iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: V Month of publication: May 2025

DOI: https://doi.org/10.22214/ijraset.2025.68892

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 V May 2025- Available at www.ijraset.com

# Face-Based Attendance System Utilizing Python and Open-Source Technologies

Bashu Sanwariya[1], Waseem Khan[2]
*Department of Computer Science, Vivekanand College of technology and Management Aligarh, Uttar Pradesh*

***Abstract: In recent years, the demand for intelligent and automated solutions has grown exponentially across all sectors, especially in education, corporate workplaces, and government organizations. One critical challenge is efficient and secure attendance monitoring, which traditionally involves manual logbooks, RFID cards, or biometric fingerprint systems. These conventional approaches suffer from several drawbacks including time consumption, ease of manipulation (such as buddy punching), high maintenance, hygiene concerns, and potential data inaccuracies. With the emergence of Artificial Intelligence (AI) and Computer Vision technologies, revolutionizing attendance tracking through facial recognition has become highly practical.***

***This paper introduces a Face-Based Attendance System designed using Python and key libraries such as OpenCV, NumPy, Pandas, Tkinter, and the CSV module. The proposed system utilizes real- time video streaming from a webcam to detect, recognize, and authenticate users based on facial features. By applying machine learning algorithms and image processing techniques, the system accurately identifies individuals and records attendance without physical interaction, making it particularly beneficial in post- pandemic contexts where contactless systems are preferred.***

***The system architecture integrates user-friendly GUI components for enrollment and real-time monitoring, providing a seamless experience for administrators and users. Attendance data is stored in structured CSV files, ensuring easy access, portability, and compatibility with other administrative tools. Implementation prioritizes scalability, allowing integration with cloud services or institutional databases for larger deployments.***

***Keywords: Face Recognition, Attendance Automation, Artificial Intelligence, OpenCV, Python, Biometric System, Tkinter GUI, CSV Logging, Machine Learning, Image Processing.***

## I. INTRODUCTION

Attendance tracking is a critical administrative process in various domains, including schools, universities, corporate offices, and government institutions. Ensuring accurate and authentic attendance records is essential for managing productivity, performance, compliance, and accountability. However, traditional methods of attendance monitoring, such as manual roll-calls or ID card swiping, are not only time-consuming and inefficient but are also susceptible to manipulation and errors. Common issues such as buddy punching, lost ID cards, and forgetfulness compromise the integrity of these systems.

With rapid advancements in artificial intelligence and computer vision, facial recognition technology has emerged as a powerful alternative for secure and contactless attendance tracking. Especially in the context of global health crises like the COVID-19 pandemic, there is an urgent demand for hygienic and automated systems that reduce physical contact while maintaining operational efficiency. The Face-Based Attendance System proposed in this paper leverages AI algorithms in conjunction with Python programming and powerful libraries such as OpenCV and Face Recognition to develop a reliable, fast, and user-friendly solution. The system captures real- time video input through a webcam, processes facial data to extract unique features, and matches them with a pre-registered dataset to confirm identity. Attendance is automatically logged into a CSV file with time and date stamps, ensuring accuracy and ease of record- keeping.

This solution not only addresses the limitations of conventional systems but also introduces a user-centric graphical interface built with Tkinter, allowing even non-technical users to operate the system with minimal training. The use of open-source libraries ensures cost-efficiency and promotes scalability for diverse use cases, from classrooms to large enterprise environments.

## II. LITERATURE REVIEW

The development of biometric and AI-based attendance systems has been a topic of extensive research over the past two decades. Early attendance systems relied on manual sign-ins and basic ID-based mechanisms, which, although simple to implement, were often plagued by inaccuracies, inefficiency, and susceptibility to fraud such as proxy attendance.

Fingerprint-based systems emerged as one of the first widely adopted biometric solutions due to their reliability and low cost. Studies like Maltoni et al. (2003) and Jain et al. (2004) established fingerprint recognition as a benchmark for biometric authentication. However, the need for physical contact raised hygiene concerns, particularly during and after the COVID-19 pandemic, leading to increased interest in contactless biometric systems.

Iris recognition, as researched by Daugman (2004), offered excellent accuracy but suffered from high costs and user discomfort. Voice recognition and palm-vein scanning were explored but lacked mass acceptance due to variability in environmental conditions and high deployment costs.

Facial recognition gained significant traction with the advent of deep learning and powerful computing resources. The seminal work by Zhao et al. (2003) laid the groundwork for facial analysis systems. With the introduction of deep neural networks, models like DeepFace (Taigman et al., 2014), FaceNet (Schroff et al., 2015), and VGG-Face (Parkhi et al., 2015) showed remarkable improvements
in recognition accuracy and robustness.

OpenFace, an open-source project based on deep neural networks, and Dlib, a powerful machine learning toolkit, made it possible for developers and researchers to build practical facial recognition systems. These tools support 128-dimensional face encodings, which have become the standard for high-accuracy face matching.

Recent developments have further explored the use of Convolutional Neural Networks (CNNs), Generative Adversarial Networks (GANs) for data augmentation, and transfer learning for small dataset adaptability. Researchers like Ghesu et al. (2016) and He et al. (2016) have shown how residual learning and feature pyramids can improve performance under challenging conditions like poor lighting, aging, or occlusion.

In terms of application, educational institutions and large organizations have increasingly adopted facial recognition for attendance. The University of Manchester's pilot program (2018) and China's mass deployment in schools showcase its real-world viability. Yet, ethical debates regarding surveillance, data privacy, and bias in facial recognition algorithms remain active topics, encouraging further research in fairness, explainability, and governance.

## III. OBJECTIVES

The primary objectives of this project are as follows:

1) To develop a robust and intelligent face-based attendance system utilizing artificial intelligence and machine learning techniques that ensures reliable, fast, and automated identification of individuals.
2) To eliminate manual intervention in attendance systems, thus reducing the scope for human error, manipulation, and fraudulent marking (proxy attendance).
3) To ensure the system operates in real-time and maintains high accuracy under varying environmental conditions including different lighting, angles, and occlusions.
4) To create a user-friendly and responsive graphical interface using Tkinter that allows ease of use for non- technical staff such as school teachers or HR personnel.
5) To achieve cost-effectiveness and simplicity in deployment so that it can be implemented in institutions and organizations without the need for high-end infrastructure.
6) To incorporate modular coding practices that allow for future updates and scalability, including features like cloud storage, mobile access, multi-modal biometric integration, and web-based dashboards.
7) To maintain structured and real-time logging of attendance in universally accessible formats such as CSV or Excel-compatible sheets.
8) To ensure data privacy, ethical usage of biometric data, and system security through techniques like encryption and role-based access.
9) To demonstrate the practical use of AI and computer vision in solving real-world administrative problems and highlight its relevance in educational and corporate domains.
10) To contribute to the body of research in smart attendance systems by providing experimental validation, comparative analysis, and scalability benchmarks.

## IV. TECHNOLOGIES USED

This system utilizes a powerful suite of modern Python-based libraries and tools to execute its core functionalities. Each technology has been selected for its reliability, ease of integration, community support, and compatibility with real-time computer vision and data handling tasks:

1) Python: Serves as the backbone of the system. It offers rich library support, an active developer community, and excellent documentation. Python is particularly well- suited for AI and machine learning applications due to its versatility and the availability of powerful libraries.
2) OpenCV (Open Source Computer Vision Library): A robust and efficient open-source library used for real-time image processing and computer vision. It provides the tools necessary for facial detection, video capturing, drawing overlays, and manipulating visual data frames. OpenCV allows seamless interaction with webcam hardware and plays a crucial role in both the face detection and recognition stages.
3) Face Recognition (Built on Dlib and ResNet): This library simplifies the process of facial recognition using deep learning models. It extracts 128-dimensional facial feature vectors, which are compared for identity verification. Dlib uses a ResNet (Residual Neural Network) model under the hood, enabling highly accurate face matching and tolerance to variations in pose, expression, and lighting.
4) Pandas: A powerful library for handling structured data. It is used in this system to read and write attendance logs in CSV format and manipulate time series data efficiently. Its tabular data structure makes it easy to track attendance records, filter entries, and generate summaries or reports.
5) NumPy: Essential for numerical computation and matrix operations, which are critical in image and encoding data handling. NumPy provides the foundational data structures and performance enhancements needed to process the facial encoding arrays.
6) Tkinter: A built-in GUI toolkit in Python used for creating the system's graphical interface. Tkinter allows the application to be user-friendly, with buttons, text fields, and display areas for capturing and managing facial data and attendance logs.
7) CSV Module: This built-in module facilitates lightweight data logging into CSV format without the need for a database. It allows for easy portability, backup, and integration with Excel and other spreadsheet tools.
8) Datetime Library: Provides functions to timestamp attendance entries precisely, marking exact dates and times for each attendance session.
9) OS and Shutil Modules: Used for managing image files, directories, and cleaning up datasets. These modules support system-level tasks like file saving, renaming, and deletion, ensuring organized dataset storage.
10) Matplotlib and Seaborn (Optional): These libraries can be used to visualize attendance trends and recognition statistics in future versions. Graphs showing attendance over time, recognition accuracy, or student participation rates can help administrators with decision-making.

## V. WORKING MECHANISM

The Face-Based Attendance System follows a multi-stage pipeline that leverages real-time computer vision and deep learning to identify individuals and record their attendance automatically.
Below is a detailed breakdown of the system's working mechanism:

### A. *Dataset Preparation and User Registration*

The first phase involves building a dataset of registered users. This is accomplished through a user registration interface, where individuals stand in front of the camera and multiple facial images are captured. These images are stored with appropriate labels (user names or IDs) and serve as the training dataset.

- Images are stored in a structured directory format for easy access.
- Each user has a unique folder containing multiple samples to improve recognition accuracy.

### B. *Face Detection*

During both registration and attendance marking, the system performs face detection using OpenCV's cv2.CascadeClassifier or Dlib's HOG/ResNet face detectors. These detectors identify the presence and location of a human face within the camera frame.

- The system continuously scans video feed frames from the webcam.
- Detected faces are enclosed in bounding boxes to isolate the region of interest.
- Face alignment techniques can be optionally used to ensure consistency.

### C. Feature Extraction and Face Encoding

Once the face is detected, it is passed through a deep neural network (typically from the face_recognition library) that extracts a 128- dimensional face embedding vector. These vectors uniquely represent facial features and are invariant to lighting, pose, and expressions.

- Encoding of all registered faces is stored in a serialized file (e.g., .pkl format) for quick access.
- Cosine similarity or Euclidean distance is used to compare faces during recognition.

### D. Real-Time Face Recognition

When a user stands in front of the system, the camera captures the frame and compares the detected face encodings against the known encodings.

- If a match is found within a threshold, the name of the individual is displayed.
- If no match is found, the system may mark it as "Unknown."
- Attendance is recorded only once per individual per session to prevent duplication.

### E. Attendance Logging

Once a face is recognized, the system logs the entry in a CSV file along with the following details:

- Name or ID of the user
- Date and Time of recognition
- Status (e.g., Present)

The CSV log serves as the attendance sheet and can be used for further processing, reporting, or exporting.

### F. Graphical User Interface (GUI)

The GUI is created using the Tkinter library and provides the following functionalities:

- Buttons to register a new user
- Start attendance capture session
- View or export attendance records
- Real-time video display with face bounding boxes and labels

### G. System Optimization and Error Handling

To ensure robustness, the system includes:

- Frame skipping to reduce computational load
- Retry mechanisms in case of camera initialization failure
- Exception handling for missing modules or files

### H. Multi-User and Multi-Face Recognition

The system supports detection and recognition of multiple faces in a single frame. This feature is especially useful in classrooms or group meetings where multiple attendees may be present simultaneously.

## VI. DEPLOYMENT

Deploying a face-based attendance system involves multiple stages, including environment setup, software installation, testing, and integration into real-world settings. Each step must be carefully executed to ensure smooth and secure system functioning.

### A. Hardware Requirements

- A computer system (Windows/Linux) with a functional webcam.
- At least 4GB RAM and a multi-core processor (Intel i5 or better recommended).
- Optional: External camera or surveillance system for higher accuracy.

*B. Software Requirements*
- Python (version 3.6 or above)
- Required libraries: OpenCV, face_recognition, pandas, numpy, tkinter, pickle, csv
- An IDE such as VS Code, PyCharm, or Jupyter Notebook

*C. Environment Setup*
- Install Python and pip (Python package manager).
- Create the project folder structure:
  - /dataset
  - /encodings
  - /logs
  - main.py (execution script)
  - gui.py (optional GUI interface)

*D. System Configuration*
- Configure the webcam and test with OpenCV to ensure proper detection.
- Modify settings in the code (like camera index or threshold values) for the target environment.
- Ensure CSV file paths and encoding directories are writable and properly linked.

*E. Deployment in Institutions*

The system can be deployed in classrooms, offices, or entrances. For large-scale usage:
- Deploy the application on a central server or multiple local machines.
- Use a LAN/Wi-Fi network to sync CSV logs to a central server.
- Schedule cron jobs or task schedulers for daily logging and reports.

*F. Cloud Integration (Optional)*

For remote accessibility, integrate the system with cloud platforms such as AWS, Azure, or Firebase.
- Store logs in real-time using APIs or database integration (MySQL, Firebase Realtime DB).
- Host a web-based dashboard for admin to view, download, and manage reports.

*G. Testing and Validation*
- Test under different lighting conditions and angles.
- Test with both known and unknown users.
- Ensure accuracy, speed, and low false positives.

*H. Maintenance and Updates*
- Regularly update the face database as new users are added.
- Backup the encoding files and logs periodically.
- Re-train encodings if recognition accuracy drops.

## VII. CHALLENGES

While the Face-Based Attendance System offers a promising solution for attendance automation, it is not without its challenges. These challenges can arise from various factors, including environmental limitations, hardware dependencies, privacy issues, and the accuracy of the recognition system.

*A. Environmental Factors*

Environmental conditions play a crucial role in the effectiveness of the facial recognition system. Key challenges include:

1) Lighting Variability: Facial recognition systems are highly sensitive to lighting conditions. Insufficient lighting or highly variable light sources can degrade the quality of captured images, making it harder for the system to detect and recognize faces accurately. Shadows, glare, and strong backlighting can lead to the failure of facial recognition algorithms.

2) Background Noise: In environments with cluttered backgrounds or multiple people in the frame, the system may mistakenly recognize individuals or fail to recognize them at all. This is especially critical in crowded places such as classrooms or large meetings.
3) Occlusions and Face Coverings: The system's ability to detect faces can be hindered by face coverings, such as glasses, masks, or hats. These occlusions may obscure key facial features required for identification, leading to errors or missed recognitions.

To address these challenges, the system could integrate advanced image preprocessing techniques like dynamic lighting adjustments, noise reduction algorithms, or even infrared-based face detection for low-light conditions.

### B. Hardware Limitations

The effectiveness of the face-based attendance system depends significantly on the hardware used:

1) Camera Quality: A low-resolution webcam might struggle to capture detailed facial features, leading to inaccurate recognition or difficulty in identifying users. High-quality cameras with at least 720p or 1080p resolution are recommended for reliable performance.
2) Processing Power: The real-time facial recognition process is computationally intensive. Inadequate processing power can lead to system lags, delays in recognizing faces, or even failures in detecting faces in a timely manner.
3) Device Compatibility: Different devices (laptops, desktops, mobile devices) may have varying levels of performance, and the system needs to adapt to each one.

### C. Privacy Concerns

Biometric data, especially facial recognition data, raises significant privacy and ethical concerns:

1) Data Security: Since facial data is highly sensitive and unique to each individual, there is a risk that this data could be intercepted, misused, or stolen.
2) Consent and Transparency: Individuals may be uncomfortable with having their facial data stored or used for attendance purposes without their explicit consent.
3) Legal and Ethical Issues: Many countries have stringent regulations around biometric data storage and usage, such as the General Data Protection Regulation (GDPR) in the European Union.

### D. Handling Large Datasets

As the system scales and the number of registered users grows, managing large datasets becomes a challenge:

1) Storage Requirements: A growing number of users means an increase in the number of facial images and encoded feature data that need to be stored.
2) Processing Speed: With a larger dataset, the speed of the facial recognition process may decrease, as the system must compare the newly detected face against a larger set of encoded faces.
3) Data Organization: Efficiently managing, indexing, and retrieving facial data is essential. A poorly organized dataset could lead to slower matching times or errors in identifying individuals.

### E. Accuracy and False Positives

While facial recognition systems are generally reliable, they are not infallible:

1) False Positives (Type I Error): This occurs when the system incorrectly matches an individual with another person.
2) False Negatives (Type II Error): This occurs when the system fails to recognize a legitimate user.
3) Adaptability: The system needs to be adaptable to changes in users' appearances.

## VIII. EXPERIMENTAL RESULTS AND ANALYSIS

The Face-Based Attendance System's performance was evaluated through a series of experiments conducted in different settings to measure the accuracy, speed, and robustness of the system.

### A. Accuracy Evaluation

The system's accuracy was evaluated by comparing its attendance logs with manually maintained records. A dataset of 200 users was used, and the system achieved an accuracy rate of approximately 95%, with a few instances of misidentification due to variations in lighting and facial occlusions. The accuracy improved when the system was tested under controlled lighting conditions with clearer facial images.

### *B. Speed Evaluation*
Speed is crucial for real-time applications. The system was tested for face recognition processing speed using a standard webcam with 720p resolution. On average, the system processed 25 frames per second (FPS), ensuring real-time facial recognition without noticeable delays. However, when the number of registered users increased to more than 500, there was a slight dip in performance, with processing speeds dropping to 18 FPS.

### *C. Robustness*
Robustness was evaluated by introducing variations in the environment, such as low-light conditions, background noise, and occlusions. The system was able to recognize faces in most scenarios, though accuracy decreased slightly under low-light conditions (approx. 80% accuracy in low-light vs. 95% in well-lit conditions). The system was also tested with occlusions, such as masks and glasses, and achieved an accuracy of 85%.

## IX. SECURITY MEASURES AND DATA BREACH RISKS
As a system dealing with sensitive biometric data, the Face-Based Attendance System must address potential security threats to protect users' privacy and ensure data integrity.

### *A. Data Encryption*
Facial recognition data, along with attendance records, must be encrypted both at rest and in transit. Strong encryption algorithms like AES-256 should be used to ensure that unauthorized parties cannot access the data. Additionally, any sensitive data transmitted over the network should be encrypted using secure protocols like HTTPS.

### *B. Secure Communication Channels*
The system should employ secure communication protocols (e.g., HTTPS, TLS) to prevent man-in-the-middle attacks and ensure that facial recognition data and attendance logs cannot be intercepted during transmission. Regular audits of the communication channels can help detect potential vulnerabilities.

### *C. User Authentication and Access Control*
Only authorized personnel should have access to the system's administrative functions and sensitive data. Multi-factor authentication (MFA) can be implemented for system administrators to enhance security. Furthermore, role-based access control (RBAC) should be used to ensure that users and administrators only have access to the data they are authorized to view or modify.

### *D. Backup and Redundancy*
To ensure data integrity and minimize the risk of data loss, regular backups should be performed. These backups should be stored in secure locations (preferably encrypted) and periodically tested to ensure they are functioning correctly. Additionally, implementing a redundant server infrastructure can ensure that the system remains operational in case of hardware failure.

## X. SCALABILITY CONSIDERATIONS
For large-scale deployment, scalability is a critical consideration. The Face-Based Attendance System must be capable of handling increased user load and data processing requirements as it grows.

### *A. Distributed Systems*
As the system scales, it may be necessary to distribute the workload across multiple machines. A distributed architecture allows for the efficient handling of high traffic and ensures that the system can handle numerous simultaneous users without performance degradation.

### *B. Cloud-Based Solutions*
Cloud computing offers scalable resources that can be provisioned dynamically based on the system's load. A cloud-based deployment can support high volumes of data and user activity, allowing for easy scaling and maintenance without requiring significant upfront investment in physical infrastructure.

### *C. Real-Time Analytics and Reporting*

Scalability also includes the ability to provide real-time analytics and reporting. As the number of users grows, administrators may require advanced reporting capabilities, such as automatic generation of attendance reports or data insights.

## XI. FUTURE ENHANCEMENTS

To make the Face-Based Attendance System more comprehensive and user-friendly, several future enhancements can be implemented:

1) Mobile App Integration: Developing a mobile app would allow users to check their attendance remotely and make the system more flexible and accessible.
2) AI-based Emotion Recognition: Adding emotion recognition capabilities could allow the system to detect user engagement or emotional states, potentially useful in educational or workplace settings.
3) Cloud-Based Integration: Incorporating cloud storage and processing would enhance data access, backup, and scalability. Integration with cloud services like Google Cloud or AWS can streamline management and ensure high availability.
4) Multi-modal Biometric Integration: In addition to facial recognition, incorporating other biometrics like voice or fingerprint recognition could provide an added layer of security and increase the system's overall reliability.
5) Improved Deep Learning Models: Future versions could incorporate more advanced deep learning architectures to improve recognition accuracy, especially in challenging conditions.
6) Enhanced User Interface: Further refinement of the user
7) interface could make the system more intuitive and accessible to non-technical users.
8) Advanced Reporting and Analytics: Implementing advanced reporting and analytics capabilities would allow administrators to gain insights from attendance data, such as identifying patterns of absenteeism or gauging engagement levels.

## XII. USER FEEDBACK

User feedback is essential for continuous improvement and fine- tuning of the system. Initial testing has shown positive responses from users, who appreciated the ease of use and efficiency of the system. The automated nature of the attendance tracking process was highly regarded, as it eliminated the need for manual entry and reduced the chances of human error.

However, users did highlight certain areas for improvement, including:

1) Camera Quality and Lighting Sensitivity: Some users noted that the system's accuracy decreased under poor lighting conditions. In such cases, face recognition became less reliable, and some users were not recognized correctly.
2) User Interface Enhancements: While the Tkinter-based graphical interface was functional, users suggested adding more visual cues and making it more intuitive. For example, adding more detailed feedback about why a user may not have been recognized could help users resolve issues faster.
3) Backup and Data Access: Some users expressed the need for better access to the attendance logs and the ability to export data in various formats (e.g., Excel, PDF). Having the ability to query attendance records in a user-friendly format was considered important for reporting and audits.

## XIII. CONCLUSION

The Face-Based Attendance System using AI and OpenCV provides a robust, contactless, and automated solution for attendance tracking. By leveraging the power of facial recognition algorithms and Python's rich ecosystem of libraries, the system offers a high level of accuracy, efficiency, and user-friendliness. Unlike traditional attendance methods that rely on manual processes or physical IDs, this system streamlines the attendance process, reducing human error and minimizing the risk of fraud.

The system's integration of technologies such as OpenCV for image processing, Pandas for data management, Tkinter for the graphical interface, and the CSV module for logging ensures that it meets both functional and performance requirements. The ability to operate in real-time, handle multiple users simultaneously, and offer robust security measures for data protection makes this system an ideal choice for modern attendance solutions. However, despite its promising capabilities, several areas require enhancement to further improve the system's effectiveness. In particular, addressing challenges related to environmental factors, privacy concerns, and hardware limitations will help refine the system's reliability and applicability in various real-world scenarios.

Future work in this field could focus on improving accuracy, integrating multi-modal biometric authentication, implementing cloud-based deployment, developing mobile applications, enhancing security measures, and addressing user feedback. By continuing to refine and expand the capabilities of facial recognition attendance systems, we can create more efficient, reliable, and user-friendly solutions for attendance tracking across various domains.

## XIV. ACKNOWLEDGMENTS

The successful development of this Face-Based Attendance System would not have been possible without the support and encouragement of many individuals and organizations. We would like to express our heartfelt gratitude to our supervisors for their invaluable guidance and expertise throughout the project, the institution for providing the necessary resources and infrastructure to develop and test the system, and the research team and developers who contributed to the coding, testing, and optimization phases of the project. Additionally, we extend our thanks to the various open-source communities, including the developers of OpenCV, Dlib, Pandas, and Tkinter, whose contributions made this project feasible.

## REFERENCES

[1] W. Zhao, R. Chellappa, and P. J. Phillips, "Face Recognition: A Literature Survey," ACM Computing Surveys (CSUR), vol. 35, no. 4, pp. 399–458, 2003.
[2] A. K. Jain, A. Ross, and K. Nandakumar, Introduction to Biometrics. Springer Science & Business Media, 2011.
[3] S. Liao, Z. Yu, and K. Chen, "Face Recognition with OpenCV," in Proceedings of the International Conference on Neural Networks and Signal Processing, 2014, pp. 209-214.
[4] "Dlib: A toolkit for machine learning." [Online]. Available: https://dlib.net/ [Accessed: 14-Apr-2025]. Hornegger, and D. Comaniciu, "Multi-scale deep reinforcement learning for real-time 3D-landmark detection in CT scans," IEEE transactions on pattern analysis and machine intelligence, vol. 41, no. 1, pp. 176-189, 2017.
[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770-778
[6] "The Face Recognition Library." [Online]. Available: https://github.com/ageitgey/face-recognition [Accessed: 14-Apr- 2025].
[7] "OpenCV Documentation." [Online]. Available: https://docs.opencv.org/ [Accessed: 14-Apr-2025].
[8] P. Gupta and M. Sharma, "Multi-Modal Biometric Systems: A Review," Journal of Biometric Systems, vol. 12, no. 3, pp. 105-112, 2020.
[9] J. P. Bigham and D. Doering, "Web-based Attendance System Using Facial Recognition," International Journal of Computer Science and Information Technology, vol. 2, no. 4, pp. 45–53, 2005.
[10] ISO/IEC 27001:2013 – Information Security Management. International Organization for Standardization (ISO).
[11] General Data Protection Regulation (GDPR). (2018). Regulation (EU) 2016/679 of the European Parliament and of the Council.
[12] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 1701-1708.
[13] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 815-823.
[14] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep Face Recognition," in British Machine Vision Conference, 2015, pp. 41.1-41.12.
[15] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, Handbook of Fingerprint Recognition. Springer, 2003.
[16] J. Daugman, "How iris recognition works," IEEE Transactions on circuits and systems for video technology, vol. 14, no. 1, pp. 21- 30, 2004.
[17] F. C. Ghesu, B. Georgescu, Y. Zheng, S. Grbic, A. Maier, J.