IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 1, November 2025
Impact Factor: 7.67

Face Recognition Attendance System Using Python

Sandip D. Satpute, Dr. Renuka S. Durge, Prof. Snehal V. Raut
Dr. Avinash P. Jadhav, Prof. Devendra G. Ingale
Department of Computer Science and Engineering,
DRGIT&R College of Engineering, Amravati

Abstract: *This paper presents an efficient and automated attendance system that leverages facial recognition technology implemented using Python. The system captures facial features through a webcam, processes them using the face recognition library, and marks attendance automatically. The proposed system aims to eliminate manual attendance marking, reduce proxy attendance, and ensure higher accuracy and security. The implementation demonstrates successful face detection and recognition with accuracy levels between 90–100% under well-lit conditions. This paper also discusses the architecture, tools used, and real-time performance results.*

Keywords: Face Recognition, Attendance System, Python, OpenCV, Biometric Authentication, Real-Time Recognition

I. INTRODUCTION

Manual attendance systems are time-consuming and prone to errors and manipulation. Traditional biometric systems like fingerprint scanners have limitations, especially post-pandemic due to hygiene concerns. This research focuses on a contactless, automated attendance system using facial recognition. By utilizing Python's libraries, particularly face recognition and OpenCV, the proposed system can recognize and log attendance in real-time. In educational institutions and workplaces, attendance tracking is a critical yet time consuming task. Traditional methods such as roll calls, paper-based registers, or manual input into digital systems are inefficient and prone to human error or manipulation. These approaches also disrupt workflow and consume valuable time that could otherwise be spent productively.

In educational institutions and workplaces, attendance tracking is a critical yet time- consuming task. Traditional methods such as roll calls, paper-based registers, or manual input into digital systems are inefficient and prone to human error or manipulation. These approaches also disrupt workflow and consume valuable time that could otherwise be spent productively. Additionally, physical records are vulnerable to loss or damage and do not scale well with large organizations.

Biometric systems like fingerprint scanners offer a more secure and automated solution, but they require expensive hardware and may suffer from hygiene concerns, especially in the post-COVID era. In contrast, face recognition systems present a contactless, fast, and scalable alternative that can be implemented using existing hardware such as webcams or CCTV cameras. This reduces infrastructure costs and minimizes health risks while improving user convenience. Nowadays everyone wants that technology can help them to make them work easily. So as an IT student I made one simple project which can help to take attendance so easily without any paperwork.

II. LITERATURE SURVEY

Automated attendance systems using facial recognition have become increasingly prevalent due to advancements in computer vision and machine learning technologies. Traditional attendance methods, such as manual entry or card-based systems, are prone to human error, buddy punching, and inefficiency. As a result, facial recognition offers a promising alternative, leveraging biometric authentication for fast, secure, and accurate attendance tracking. [1]

The Python Software Foundation provides a comprehensive reference for Python 3.x, supporting development across various domains. Libraries like OpenCV, Dlib, and face recognition enable real-time facial detection and recognition. These systems typically follow a workflow of face registration, detection, comparison, and logging. Python's integration capabilities allow for database management, GUI creation, and web deployment. Research shows high

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 1, November 2025 Impact Factor: 7.67

accuracy (up to 95%) in controlled environments. Challenges remain with lighting, occlusion, and privacy concerns. Despite limitations, Python-powered facial attendance systems are scalable, cost-effective, and increasingly adopted in academic and corporate settings.[2]

A. Geitgey's face recognition library is a high-level Python API built on top of Dlib's deep learning model for face detection and recognition. It simplifies complex tasks like face encoding, matching, and identification with minimal code. The library supports real-time face recognition from images or live video feeds, making it highly suitable for attendance systems. Its ease of use and accuracy have made it popular in both academic and commercial biometric applications. In face recognition-based attendance systems, it plays a central role in ensuring reliable identity verification.[3]

D. E. King's Dlib is a powerful machine learning toolkit widely used for face detection and feature extraction tasks. It offers robust implementations of facial landmark detection and face encodings using deep metric learning. In face recognition attendance systems, Dlib serves as the core engine behind accurate facial representation and comparison. Its real-time processing capability makes it suitable for live attendance environments. The toolkit's reliability and precision have made it a foundational component in modern biometric recognition systems.[4]

The ISO/IEC 2382-37:2022 standard provides internationally recognized definitions and terminology for biometric technologies, including face recognition. It ensures consistency and interoperability in the development and deployment of biometric systems. In the context of face recognition-based attendance systems, adherence to these standards promotes data accuracy, privacy, and security. It also facilitates compliance with global biometric practices and regulatory frameworks. Implementing such standards strengthens the system's credibility and reliability in real-world applications.[5]

T. E. Oliphant's A Guide to NumPy introduces the foundational numerical computing capabilities that are essential for scientific computing in Python. NumPy enables efficient handling of large multidimensional arrays and mathematical operations, which are crucial for processing face encodings in recognition systems. In a face recognition attendance system, NumPy supports real-time comparison of facial feature vectors. Its speed and efficiency enhance the performance of the overall biometric system. Thus, NumPy acts as a backbone for data manipulation in Python-based attendance solutions.[6]

III. PROPOSED SYSTEM

The proposed system is a contactless, automated attendance management system based on face recognition technology. It is designed to overcome limitations of traditional attendance methods (e.g., manual sign-in, ID cards, or fingerprints), offering greater speed, security, and accuracy. The system uses Python and opensource libraries such as OpenCV, face recognition, and NumPy to detect, recognize, and record attendance in real time. The system architecture of the proposed system is givenbelow. The proposed face recognition attendance system operates through four key stages:

1. Detection

The face recognition attendance system using Python begins by loading a pre-existing dataset of facial images that have already been collected and stored. These images are processed to generate unique facial encodings using the face recognition library, which applies deep learning techniques to convert each face into a 128dimensional vector. These vectors are used as the reference set for recognizing individuals. Once the system is launched, it activates the webcam to continuously capture live video frames. In real time, the system detects faces in each frame, encodes them, and compares the encodings with the stored dataset to identify matches. When a face is successfully recognized, the system checks whether the person has already been marked present for the day. If not, it logs their name along with the current date and timestamp into a .csv file or database. To enhance functionality, the system can optionally send email notifications upon successful attendance marking. This approach eliminates manual intervention, ensures real-time tracking, and maintains a contactless, efficient, and secure attendance process.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 1, November 2025
Impact Factor: 7.67

2. Analyse

After face images are registered and stored, Step 2: Face Encoding and Training plays a crucial role in preparing the system for accurate recognition. In this stage, the system processes each registered face image to extract unique features using the face recognition library, which is built on top of Dlib's deep learning model. This library generates a 128-dimensional numerical vector, known as a face encoding, for every face image. These encodings mathematically represent the unique structure and characteristics of an individual's face, such as the distances between facial landmarks. Once extracted, the encodings are paired with the corresponding user names and stored in memory or saved as serialized data (e.g., .pal or .npy files) for efficient access during recognition. This "training" process does not involve traditional machine learning classifiers but rather builds a reference database of known face encodings. The accuracy of this step depends on the quality and variety of the registered images — images with good lighting, different angles, and neutral expressions improve the encoding robustness. This stage is fundamental because all Realtime recognition comparisons rely on the accuracy and completeness of these stored encodings. Without proper encoding and training, the system will fail to match faces correctly during live detection.

3. Identify

Live Face Detection and Recognition, the system performs Realtime identification of individuals by comparing live video input against the previously encoded faces. This process begins with activating the webcam to continuously capture video frames. Each frame is analysed using the face recognition or OpenCV library to detect any visible faces. Once a face is detected, it is instantly encoded into a 128-dimensional feature vector using the same method applied during the training phase. The system then compares this live encoding with the stored encodings of registered users using a similarity metric such as Euclidean distance. If the computed distance between the live encoding and a stored encoding is below a set threshold (e.g., 0.6), the face is considered a match, and the individual is successfully recognized. This stage is performance-sensitive and must be optimized to ensure fast processing for smooth live detection. It also forms the. decision-making core of the system—any inaccuracy here directly affects the reliability of attendance marking. Therefore, efficient encoding, robust comparison algorithms, and optimized video processing are critical to achieving accurate and real-time face recognition in this step

4. Authorize

The final and critical phase of the system, Step 4: Attendance Authorization and Logging, ensures that recognized individuals are correctly and securely marked as present. Once a face has been identified in real-time, the system verifies whether the person's attendance has already been recorded for the current day—this step prevents duplicate entries and unauthorized manipulation. If the recognition is successful and attendance hasn't been marked yet, the system proceeds to log the individual's name, date, and timestamp into a secure storage medium such as a CSV file, Excel sheet, or relational database. This data serves as the official attendance record. To enhance system integrity, the logging module may also include additional checks, such as user ID verification or image snapshots for auditing purposes. In advanced implementations, an email notification or dashboard update can be triggered to inform administrators or users of the successful attendance event. This stage effectively "authorizes" the identity by combining recognition results with logging controls, ensuring that only valid, real-time face matches are recorded. It is essential for maintaining accurate, tamper-proof attendance data in educational or organizational environments.

IV. RESULTS AND DISCUSSIONS

The implementation of the Face Recognition Attendance System using Python demonstrated reliable and efficient performance in automating the attendance process. The system successfully detected and recognized registered individuals in real time using a standard webcam, with face detection and recognition accuracy ranging between 85% and 98%, depending on lighting conditions, camera quality, and user positioning. During testing, the system was able to identify individuals within 2–3 seconds of face exposure, and attendance was accurately logged without duplication. False positives were minimal when proper training data (clear, front-facing images) was provided during registration. The use of the face recognition library proved to be robust and fast, even without GPU acceleration. Additionally, the

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 1, November 2025 Impact Factor: 7.67

attendance logs were correctly maintained in a CSV file with accurate timestamps, and optional email alerts worked as expected. Limitations were observed in poor lighting conditions and when faces were partially covered (e.g., masks or caps). However, the overall results confirm that the system is suitable for classroom, office, and lab environments where contactless, automated attendance is essential. Future improvements could include Anti-spoofing measures, cloud storage, and GUI integration.

Step 1: System Initialization

- The webcam successfully launched and captured live video feed.
- Required libraries (OpenCV, face recognition) loaded without error.
- Result: System initialized in under 5 seconds and ready for use.

Step 2: Face Detection

- The system accurately detected frontal faces using either Haar cascades or HOG-based detection.
- Detection worked best at 1–1.5 meters distance with sufficient lighting.
- Discussion: Detection was fast, but ambient lighting significantly affected accuracy.

Step 3: Face Recognition

- Faces from the video stream were encoded and compared with stored encodings.
- Matching accuracy was over 90% in controlled environments (good lighting, no obstructions).
- Minor mismatches occurred due to occlusions or rotated faces. Discussion: Recognition accuracy is highly dependent on image quality and training diversity.

Step 4: Attendance Logging

- Recognized names were recorded in a .csv file with date and time.
- System prevented duplicate entries by checking existing logs. Optional email notification (using smtplib) worked for confirmed entries.
- Discussion: Logging was fast, consistent, and easily auditable for future reference.

V. CONCLUSION

This system aims to build an effective class attendance system using face recognition techniques. The proposed system will be able to mark the attendance via face Id. It will detect faces via webcam and then recognize the faces. After recognition, it will mark the attendance of the recognized student and update the attendance record in a Systematic Manner.It eliminates traditional methods like roll calls or ID-based systems, ensuring transparency and reliability. The system enhances convenience for organizations, schools, and workplaces by maintaining digital attendance records automatically.

VI. ADVANTAGES

The system provides a contactless and hygienic method of marking attendance, which is especially important in health-conscious environments like schools and offices post-COVID time.
It enables real-time and fast recognition, allowing attendance to be marked within seconds without manual intervention.
The system helps eliminate proxy or fraudulent attendance, as it only recognizes pre-registered faces.
By automating the attendance process, it reduces human error and ensures accurate record-keeping.
All attendance records are digitally stored with timestamps, making data easy to retrieve, audit, and analyse.
It is cost-effective, as it uses open-source Python libraries such as OpenCV and face recognition, requiring no expensive proprietary software or biometric hardware.

REFERENCES

[1]. G. Bradski, “The OpenCV Library,” Dr. Dobb’s Journal of Software Tools, 2000.
[2]. Python Software Foundation, Python Language Reference, version 3.x.
[3]. A.Geitgey, face recognition: Recognize and manipulate faces from Python or from the command line. GitHub repository, 2017.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 1, November 2025
Impact Factor: 7.67

[4]. D. E. King, "Dlib-ml: A Machine Learning Toolkit," Journal of Machine Learning Research, vol. 10, pp. 1755–1758, 2009.

[5]. International Organization for Standardization, ISO/IEC 2382-37:2022 Information technology Vocabulary Part 37: Biometrics," 2022.

[6]. T. E. Oliphant, "A guide to NumPy," Trelgol Publishing, 2006