Attendance Management System Using Face Recognition

A Project Report

submitted in partial fulfillment of the requirements

of

AICTE Internship on AI: Transformative Learning
with
TechSaksham – A joint CSR initiative of Microsoft & SAP

by

Shubham Narvekar, snk20041130@gmail.com

Under the Guidance of

Aditya Ardak

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to everyone in Techsaksham who has contributed to the successful completion of this project, Attendance Management System Using Face Recognition.

Firstly, I would like to thank you to my mentor **Aditya Ardak** sir, for his invaluable guidance, support, encouragement throughout this project. Sir feedback and suggestion helped me stay focused and motivated. Thanks for providing expertise and assistance in the areas such as face recognition, capturing image, training images using models.

I would also like to acknowledge the developers and contributors of the libraries and frameworks I used in this project, including TensorFlow, OpenCV, Tkinter, and Numpy. Their open-source contributions were crucial to the development and implementation of the system.

Thank you all for your support.

ABSTRACT

The Face Recognition Attendance Management System aims to do automatic process of marking attendance using face recognition technology. The traditional manual attendance system makes error and can be time-consuming in large organizations or educational institutions. This project addresses these challenges by developing a solution that uses computer vision and machine learning to accurately recognize faces and record attendance.

Objectives:

1. Develop a system that can capture and store facial data for each individual.
2. Train a face recognition model capable of identifying users in real-time.
3. Create a graphical user interface (GUI) for interacting with the system.
4. Store attendance records and provide a mechanism for viewing past attendance data.

Methodology:

The system uses Python and several libraries, including **OpenCV, TenserFlow,** and **Tkinter.** First, the system captures facial images using a webcam, with a user ID and user name, and stores in the dataset. Next to train the model on these facial images using algorithm. Once the model is trained, it can recognize faces from the webcam and mark the attendance accordingly. The system also features a GUI for easy interaction capturing faces, training the model and displaying attendance.

Key Results:

The system successfully recognized faces and record attendance in real-time. It efficiently handled multiple users and stored accurate records in a CSV file.

Conclusion:

The face recognition attendance management system is a effective solution for automating attendance tracking, offering a more efficient, reliable, and secure alternative to traditional methods.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

1.1Problem Statement:

The process of manually recording attendance is time-consuming, prone to errors, and susceptible to manipulation. Traditional methods, such as calling out names or signing attendance sheets, can lead to inefficiencies in institutions and workplaces. Additionally, such methods lack reliability and security, as attendance can be falsified through proxy marking. This project addresses these challenges by automating attendance tracking using facial recognition technology, which ensures accuracy, security and convenience. The ability to identify individuals in real-time without physical interaction is especially relevant in the modern era, where efficiency and contactless systems are becoming necessity

1.2Motivation:

The motivation for this project comes from the growing need for efficient, secure, and automated systems in institutions and organizations. Facial recognition technology has gained widespread attention due to its high accuracy in identifying individuals. By integrating this technology into attendance management, the project eliminates traditional methods while offering potential applications:

- **Educational Institutions:** Automating student attendance in classrooms and exams.
- **Corporate Environments:** Tracking employee attendance and ensuring compliance with workplaces policies.
- **Public and Private Events:** Monitoring attendance at conferences, seminars, and gatherings.

The impact of this project extends beyond operational efficiency, as it promotes technological adoption, enhances security, and support a seamless user experiences.

1.3 Objective:

The primary objective of the project are:

1. To develop an automated attendance system using facial recognition technology.
2. To create a user-friendly interface for capturing, storing and managing facial data.
3. To design and train a facial recognition model capable of identifying individuals in real time with high accuracy.
4. To maintain attendance records in a structured format for easy retrieval and analysis
5. To ensure data privacy and security while providing reliable performance under varied conditions.

1.4 Scope of the Project:

The scope of this project includes:

- Development of a facial recognition system using python and key libraries like OpenCV and TensorFlow.
- A GUI for interaction, including features for capturing images, training the model, and displaying attendance records.
- A local storage mechanism (CSV) for managing attendance data.

Limitations:

- The system requires adequate lighting and proper positioning of the user for accurate facial recognition.
- It is currently designed for small to medium-scale usage and may need optimization for large-scale deployments.
- The project does not include advanced features such as cloud integration or mobile compatibility, which can be explored in future times.

CHAPTER 2

Literature Survey

2.1 Review relevant literature or previous work in this domain.

The domain of face recognition and attendance management has been extensively studied and developed over the years, leveraging advancements in machine learning, image processing, and computer vision. Key research and works in this area include:

1. Viola-Jones Algorithm for Face Detection:
   Viola and Jones (2001) introduced a robust and rapid face detection algorithm using a cascade classifier. It remains a foundational technique for detecting faces in real-time applications. [1]
2. Face Recognition using Eigenfaces and Fisherfaces:
   Early models utilized Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) for face recognition. While effective, these methods struggled with variations in lighting, pose, and occlusions.[2]
3. Deep Learning for Face Recognition:
   Deep learning models, such as FaceNet, DeepFace, and VGGFace, have significantly improved recognition accuracy by extracting high-dimensional feature representations from facial images.[3]
4. Applications in Biometrics and Security:
   Face recognition systems are widely used in security systems, biometrics, and surveillance, as discussed in works by [4]

2.2 Mention any existing models, techniques, or methodologies related to the problem.

Several models and methodologies have been developed to solve the problem of face recognition-based attendance:

1. **Dlib's Face Recognition Library:**
   This library uses Histogram of Oriented Gradients (HOG) for feature extraction and Support Vector Machines (SVM) for classification. It provides high accuracy and ease of implementation.
2. **DeepFace by Facebook:**
   DeepFace introduced a deep learning-based model for face recognition, achieving human-level accuracy.
3. **FaceNet by Google:**
   FaceNet uses deep convolutional neural networks to project facial images into a Euclidean space, simplifying the recognition process.[5]
4. **OpenCV Integration:**
   OpenCV provides comprehensive tools for face detection, alignment, and recognition using pre-trained Haar cascades and deep learning models.[6]

2.3 Highlight the gaps or limitations in existing solutions and how your project will address them.

Despite significant advancements, the following gaps or limitations exist in current face recognition-based attendance systems:

1. **High Computational Requirements:**
   Many advanced face recognition models demand high computational power, making them unsuitable for resource-constrained systems.
2. **Data Privacy Concerns:**
   Systems storing facial data locally or on servers pose potential privacy risks if data is not properly secured.
3. **Recognition Challenges in Real-World Scenarios:**
   Variations in lighting, pose, facial expressions, and occlusions (e.g., masks) reduce recognition accuracy in uncontrolled environments.
4. **Scalability Issues:**
   Existing systems may struggle to handle a large number of users efficiently.

This project aims to overcome the identified gaps by:

1. **Efficient Algorithms:** Using lightweight algorithms such as the Viola-Jones method for face detection and Local Binary Patterns Histograms (LBPH) for recognition to ensure compatibility with low-resource systems.
2. **Enhanced Privacy**: Storing attendance data securely in local databases and limiting unnecessary data exposure.
3. **Robust Design**: Ensuring the system is adaptable to varying environmental conditions by preprocessing images for better recognition accuracy.
4. **Scalability:** Implementing a modular design that can handle multiple users and integrate with larger systems in the future.

CHAPTER 3

Proposed Methodology

3.1 System Design

System Architecture

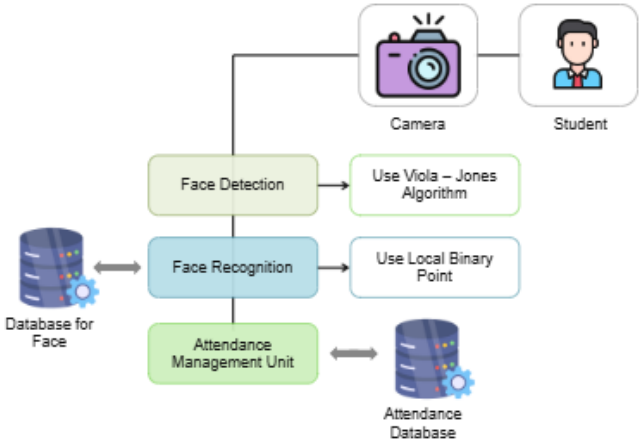

Figure 1. System Design of the project

The provided system design illustrates the workflow of Face Recognition Attendance Management System, breaking it into different modules.

1. **Input: Camera and Student**
    - **Camera**
        - Acts as the primary input device for the system.
        - Capture live images of the students/users.
        - Serves as the starting point for face detection and recognition.
    - **Student**
        - Represents the subject whose face is to be recognized and attendance is to be recorded.
2. **Face Detection Module**
    - **Process:**
        - The camera feed is sent to the Face Detection module.
        - The system identifies and isolates faces from the captured frame.
        - Non-face elements in the frame are ignored.
        - The Viola-Jones Algorithm is used for face detection.

3. **Face Recognition Module**
    - **Process:**
        - The detected faces are passed to the Face Recognition module for further analysis.
        - This module compares the detected face with the data stored in the Database for Face.
        - Local Binary Patterns are used for recognition.
    - **Database for Face:**
        - Stores facial features and user data during the initial enrollment phase.
        - Acts as the reference data for recognition.
4. **Attendance Management Unit**
    - **Process:**
        - Once the face is recognized, the Attendance Management Unit marks the attendance of the identified individual.
        - Records the user's attendance in the Attendance Database with the corresponding timestamp.
    - **Attendance Database:**
        - A centralized storage system where all attendance records are maintained.
        - Allows retrieval and analysis of attendance data, such as checking attendance history.

3.2 Requirement Specification

To implement the Face Recognition Attendance Management System effectively, the following tools and technologies are required:

3.2.1 Hardware Requirements:

1. **Camera:**
    - A webcam or external camera for capturing clear facial images.
2. **Computer System:**
    - Processor: Intel i5 or higher.
    - RAM: Minimum 8 GB (16 GB recommended for better performance during model training).

- Storage: Minimum 256 GB for storing datasets and attendance records.

3. **Power Supply:**
   - A stable power source to ensure uninterrupted system operation.

3.2.2 Software Requirements:

1. **Programming Languages:**
   - **Python:** The primary language for implementing face detection, recognition, and GUI development.
2. **Libraries and Frameworks:**
   - **OpenCV:** For real- time computer vision tasks, including face detection and image processing.
   - **TensorFlow:** For training and deploying machine learning models.
   - **Numpy:** For numerical computation and array manipulation.
   - **Tkinter:** For building the graphical user interface (GUI).
3. **Operating System:**
   - Windows 10 or later/ Linux (Ubuntu 18.04 or higher) / macOS.
4. **Development Tools:**
   - **Python IDE/Code Editor:**
     - Examples: PyCharm, VS Code, or Jupyter Notebook for coding and debugging.
   - **Pip:** For installing required Python Libraries.
5. **Database:**
   - **CSV files:** Used for storing attendance records.
6. **Other Dependencies:**
   - **Haarcascade Pre-trained Model:** Used with OpenCV for face detection.
   - **LBPH Algorithm:** For face recognition.

CHAPTER 4

Implementation and Result

4.1 Snap Shots of Result:

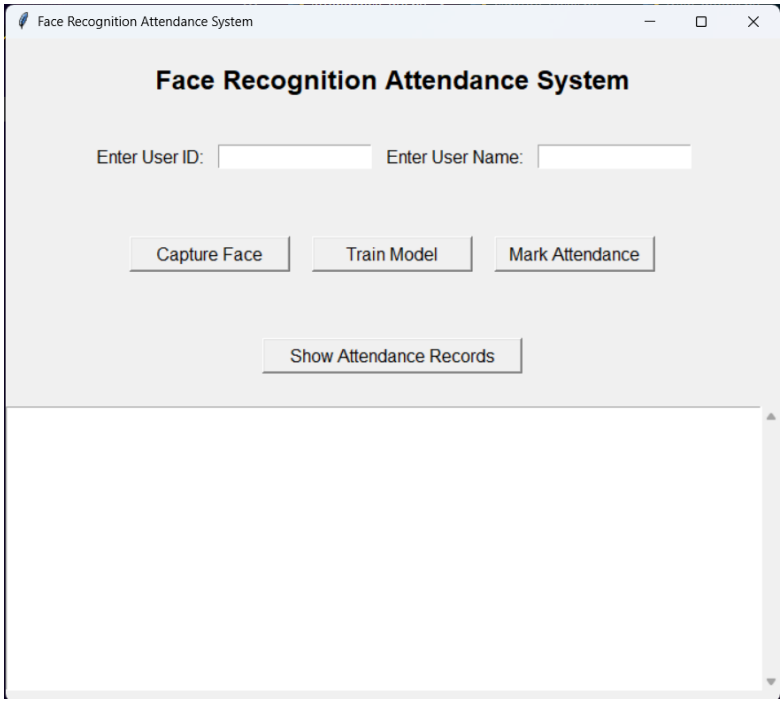

Figure 2. GUI

This is snapshot that represents the main GUI (Graphical User Interface) of the Face Recognition Attendance System.

- The title "Face Recognition Attendance System" is prominently displayed at the top of the window, indicating the application's purpose.
- A text box where the user inputs their unique User ID during face data registration.
- A text box where the user provides their name for identification and attendance purposes.
- The Capture Face button initiates the process of capturing the user's facial data via the connected camera. The data is stored for future recognition.

- The Train Model button triggers the training of the facial recognition model. It uses the stored facial data to improve the system's recognition accuracy.
- The Mark Attendance button starts the process of recognizing a user's face in real-time and marking their attendance in the database.
- The Show Attendance Records button loads data in the bottom text area displaying all attendance records. The section is blank in the current snapshot because no data is yet loaded or displayed.

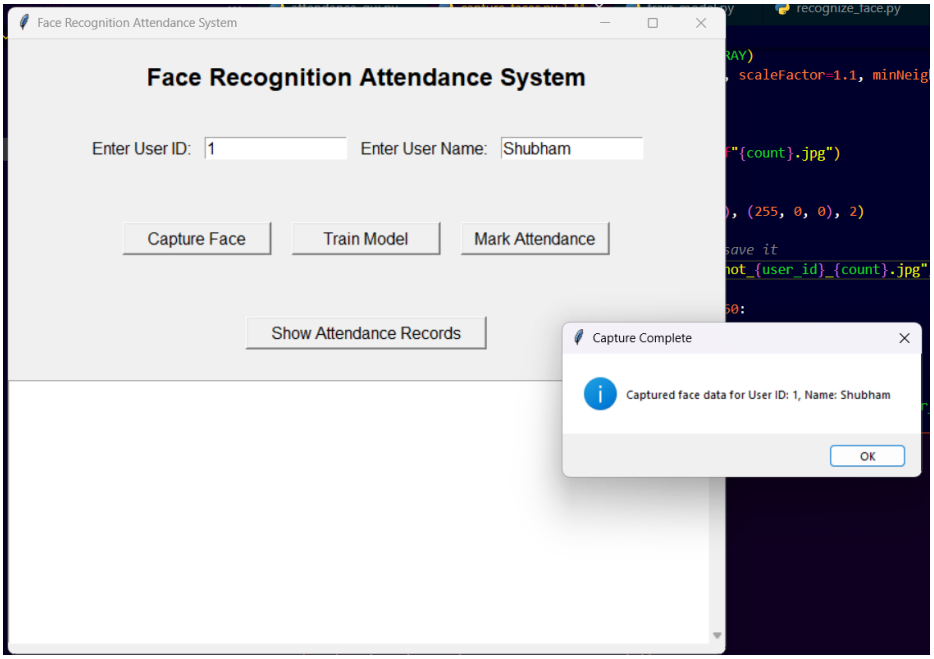

Figure 3. Capture Face with pop-up notification

Figure 4. Capture Face Image

This screenshot showcases the Face Recognition Attendance System's GUI, specifically highlighting the functionality of capturing face data for a new user.

- The user has entered the value 1 in this field, indicating the unique identifier for the individual whose face data is being captured.
- The user has entered Shubham, providing the name associated with the User ID.
- The Capture Face button is used to initiate the process of capturing face data for the given User ID and User Name. Clicking this button triggers the camera to capture and store face images for training.
- Above the facial image which is captured by the camera.
- The message reads: "Captured face data for User ID: 1, Name: Shubham". This ensures the user that the system has successfully recorded the face data.

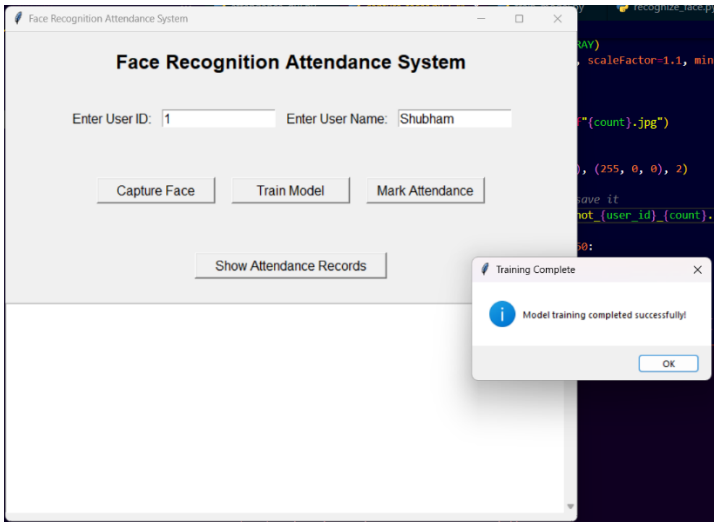

Figure 5. Train Model

This screenshot showcases the Face Recognition Attendance System's GUI, specifically highlighting the functionality of training model of capture image of a new user.

- The model employs a machine learning algorithm such as Local Binary Patterns Histogram (LBPH) or any other face recognition algorithm to train on the stored face images.
- The training process involves analyzing unique facial features and encoding them into a format that the model can recognize.

- Once the training is successfully completed, the system outputs a confirmation message.

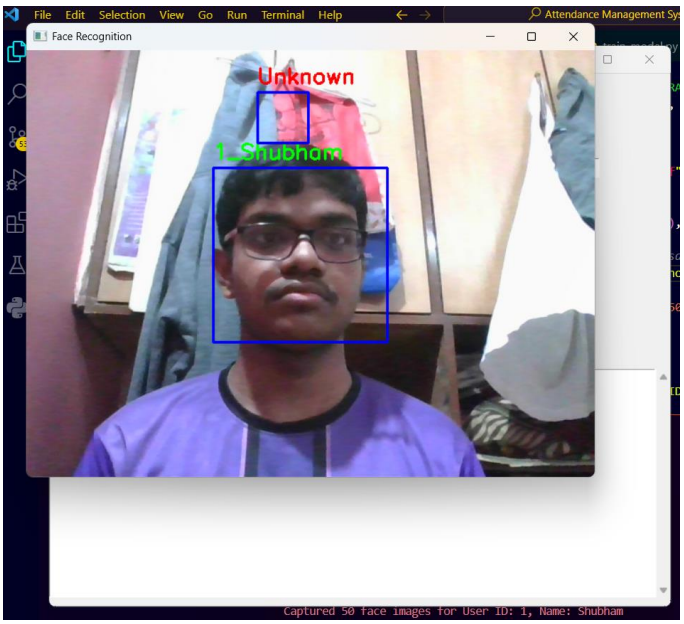

Figure 6. Marking Attendance

This screenshot demonstrates the execution of the Mark Attendance feature in the Face Recognition Attendance System.

- The user initiates the attendance process by selecting the Mark Attendance button in the GUI.
- The system activates the webcam to capture the live video feed.
- The trained face recognition model processes the video feed in real time, identifying any faces visible in the frame.
- It matches the detected faces against the previously trained data using unique facial features.
- Upon a successful match, the system retrieves the corresponding user ID and name from the database.
- The attendance is marked in the attendance database, along with the current date and time.
- After logging the attendance, a popup or message box is displayed, confirming that the attendance for the specific user has been successfully marked.

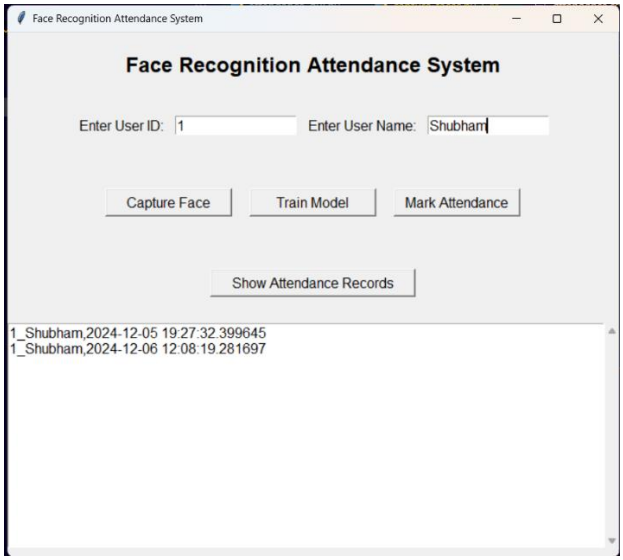

Figure 7. Show Attendance Records

This screenshot highlights the functionality of the Show Attendance Records feature in the Face Recognition Attendance System.

- The user selects the Show Attendance Records button from the GUI.
- This triggers the system to fetch attendance data from the attendance database.
- The system retrieves all recorded attendance entries, including the User ID, User Name, Date, and Time of attendance.
- The retrieved data is displayed in a neatly formatted manner, such as a table or list, within the GUI.

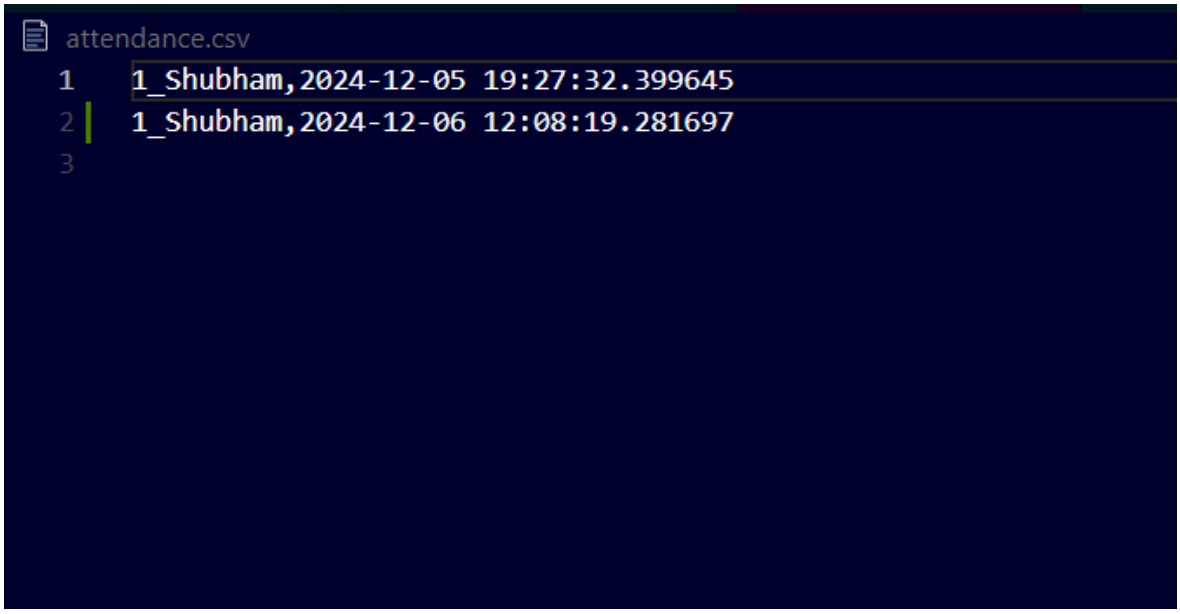

Table 1. Attendance Table

This snapshot showcases the attendance.csv file, which serves as the primary data storage for all attendance records in the Face Recognition Attendance System.

4.2 GitHub Link for Code:

https://github.com/SNKPSEUDO11/Attendance-Management-System-using-Face-Recognition.git

CHAPTER 5

Discussion and Conclusion

5.1 Future Work:

1. **Enhancing Recognition Accuracy:**
    - Implement advanced deep learning models such as Convolutional Neural Networks (CNNs) for better accuracy in face recognition, especially in challenging scenarios like varying lighting conditions, facial expressions, and partial occlusions.
    - Integrate pre-trained models such as FaceNet or Dlib for more robust and faster recognition.
2. **Integration with IoT Devices:**
    - Incorporate IoT-enabled devices like smart cameras for real-time attendance tracking in larger environments such as classrooms or offices.
3. **Cloud-Based Storage and Analysis:**
    - Migrate the database to a cloud platform to enable real-time, scalable, and secure data storage and analytics.
4. **Data Privacy and Compliance:**
    - Incorporate data encryption and ensure compliance with privacy regulations like GDPR to safeguard sensitive information.
5. **Attendance Analytics Dashboard:**
    - Develop a comprehensive analytics dashboard to provide insights such as attendance trends, punctuality statistics, and user behavior analysis.

5.2 Conclusion:

The Face Recognition Attendance System is a significant step towards automating and modernizing attendance management processes. By leveraging face detection and recognition technologies, the system eliminates the need for traditional manual or biometric-based attendance mechanisms, providing a touchless and efficient alternative.

The project successfully achieved its objectives, including capturing and storing user facial data, training a recognition model, and accurately marking attendance. The system offers a user-friendly interface and ensures minimal human intervention, thereby reducing errors and saving time.

The implementation demonstrates the potential of combining machine learning, computer vision, and GUI-based applications to solve real-world problems. While the current system is functional and effective, there is ample scope for enhancements such as improved recognition accuracy, mobile integration, and cloud-based deployment.

In conclusion, this project contributes to the growing adoption of AI-driven solutions in administrative tasks, paving the way for smarter, faster, and more reliable attendance management systems in educational institutions, workplaces, and beyond.

REFERENCES

[1] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, IEEE Comput. Soc, pp. I-511-I–518. doi: 10.1109/CVPR.2001.990517.

[2] M. Turk and A. Pentland, “Eigenfaces for Recognition,” *J Cogn Neurosci*, vol. 3, no. 1, pp. 71–86, Jan. 1991, doi: 10.1162/jocn.1991.3.1.71.

[3] O. M. Parkhi, A. Vedaldi, and A. Zisserman, “Deep Face Recognition,” in *Procedings of the British Machine Vision Conference 2015*, British Machine Vision Association, 2015, pp. 41.1-41.12. doi: 10.5244/C.29.41.

[4] S. Z. Li and A. K. Jain, Eds., *Handbook of Face Recognition*. London: Springer London, 2011. doi: 10.1007/978-0-85729-932-1.

[5] F. Schroff, D. Kalenichenko, and J. Philbin, “FaceNet: A unified embedding for face recognition and clustering,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2015, pp. 815–823. doi: 10.1109/CVPR.2015.7298682.

[6] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, “Deep Learning for Computer Vision: A Brief Review,” *Comput Intell Neurosci*, vol. 2018, pp. 1–13, 2018, doi: 10.1155/2018/7068349.