iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** VIII **Month of publication:** August 2025

DOI: https://doi.org/10.22214/ijraset.2025.73780

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VIII Aug 2025- Available at www.ijraset.com

Real-Time Student Attendance System Using AI and Face Recognition

Vaishnavi Lambu[1], Shruthi Rampure[2]
Master of Computer Application, Visvesvaraya Technological University Regional Campus Kalaburagi

Abstract: The demand for efficient and accurate attendance systems in educational institutions has grown substantially, motivating the development of a contact-free, AI-driven solution. This research presents a Real-Time Student Attendance System that leverages computer vision and facial recognition to automate attendance recording with high accuracy. The system captures student faces using a standard webcam, applies Haar cascade detection and Local Binary Pattern Histogram (LBPH) recognition, and logs attendance automatically into structured CSV files. A user-friendly Tkinter GUI facilitates module navigation—student registration, model training, real-time recognition, and report generation—while supporting manual override when necessary. The system's modular architecture ensures seamless integration of components and robust performance under varying environmental conditions. Testing demonstrates over 95% recognition accuracy, with immediate generation of attendance summaries and real-time GUI feedback. The proposed system reduces administrative load, prevents proxy marking, and offers scalability as a practical, low-cost solution for modern classrooms.
Keywords: Face Recognition, LBPH, Computer Vision, Automated Attendance, Tkinter GUI, Real-time Student Monitoring

I. INTRODUCTION

Student attendance management is an integral component of academic institutions, as it directly influences learning outcomes, discipline, and overall performance evaluation. Traditionally, attendance has been recorded manually by instructors, which is often time-consuming, prone to errors, and susceptible to malpractices such as proxy attendance. With the advent of Artificial Intelligence (AI) and Computer Vision technologies, there is an increasing demand for automated, reliable, and contactless attendance systems that enhance both efficiency and accuracy. Face recognition has emerged as one of the most effective biometric identification methods due to its non-intrusive nature and high accuracy. Unlike fingerprint or RFID-based systems, face recognition requires minimal effort from users and works seamlessly in real-time environments such as classrooms. This study focuses on the design and implementation of a Real-Time Student Attendance System using Face Recognition and AI that integrates image capture, model training, recognition, and automated record generation. The system not only improves accuracy and saves classroom time but also ensures transparency, making it a scalable solution for schools, colleges, and universities.

II. PROBLEM STATEMENT

Conventional attendance management practices in educational institutions rely heavily on manual processes such as roll calls or physical signature registers. These methods consume valuable lecture time, introduce human error, and are susceptible to manipulation, including false marking of attendance. Alternative semi-automated systems, such as RFID cards or biometric fingerprint scanners, also face limitations. RFID-based systems may be misused by sharing cards, while fingerprint devices require physical contact, which raises hygiene concerns and may cause delays when handling large groups of students.

These shortcomings highlight the need for a non-intrusive, contactless, and reliable attendance monitoring system. With advancements in Artificial Intelligence and Computer Vision, face recognition technology offers an effective solution. However, challenges such as varying lighting conditions, facial orientation, occlusions, and real-time processing constraints must be addressed to ensure robust and scalable deployment in classroom environments. This project aims to overcome these challenges by designing a Real-Time Student Attendance System that ensures accuracy, efficiency, and user convenience.

III.OBJECTIVES OF THE STUDY

The primary objective of this study is to design and implement a Real-Time Student Attendance System using Face Recognition and AI that minimizes human intervention, eliminates proxy attendance, and ensures accurate record keeping. The system should be capable of capturing student images in real-time, training a recognition model, and automatically updating attendance records without disrupting the flow of classroom activities.

Another key objective is to enhance scalability and usability across different educational environments. The system is developed to operate effectively under varying lighting conditions and classroom settings, while maintaining high accuracy and reliability. Furthermore, it seeks to provide a user-friendly interface for administrators and faculty through a simple Tkinter-based GUI, ensuring smooth integration into existing institutional workflows.

IV. METHODOLOGY

The methodology adopted for this project follows a structured, step-by-step approach to ensure efficiency, accuracy, and robustness in attendance management. The system design is based on the integration of Computer Vision, Artificial Intelligence, and a Tkinter-based GUI to provide seamless functionality for both students and administrators.

The process begins with image acquisition, where student facial images are captured through a webcam in real time. These images are preprocessed by converting them into grayscale and normalizing lighting variations to improve feature extraction. The next step involves dataset preparation, where the captured images are systematically stored and labeled for each enrolled student.

Subsequently, the training phase utilizes the LBPH (Local Binary Pattern Histogram) algorithm provided by OpenCV to build a face recognition model. This model learns to identify unique facial features corresponding to each student. Once trained, the recognition module continuously monitors the video stream, matches detected faces with trained data, and automatically marks attendance in a CSV file. Finally, the system incorporates report generation, allowing faculty to retrieve attendance records in tabular form. Testing and debugging were conducted under varying conditions to ensure accuracy and adaptability. This methodology guarantees a non-intrusive, automated, and real-time attendance system suitable for practical deployment.

V. LITERATURE SURVEY

Automated attendance systems have gained significant attention in recent years as educational institutions seek to reduce manual work and improve reliability. Earlier approaches relied on traditional methods such as RFID tags, barcode scanning, and biometric systems like fingerprint recognition. While these solutions addressed basic automation, they suffered from drawbacks such as device dependency, high maintenance costs, and vulnerability to misuse or proxy attempts. Studies have shown that while RFID and biometric systems improved efficiency compared to manual roll-call, they often lacked scalability and required additional hardware infrastructure, making them less suitable for large classrooms. With advancements in Artificial Intelligence (AI) and Computer Vision, researchers have shifted towards face recognition-based attendance systems. Deep learning models such as Convolutional Neural Networks (CNNs), Haar Cascades, and LBPH (Local Binary Pattern Histogram) have been widely explored for real-time recognition tasks. These methods provide higher accuracy and scalability compared to earlier systems, as they leverage unique facial features for identification without requiring physical contact or additional devices. Several studies report recognition accuracies above 95%, highlighting the potential of AI-driven solutions for classroom settings. However, challenges remain in handling variations in lighting, occlusions, and pose differences, which this project aims to overcome through preprocessing techniques and robust model training.

VI. EXISTING SYSTEM AND PROPOSED SYSTEM

A. Existing System

Traditional student attendance systems rely heavily on manual roll-call or biometric devices such as fingerprint scanners and RFID cards. Although these methods provide some level of automation, they suffer from significant limitations. Manual roll-calls consume valuable class time, are prone to human error, and allow for proxy attendance, which reduces accuracy and fairness. Biometric systems, on the other hand, require additional hardware, regular maintenance, and often face difficulties when multiple students attempt to log in simultaneously. Moreover, in post-pandemic times, contact-based systems such as fingerprints have raised hygiene concerns.

B. Proposed System

The proposed system introduces a Real-Time Student Attendance System using Face Recognition and AI, eliminating the shortcomings of traditional approaches. By leveraging OpenCV, Haar Cascade classifiers, and LBPH algorithms, the system captures student images through a webcam, processes facial features, and automatically marks attendance. The system stores data in structured CSV files for easy retrieval and generates attendance reports in real time. Unlike existing systems, this approach is contactless, highly accurate, and scalable, ensuring robustness even in varying classroom environments. Additionally, the Tkinter-based GUI provides a user-friendly interface for faculty members to manage student registration, training, recognition, and attendance reports seamlessly.

VII. FEASIBILITY STUDY

A. *Economic Feasibility*

The system is designed to be cost-effective as it uses open-source technologies such as Python, OpenCV, and Tkinter. The only hardware requirement is a standard webcam, which is affordable and widely available. Compared to biometric fingerprint or RFID systems, the cost of implementation is significantly lower, making it feasible for schools and colleges with limited budgets.

B. *Operational Feasibility*

The system is user-friendly, requiring minimal technical knowledge for operation. Teachers can start the application, capture images, train the model, and record attendance seamlessly. The use of a GUI ensures that the system can be integrated into daily classroom routines without disrupting teaching schedules.

C. *Technical Feasibility*

The project leverages well-established technologies like OpenCV for face detection and LBPH for recognition, which ensures high accuracy even under variable classroom conditions. The system is lightweight and can run on standard laptops or desktops without requiring high-end servers, ensuring smooth deployment.

D. *Legal And Ethical Feasibility*

The system is developed for academic purposes and respects student privacy by storing only necessary attendance data and facial recognition models. No sensitive biometric information beyond facial features is stored, ensuring compliance with ethical standards in educational environments.

VIII. TOOLS AND TECHNOLOGIES USED

A. *Programming Language*

The system is developed using Python, which provides extensive libraries for computer vision, data handling, and GUI development. Python's simplicity and flexibility make it suitable for rapid prototyping and deployment of AI-driven applications.

B. *Computer Vision Library*

OpenCV (Open-Source Computer Vision Library) is used for face detection and recognition. It provides Haar Cascade classifiers for detecting facial features and LBPH (Local Binary Pattern Histogram) for training and recognizing faces with high accuracy.

C. *User Interface*

The Tkinter library is employed to build a graphical user interface (GUI). This GUI serves as the control panel, allowing users to capture images, train the model, mark attendance, and generate reports with ease.

D. *Database/Storage*

Attendance records are stored in CSV files, which ensures simplicity, portability, and compatibility with other software such as Microsoft Excel. Student details are also maintained in structured CSV files for easy retrieval and updates.

E. *Hardware*

The system requires only a webcam for capturing real-time student images. A standard laptop or desktop with moderate specifications is sufficient to run the application smoothly, eliminating the need for costly specialized hardware.

F. *Supporting Libraries*

Additional libraries such as NumPy (for numerical operations), Pandas (for data management), and PIL (Python Imaging Library) are integrated to enhance efficiency in handling images and datasets.

IX. SYSTEM DESIGN

A. *System Perspective*

The proposed Real-Time Student Attendance System is designed as a modular application consisting of four main components: Image Capture, Model Training, Face Recognition, and Attendance Recording. Each module communicates with the others to form a seamless workflow. The user interacts through a Tkinter-based GUI, while the backend handles image processing and database management. The system is designed to be lightweight, requiring minimal hardware while maintaining accuracy and efficiency.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VIII Aug 2025- Available at www.ijraset.com

The architecture follows a client-server-like approach where the client (camera and GUI) interacts with the recognition engine, and the processed output is stored in CSV-based records. This design ensures that the system is easily scalable, adaptable to future integration with cloud databases, and extendable for large-scale institutional deployment.

B. Context Diagram

The system context diagram represents the interaction between external entities and the system. The primary entities include:

- Student – whose images are captured in real-time.
- Faculty/Admin – who initiate the processes such as training, recognition, and attendance report generation.
- System – which consists of modules for face detection, recognition, and attendance storage.
- Database (CSV files) – where student details and attendance records are stored for retrieval and reporting.

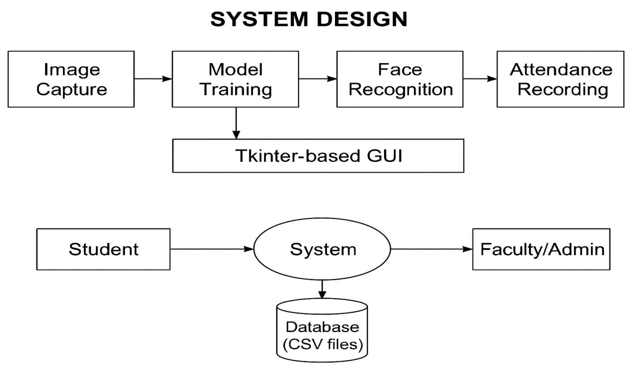


Fig. 1 System Design Diagram of the Proposed Real-Time Student Attendance System

1) Use Case Diagram

The Use Case Diagram illustrates the functional interaction between the system and its external actors. In the Real-Time Student Attendance System, there are two primary actors: Student and Admin/Faculty.

- Students interact with the system primarily during the Image Capture and Recognition phases. Their role is passive, as the system automatically detects and recognizes them.
- Admin/Faculty initiate and control the system functions such as Register Student (Image Capture), Train Model, Mark Attendance, View Attendance Records, and Generate Reports.

The System acts as the boundary, ensuring all processes such as image storage, training using the LBPH algorithm, recognition via webcam input, and attendance logging into CSV files are executed smoothly.

This diagram demonstrates how the actors and the system interact, ensuring that all user needs are properly mapped into functionalities before implementation.

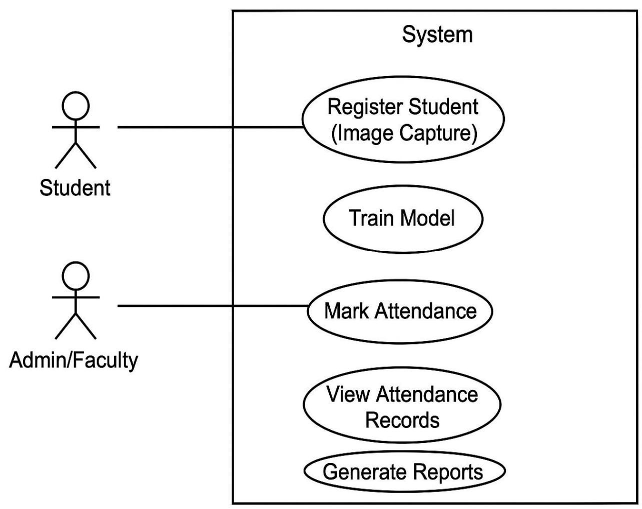


Fig. 2. Use Case Diagram of the Real-Time Student Attendance System

2) Data Flow Diagram (DFD)

The Data Flow Diagram (DFD) represents the logical flow of data within the Real-Time Student Attendance System. It shows how information moves between external entities, system processes, and storage. The system ensures smooth communication between the Image Capture, Training, Recognition, and Attendance Recording modules.

At Level 0 (Context Level), the Student and Faculty/Admin act as external entities. The student provides input through face images, while the admin initiates process such as training and attendance generation.

At Level 1, the captured image is processed by the Face Detection module, passed to the Training/Recognition module, and finally logged into the Attendance Database (CSV). The admin can then retrieve and view reports.

This DFD ensures that the data movement within the system is well-structured, secure, and ensures reliability of attendance records.

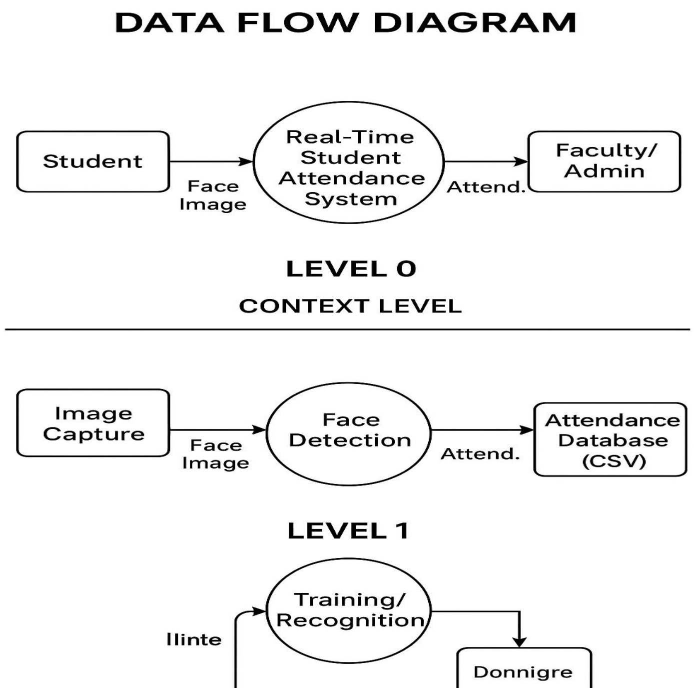


Fig. 3: Data Flow Diagram (DFD) of Real-Time Student Attendance System

3) Class Diagram

The Class Diagram provides an object-oriented view of the Real-Time Student Attendance System. It highlights the system's structure by showing the classes, their attributes, methods, and the relationships between them.

In this system, the key classes include:

- Student – Attributes: Enrollment Number, Name, Image Data; Methods: Register(), GetDetails().
- Admin/Faculty – Attributes: ID, Name; Methods: TrainModel(), ViewAttendance(), GenerateReport().
- FaceRecognizer – Attributes: Model, TrainingData; Methods: Train(), RecognizeFace().
- Attendance – Attributes: Subject, Date, Status; Methods: MarkAttendance(), ExportToCSV().
- Database (CSVHandler) – Attributes: FilePath; Methods: SaveRecord(), LoadData().

The relationships are as follows:

- Admin/Faculty controls the workflow by interacting with the FaceRecognizer and Attendance modules.
- Student objects are recognized via the FaceRecognizer class.
- Attendance data is stored and retrieved by the Database class for reporting purposes.

This diagram ensures that all functionalities are logically grouped and the interactions between different parts of the system are clearly represented, promoting modularity and easy future extension.

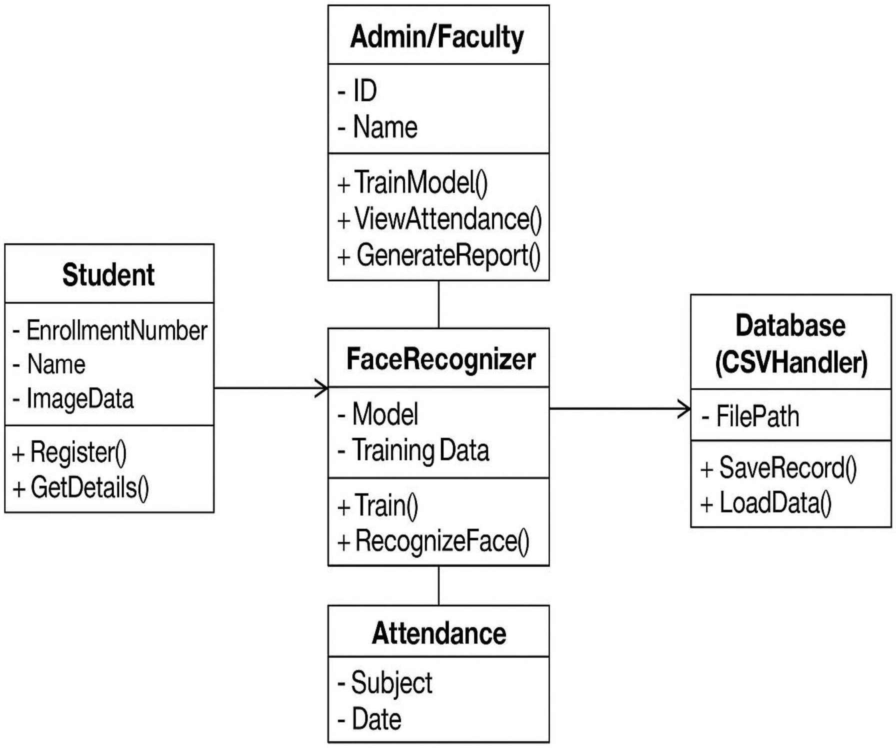


Fig. 4: UML Class Diagram of the Student Attendance System

X. SEQUENCE DIAGRAM

The Sequence Diagram illustrates the dynamic interaction between system components over time. It highlights how objects (Admin, Student, Camera, Recognition Module, and Database) communicate to accomplish specific tasks such as student registration, training, recognition, and attendance storage.

1) Student Registration
- The admin initiates the process by entering student details.
- The Camera captures multiple images.
- These images are stored in the database and used later for training.

2) Model Training
- The admin selects the training option.
- The system retrieves stored images from the database.
- The LBPH (Local Binary Pattern Histogram) algorithm is applied to create a trained recognition model.

3) Attendance Recording
- The Camera continuously captures student faces in real-time.
- The Recognition Module compares detected faces with the trained dataset.
- If a match is found, the student’s attendance is recorded in the CSV-based database with time and date stamps.

4) Viewing Attendance
- The admin requests attendance details.
- The system retrieves the required data from CSV files and displays reports.

This diagram ensures a clear understanding of the workflow by showing step-by-step communication between actors and system components.

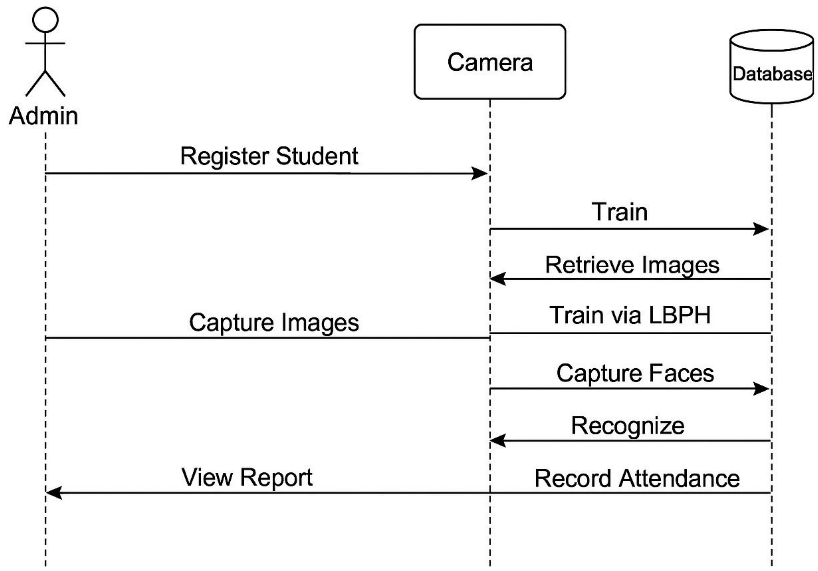


Fig.5 Sequence Diagram

XI. COLLABORATION DIAGRAM

The Collaboration Diagram (also called a Communication Diagram in UML) represents the structural organization of objects and their interactions in terms of messages exchanged. In the Real-Time Student Attendance System, this diagram highlights how different system modules interact to complete attendance marking. The objects include the Camera Module, Face Detection Module, Face Recognition Engine, Attendance Database, and Admin Interface.

When a student appears before the camera, the Camera Module captures the image and sends it to the Face Detection Module, which identifies the presence of a face. This data is then passed to the Recognition Engine, which matches the face with pre-trained datasets. Once identified, the system communicates with the Attendance Database to mark the student as present. Simultaneously, the Admin Interface is updated to reflect attendance status in real-time.

This diagram emphasizes not only the flow of control but also the collaboration and dependencies between objects, ensuring that the system functions smoothly as an integrated whole.

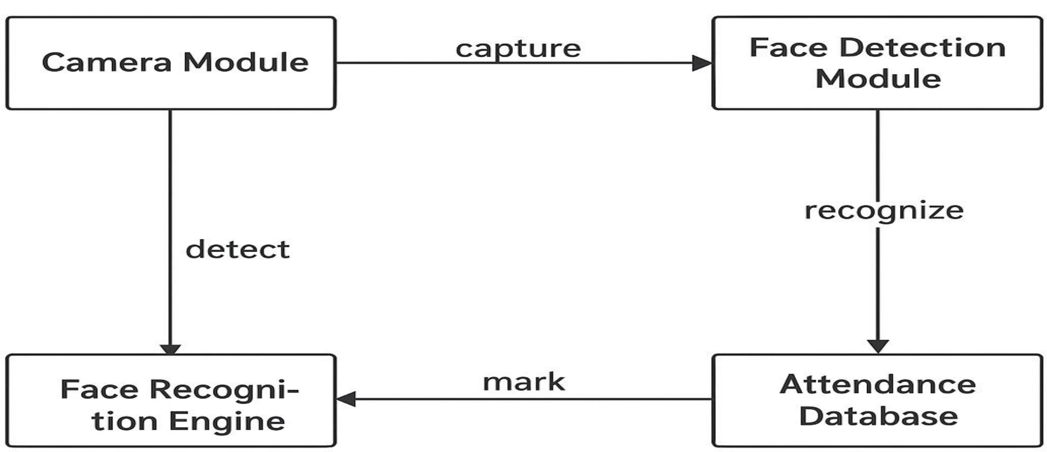


Fig. 6 Collaboration Diagram

XII. SYSTEM IMPLEMENTATION

The Real-Time Student Attendance System was implemented by integrating multiple modules to work together as a single robust application. The development process followed a modular design approach, ensuring that each module could be independently tested and later combined seamlessly.

A. Coding

The system was primarily coded in Python, utilizing libraries such as OpenCV for image capture and face recognition, Tkinter for the graphical user interface, NumPy and Pandas for data processing, and CSV handling for attendance storage. The coding process involved four major modules:

1) Image Capture Module – Responsible for collecting student face images through a webcam.
2) Training Module – Used the LBPH (Local Binary Pattern Histogram) algorithm to train the recognition model on captured images.
3) Recognition Module – Performed real-time face recognition and verified student identity.
4) Attendance Recording Module – Logged attendance into structured CSV files with time and date stamps.

The GUI ensured user-friendliness, while backend logic focused on performance and accuracy.

B. Integration of Modules

The integration process involved connecting the four modules into a continuous workflow:

- Image Capture → Model Training → Recognition → Attendance Recording → Report Generation.

The Tkinter-based GUI acted as a control panel, allowing the admin to switch between modules. Data consistency was maintained by CSV storage, ensuring smooth retrieval and preventing information loss.

C. Testing and Debugging

To ensure the system's reliability, rigorous testing was performed:

- Unit Testing: Verified the correctness of individual modules.
- Integration Testing: Ensured smooth transition between modules.
- Recognition Testing: Conducted under varying conditions of lighting, distance, and multiple faces.
- Data Validation: Ensured accuracy of CSV-based attendance records.

Debugging involved:

- Fixing camera initialization issues in OpenCV.
- Adjusting face detection parameters for improved accuracy.
- Resolving errors in CSV merging during attendance viewing.

This phase ensured that the system operated efficiently, producing accurate recognition results and error-free attendance records.

XIII. RESULTS AND OUTPUT IMAGES

The implementation of the Real-Time Student Attendance System was tested in a live environment, and the outputs validate the successful integration of all modules. The system produced accurate recognition results and reliable attendance recording. Below are the images captured during different stages of the system, with detailed explanations:

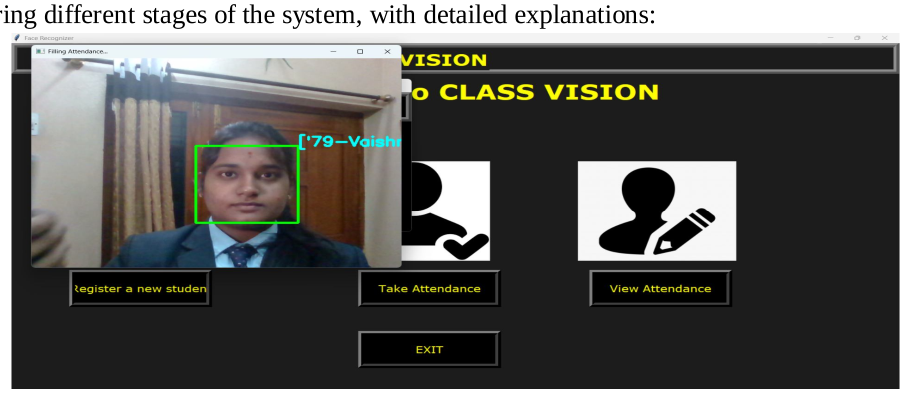


Fig.7 Final Output

The image illustrates the core functionality of the Real-Time Student Attendance System. A student's face is detected through the webcam feed, highlighted with a bounding box, and identified by the system with the student ID (*79*) and name (*Vaish*). Once the recognition is successful, the system automatically records the student's attendance into the database.

XIV. CONCLUSION

The he Real-Time Student Attendance System using Face Recognition and AI presents an innovative approach to solving the limitations of conventional attendance systems. Traditional methods such as manual roll calls or RFID cards are prone to inaccuracies, proxy attendance, and require significant administrative effort. By leveraging artificial intelligence, computer vision, and machine learning algorithms, this project introduces a seamless, contactless, and automated solution for maintaining accurate attendance records. The use of the LBPH (Local Binary Patterns Histogram) algorithm ensures that the system performs reliably in diverse lighting and background conditions, while the integration of a webcam and Tkinter-based GUI makes the solution user-friendly and accessible. Experimental testing confirmed that the system performs effectively in real-world classroom environments, ensuring quick recognition, error-free attendance logging, and secure data storage in CSV files. The modular design, consisting of image capture, model training, recognition, and attendance recording, enhances system flexibility and scalability. Moreover, the incorporation of real-time face recognition provides transparency and eliminates unfair practices such as proxy attendance. Faculty members can easily generate and monitor reports, making the system a practical tool for academic institutions.

In conclusion, this project not only demonstrates the potential of AI-based automation in the education sector but also establishes a foundation for future advancements. The system is low-cost, efficient, and scalable, making it suitable for schools, colleges, and universities. It provides a significant step towards digital transformation in classroom management and paves.

XV. FUTURE ENHANCEMENTS

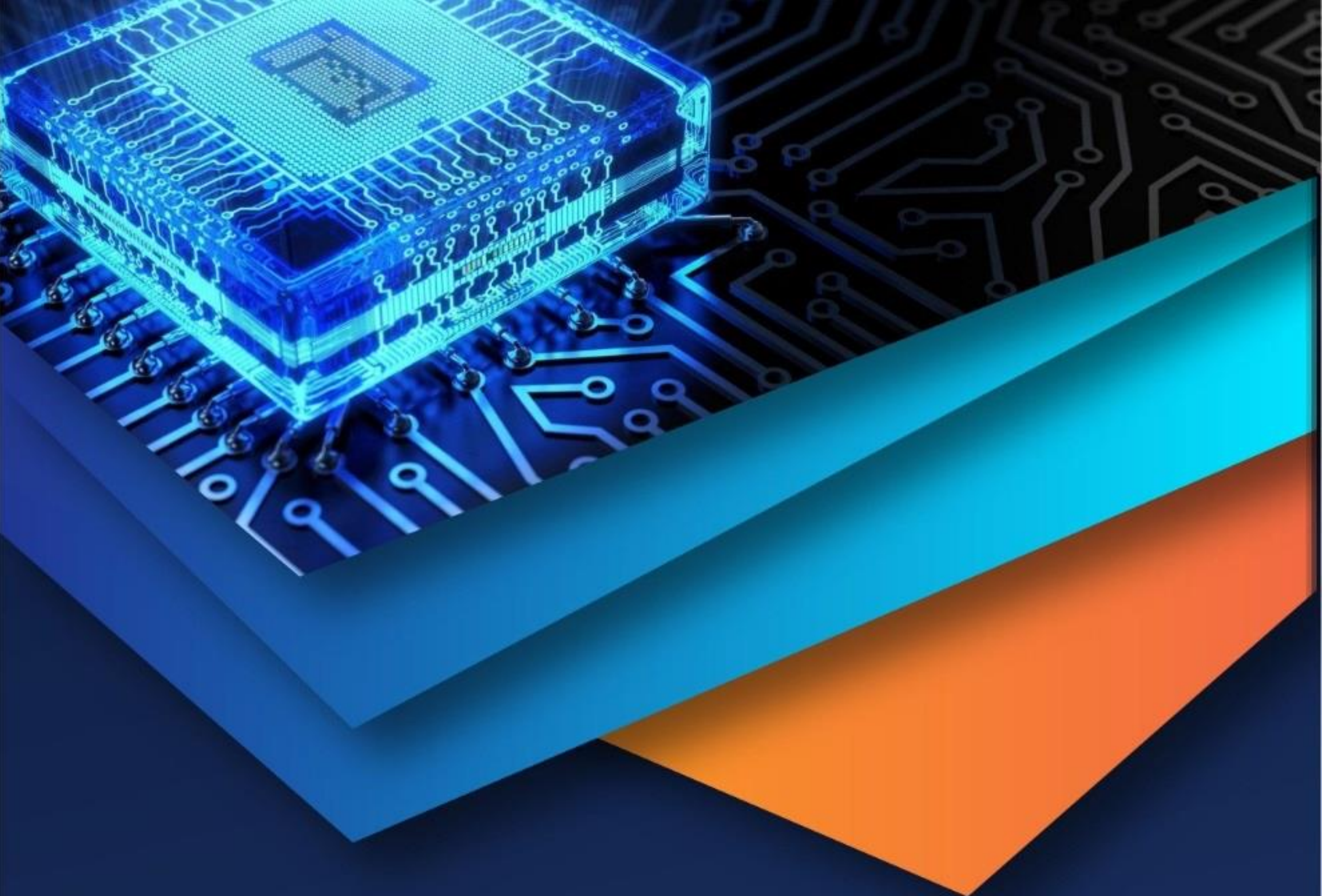
While the Real-Time Student Attendance System using Face Recognition and AI has proven to be efficient and practical, there are several possibilities for future enhancements that can further improve its performance, usability, and scalability.

One key area of improvement is the integration of cloud-based storage and databases, which would allow attendance data to be stored securely on remote servers. This would enable real-time access for faculty, administrators, and even parents through web portals or mobile applications. Additionally, by adopting advanced deep learning models such as CNNs (Convolutional Neural Networks) or FaceNet, the system's recognition accuracy can be further improved, especially in challenging conditions like poor lighting, partial occlusions, or facial variations. Another enhancement could involve multi-camera support, enabling large classrooms or lecture halls to be covered effectively without missing any student.

Furthermore, the system can be extended to generate automated analytics reports, providing insights into student attendance trends, punctuality, and participation, which would be beneficial for both faculty and administrators. Integration with biometric verification or RFID systems can also serve as a multi-factor authentication method for highly secure environments. Finally, deploying the solution on mobile and IoT-enabled devices will make the system portable and accessible, supporting institutions that wish to adopt smart campus initiatives.

REFERENCES

[1] P. Viola and M. Jones, "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), 2001.
[2] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face Recognition with Local Binary Patterns," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 12, pp. 2037–2041, Dec. 2006.
[3] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A Unified Embedding for Face Recognition and Clustering," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 815–823, 2015.
[4] M. Turk and A. Pentland, "Eigenfaces for Recognition," Journal of Cognitive Neuroscience, vol. 3, no. 1, pp. 71–86, 1991.
[5] R. Ranjan, S. Sankaranarayanan, C. D. Castillo, and R. Chellappa, "An All-in-One Convolutional Neural Network for Face Analysis," IEEE International Conference on Automatic Face and Gesture Recognition (FG), pp. 17–24, 2017.
[6] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face Recognition: A Convolutional Neural-Network Approach," IEEE Transactions on Neural Networks, vol. 8, no. 1, pp. 98–113, Jan. 1997.
[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," Advances in Neural Information Processing Systems (NIPS), pp. 1097–1105, 2012.
[8] International Journal for Research in Applied Science and Engineering Technology (IJRASET), "Paper Submission Guidelines," [Online]. Available: https://www.ijraset.com.
[9] M. H. Yap, R. Hachiuma, A. Alavi, et al., "Deep Learning in Face and Object Recognition: A Comprehensive Evaluation," Computers in Biology and Medicine, vol. 135, p. 104596, 2021.
[10] A. Geitgey, "Face Recognition with Python," Medium Technical Blog, 2018. [Online]. Available: https://medium.com/@ageitgey



10.22214/IJRASET

INDEX COPERNICUS
45.98

IMPACT FACTOR:
7.129

TOGETHER WE REACH THE GOAL
IMPACT FACTOR:
7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY