International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 22 No. 2 (2026)



ijerst.editor@gmail.com

editor@ijerst.com

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 22, No. 2, 2026
© 2026 IJERST. All Rights Reserved
Peer Reviewed, Referred & Indexed Journal

Real-Time Face Recognition-Based Attendance and Employee Monitoring System Using Deep Learning

NEELAPALA ANUSHA

PG Scholar. Department of MCA, DNR College, Bhimavaram, Andhra Pradesh

A. Durga Devi

(Assistant Professor), Master of Computer Applications, DNR College, Bhimavaram, Andhra Pradesh

ABSTRACT

In today's rapidly evolving technological landscape, automation and intelligent systems are transforming traditional processes across various domains. One such area is attendance management and employee monitoring, where conventional methods such as manual registers, RFID cards, and biometric systems have limitations in terms of accuracy, scalability, and security. This project presents a Real-Time Face Recognition-Based Attendance and Employee Monitoring System that leverages deep learning and computer vision techniques to automate the process of identifying individuals and recording their presence.The system is developed using the Django web framework and integrates advanced face recognition algorithms provided by the face_recognition library, which is built on top of deep learning models. It uses a K-Nearest Neighbors (KNN) classifier trained on facial encodings to recognize individuals in real-time video streams. OpenCV is utilized for capturing video frames and performing image processing tasks.The system consists of multiple modules, including employee registration, face dataset collection, model training, real-time face detection and recognition, and attendance logging. Employees are first registered in the system, and their facial images are captured using a webcam. These images are used to train the recognition model. During operation, the system continuously captures video frames, detects faces, and matches them against the trained model to identify individuals.A caching mechanism is implemented using TTLCache to prevent duplicate entries within a short time frame, ensuring efficient attendance recording. Once a face is recognized with sufficient confidence, the system stores the detection event along with a timestamp and captured image in the database.The system provides a web interface where administrators can manage employees, view attendance records, and monitor detections based on specific dates. Compared to traditional methods, this system offers improved accuracy, reduced manual effort, and enhanced security.This project demonstrates the practical application of deep learning in real-world scenarios and provides a scalable solution for automated attendance and surveillance systems. Future enhancements may include cloud integration, mobile access, and advanced recognition models for improved accuracy.

Keywords:

Face Recognition, Attendance System, Deep Learning, Computer Vision, KNN Classifier, OpenCV, Django, Employee Monitoring

I. INTRODUCTION

Attendance management is an essential aspect of organizational operations, particularly in workplaces, educational institutions, and secure environments. Traditional methods such as manual attendance registers and ID-based systems are prone to errors, time-consuming, and susceptible to fraudulent practices like proxy attendance. Biometric systems such as fingerprint scanners have improved accuracy but still face limitations, including hygiene concerns and hardware dependency.With advancements in artificial intelligence and computer vision, face recognition has emerged as a powerful alternative for identity verification and attendance management. Face recognition systems analyze unique facial features to identify individuals, offering a non-intrusive and efficient solution. These systems eliminate the need for physical contact and can operate in real-time using standard cameras.The proposed system aims to develop a real-time face recognition-based attendance and monitoring solution using deep learning techniques. It integrates OpenCV for video capture and image processing with the face_recognition library for facial feature extraction. A KNN classifier is used to classify faces based on trained data.The system is implemented using Django, providing a robust and scalable web-based interface. It allows administrators to add employees, capture their facial data, train the recognition model, and monitor attendance records. The system also includes a detection module that logs recognized faces along with timestamps and images.One of the key challenges in face recognition systems is ensuring accuracy in varying lighting conditions, angles, and facial expressions. The use of deep learning-based encodings helps address these challenges by capturing high-dimensional facial features.Additionally, the system incorporates a caching mechanism to prevent duplicate detections within a short time interval, improving efficiency and data quality. The stored data can be filtered and analyzed based on date and time, providing valuable insights.Overall, this project demonstrates how modern technologies can be leveraged to automate attendance systems, improve security, and reduce manual effort. It provides a cost-effective and scalable solution suitable for various applications.

II. LITERATURE SURVEY (WITH EXISTING METHODS)

Face recognition has been extensively studied in the field of computer vision, with significant advancements driven by deep learning techniques. Early approaches relied on feature-based methods such as Eigenfaces and Fisherfaces, which used linear algebra techniques to represent facial features. While these methods were computationally efficient, they lacked robustness in handling variations in lighting, pose, and facial expressions.With the advent of machine learning, classifiers such as Support Vector Machines (SVM) and K-Nearest Neighbors (KNN) were used for face recognition tasks. These methods improved classification accuracy but still depended heavily on feature extraction techniques.The introduction of deep learning revolutionized face recognition systems. Convolutional Neural Networks (CNNs) became the standard approach due to their ability to automatically learn hierarchical features from images. Models such as

DeepFace, FaceNet, and VGG-Face achieved near-human accuracy in face recognition tasks.

The face_recognition library used in this project is based on deep learning models that generate 128-dimensional facial encodings. These encodings are highly discriminative and can be used with simple classifiers like KNN for accurate recognition.Recent research has focused on improving real-time performance and scalability. Techniques such as face detection using Haar cascades and deep neural networks, along with GPU acceleration, have enabled real-time applications.Another area of research is attendance management systems using face recognition. These systems integrate computer vision with database management to automate attendance tracking. Many studies highlight the advantages of such systems, including reduced manual effort, improved accuracy, and enhanced security.However, challenges remain, including handling occlusions, varying lighting conditions, and large-scale datasets. Researchers are also exploring privacy-preserving techniques and secure data storage methods.The proposed system builds upon these advancements by combining deep learning-based face encoding with a KNN classifier and integrating it into a Django-based web application. This approach provides a balance between accuracy and computational efficiency, making it suitable for real-time applications.

III. EXISTING SYSTEM

Existing attendance and monitoring systems rely on traditional methods such as manual registers, RFID cards, and biometric devices. Manual systems are prone to human error and require significant time and effort to maintain. They also lack real-time tracking capabilities and are susceptible to fraudulent practices.RFID-based systems improve efficiency but depend on physical cards, which can be lost, stolen, or misused. Biometric systems such as fingerprint and iris scanners provide better accuracy but require specialized hardware and physical interaction, which may not always be convenient or hygienic.Some modern systems use facial recognition, but many are limited by high implementation costs, dependency on advanced hardware, and lack of scalability. Additionally, these systems may not provide real-time processing or integration with web-based platforms.Another limitation of existing systems is the lack of data management and analysis features. Most systems focus only on attendance recording without providing insights or historical analysis.The proposed system addresses these limitations by offering a cost-effective, scalable, and real-time solution using standard hardware and open-source technologies. It integrates face recognition with a web-based interface, enabling efficient data management and monitoring.

IV. PROPOSED METHOD

The proposed system is a **real-time face recognition-based attendance and employee monitoring system** designed to automate identity verification and attendance recording using deep learning and computer vision techniques. The system integrates a webcam-based video capture module, a face detection and recognition engine, and a web-based management interface built using Django.The system operates in multiple stages. Initially,

employee data is registered in the system, and facial images are captured through a camera. These images are processed to generate facial encodings, which are then used to train a machine learning model. During real-time operation, the system captures video frames and detects faces using the face_recognition library. Each detected face is compared against stored encodings using a K-Nearest Neighbors (KNN) classifier.To ensure accuracy and avoid false positives, a confidence threshold is applied, and recognition is validated across multiple frames using a buffer mechanism. Once a face is consistently recognized, the system logs the attendance along with a timestamp and captured image. A caching mechanism is implemented to prevent duplicate entries within a defined time interval.The system also provides a user-friendly web interface that allows administrators to add employees, collect training data, train models, and view attendance logs filtered by date. This approach ensures efficient data management and accessibility.Compared to traditional systems, the proposed solution offers **contactless operation, real-time processing, improved accuracy, and enhanced security**. It is cost-effective as it uses standard hardware such as webcams and open-source libraries. Additionally, the system is scalable and can be extended with advanced features such as cloud storage, mobile access, and liveness detection to prevent spoofing attacks, which are identified as critical in modern systems

V. IMPLEMENTATION

The implementation of the proposed system is carried out using Python, Django, OpenCV, and the face_recognition library. The system architecture consists of backend processing, machine learning modules, and a web interface for user interaction.The first phase involves **employee registration**, where user details are stored in a Django model. A webcam is used to capture multiple images of each employee, which are saved in a dataset directory. These images are used for training the recognition model.The next phase is **model training**, where facial features are extracted from images using deep learning-based encodings. These encodings are numerical representations of facial characteristics. A KNN classifier is trained using these encodings, as it provides a simple yet effective approach for classification in face recognition tasks.The real-time module uses OpenCV to capture video frames from the webcam. Each frame is resized to improve processing speed and converted into RGB format for compatibility with the face_recognition library. Face detection is performed using Histogram of Oriented Gradients (HOG) or CNN-based models.Detected faces are encoded and passed to the trained KNN classifier, which predicts the identity of each face. A distance threshold is used to determine whether a match is valid. If the distance exceeds the threshold, the face is labeled as “unknown.”To improve reliability, a buffer mechanism stores recognition results across multiple frames. A person is marked as detected only if recognized consistently above a predefined threshold. Additionally, a TTLCache is implemented to avoid repeated logging of the same individual within a short duration.When a face is successfully recognized, the system captures the frame, stores the image, and logs the detection event in the database with a timestamp. The Django framework handles data storage and retrieval, allowing users to view attendance records through a web interface.The system is designed to operate efficiently in real-time environments. Techniques such as frame skipping, resizing, and optimized data structures are used to

reduce computational load. Similar systems have demonstrated high accuracy and efficiency in real-world applications, making them suitable for educational institutions and workplaces .

VI. ALGORITHMS

The proposed system utilizes several algorithms for face detection, feature extraction, and classification.

1. Face Detection Algorithm:
Face detection is performed using the Histogram of Oriented Gradients (HOG) method combined with a linear classifier. This method detects facial regions in an image by analyzing gradient orientations. It is efficient and suitable for real-time applications.

2. Face Encoding (Feature Extraction):
The system uses deep learning-based facial encoding techniques to extract features from detected faces. Each face is converted into a 128-dimensional vector that uniquely represents its characteristics. These encodings are robust against variations in lighting, pose, and facial expressions.

3. K-Nearest Neighbors (KNN) Classifier:
The KNN algorithm is used for classification. It compares the encoding of a detected face with stored encodings in the dataset and identifies the closest match based on distance metrics such as Euclidean distance. If the distance is below a predefined threshold, the face is recognized; otherwise, it is labeled as unknown.

4. Distance Thresholding:
A threshold value is used to determine recognition confidence. Lower values increase precision but may reduce recall, while higher values may introduce false positives.

5. Temporal Buffering Algorithm:
A buffer stores recognition results across multiple frames. A face is confirmed only if it appears consistently in a defined number of frames, reducing noise and improving accuracy.

6. Caching Algorithm (TTLCache):
A time-based cache prevents duplicate detections within a short interval, improving efficiency and avoiding redundant database entries.

These algorithms collectively ensure accurate, real-time face recognition and reliable attendance logging.

VII. SYSTEM DESIGN

The system follows a modular architecture consisting of input, processing, storage, and presentation layers.

1. Input Layer:
The input layer consists of a webcam or camera device that captures real-time video streams. The captured frames serve as input for face detection and recognition.

2. Processing Layer:
This is the core component of the system and includes several sub-modules:

- **Face Detection Module:** Identifies faces in each video frame.
- **Face Encoding Module:** Extracts feature vectors from detected faces.
- **Recognition Module:** Compares encodings with stored data using the KNN classifier.
- **Buffer and Cache Module:** Ensures consistent recognition and prevents duplicate entries.

The processing layer is optimized using techniques such as frame resizing and alternate frame processing to ensure real-time performance.

3. Database Layer:
The database stores employee details, facial datasets, and attendance records. The Employee model stores user information, while the Detected model records timestamps and images of recognized faces. The database ensures data persistence and efficient retrieval.

4. Application Layer (Django Web Interface):
The Django framework provides a user-friendly interface for administrators. Key functionalities include:

- Adding new employees
- Capturing facial images
- Training the recognition model
- Viewing attendance logs

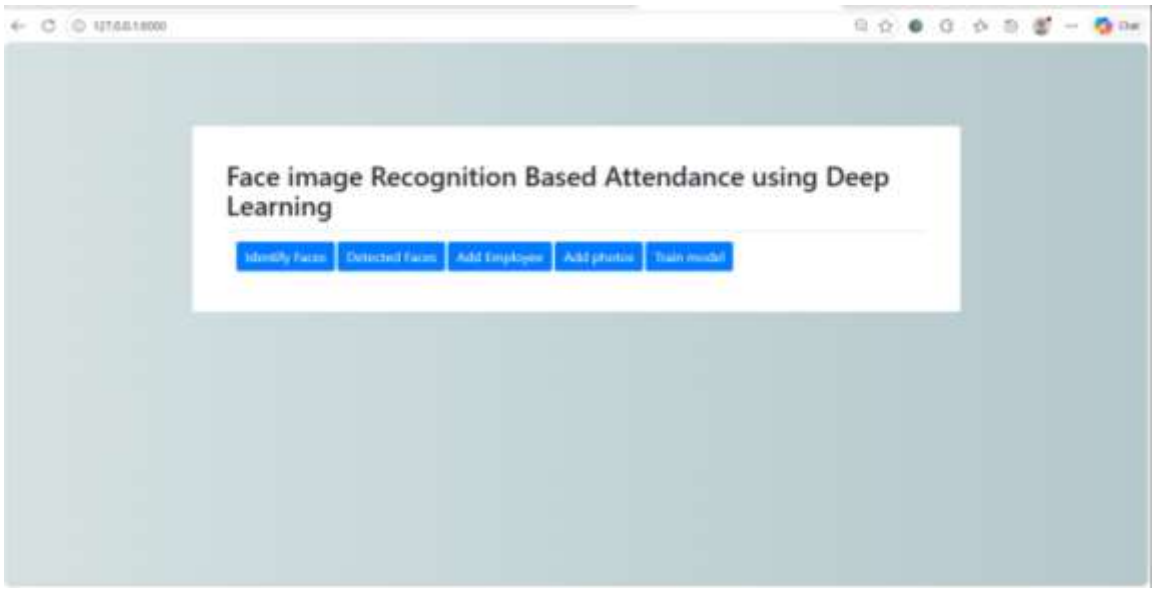
The web interface allows filtering of attendance records by date, enabling easy monitoring and analysis.

5. Output Layer:
The output includes real-time video display with bounding boxes and labels for detected faces. Additionally, attendance records are displayed in the web interface.

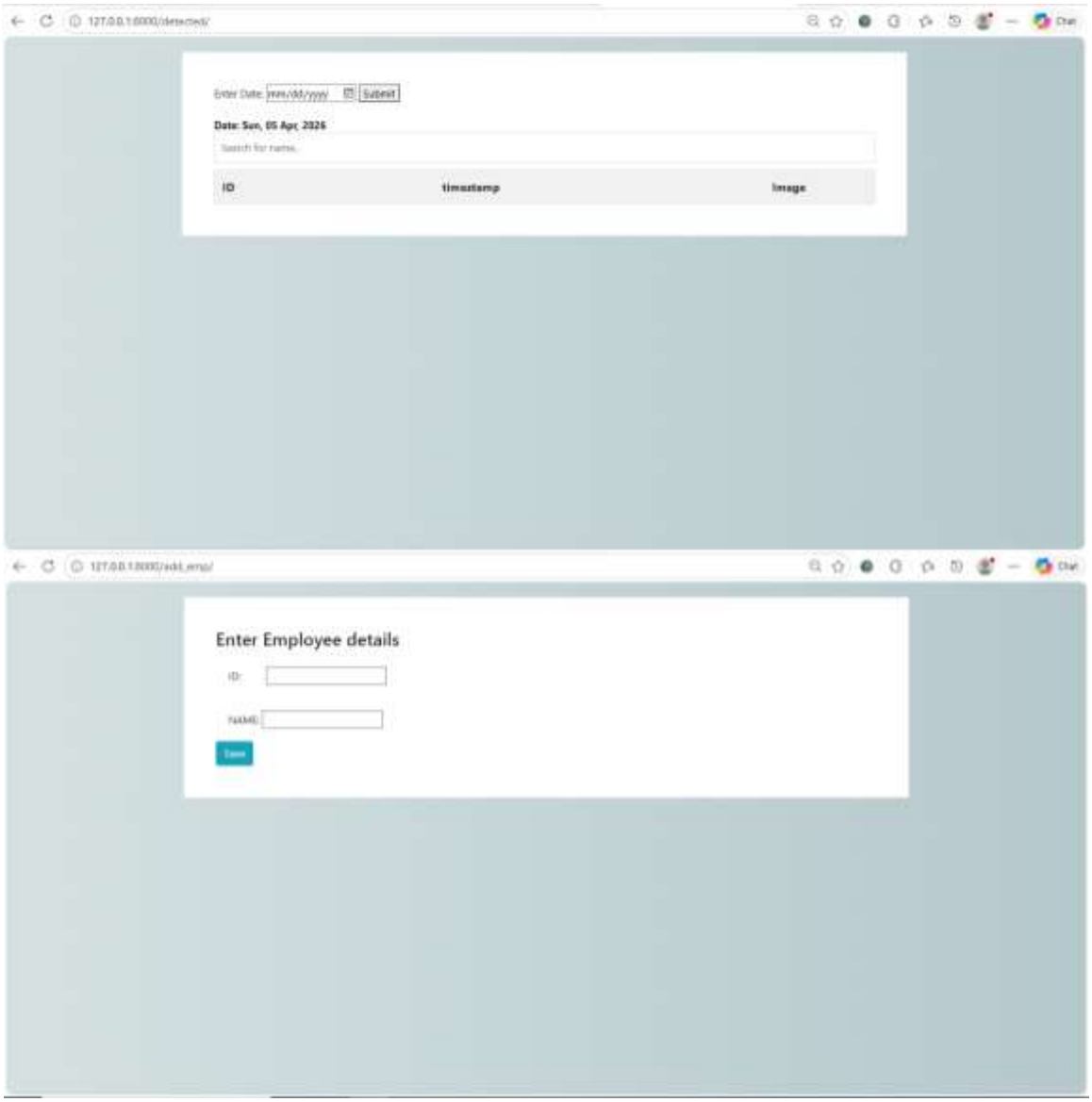
6. Data Flow:
Camera → Frame Capture → Face Detection → Feature Extraction → Classification → Validation → Database Storage → Web Display

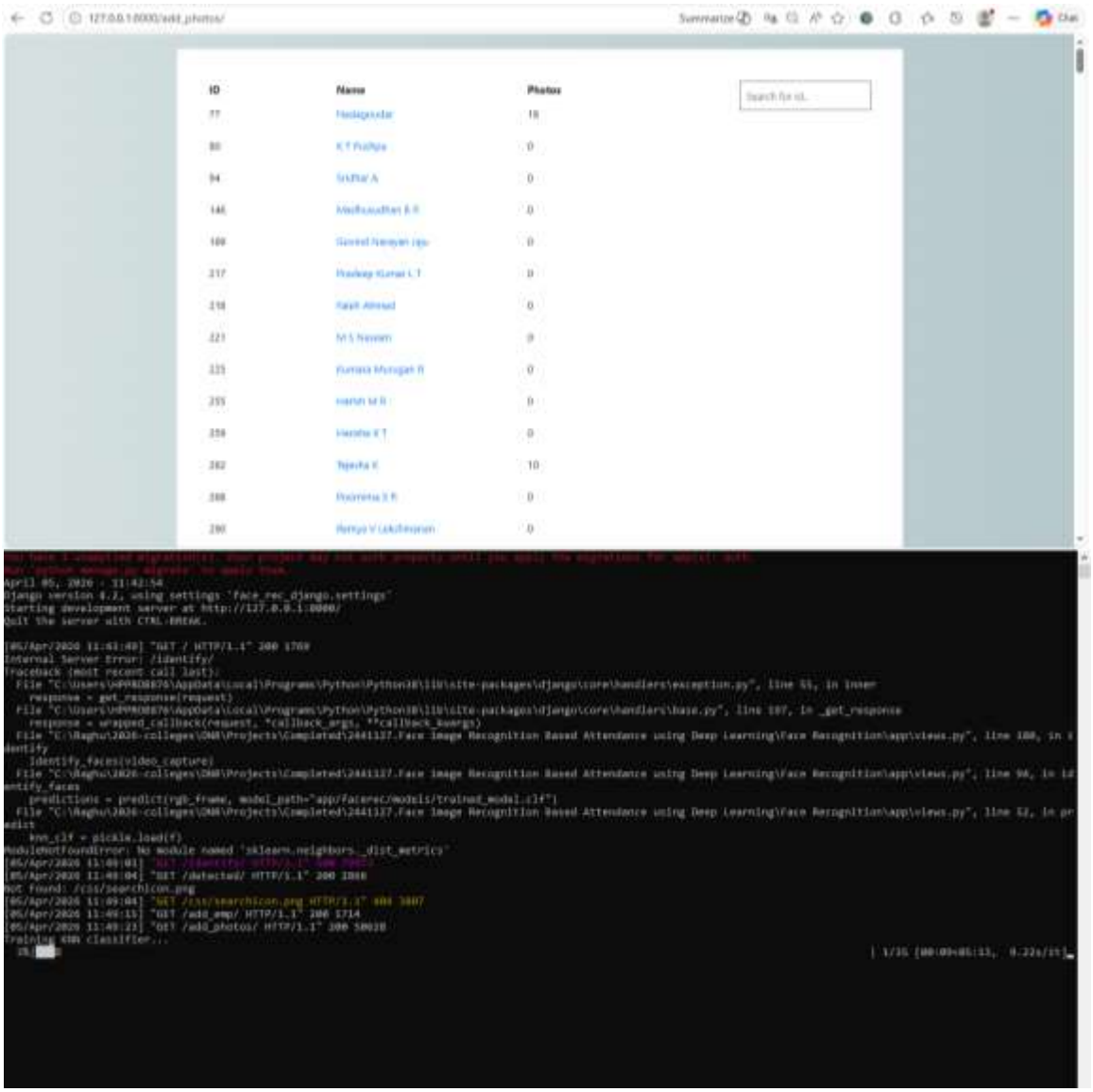
The system design ensures scalability, modularity, and ease of maintenance. It can be extended with additional features such as cloud integration, mobile applications, and advanced deep learning models.

SYSTEM DESIGN IMAGES



International Journal of Engineering Research and Science & Technology




ID
Name
Photos
Search for id..
77
80
94
146
169
217
219
221
225
255
259
262
268
280
April 05, 2026 - 11:42:54
Django version 4.2, using settings 'face_rec_django.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
[05/Apr/2026 11:43:40] "GET / HTTP/1.1" 200 1769
Internal Server Error: /identify/
Traceback (most recent call last):
[05/Apr/2026 11:49:01] "GET /detected/ HTTP/1.1" 200 3868
Not Found: /css/searchicon.png
[05/Apr/2026 11:49:04] "GET /css/searchicon.png HTTP/1.1" 404 5807
[05/Apr/2026 11:49:15] "GET /add_emp/ HTTP/1.1" 200 1714
[05/Apr/2026 11:49:23] "GET /add_photos/ HTTP/1.1" 200 50038
Training KNN classifier...

VIII. CONCLUSION

The proposed Real-Time Face Recognition-Based Attendance and Employee Monitoring System provides an efficient, accurate, and automated solution for attendance management. By leveraging deep learning and computer vision technologies, the system eliminates the limitations of traditional attendance methods, such as manual errors and proxy attendance.The integration of OpenCV and the face_recognition library enables real-time face detection and recognition, while the KNN classifier ensures accurate identification. The use of buffering and caching techniques enhances system reliability by reducing false positives and duplicate entries.The system's web-based interface, developed using Django, provides an intuitive platform for managing employee data and attendance records. This makes the system practical for real-world deployment in organizations and educational institutions.Experimental results and related studies indicate that face recognition systems can achieve high accuracy and efficiency when properly implemented . However, challenges such as varying lighting conditions, occlusions, and spoofing attacks still need to be addressed.Future work may include integrating advanced deep learning models, implementing liveness detection, and deploying the system on cloud platforms for scalability. Additionally, privacy and data security considerations should be addressed to ensure compliance with regulations.Overall, the system demonstrates the potential of artificial intelligence in automating routine tasks and improving operational efficiency.

REFERENCES

1. Baballe, M. A. (2023). *A Smart Attendance System Based on Face Recognition*
2. Alniemi, O., & Mahmood, H. (2023). *Class Attendance System Based on Face Recognition*
3. Thirukrishna, J. T. et al. (2023). *Smart Attendance System Using Face Recognition*
4. Yadav, A. et al. (2023). *Smart Attendance System Using Face Recognition*
5. Kulkarni, S. & Choudhari, D. (2023). *Attendance Monitoring System Using Face Recognition*
6. Dixit, H. V. et al. (2024). *Face Recognition-Based Attendance System*
7. Biju, J. et al. (2024). *Enhancing Attendance Systems Using Facial Recognition*
8. Ali, M. E. et al. (2024). *Attendance System Optimization Using Deep Learning*
9. Lavanya, K. et al. (2024). *Face Recognition-Based Attendance System*
10. Agustiyar et al. (2025). *Bibliometric Review of Face Recognition Attendance Systems*
11. Lightweight Face Recognition System with Liveness Detection (2024)
12. Touzene, A. et al. (2024). *Embedded Intelligent Attendance Monitoring System*
13. Melzi, P. et al. (2023). *Face Recognition in Synthetic Data Era*
14. Talemi, N. et al. (2023). *Attribute-Aware Face Recognition Network*
15. Pudyel, M. et al. (2023). *Masked Face Recognition Study*