International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 13, Issue 3, 2025

Face- Recognition Based Attendance System Using Machine Learning

G Jyothi, Amala Kura, L Bhavya Sri, Pranathi Poloju

[1]Associate Professor, Department of IT Bhoj Reddy Engineering College for Women, India.

[2,3,4]B. Tech Students, Department Of IT, Bhoj Reddy Engineering College For Women, India.

pranathipoloju27@gmail.com

ABSTRACT

In The rapid advancement of digital technology and the growing need for automation have led to the increased adoption of biometric systems for tasks such as identity verification and attendance tracking. Among these, facial recognition has emerged as a non-intrusive and efficient method. This paper presents a facial recognition-based automatic attendance system that eliminates the need for manual processes. The system uses Haar Cascade Classifier for face detection and a trained classifier model for recognizing individuals from real-time video input. It integrates image acquisition, feature extraction, face recognition, and automated attendance logging into a unified framework. A performance evaluation of the system is conducted using metrics such as accuracy, processing efficiency, and robustness under varying conditions. The proposed system demonstrates high accuracy and reliability, offering a practical solution to automate attendance systems while reducing manual intervention and improving data management.

Key Words*: Facial Recognition, Biometric Attendance System, Haar Cascade Classifier, Computer Vision, Machine Learning, Real-Time Identification, Automated Attendance, Image Processing, Feature Extraction, Attendance Management System, Identity Verification, Contactless Technology, AI-based Attendance, Face Detection, Efficiency and Accuracy.*

1-INTRODUCTION

A well-known saying in biometrics is, "Your face is your identity," emphasizing the uniqueness of facial features for identification. With advances in AI and image processing, facial recognition has become widely used across sectors. Automated, contactless attendance systems are increasingly in demand, especially in schools and workplaces, as traditional methods are often slow, error-prone, and easy to manipulate.Humans recognize faces naturally, but replicating this in machines is challenging due to factors like lighting, expressions, occlusions, and head poses. To overcome these, biometric attendance systems using facial recognition have emerged as reliable and efficient alternatives. These systems use computer vision and machine learning to detect and recognize faces in real time, offering a smooth, user-friendly experience.The Haar Cascade Classifier is a popular technique for fast and effective face detection. After detecting a face, a trained model compares it to stored data to identify individuals. The process involves image acquisition, face detection, feature extraction, classification, and attendance logging.This project aims to design and implement a real-time, automatic attendance system using facial recognition. By integrating these technologies, it seeks to accurately identify individuals and mark attendance with minimal human involvement, improving efficiency, accuracy, and record- keeping while reducing administrative work. The system highlights the potential of facial recognition as a practical, scalable solution for modern attendance management

Existing System:

Traditional attendance systems primarily rely on manual entry or card-based identification methods, which are time-consuming, error-prone, and vulnerable to manipulation. Some systems use biometric methods like fingerprint or iris scanning; however, these require physical contact and can be unhygienic or inconvenient. Recently, facial recognition has emerged as a contactless biometric solution, but many existing systems lack real-time capability, accuracy under varied conditions, and integration with automated attendance logging. These limitations highlight the need for a more efficient and scalable solution.

Proposed System:

A real-time, contactless attendance system is developed using facial recognition technology. utilizes face detection, feature extraction, and classification to identify individuals accurately. The system automates attendance logging, ensuring higher efficiency and reduced manual effort.

2-RELATED WORK

A comprehensive study by Satpathy and Mishra introduces a smart attendance management system designed to automate the conventional roll-call procedure using facial recognition technology [4]. This system leverages the OpenCV library [3] for real-time face detection and recognition, employing Local Binary Pattern Histograms (LBPH) as the core algorithm [2]. The authors demonstrate how LBPH enhances accuracy, even in environments with inconsistent lighting and varied facial orientations. The system captures facial data through webcams and stores attendance records in a centralized MySQL database [7], making the data easily retrievable and reportable. The system not only minimizes manual effort but also eliminates common issues such as proxy attendance. Its ability to operate efficiently in both classroom and workplace environments, combined with real-time processing and hygienic contactless operation, positions it as a practical and scalable solution for modern attendance tracking...

Literature on Biometric-Based Educational Systems:

A TiwariIn their review of biometric authentication methods in education, Tiwari and Verma analyze the implementation and effectiveness of various biometric technologies including fingerprint, iris, and facial recognition for automating attendance processes [5]. They assess each modality based on usability, accuracy, deployment cost, and user acceptability. The study concludes that facial recognition stands out as the most scalable and non-intrusive solution, particularly suitable for educational environments. Despite challenges such as environmental factors, variations in facial expressions, and privacy concerns, the paper emphasizes the potential for biometric attendance systems to integrate seamlessly with learning management systems and student databases. The authors also highlight that with advancing cloud infrastructure, such systems could be extended to support smart campuses, thereby enabling real-time analytics, centralized control, and enhanced administrative efficiency.

Literature on AI-Powered Recognition Systems:

Ahonen et al. present a comprehensive comparative evaluation of both classical and modern facial recognition techniques, placing specific emphasis on algorithms such as Eigenfaces, Fisherfaces, Local Binary Pattern Histograms (LBPH), and advanced deep learning architectures like Convolutional Neural Networks (CNNs) [2]. The study methodically assesses these approaches based on multiple performance parameters, including recognition accuracy, robustness under varying environmental conditions, training time, and computational requirements. It highlights that while deep learning models—especially CNNs and Siamese networks—achieve superior accuracy in complex recognition tasks, their effectiveness comes at the cost of requiring large annotated datasets, powerful GPU-based infrastructure, and longer training durations. These prerequisites make them less feasible for deployment in educational settings with limited technical resources and budget constraints.

In contrast, LBPH emerges as a practical and efficient solution for real-time face recognition applications. Its architecture is relatively lightweight, relying on local texture descriptors that are resistant to changes in illumination and facial expression. The study finds that LBPH offers a favorable trade-off between computational efficiency and recognition reliability, making it well-suited for low-cost, desktop-based attendance systems. When integrated with the OpenCV framework [3], LBPH can be easily implemented using standard webcams without the need for additional sensors or complex preprocessing pipelines.

Moreover, the paper stresses the importance of maintaining consistency in preprocessing steps such as face alignment, grayscale conversion, and histogram equalization to enhance overall system performance. It also addresses potential challenges like dataset imbalance and overfitting, particularly in academic institutions where facial datasets may be limited in diversity and size. Despite these challenges, the authors argue that with proper design and optimization, LBPH-based systems can deliver consistent and real-time recognition performance, reducing dependency on manual attendance processes and enhancing operational transparency. Overall, the study underscores the relevance of LBPH as a reliable backbone for intelligent attendance systems, especially in resource-constrained environments such as schools, colleges, and training institutes.

3. REQUIREMENT ANALYSIS

Functional Requirements:

- **Image Capture**: Capture facial images using a webcam to initiate the recognition process.
- **Feature Extraction**: Detect and isolate key facial features required for unique identification.
- **Data Encoding**: Convert extracted facial features into numerical vectors for comparison.
- **Face Matching**: Match encoded vectors with the stored face data in the database.
- **Identification**: Identify the individual based on successful face matching results.
- **Attendance Logging**: Record attendance details such as name, time, and date into the system.
- **Real-Time Processing**: Ensure immediate recognition and attendance marking without delay.
- **Accuracy**: Maintain high accuracy across different lighting conditions and facial variations.
- **Contactless Operation**: Eliminate the need for physical interaction during the attendance process.
- **Automation**: Minimize manual intervention through a fully automated recognition system.

Non-Functional Requirements:

Non-functional requirements define the overall quality attributes and performance expectations of the Facial Recognition Based Automatic Attendance System. While functional requirements describe *what* the system does, non-functional requirements address *how well* it performs those tasks. These include scalability to handle a large number of students, performance for real-time recognition, system reliability, data security, and ease of use for both administrators and students. Ensuring these quality standards makes the system robust, efficient, and adaptable for future upgrades.

Scalability:

- Handles large numbers of student records and facial images without affecting performance.

- Supports scaling as more modules (e.g., chatbot, learning resources) are added.
- Ensures system efficiency even when attendance is recorded for multiple users simultaneously.

Performance:

- Face recognition and attendance logging complete with minimal delay.
- Recognition and data retrieval respond within acceptable time (e.g., under 2 seconds per student).
- Maintains system responsiveness even during peak usage such as class start time.
- **Software Requirements:**

Frontend (GUI): Python Tkinter

Backend &Image Processing: Python3.10+, penCV (cv2), NumPy

Face Recognition Algorithm: OpenCV LBPH (Local Binary Patterns Histogram)

Database & Storage: MySQL 8.0 via MySQLWorkbench, MySQLdbconnector.

Data Storage: CSV files for attendance, local folders for face datasets

AI Chatbot Module: Python (rule-based chatbot using if-else and keyword.

Visualization & Exporting: Python CSV module, Pandas for tabular viewing

Development Environment: VS Code / PyCharm

Others: PIL (for image rendering), os, datetime, tkinter

Hardware Requirements: Processor: Intel Core i5 or higher. RAM: 8GB.

Storage: At least 500GB HDD or SSD

4. DESIGN

System Architecture:

The system architecture of the Facial Recognition-Based Automatic Attendance System provides a modular, multi-layered view of how facial data is processed, recognized, and logged in real-time for educational or organizational use. It also integrates additional learning modules such as a chatbot and resource delivery system. This layered structure ensures scalability, maintainability, and smooth functionality across different tasks like attendance marking, student interaction, and data analysis.

At the top of the architecture is the Admin/Teacher user, who interacts directly with the system through the Application Layer, which is built using Python's Tkinter-based GUI. The GUI provides an intuitive interface with menu options including "Train," "Mark Attendance," "View Data," "Learning Resources," and "Chatbot." Each button corresponds to a specific operation that triggers backend logic.

The next tier is the Processing Layer, which handles the core logic.

- The Face Detection and Training module utilizes OpenCV's Haar Cascade classifier for detecting faces and the LBPH algorithm for feature extraction and training.
- The Attendance Logging module leverages Pandas and datetime modules to record timestamped entries of recognized users into CSV files and databases.
- The View Data NLP module parses, formats, and presents stored attendance or chatbot data using simple rule-based logic and basic Natural Language Processing.
- The Chatbot + Note Generator supports student queries and generates brief notes or answers using pre-configured PDF/Text responses.

All modules interact with the Data Layer, which manages datasets (face images in dataset/ folder), attendance logs (present CSVs), and chatbot

International Journal of Information Technology & Computer Engineering
ISSN 2347-3657
Volume 13, Issue 3, 2025

responses (notes/ folder). These components ensure persistent storage and retrieval without data loss.

Below this lies the Storage Layer, which comprises a structured MySQL database that stores registered user information like names, roll numbers, timestamps, and attendance status. This relational layer ensures reliability, easy querying, and structured analytics

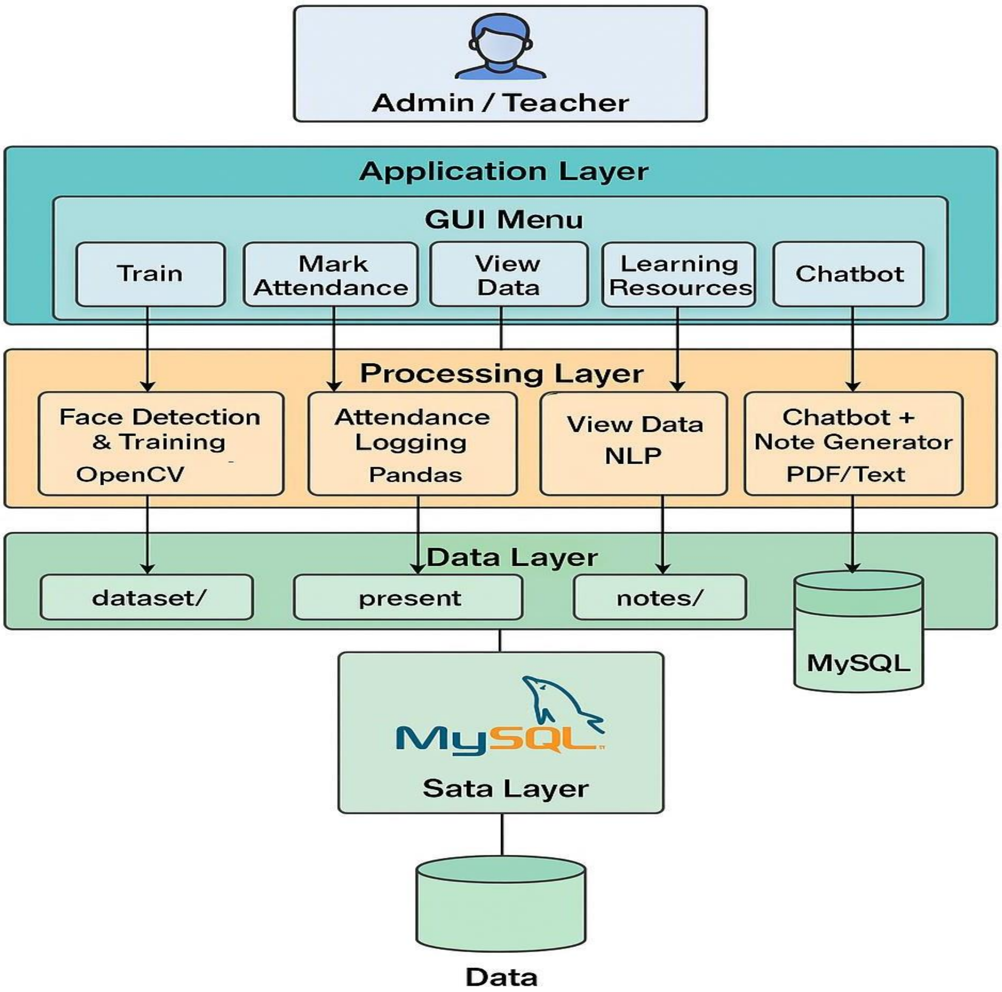


Fig. 4.1.1 System Architecture

Technical Architecture:

Technical Architecture refers to the structured design of technological components that form the core foundation of the Facial Recognition-Based Attendance and Learning Support System. It focuses on real-time processing, contactless operation, modular development, and seamless integration of both learning and administrative utilities. The architecture is built to offer high accuracy in facial recognition, easy extensibility for additional tools like chatbots and learning dashboards, and smooth data flow across various system layers.

The application begins with a Python-based GUI, built using Tkinter, that acts as the primary interface for the admin or teacher. This interface allows users to perform key operations such as Training, Marking Attendance, Viewing Data, accessing Learning Resources, and interacting with the Chatbot Module. Each button in the GUI triggers a well-defined backend process and ensures intuitive interaction for non-technical users.

The real-time facial recognition pipeline is powered by OpenCV. The system uses Haar Cascade classifiers and LBPH algorithms to capture and process live facial images. This module performs both Face Detection and Feature Extraction,

International Journal of Information Technology & Computer Engineering
ISSN 2347-3657
Volume 13, Issue 3, 2025

followed by model training using the extracted data. During attendance marking, the camera feed is used to match incoming faces with the trained model, ensuring identity verification and minimizing manual errors.

For backend processing and logging, Pandas is used to manipulate and write data into structured CSV files, while datetime modules handle timestamping. These attendance logs are stored in a present folder and are periodically pushed into a MySQL database for persistent, centralized storage. All facial datasets are maintained in a dataset/ folder, while chatbot responses and learning content are stored under notes/ in PDF or text format.

Application Layer: GUI and user input interface (Tkinter-based) **Processing Layer**: Core logic using OpenCV, Pandas, and NLP **Data Layer**: Handles CSVs, notes, datasets, and MySQL interactions

Storage Layer: Central MySQL server for long-term data preservation

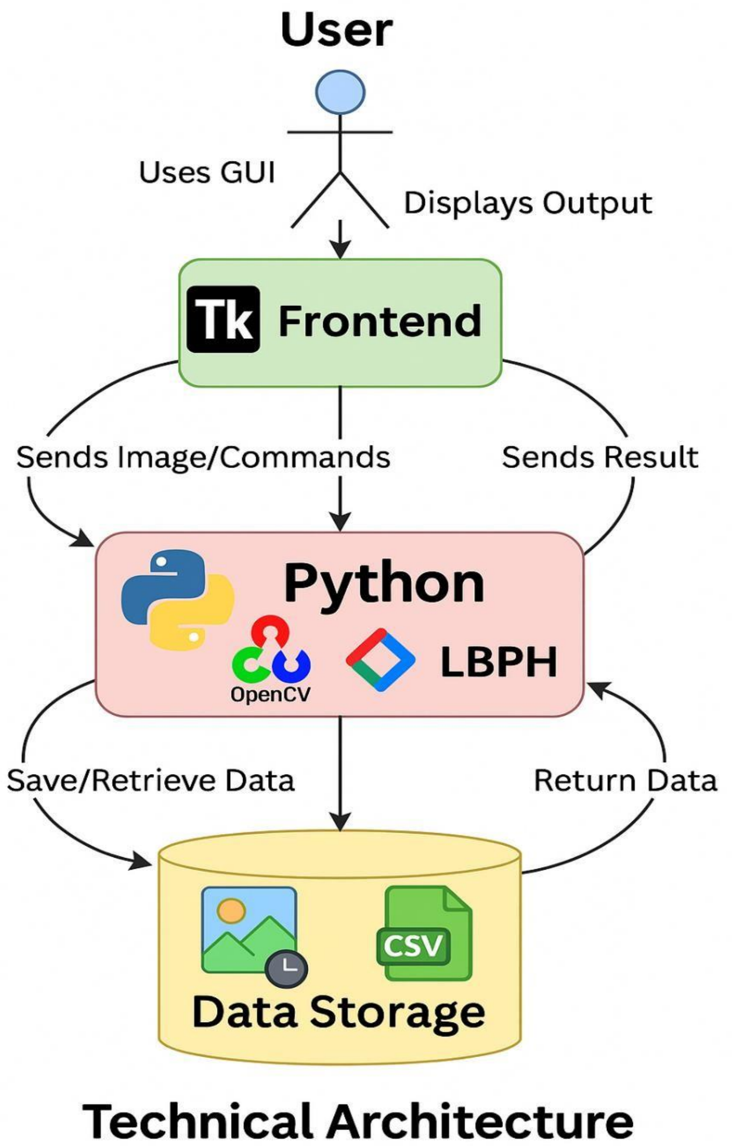


Fig. 4.1.2. Technical Architecture

5. IMPLEMENTATION

- **OpenCV**: Used for real-time webcam access, face detection using Haar Cascade, face recognition using LBPH algorithm, image preprocessing (grayscale conversion), saving training images, and annotating detected faces.

International Journal of Information Technology & Computer Engineering

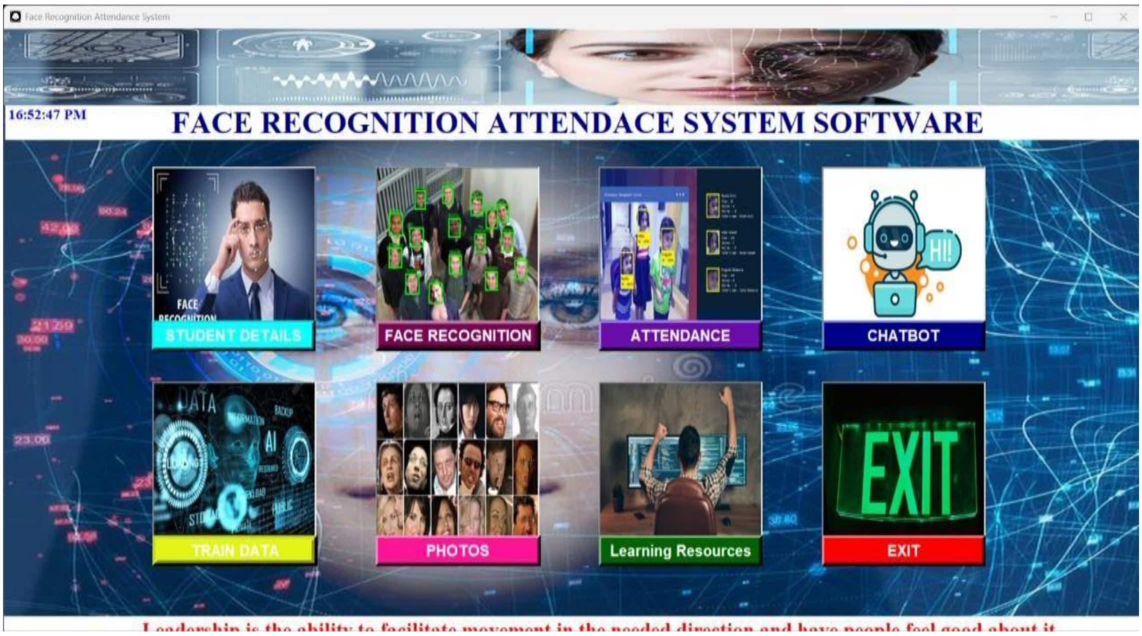
ISSN 2347-3657

Volume 13, Issue 3, 2025

- **NumPy**: Handles image data as arrays, performs matrix operations, and supports fast numerical computations needed for LBPH-based recognition.
- **Pandas**: Used to manage tabular student data (student.csv), record and display attendance (present.csv), and streamline reading/writing of structured datasets.
- **CSV (Python Standard Library)**: Helps in reading from and writing to attendance and student detail CSV files for logging and retrieval.
- **OS (Python Standard Library)**: Supports file and directory operations like saving images, accessing datasets, and managing paths (e.g., trainer file, dataset folder).

6. SCREENSHOTS

Screenshot 6.1 Main Login Page

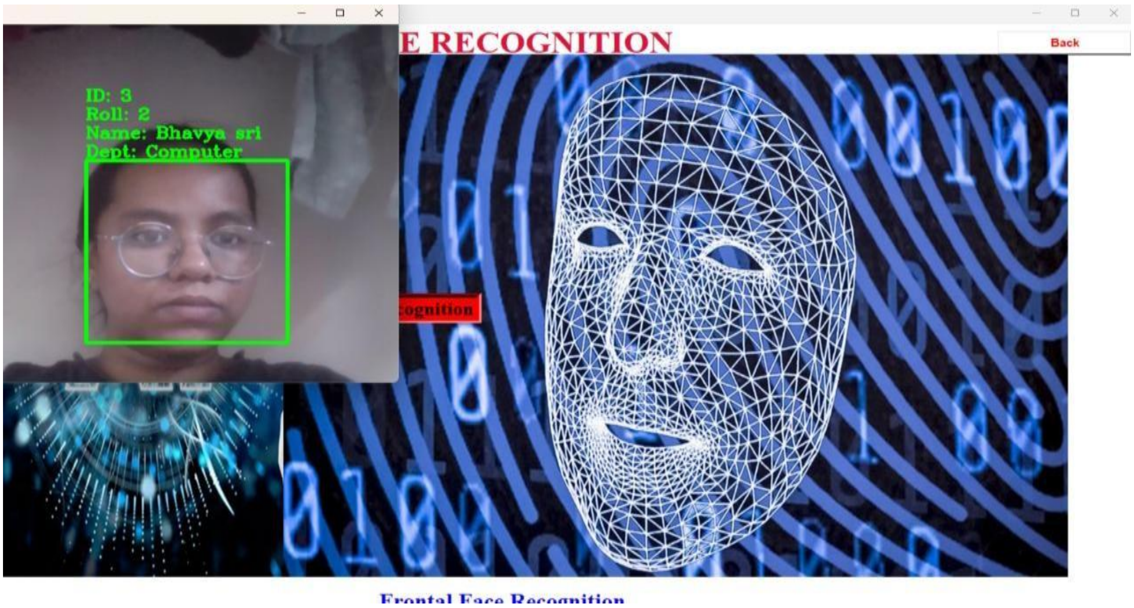


Screenshot 6.2: Main Dashboard

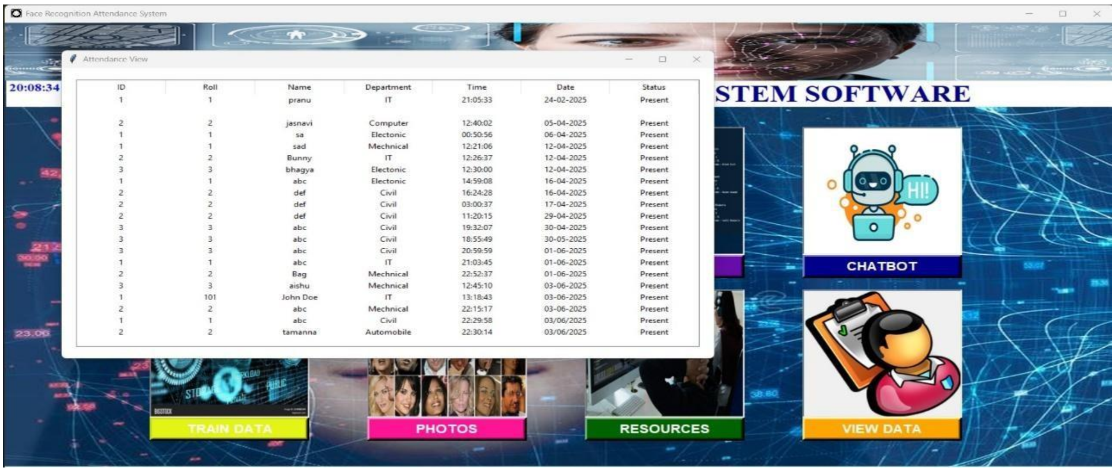
International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 13, Issue 3, 2025

Screenshot 6.3: Student management module

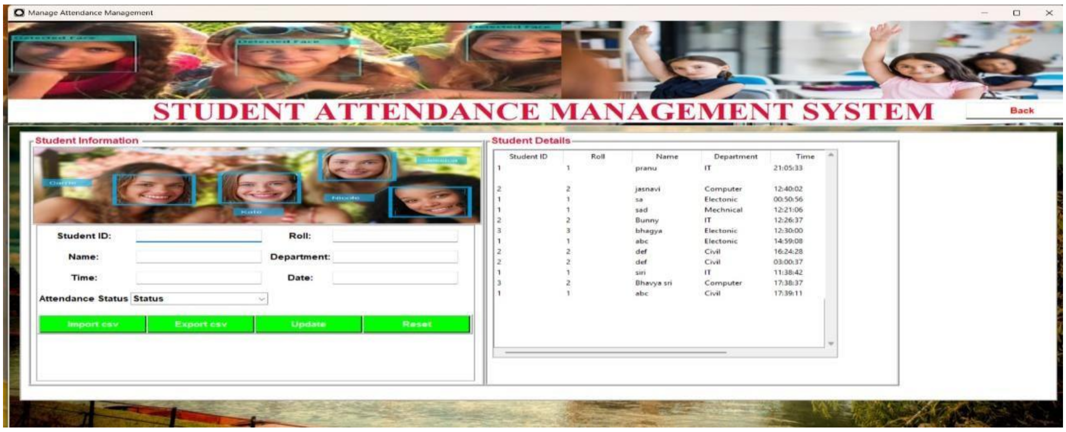
Screenshot 6.4: Face Recognition Module



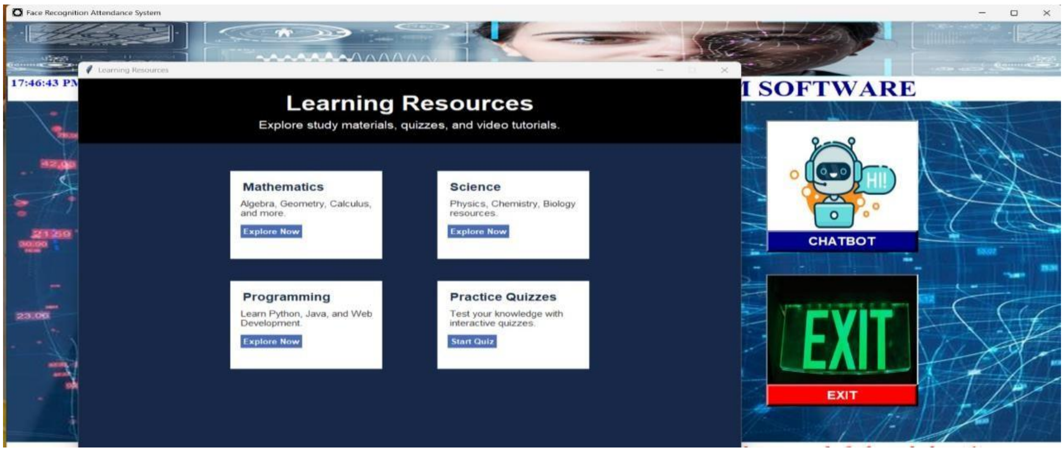
Screenshot 6.5: Training Data Module



Screenshot 6.6: Training Data Module



Screenshot 6.7: Training Data Module

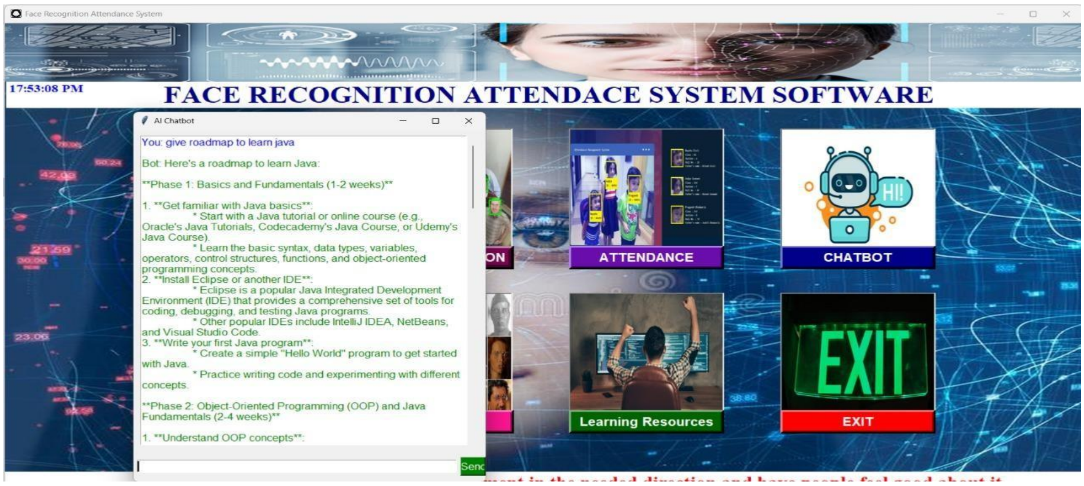


Screenshot 6.8: Learning Resources

International Journal of Information Technology & Computer Engineering

ISSN 2347-3657

Volume 13, Issue 3, 2025



Screenshot 6.9: Chatbot module

7-CONCLUSION

ThThe face recognition system using the LBPH (Local Binary Pattern Histogram) method is a highly effective and efficient approach for identifying individuals based on distinct facial features. Leveraging a structured pipeline that includes image preprocessing, feature extraction through local binary patterns, and histogram-based comparison, the system is capable of delivering accurate face detection and recognition across a range of environmental conditions and facial variations. Its performance remains consistent even with moderate lighting changes and minor occlusions, making it a robust choice for practical scenarios.

One of the key strengths of the LBPH algorithm lies in its simplicity, low memory footprint, and minimal computational overhead, allowing real-time processing on standard hardware without the need for high-end GPUs or cloud-based infrastructure. The system is capable of recognizing known individuals with high accuracy, while also efficiently handling and flagging unknown faces for further verification. These capabilities make it particularly suitable for implementation in real-time applications such as automated attendance systems, secure access control mechanisms, time-tracking in workplaces, and identity verification in restricted environments.

This project successfully demonstrates the real-world applicability of face recognition technology, highlighting its potential to replace traditional authentication methods such as manual logs, RFID cards, or biometric fingerprints, especially in settings where hygiene, speed, and scalability are critical. Furthermore, it opens avenues for future enhancements through integration with databases, remote monitoring tools, and machine learning-based decision systems to further improve accuracy and adaptability. Overall, the use of LBPH in this system showcases a balanced blend of efficiency, reliability, and affordability—making it an ideal solution for educational institutions, corporate offices, and public sector deployments alike.

REFERENCES

[1] Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition.

[2] 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.

[3] OpenCV Library. (2023). Open Source Computer Vision Library. https://opencv.org

[4] Satpathy, S., & Mishra, A. (2020). Automated Attendance System using Face Recognition. International Journal of Scientific & Technology Research, 9(4), 601–605.

[5] Tiwari, A., & Verma, R. (2021). A review paper on face recognition technologies.
Journal of Emerging Technologies and Innovative Research, 8(3), 1254–1261.

[6] Ghosh, S. (2022). Security and Ethical Challenges of Face Recognition. ACM Computing Surveys.

[7] MySQL Documentation. (2023). MySQL 8.0 Reference Manual. https://dev.mysql.com/doc

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