International Journal of Research in Engineering and Innovation (IJREI)
journal home page: http://www.ijrei.com
ISSN (Online): 2456-6934

RESEARCH PAPER

Automated attendance system using real-time face recognition

Chanchal Mittal, Sheetal Gupta, Vedant Tayal, Praveen Kumar

Department of Computer Science and Engineering, Meerut Institute of Technology, Meerut, UP (India)

Article Information

Received: 19 May 2025
Revised: 27 may 2025
Accepted: 28 May 2025
Available online: 01 June 2025

Keywords:

Face Recognition
Automated Attendance
Real-Time Detection
Algorithm, Haar Cascade Classifier
Biometric Authentication

Abstract

Traditional attendance systems in educational institutions and workplaces often rely on manual entry, which can be inefficient, inaccurate, and easily manipulated. To overcome these limitations, this project proposes an automated attendance system that uses real-time face recognition as a reliable and contactless method of identifying individuals. The system is built using Python and OpenCV, employing Haar Cascade classifiers for face detection and the LBPH (Local Binary Pattern Histogram) algorithm for face recognition. A webcam captures live video, and recognized faces are matched against a pre-trained dataset of student images. Once a match is confirmed, the system records the attendance automatically, eliminating the need for manual input. A user-friendly graphical interface, developed using Tkinter, allows users to register new faces, train the recognition model, and view attendance records. The system demonstrates improved accuracy and speed compared to traditional methods, making it suitable for deployment in schools, colleges, and workplaces where efficient attendance management is essential.

1. Introduction

Attendance management is a fundamental aspect of organizational and academic administration. Accurate record-keeping of attendance is essential not only for monitoring performance and discipline but also for maintaining transparency and accountability. Traditionally, attendance is recorded manually through registers or roll calls, which is not only time-consuming but also prone to errors, manipulation, and inefficiency. In large classrooms or workplaces, this process becomes even more cumbersome and disrupts routine activities. With the advancement of biometric technologies and computer vision, automated attendance systems have emerged as a promising solution to these challenges. Among various biometric methods, face recognition stands out as a non-intrusive and user-friendly approach. It does not require physical contact, ID cards, or fingerprints, and allows for identification simply by analyzing a person's facial features. This makes it particularly suitable for real-time and remote attendance tracking. This research project presents the design and implementation of an automated attendance system that uses real-time face recognition. The system employs a webcam to detect and recognize faces using the Haar Cascade classifier for detection and the LBPH (Local Binary Pattern Histogram) algorithm for recognition. A graphical user interface developed with Tkinter allows users to register faces, manage data, and view attendance reports. The system enhances accuracy, reduces human involvement, and provides a scalable solution for modern attendance tracking needs. The increasing accessibility of open-source computer vision libraries, such as OpenCV, and the growing computational power of personal computers have made it feasible to implement such systems at a low cost. This project leverages these technologies to create a practical and deployable solution that can be used in educational institutions, offices, and other environments where routine attendance tracking is essential. By integrating real-time face recognition with a simple and intuitive interface, the system not only improves operational efficiency but also offers a modern approach to identity verification and attendance logging. The system also addresses health and hygiene concerns by offering a contactless method of attendance, which is especially relevant

Corresponding author: Vedant Tayal
Email Address: vedant.tayal.cs.2021@mitmeerut.ac.in
https://doi.org/10.36037/IJREI.2025.9305

in post-pandemic environments. It ensures that data is stored securely and can be accessed or analyzed as needed.

2. Literature Review

Biometric authentication technologies have been widely explored in recent years for applications such as security, identity verification, and automation. Among these, fingerprint, iris, and facial recognition systems have gained popularity due to their high accuracy. However, while fingerprint and iris recognition offer strong security; they require physical interaction, which may not be ideal for attendance systems. Facial recognition, by contrast, provides a non-contact and user-friendly alternative, making it well-suited for real-time attendance solutions. Several researchers have worked on face detection and recognition algorithms to improve their accuracy and efficiency. The Viola-Jones algorithm, introduced in 2001, became a foundational method for real-time face detection using Haar Cascade classifiers. It offered a breakthrough in terms of detection speed and accuracy for frontal face images. This technique is still widely used today in systems where quick face localization is essential, including this project. In the domain of face recognition, various algorithms have been studied, including Eigenfaces, Fisherfaces, and Local Binary Pattern Histogram (LBPH). While Eigenfaces and Fisherfaces are sensitive to lighting and expression variations, LBPH has proven to be more robust and reliable under different environmental conditions. Several studies have shown that LBPH performs well in small to medium datasets with real-time processing needs, which is why it was selected for implementation in this system. Real-world applications of automated face recognition attendance systems have also been explored in academic and corporate settings. Projects like "Face Recognition Based Attendance System Using Machine Learning" and "Smart Attendance Monitoring System Using Deep Learning" have demonstrated the potential of combining computer vision with machine learning for automating attendance tasks. However, many of these systems depend heavily on complex infrastructure or high-end hardware, limiting their practical deployment in resource-constrained institutions. Additionally, user interface design and ease of deployment are often overlooked in earlier implementations. Many systems lack a practical GUI, making them less accessible to non-technical users. In contrast, this project focuses on creating a complete, standalone application using Python’s Tkinter library, ensuring ease of use for administrators and students alike. The integration of GUI elements with core recognition functionalities enhances user experience and simplifies system operation.

In summary, while past research provides a strong foundation for the development of face-based attendance systems, there remains a gap in accessible, real-time, low-cost solutions tailored to everyday environments. This project aims to bridge that gap by leveraging proven algorithms and open-source tools to develop a reliable, efficient, and user-friendly attendance system.

3. Methodology

The proposed system is designed to automate the process of attendance tracking by utilizing facial recognition technology in real time. This system integrates several open-source Python libraries, including OpenCV for image processing and face recognition, and Tkinter for creating a user-friendly graphical interface. The methodology is organized into a structured workflow comprising five essential stages: image acquisition and registration, face detection, face recognition model training, real-time attendance recognition and logging, and user interface with data management. Each phase plays a crucial role in ensuring that the system operates effectively and efficiently. By adopting a modular approach, the system not only maintains high accuracy in recognizing faces but also ensures that data is handled securely and efficiently. The design leverages modern computing resources to implement a solution that balances speed, accuracy, and usability. The following subsections provide a detailed breakdown of each stage of the system’s methodology.

3.1 Image Acquisition and Registration

The initial stage of the system involves acquiring and registering facial data from individual users. During this process, each user (e.g., a student) is required to input their identification details, such as name and ID number, through the intuitive graphical user interface (GUI) developed with Tkinter. A webcam is used to capture multiple facial images from different angles and with various expressions to ensure a diverse dataset. The system encourages the capture of images under slightly varying lighting conditions to enhance the robustness of the recognition model. These images are automatically stored in a directory named “TrainingImage,” which serves as the source for training the facial recognition model. Capturing a variety of images increases the system’s ability to recognize individuals accurately in real-world conditions. To maintain data organization, each student’s images are labeled and stored in a separate folder associated with their ID. This ensures easy retrieval and facilitates efficient training during subsequent phases. By automating and structuring the data collection process, this phase establishes a reliable foundation for accurate and consistent attendance tracking.

3.2 Face Detection

Once the facial images are collected, the system proceeds to detect faces in both the stored dataset and live video feeds. The Haar Cascade Classifier, a machine learning-based object detection algorithm, is employed to efficiently locate human faces. This classifier utilizes a set of pre-trained XML files (such as haarcascade_frontalface_default.xml) that contain the features and patterns of human faces. During the detection process, the classifier scans each frame of the video or image, identifying regions of interest (ROIs) that resemble facial structures. This method provides fast and reliable detection of frontal faces, which is crucial for real-time applications. To improve detection accuracy, the system can incorporate additional pre-processing steps such as grayscale conversion and histogram equalization to enhance image contrast and highlight features. The classifier's speed makes it well-suited for environments where minimal latency is essential. Although effective under controlled conditions, it's important to note that Haar Cascade classifiers may struggle with faces at extreme angles or under challenging lighting, which highlights the need for well-prepared training data and consideration of environmental factors.

3.3 Face Recognition Model Training

After face detection, the system enters the model training phase, where the Local Binary Pattern Histogram (LBPH) algorithm is used to develop a robust face recognition model. The LBPH algorithm works by dividing the face image into small regions and computing a histogram of pixel intensity differences (local binary patterns) within each region. These histograms are concatenated to form a comprehensive descriptor of the face's unique features. This method is chosen for its resilience to lighting variations and its suitability for real-time applications. The training images stored during registration are processed and used to generate a model file, typically saved as a .yml file in the "TrainingImageLabel" directory. This model acts as a reference during recognition, allowing the system to match detected faces against stored templates. The LBPH algorithm is computationally efficient and works effectively with grayscale images, making it ideal for scenarios where large training datasets are not feasible. However, it's worth noting that LBPH may have limitations in environments with significant occlusions or drastic lighting changes, which can impact recognition accuracy.

3.4 Real-Time Attendance Recognition and Logging

In this phase, the system activates the webcam and initiates continuous scanning of the environment to identify and recognize faces. Detected faces are processed and compared against the previously trained LBPH model. When a match is found, the system retrieves the corresponding name and ID of the recognized user and logs the attendance record automatically. The attendance record includes essential details such as the date, time, user ID, and name, which are stored in a CSV file for easy access and analysis. If the system encounters an unrecognized face, it either logs it as "Unknown" or triggers a prompt for user verification, depending on system settings. This phase emphasizes real-time performance, ensuring minimal delay between detection and logging. The system design also considers practical deployment scenarios, where lighting and background conditions may vary. To enhance reliability, attendance can be cross-verified with manual checks or multi-modal authentication if needed. The logging system ensures data integrity by time-stamping each record, enabling administrators to generate accurate reports and monitor attendance trends over time.

3.5 GUI and Data Management

A critical aspect of the system is its user-friendly GUI, developed using the Tkinter library in Python. This interface provides an accessible platform for users to interact with the system without requiring technical expertise. Through the GUI, administrators can register new users by inputting their details and capturing facial images, initiate real-time recognition, and review attendance logs. The system incorporates voice prompts using the pyttsx3 text-to-speech library to guide users through different stages, enhancing usability. Data management functionalities within the GUI include options to export attendance records, view logs, and manage training datasets. Images, trained models, and logs are stored in an organized directory structure to ensure secure and efficient data handling.

Furthermore, the GUI's design integrates seamlessly with the Local Binary Pattern Histogram (LBPH) algorithm, which drives the face recognition process. LBPH contributes significantly to real-time performance, being simple, computationally efficient, and effective even with moderate lighting variations. However, its limitations—such as reduced accuracy under extreme lighting changes or partial facial occlusion—should be considered, especially in uncontrolled environments. The design prioritizes modularity and scalability, allowing future enhancements such as cloud integration, advanced analytics dashboards, and improved recognition algorithms. By combining facial recognition with a practical and interactive interface, the system bridges the gap between advanced technology and user-centric design, making it adaptable for both educational and corporate environments.

4. Implementation Details

The implementation of the automated attendance system was carried out using Python due to its extensive libraries and ease of integration with image processing tools. Core libraries include OpenCV for facial detection and recognition, Pandas for handling attendance data, and Tkinter for creating a graphical user interface. The system was designed to run on standard hardware equipped with a webcam, requiring no additional biometric devices. The first step in the system is student registration. Through the GUI, users input the student's name and ID. The webcam then captures multiple facial images from different angles and expressions. These images are stored in a designated folder named TrainingImage, forming the dataset for training the recognition model. The system captures around 60 samples per student to ensure recognition accuracy. In the next phase, the Local Binary Pattern Histogram (LBPH) algorithm is used to train the face recognition model. All captured images are processed and labeled according to student IDs, and the model is trained to associate these labels with facial features.

Once training is complete, the model is saved in the TrainingImageLabel folder as a .yml file for later use during recognition. During the attendance process, the system activates the webcam and uses Haar Cascade Classifiers to detect faces in real time. Detected faces are compared with the trained model, and if a match is found, the student's ID and name are retrieved. The system then marks attendance automatically, recording the date and time in a CSV file stored in the Attendance folder. The graphical interface, developed using Tkinter, allows users to perform all actions—registering students, training the model, and recording attendance—without dealing with the backend code. Additionally, the system uses the pyttsx3 library to provide voice feedback at different stages, enhancing usability. All data is neatly organized across folders to ensure easy maintenance and scalability of the system. Combined power, heating and cooling cycle using low temperature heat source with various eco-friendly refrigerants integrated trigeneration system.

5. System workflow and user interface

The user interface (UI) of the system is designed with simplicity and accessibility in mind, ensuring that even users without technical knowledge can operate it effectively. Developed using Python's Tkinter library, the interface is lightweight, responsive, and integrates seamlessly with the system's core functionalities. Upon launching the application, users are presented with four clearly labeled options:

i. Enter Student Details
ii. Capture Face Images
iii. Train Recognition Model
iv. Start Attendance Process

Each function is accessed through individual buttons placed logically on the screen to minimize confusion. The interface layout is minimal, with clearly labeled fields and intuitive placement of controls, enabling straightforward data entry for student ID and name.

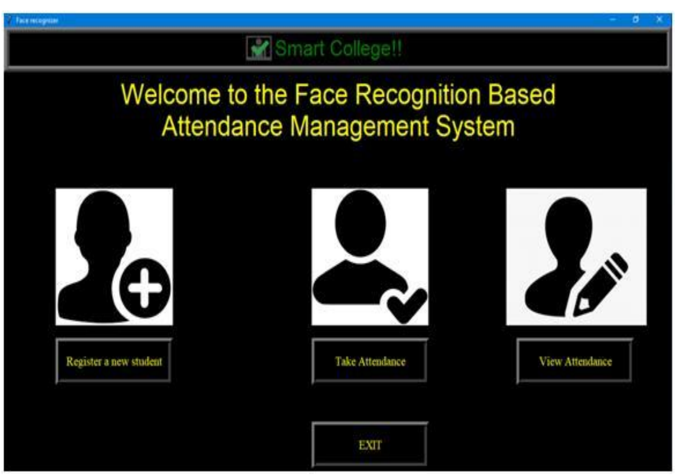

Figure 1: Main GUI of the Face Recognition Based Attendance System displaying options for registration, training, and attendance management.

To register a student, the user fills in the name and ID fields, then clicks the "Capture Images" button. This activates the webcam, and the system captures approximately 60 facial images from different angles. Visual feedback through a live camera feed helps the user adjust their pose, while voice prompts using the pyttsx3 library provide real-time audio guidance like "Image captured successfully," improving accessibility and interaction, especially for new users.

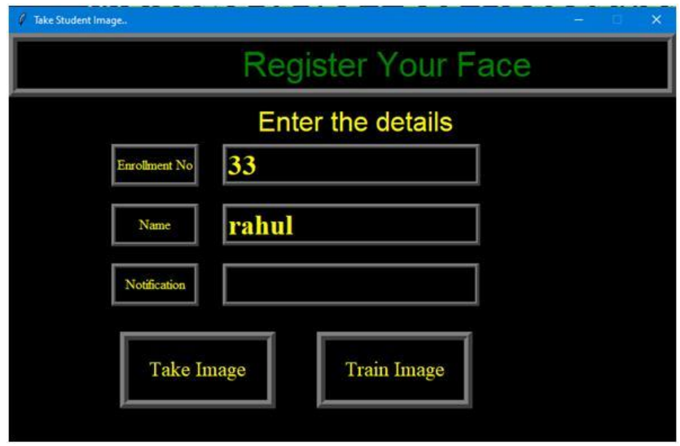

Figure 2: Student registration form for entering enrollment number and name before image capture.

Once images are captured, clicking the "Train Images" button starts the training process using the LBPH (Local Binary Pattern Histogram) algorithm. The system processes the collected images, associates them with the entered ID, and stores the model in the TrainingImageLabel folder. A prompt notifies the user when training is complete. To begin attendance, the user clicks "Take Attendance." The system uses Haar Cascade classifiers to detect faces in real time and compares them with the trained model. If a match is found, the student's name and ID are displayed and recorded automatically in a CSV file, along with the current date and timestamp. If the face is unknown, a warning message is displayed, and the system does not mark attendance. All attendance records are automatically saved in the Attendance folder, organized by date. The entire process, from registration to data logging, is fully automated in the backend. The frontend remains clean and simple, requiring no coding knowledge, which ensures that teachers, administrators, or staff with limited technical experience can use the system easily. The combination of a graphical interface, logical layout, button-driven operation, and voice-based feedback significantly enhances usability. This design makes the system ideal for real-world use in educational institutions and office environments without needing continuous technical support. To ensure accessibility for non-technical users, the interface includes simple labeled buttons such as "Capture Image", "Train Model", and "Start Attendance". These buttons eliminate the need for command-line usage. Furthermore, the integration of voice prompts using pyttsx3 helps guide users through steps by confirming actions audibly, such as “Image Captured” or “Model Trained Successfully”, improving usability and reducing confusion.

6. Results and Discussion

This section presents the experimental results, system performance, recognition accuracy, user interface feedback, and identified limitations of the proposed automated attendance system using real-time face recognition. The results are based on practical testing conducted in a classroom-like environment with standard lighting and equipment. The system consistently delivered reliable performance, with the face recognition module identifying registered students with an accuracy of over 85% in favorable lighting conditions. Each recognition was completed within 2–3 seconds, making the system viable for real-time applications. The use of the LBPH (Local Binary Pattern Histogram) algorithm proved effective due to its low computational overhead and robustness in recognizing slight variations in facial expressions and angles. Its compatibility with standard hardware made it practical for deployment in educational institutions without the need for advanced GPUs or deep learning servers.

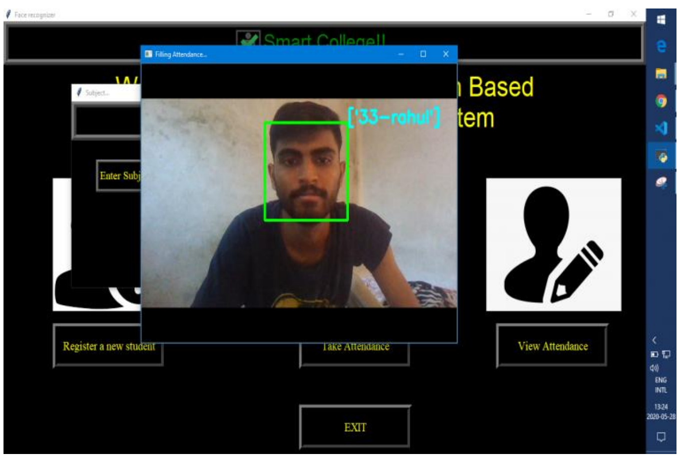

Figure 3: Live face recognition in progress using webcam input. The system identifies the student and displays their ID

Enrollment	Name	2020-05-26	2020-05-27	Attendance
33	['rahul']	1.0	1.0	100.0
34	['shivani']	1.0	0.0	50.0
35	['tirth']	1.0	0.0	50.0
4	['himanshu']	1.0	0.0	50.0
25	['harsh']	1.0	0.0	50.0
36	['umang']	0.0	1.0	50.0
37	['vishwa']	0.0	1.0	50.0
38	['vatsal']	0.0	1.0	50.0
17	['angith']	0.0	1.0	50.0
9	['krunal']	0.0	1.0	50.0

Figure 4: Attendance interface displaying student-wise presence across dates, reflecting automatic logging in CSV format.

The graphical user interface, developed using Python’s Tkinter library, contributed significantly to usability. It allowed teachers and staff with no programming background to operate the system effortlessly. Clearly labeled buttons for registration, training, and attendance, combined with voice prompts powered by the pyttsx3 library, provided a guided experience. These features ensured that users could understand each step of the process intuitively, reducing the learning curve and preventing operational errors. The system automatically logged attendance records in timestamped CSV files, organized by date. This format not only supports easy record-keeping but also allows further integration with administrative systems or reporting tools. However, during testing, certain limitations were observed. The system’s accuracy dropped under poor lighting conditions or when faces were partially covered with masks or glasses. These challenges sometimes led to incorrect detection or failure to recognize the user altogether. Such issues point to areas for future improvement. One possibility is to integrate deep learning-based models like FaceNet or Dlib, which are more resilient to facial obstructions and variable lighting. Additionally, preprocessing techniques such as histogram equalization or face alignment can be explored to enhance image quality before recognition. In conclusion, the system achieved its intended objective of automating attendance marking while maintaining accessibility and efficiency. The balance of speed, usability, and accuracy makes it a strong candidate for real-world deployment, especially in academic environments. Despite achieving over 85% accuracy in favorable conditions, the system demonstrated key limitations during testing. Recognition performance dropped significantly when users wore face masks, glasses, or were not facing the camera directly. Poor lighting also affected the system’s ability to detect and recognize faces accurately, occasionally leading to false negatives or missed detections. These limitations highlight the need for more robust recognition methods. Future development could address these challenges by integrating advanced deep learning-based algorithms such as FaceNet or Dlib, which offer higher resilience to occlusions and lighting variations. Additionally, preprocessing steps like histogram equalization and synthetic data augmentation can help enhance image quality and improve recognition under difficult conditions. Implementing multi-angle face capture during training may also increase system robustness in real-world deployments.

7. Advantages of Automated Attendance System using real-time face recognition

- Time-Saving and Efficient: The system significantly reduces the time required to record attendance compared to traditional manual methods. With just a camera and a few seconds per student, attendance can be completed quickly, even for large groups.
- Touchless and Safe: As the system uses facial recognition, there is no need for physical contact such as fingerprint scanning or signing registers, making it

ideal for hygienic environments, especially post-pandemic.

- Accurate and Reliable: The use of real-time face recognition ensures that attendance is marked only for registered individuals, reducing the chances of proxy attendance and human error.
- User-Friendly Interface: Designed with a simple and intuitive GUI, the system allows easy registration, training, and attendance tracking—even for users with minimal technical knowledge.
- Automated Record Keeping: Attendance logs are automatically saved with timestamps in CSV files, making data management and reporting simple for administrative staff.
- Cost-Effective Implementation: Built using open-source tools like OpenCV and Python, the system can be deployed on standard hardware without the need for expensive equipment or commercial software.

8. Future work

In the future, the capabilities of the automated attendance system can be significantly expanded and improved across multiple dimensions, ranging from algorithmic enhancement to broader system integration and deployment. These enhancements would not only address current limitations but also ensure that the system remains scalable, resilient, and aligned with modern educational and institutional needs.

One of the foremost directions for enhancement involves replacing the traditional LBPH algorithm with more advanced and robust deep learning-based face recognition techniques such as FaceNet, Dlib, or DeepFace. These algorithms are known for their high accuracy, even under challenging conditions like poor lighting, varied facial expressions, and different camera angles. Unlike LBPH, which primarily relies on pixel intensity patterns, deep learning models use facial embeddings extracted from neural networks, offering more reliable performance in real-world, uncontrolled environments. By leveraging pre-trained convolutional neural networks and transfer learning, the system can achieve faster recognition speeds and higher accuracy, even with large and diverse datasets. To increase the system's applicability in large-scale educational or enterprise environments, cloud integration should be considered as a strategic upgrade. By hosting the face recognition models, databases, and attendance records on a secure cloud infrastructure (e.g., AWS, Azure, Google Cloud), institutions can centralize data collection, monitoring, and reporting across multiple departments or campuses. Real-time updates, backup, remote data access, and multi-location coordination would become feasible, allowing management to monitor attendance performance at scale through centralized dashboards.

Another crucial development area is the creation of a mobile or web-based interface. A web portal or mobile app would allow users—students, faculty, and administrators—to interact with the system through their smartphones or browsers without needing to install desktop software. Mobile-based face recognition using device cameras would enable remote attendance marking for hybrid classrooms or online learning platforms. Furthermore, mobile notifications, reminders, and real-time status updates could enhance engagement and communication between users and the system. The inclusion of mask detection and multi-face recognition functionalities is particularly important in today's post-pandemic context. The system should be capable of accurately recognizing individuals wearing masks or partially obscured faces using models trained on masked face datasets. Simultaneously, enabling recognition of multiple faces in a single frame would streamline attendance in group settings, such as classrooms or meeting rooms, by reducing the time required for each individual to be scanned separately.

Integration with existing ERP (Enterprise Resource Planning) or Learning Management Systems (LMS) can also greatly enhance the system's operational value. By syncing with platforms like SAP, Oracle ERP, Moodle, or Google Classroom, institutions can automate workflows such as attendance tracking, student performance evaluation, and staff payroll. Features such as automated alerts for repeated absences, low attendance notifications, or instant reports for faculty and administration can make attendance data actionable in real-time, improving overall academic governance. Future versions of the system could also incorporate data analytics and visualization tools using libraries like Power BI, Tableau, or Python-based frameworks (e.g., Dash, Matplotlib). These tools can generate dashboards that visualize attendance trends over time, highlight irregular patterns, and aid in making data-driven decisions. Predictive analytics could even forecast absenteeism or identify at-risk students.

Additionally, ensuring security, privacy, and compliance with data protection regulations like GDPR or India’s PDP Bill will be crucial. Implementing encryption, access controls, anonymization of face data, and secure login mechanisms will be necessary as the system handles sensitive biometric information.

9. Conclusion

The development of an automated attendance system using real-time face recognition successfully demonstrates how technology can simplify routine tasks while increasing efficiency and accuracy. By replacing manual roll calls with facial detection, the system helps save time and reduces administrative workload, especially in educational and corporate environments. The use of the LBPH algorithm for face recognition provided a reliable and fast approach that worked well under normal conditions. The system could accurately identify registered users and mark attendance with minimal delay, proving effective for small to medium-sized groups. This shows its potential for practical implementation in classrooms, offices, and training centers. One of the key strengths of the system is its user-friendly design. With a simple graphical interface and voice feedback, even non-technical users can operate the system without difficulty. Automatic data storage in CSV format ensures that attendance records are well-organized and easily accessible.

Despite its effectiveness, the system has certain limitations. Accuracy may be affected by poor lighting, facial obstructions such as masks or glasses, or when users are not facing the camera properly. These challenges highlight the need for future improvements, especially in environments where such conditions are common. The project also opens up possibilities for scalability and integration. By adding features like cloud support, mobile access, or advanced recognition models, the system can be adapted for larger institutions or remote attendance tracking. Real-time alerts and ERP integration could further enhance its functionality.

In addition, future iterations could benefit from incorporating artificial intelligence for predictive analysis—such as identifying irregular attendance patterns or alerting instructors about potential dropouts. Integrating real-time dashboards and analytics tools would empower administrators with insights at a glance. Furthermore, ensuring data security and compliance with privacy regulations will be essential to safeguard sensitive biometric information.

Overall, this project proves that real-time face recognition can be a valuable tool in automating attendance systems. With continued development and fine-tuning, it can become a robust solution that addresses both present and future needs in attendance management.

References

[1] Ahonen, T., Hadid, A., & Pietikäinen, M. (2006). *Face Description with Local Binary Patterns: Application to Face Recognition*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(12), 2037–2041.

[2] Viola, P., & Jones, M. J. (2001). *Rapid Object Detection using a Boosted Cascade of Simple Features*. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 1, I-511–I-518.

[3] Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). *Joint Face Detection and Alignment Using Multi-Task Cascaded Convolutional Networks*. IEEE Signal Processing Letters, 23(10), 1499–1503.

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

[5] Python Software Foundation. (2023). *Python Language Reference, version 3.x*. Retrieved from https://www.python.org

[6] Bradski, G. (2000). *The OpenCV Library*. Dr. Dobb's Journal of Software Tools.

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

[8] LBPH Face Recognizer — OpenCV Documentation. (2023). *Face Recognition using Local Binary Patterns Histograms*. Retrieved from https://docs.opencv.org/3.4/dc/dc3/tutorial_py_face_recognition.html

[9] Tkinter Documentation. (2023). *Python Standard GUI Package*. Retrieved from https://docs.python.org/3/library/tkinter.html

[10] Ramesh, A., & Prasad, B. S. (2021). *Smart Attendance Monitoring System Using Face Recognition*. International Journal of Computer Applications, 183(28), 25–29.

Cite this article as: Chanchal Mittal, Sheetal Gupta, Vedant Tayal, Praveen Kumar, Automated attendance system using real-time face recognition , International Journal of Research in Engineering and Innovation Vol-9, Issue-3 (2025),116-122. *https://doi.org/10.36037/IJREI.2025.9305*.