Face Recognition System

Project Report

BACHELORS OF TECHNOLOGY

in

CSE

To

K.R Mangalam University

by

Utkarsh Thapliyal(2401010052)

Pranjal Dutta(2401010051)

Jony Gautam(2401010032)

Ritik Sharma(2401010058)

Under the supervision of

Dr. Rishika Mehta

Department of Computer Science and EngineeringSchool of

Engineering and Technology

INDEX

ABSTRACT

The traditional methods of recording attendance, whether through manual registers or RFID-based systems, are often time-consuming, error-prone, and vulnerable to manipulation such as proxy attendance. In the digital era, the need for a more efficient, accurate, and contactless attendance solution is critical, especially in educational institutions, workplaces, and secured environments. This project presents a **Face Recognition-Based Attendance System** implemented using **Python**, the **Flask web framework**, and the **face_recognition** library, aiming to automate and enhance the process of recording attendance.

The system captures an image of the individual through a webcam embedded in a web interface, processes the image using OpenCV, and compares the detected face with pre-encoded images of known individuals stored in the system. If a match is found, the individual's attendance is marked and logged into a CSV file, uniquely generated for each date. This approach significantly reduces the need for manual supervision and provides a secure, real-time, and user-friendly way of taking attendance.

The project is composed of several key modules: a **web interface** for image capture, a **backend processing engine** for face detection and recognition, and a **data logging component** for attendance storage. Images of known individuals are stored in a predefined directory. During system initialization, these images are loaded, verified to be in proper RGB 8-bit format, and encoded into numerical face embeddings using the face_recognition library. The Flask server handles incoming POST requests containing webcam-captured images encoded in Base64 format, decodes and processes them, and performs face matching. If the captured face matches one of the known encodings with a confidence threshold, attendance is marked by recording the person’s name and timestamp into a CSV file for the day.

This project also emphasizes robustness in handling errors and edge cases, such as invalid image types, no faces found, or multiple faces in a frame. The system uses appropriate logging and error messaging to guide users during operation. For performance and accuracy, the face recognition model used is built upon deep learning and returns reliable matches under good lighting and frontal face angles.

The Face Recognition Attendance System represents a scalable and adaptable solution. It can be extended by integrating a database instead of CSV storage, providing role-based access (admin and user), and adding a dashboard to display attendance reports.

1. INTRODUCTION

Attendance management is a critical function in educational institutions, corporate offices, and organizations where monitoring employee or student presence is essential for productivity, discipline, and record-keeping. Traditionally, attendance is recorded manually using registers or electronic methods such as RFID cards and biometric systems like fingerprint scanners. However, these methods have limitations including susceptibility to proxy attendance, hygiene concerns (especially in post-pandemic environments), and dependency on human supervision.
With the advancements in artificial intelligence and computer vision, face recognition has emerged as a reliable and efficient technology for identity verification. Face recognition offers a **contactless, non-intrusive**, and user-friendly alternative to conventional attendance systems. It leverages facial features that are unique to each individual, reducing the chances of impersonation and ensuring accurate identification.

This project aims to build a **Face Recognition-Based Attendance System** using **Python**, the **Flask web framework**, **OpenCV**, and the **face_recognition** library. The system captures an image from the user's webcam via a browser interface, processes it to extract facial features, and compares it against a database of known faces. If a match is found, the system marks the individual's attendance by saving their name and timestamp in a daily CSV file.

The entire process is automated and occurs in real time through a web interface, making it accessible and scalable. The system also includes error handling for unsupported image types, absence of detectable faces, and mismatch cases. By integrating face recognition with a web-based UI, the project demonstrates a practical, modern solution for digital attendance management that is both secure and efficient.

This system not only reduces administrative overhead but also ensures the integrity and accuracy of attendance data, setting the foundation for smarter.

2. MOTIVATION

In today's fast-paced and technology-driven world, the efficiency and accuracy of administrative tasks such as attendance management are more critical than ever. Traditional methods of attendance—such as paper-based sign-in sheets, manual roll calls, or even RFID and biometric systems—often come with significant drawbacks. Manual processes are time-consuming, prone to human error, and susceptible to manipulation such as proxy attendance. Biometric methods like fingerprint scanners, while more secure, raise hygiene concerns due to physical contact, especially in light of recent global health crises like COVID-19.

These limitations highlight the need for an innovative, contactless, and intelligent attendance system that minimizes human involvement, enhances reliability, and streamlines daily operations. The emergence of **face recognition technology**, powered by artificial intelligence and deep learning, provides an excellent opportunity to address these issues. Face recognition systems offer a non-intrusive, fast, and accurate method of verifying identity by analyzing unique facial features.

The motivation behind this project stems from a desire to apply cutting-edge AI technologies to real-world problems. Automating attendance through face recognition not only saves time but also improves security and accountability. It eliminates the need for physical interaction, reduces the administrative workload, and ensures that records are maintained systematically without tampering or oversight.

Furthermore, the use of a **web-based interface via Flask** makes the system highly accessible and platform-independent, allowing it to be used on any device with a webcam and browser. This increases the practicality and scalability of the system for deployment in schools, universities, offices, and other institutions.

By integrating **Python**, **OpenCV**, and the **face_recognition** library into a cohesive solution, this project aims to demonstrate how modern technologies can replace outdated systems in a meaningful way. The long-term vision is to evolve this project into a fully-featured attendance platform with database integration, analytics, and cloud support.

Ultimately, this project is motivated by the need for smarter, safer, and more efficient solutions in everyday operations—and showcases how artificial intelligence can contribute significantly to building those systems.

3. LITERATURE REVIEW

Face recognition technology has garnered significant attention over the past two decades due to its potential applications in security, surveillance, and automation. With rapid advancements in computer vision and machine learning, researchers and developers have explored various methods to enhance the accuracy, reliability, and efficiency of facial recognition systems. This literature review highlights key studies, frameworks, and technologies that have influenced the development of face recognition-based attendance systems.

Traditional Attendance Systems relied heavily on manual inputs or biometric methods like fingerprint and RFID cards. While these systems were functional, they suffered from challenges such as proxy attendance, card theft, wear and tear, and hygiene concerns. To overcome these limitations, researchers began exploring **biometric-based attendance systems**, particularly facial recognition due to its contactless nature and high accuracy under controlled conditions.

Ming-Hsuan Yang et al. (2002) provided foundational work in face detection and recognition by evaluating appearance-based methods such as Eigenfaces and Fisherfaces. These early methods used principal component analysis (PCA) and linear discriminant analysis (LDA) for dimensionality reduction and feature extraction. Although innovative at the time, they lacked robustness in varying lighting conditions and facial orientations.

Later, **Viola and Jones (2004)** proposed a real-time face detection framework using Haar cascades, enabling faster detection speeds. This method became widely adopted in early face recognition applications but was still limited by its sensitivity to lighting and occlusion.

With the advent of **deep learning**, significant improvements emerged. The **DeepFace model by Facebook (2014)** and **FaceNet by Google (2015)** revolutionized the field by introducing deep convolutional neural networks (CNNs) that achieved human-level accuracy in facial verification tasks. These models use facial embeddings—a compact numerical representation of facial features—which can be compared using Euclidean distances.

The `face_recognition` Python library, built on top of dlib and implementing FaceNet-style deep learning techniques, made it easier for developers to implement real-world face recognition systems without needing to train complex models. This library has been adopted in several academic and commercial projects for tasks such as identity verification and attendance tracking.

In conclusion, existing literature and technologies provide a strong foundation for implementing accurate, contactless, and automated attendance systems. The current project builds upon these advancements to offer a reliable solution tailored for educational and organizational environments.

.

4. GAP ANALYSIS

Despite numerous advancements in biometric-based attendance systems, several gaps still exist in their implementation, efficiency, and practical usage. Traditional attendance systems—ranging from manual entry to fingerprint and RFID-based solutions—pose issues related to security, hygiene, and user convenience. Although face recognition technology has emerged as a promising alternative, existing implementations often fall short in several critical areas, particularly in real-time application, affordability, ease of deployment, and adaptability to uncontrolled environments.

One of the primary gaps identified is the **lack of contactless, automated attendance solutions** that are both cost-effective and easy to integrate. Many existing commercial face recognition systems are expensive, require proprietary hardware, or are limited to specific platforms. This restricts their adoption, especially in educational institutions and small-to-medium enterprises with limited resources.

Another major gap lies in **data handling and user privacy**. While several systems store attendance records, they often do not ensure secure, structured, and retrievable formats. Moreover, many do not incorporate local storage alternatives, defaulting to cloud systems that may not meet privacy requirements, particularly in educational or governmental use cases. This calls for a more flexible architecture that can support local data storage in widely accepted formats like CSV or can be integrated with databases as needed.

From a technical standpoint, many systems do not robustly handle edge cases, such as **unsupported image formats**, **poor lighting**, **multiple faces in a frame**, or **unregistered users**. These deficiencies result in unreliable performance in real-world settings, where environmental factors cannot always be controlled. Furthermore, many implementations lack **real-time feedback mechanisms** to inform users of success or failure in recognition, reducing usability. Additionally, **integration with web-based platforms** remains limited. Desktop-based systems are more common but lack portability and ease of access. A system that leverages a browser interface, using frameworks like Flask, provides better scalability and device independence, yet such approaches are underutilized in practical deployments.

This project addresses these gaps by developing a **web-based, real-time, contactless attendance system** that uses open-source technologies. It ensures compatibility with standard image formats, offers clear error handling, supports local attendance logging, and provides a user-friendly interface accessible from any webcam-enabled device. By bridging these technical and practical shortcomings, the system offers a more complete and adaptable solution suitable for modern attendance tracking needs.

5. PROBLEM STATEMENT

Attendance management is a fundamental yet often time-consuming task in educational institutions, corporate organizations, and training centers. The accuracy and integrity of attendance data play a crucial role in performance evaluation, payroll, grading, and security. However, existing attendance systems—ranging from manual sign-in sheets to biometric-based solutions—pose several limitations that affect their efficiency, reliability, and user experience.

Manual attendance systems are inherently flawed due to their dependence on human supervision. They are time-consuming, prone to human error, and highly susceptible to fraudulent practices such as proxy attendance or buddy punching. These shortcomings not only compromise the authenticity of attendance records but also waste valuable time that could be spent on productive activities.

Biometric systems such as fingerprint scanners and RFID-based cards were introduced to address some of these challenges. While they improve accuracy and reduce fraud, they are not without limitations. Fingerprint scanners require physical contact, which raises hygiene concerns—especially in the post-pandemic era. RFID cards can be lost, stolen, or swapped, and their use still often requires physical interaction or close proximity to the scanner. Both systems also involve additional hardware, increasing cost and maintenance requirements.

Moreover, many existing systems lack scalability and real-time feedback, do not provide accessible web interfaces, and offer poor error handling in cases such as unreadable images, multiple faces, or no face detected. These limitations hinder their usability and reliability in dynamic environments with varying lighting conditions, camera angles, and user behaviors.

The need of the hour is a **contactless, automated, real-time, and user-friendly attendance system** that ensures security, reduces human involvement, and is easy to deploy using commonly available hardware like webcams. Furthermore, the system must be capable of accurately identifying individuals based on unique facial features, storing attendance records efficiently, and providing immediate feedback to users.

This project aims to solve these problems by designing and developing a **face recognition-based attendance system using Python, Flask, OpenCV, and the face_recognition library**. The solution is intended to be platform-independent, cost-effective, and accessible through a web browser, making it suitable for schools, universities, offices, and other institutions seeking a smarter and more reliable attendance solution.

6. OBJECTIVES

- To develop a contactless and automated attendance system that uses facial recognition to eliminate the need for manual or physical input methods such as fingerprint scanning or RFID cards.
- To accurately detect and recognize individual faces from live webcam input or uploaded images using deep learning-based face encoding and comparison techniques.
- To create a user-friendly, web-based interface using Flask that allows seamless interaction with the system from any device equipped with a camera and internet browser.
- To ensure secure and efficient storage of attendance data by maintaining daily logs in structured formats like CSV, with options for future database integration
- To provide real-time feedback to users during the attendance process, including success, failure, and error messages for improved usability.
- To handle edge cases and errors robustly, including situations where no face is detected, multiple faces are present, or unsupported image formats are used.
- To offer a scalable and open-source solution that can be easily customized and deployed across educational institutions, offices, and training environments with minimal hardware requirements.

7. Tools/Technologies Used

- **Python**
  The core programming language used for implementing backend logic, facial recognition, and data handling due to its simplicity and robust libraries.

- **Flask**
  A lightweight Python web framework used to build the web-based user interface and handle HTTP requests for real-time image capture and processing.

- **OpenCV (cv2)**
  An open-source computer vision library used for image manipulation, conversion (e.g., BGR to RGB), and decoding image data from base64 format.

- **face_recognition (dlib)**
  A powerful Python library built on dlib, used for detecting and encoding facial features and comparing them to known faces using deep learning models.

- **NumPy**
  A fundamental package for numerical computing in Python, used for handling image arrays and mathematical operations.

- **Pandas**
  A data analysis library used for managing attendance records in CSV format, allowing efficient reading, writing, and updating of tabular data**.**

- **HTML/CSS/JavaScript (within Flask templates)**
  Used to create the front-end interface that captures webcam images, sends them to the server, and displays real-time feedback to users.

8.METHODOLOGY

The development of the Face Recognition-Based Attendance System follows a structured, step-by-step methodology to ensure accuracy, usability, and scalability. This methodology consists of the following key stages:

---

1. Requirement Analysis

In this phase, the project's functional and non-functional requirements were identified. The core requirements included real-time face recognition, attendance logging, contactless operation, and a web interface. Hardware and software constraints were also considered to ensure the solution would be accessible using standard webcams and open-source tools.

---

2. Technology Selection

Based on the requirements, open-source technologies such as Python, Flask, OpenCV, and the face_recognition library were selected. These tools were chosen for their compatibility, ease of use, and powerful capabilities in handling images, web development, and machine learning-based face detection.

---

3. Data Collection and Preprocessing

Facial images of known individuals were collected and stored in a predefined folder (images). These images were preprocessed using OpenCV to ensure they were in the correct RGB format and of 8-bit depth. Face encodings were extracted from these images using the face_recognition library and stored in memory for comparison during real-time recognition.

---

4. Web Application Development

A Flask-based web server was developed to serve the HTML frontend and handle image submissions. Users access the system via a web browser, which uses JavaScript and the device's webcam to capture a photo and send it to the server in base64 format.

---

5. Image Decoding and Face Recognition

The server decodes the received base64 image, processes it using OpenCV to ensure proper format, and converts it into an RGB image. Using face_recognition, the system detects and encodes the face from the submitted image. This encoding is then compared with known face encodings to determine a match.

---

6. Attendance Logging

If a match is found, the user’s name is logged into a CSV file with a timestamp. This file is generated daily and stored in a designated Attendance folder.

REFERENCES

1. Geitgey, A. (2017). face_recognition [Python library]. GitHub. Retrieved from https://github.com/ageitgey/face_recognition
2. Bradski, G. (2000). The OpenCV Library. Dr. Dobb's Journal of Software Tools. Retrieved from https://opencv.org/
3. Grinberg, M. (2018). Flask Web Development: Developing Web Applications with Python (2nd ed.). O'Reilly Media.
4. McKinney, W. (2010). Data Structures for Statistical Computing in Python. Proceedings of the 9th Python in Science Conference, 51–56.
5. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825–2830.
6. dlib.net. (n.d.). Machine Learning Library. Retrieved from http://dlib.net/
7. Zhao, W., Chellappa, R., Phillips, P. J., & Rosenfeld, A. (2003). Face recognition: A literature survey. ACM Computing Surveys (CSUR), 35(4), 399–458.
8. Convolutional Networks. IEEE Signal Processing Letters, 23(10), 1499–1503.