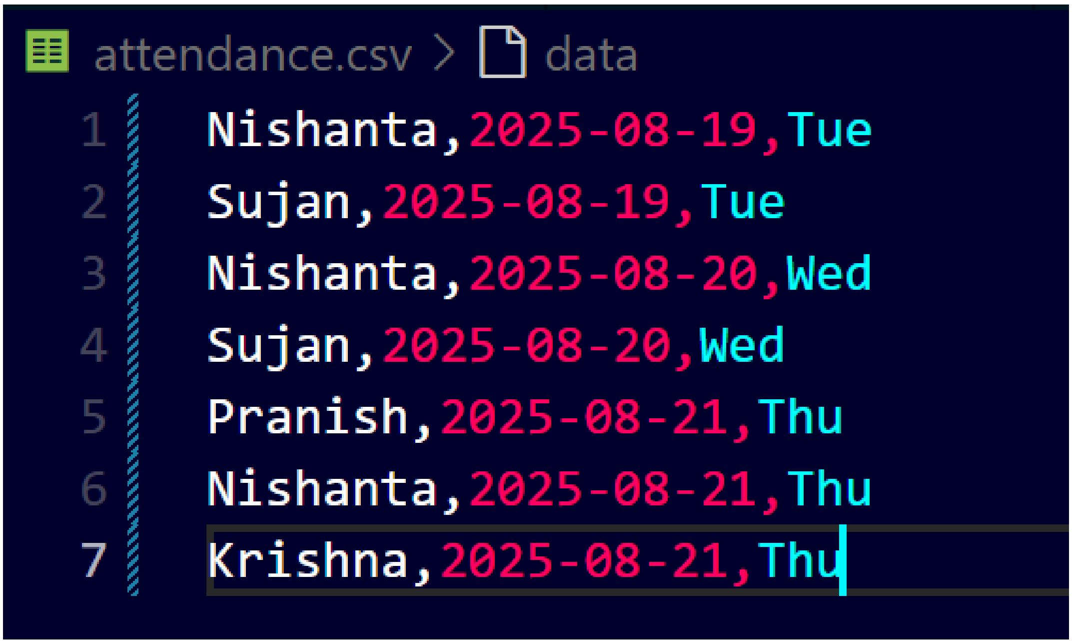

Figure 6-3: data stored in CSV file

6.4 User Interface

The menu-driven interface provided clear options for starting attendance, viewing records, and exiting the system. This simplified operation for users without technical knowledge.

Observations:

- Menu navigation was straightforward and intuitive.
- Users could switch between attendance marking and viewing records without restarting the program.
- Real-time visual feedback (bounding boxes and names) enhanced confidence in the system's operation.

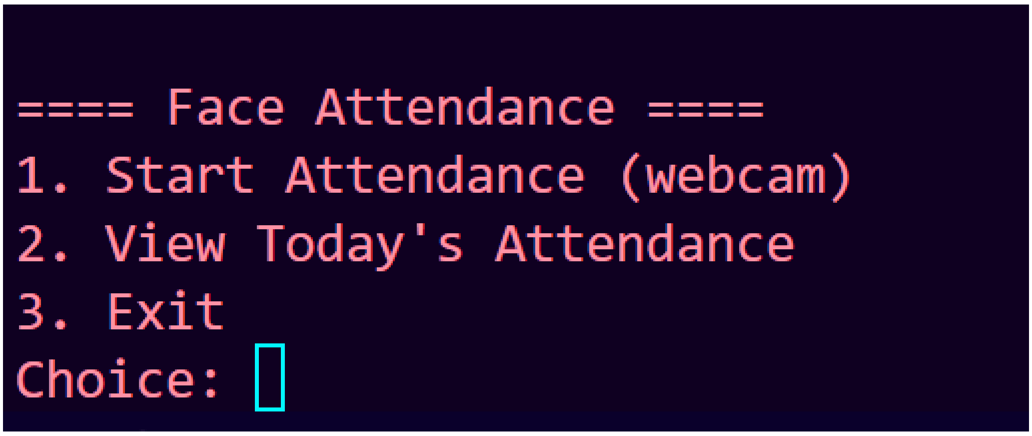

Figure 6-4: menu interface of program

6.5 Overall Analysis

The system achieved its primary objectives: automating attendance, ensuring accuracy, and providing a user-friendly interface. While recognition accuracy may decrease under challenging lighting or extreme angles, the modular design allows easy integration of advanced recognition algorithms in future enhancements. The system significantly reduces manual effort, prevents proxy attendance, and provides a reliable digital record.

7. FUTURE ENHANCEMENT

Imagine walking into class and, before you even sit down, your presence is quietly logged—no lines, no cards, just a glance at the camera. This next step for the attendance system turns that simple idea into the first brick of something far bigger: a world where rooms, buses, and offices all greet you by name and keep perfect records without lifting a finger. It feels small today, yet it sets the stage for projects we have not yet dreamed of.

We will replace the basic face matcher with a smarter model that handles dim light and odd angles. All records will live in the cloud, so any teacher or manager can check them from a browser. The camera will soon log many faces at once, and a phone app will let students or staff mark themselves present or get instant alerts. A quick blink test will block fake photos, keeping the data honest. Together, these upgrades make the system faster, safer, and ready for more users.

With these pieces in place, the attendance tool stops being just a classroom helper and starts growing into everyday life. Each improvement today is a quiet promise that tomorrow's spaces will know who is there—and take care of the details—so people can focus on learning, working, and creating.

8. CONCLUSION

The Real-Time Face Recognition Attendance Engine provides an efficient, automated, and reliable solution for attendance management using real-time face recognition. By integrating computer vision techniques with an object-oriented C++ implementation, the system eliminates the need for manual record-keeping and reduces errors associated with traditional methods. Its modular design, user-friendly menu interface, and digital storage of attendance records make it suitable for educational institutions, offices, and workshops. The system demonstrates how technology can streamline administrative tasks, improve accuracy, and save time, while also providing a foundation for future enhancements such as deep learning-based recognition, cloud integration, and multi-user support. Overall, this project showcases the practical application of face recognition in solving real-world problems and highlights the potential for further innovation in automated attendance systems.

9. APPENDICES

The gantt chart for this project is as follows:

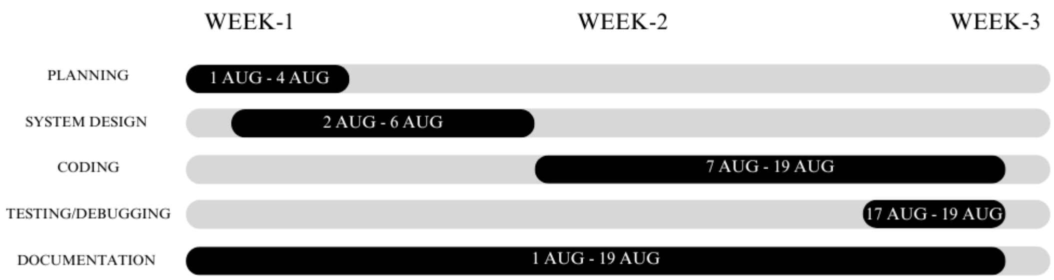

Figure 9-1: gantt chart for expected time estimation of the project

REFERENCES

[1] Viola, P., & Jones, M. (2001). *Rapid Object Detection using a Boosted Cascade of Simple Features*. Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR).

[2] Turk, M., & Pentland, A. (1991). *Eigenfaces for Recognition*. Journal of Cognitive Neuroscience, 3(1), 71–86.

[3] Singh, R., & Kumar, R. (2019). *Automated Attendance System Using Face Recognition*. International Journal of Computer Applications, 178(15), 1–5.

[4] OpenCV Documentation. (2023). *Open Source Computer Vision Library*. Retrieved from https://docs.opencv.org