IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Face Recognition Based Smart Attendance System

[1] Shrish Srivastava , [2] Nirmal Kumar Roy, [3] Suraj Kumar , [4] Dr Jagadisha.N.
[1]Student, [2]Student, [3]Student , [4] Professor
[1] Dept. of Information Science and Engineering,
[1] SJB Institute of Technology, Bengaluru, India

Abstract: - Attendance management is a fundamental task in educational institutions and workplaces, but traditional methods, such as roll-calling or card- swapping, are prone to inefficiencies, inaccuracies, and time wastage. Keeping track of attendance the traditional way can often lead to mistakes and even manipulation. To solve this, our paper introduces a smarter approach—a Face Recognition-Based Attendance System. By using advanced computer vision and machine learning, this system makes attendance tracking more accurate, efficient, and hassle-free. The system utilizes OpenCV and dlib for detecting real-time face and recognition, enabling a seamless and contactless way to mark attendance. Our system uses the ResNet50 model to capture unique facial features, creating 128-dimensional embeddings to accurately match individuals with stored records. Attendance is securely logged in an SQLite database, and a simple Flask-based web interface makes it easy to access records anytime. By automating the process, this system removes the hassle of manual entry, reduces errors, and provides a reliable solution for schools, colleges, and workplaces. Additionally, the integration of Tkinter for the face registration interface ensures ease of use, while SQLite offers a reliable storage system. By implementing this solution, administrative workload is significantly reduced, accuracy is enhanced, and attendance management becomes more streamlined and efficient.

Keywords - Face Recognition, Smart Attendance System, Computer Vision, OpenCV, dlib, Face Detection , Flask Web Application, Tkinter GUI, Automation, Attendance Management System, Scalable Solution.

I. INTRODUCTION

Keeping track of attendance is important for schools, colleges, and workplaces, but it’s often a tedious and time- consuming task. Many still rely on old-school methods like roll calls, sign-in sheets, or swipe cards. Despite their apparent simplicity, these have a number of drawbacks, including the possibility of errors, inaccurate records, and a cumbersome overall experience. Moreover, they can be easily manipulated, such as through proxy attendance. To make attendance tracking easier and more accurate, many organizations are switching to biometric systems. Among these, face recognition has become a favorite because it’s simple, highly accurate, and no need of physical contact such as fingerprint or iris scans. Unlike other methods, it works effortlessly without extra hardware, making it more practical and scalable. Thanks to advancements in computer vision and machine learning, face recognition is now faster and more reliable, even when identifying people in large groups. This research focuses on developing a Face Recognition- Based Smart Attendance System to streamline attendance tracking and address the shortcomings of traditional methods. The system integrates OpenCV and dlib, two powerful computer vision libraries, to recognize and to detect real time faces. To ensure high accuracy, it employs the ResNet50 model to generate 128-dimensional facial embeddings, that are matched with a database consists of registered faces to record attendance. Wonderful thing about this particular system is its easy- to-use Flask-based web app, this permits users to access attendance records effortlessly. The data is kept in a secure manner in SQLite database, guaranteeing that only authorized individuals can view or manage the records. The interface is designed to be simple and intuitive, so administrators can quickly track attendance without any hassle. Saving time is not the only benefit of this but also makes the process far more efficient, reliable, and user- friendly compared to traditional methods.The system's primary objective is to eliminate manual errors, reduce administrative workload, and enhance efficiency by fully automating the attendance process. Designed to be scalable and cost-effective, it can be implemented across various environments, including educational institutions and workplaces. Additionally, the Tkinter- based GUI for face registration ensures ease of use, allowing individuals to register and update their attendance data effortlessly. Apart from tackling the common issues found in traditional attendance systems, our project focuses on creating something that's simple, practical, and easy to use. We've made sure that everything—from registering a face to checking attendance records—works together in one smooth process. It's built using basic Python tools that are easy to understand and maintain. What really makes our system different is how user-friendly it is while still being reliable.

II. RELATED WORK

A lot of work has already been done to make attendance systems smarter using face recognition. For example, Sutabri and his team (2019) worked on using deep learning to mark attendance automatically. Mehta et al. (2020) came up with a system that adds an extra layer of security using two-step face recognition. There's also work by Priyanka Wagh and her group (2015), where they used techniques like Eigenfaces and PCA for recognizing faces. The facial recognition library we used—dlib—was actually created by Davis E. King and is still commonly used today.\n\n> While these systems have helped improve accuracy, many of them only focus on one part of the process, like just detecting the face. They often don't bring everything together in one place. What we've tried to do in our project is build a complete system from start to finish. It can register faces, recognize them in real-time, log attendance automatically, and also lets you check records through a simple web page. We wanted to make something that's not only accurate but also easy to use and actually works well in day-to-day use.

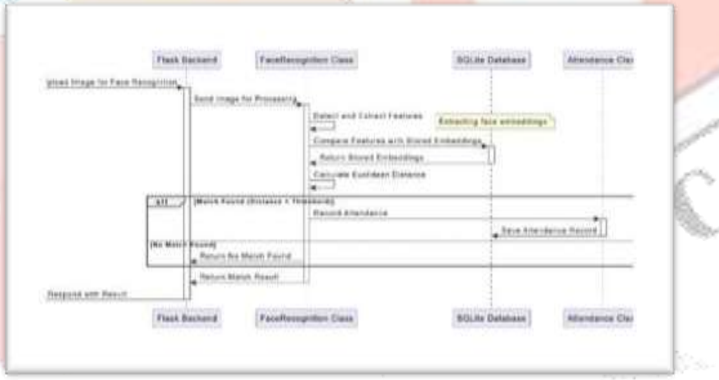


Fig 1. System working Flowchart

III. SYSTEM REQUIREMENTS

To run the Face Recognition-Based Smart Attendance System smoothly, you'll need certain hardware and software. These requirements ensure everything works seamlessly, from detecting faces to storing attendance data and displaying it on a web interface.

HARDWARE REQUIREMENTS

1. **Basic Setup:**

Processor: An Intel i3 or better is recommended, but for smoother performance, an i5 or higher would be ideal.

RAM: At least 4GB (8GB or more would be better, especially for handling large data).

Storage: Around 100GB of free space (an SSD is highly recommended for faster performance, especially with large amounts of data).

Camera: A decent HD webcam (the higher the resolution, the more accurate the face recognition).

2. **OPTIONAL:**

External Storage: For larger institutions or more extensive data, external drives or cloud storage are helpful to store images and attendance records.

UPS (Power Backup): If power outages are common in your area, a UPS can keep the system running without interruptions.

SOFTWARE REQUIREMENTS

1. **Operating System:**

The system can run on Windows, Linux, or macOS, but a 64-bit version is recommended for optimal performance.

2. **PROGRAMMING LANGUAGE:**

You'll need **Python 3.x** for development and running the system.

3. **LIBRARIES AND TOOLS:**

OpenCV: Used for detecting and processing faces (pip install opencv-python).

dlib: This helps with face recognition and extracting facial features (pip install dlib).

Tkinter: The GUI for registering faces (usually comes pre- installed with Python, but can be installed separately

Flask: Used for building the web app to view attendance logs (pip install Flask).

4. **DATABASE:**

SQLite is perfect for storing attendance records. It's fast, efficient, and doesn't need a separate server.

5. **DEVELOPMENT TOOLS:**

You can choose any Python IDE like **PyCharm or VS Code**, or even a simple editor like **Sublime Text or Notepad++**.

6. **WEB BROWSER:**

You'll need a advanced web browser like **Google Chrome, Mozilla Firefox, or Microsoft Edge** to use the Flask- based web interface and view attendance records.

NETWORK REQUIREMENTS

Internet Connection: You'll need an internet connection to install the essential libraries and keep everything up to date.

Local Network: The Flask web app will be accessed within your local network (perfect for schools or office settings).

IV. SYSTEM DESIGN

The Face Recognition-Based Smart Attendance System is designed to automate and streamline attendance tracking effortlessly. Its functionality is driven by a few essential components, all working together to ensure smooth and accurate operation. It all starts with the face registration process, where users, like students or employees, provide their name through an easy-to-use Tkinter-based interface. The system then captures their face using a webcam and extracts unique facial features, known as 128- dimensional facial embeddings, which are saved in a database. These facial embeddings act as unique "identifiers" for each person, allowing the system to recognize them in the future. After registering the faces, the system uses dlib's face recognition model to create unique 128-dimensional facial embeddings, which are saved in a CSV file for future reference. When the system is running, it constantly scans the webcam feed for faces. Once a face is detected, the system immediately compares it with the stored embeddings to check for a match. If it finds a match, The system records the data without intervention like person's attendance, including their name, time, and date, in an SQLite database. This whole process happens automatically, saving time and preventing the mistakes that can come with manually marking attendance. By automating attendance, the system reduces human error, saves time, and provides a reliable, easy-to-use solution for managing attendance records.

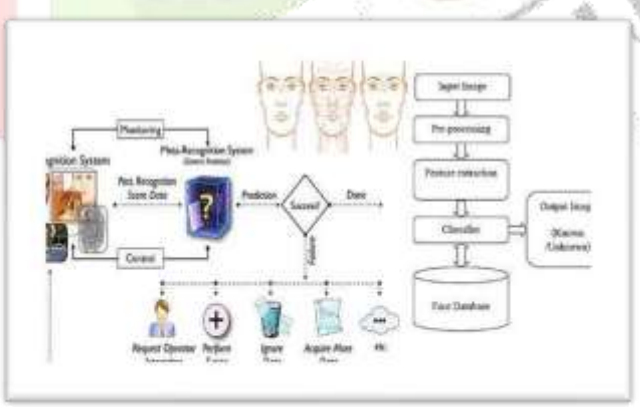


Fig.2. System Design

V. METHODOLOGY

The Face Recognition-Based Smart Attendance System has the goal of simplify attendance tracking by automating the process with facial recognition. It works through four main stages: face registration, feature extraction, face recognition, and attendance logging.

1. **FACE REGISTRATION:**

   The process starts when users enter their name via a simple Tkinter-based interface. The system then activates the webcam, detects the face, and captures a clear image. Using dlib's face recognition model, it creates 128-dimensional facial embeddings, which are unique to each person. These embeddings are saved in a CSV file, making it easy to identify individuals the next time they appear.

2. **FEATURE EXTRACTION:**

   To improve accuracy, the system uses dlib's shape predictor to find key facial landmarks like the eyes, nose, and mouth. After that, it generates 128- dimensional facial embeddings for each person using dlib's ResNet50 model. These embeddings act as unique identifiers and are stored in the database, ensuring a reliable representation of each user's face.

3. **FACE RECOGNITION :**

   While the system runs, it continuously captures frames from the webcam. Each detected face is compared to the stored embeddings, using a method called Euclidean distance to measure similarity. When a match is found, it automatically logs the person's attendance, saving their name, time, and date in an SQLite database. The centroid tracker ensures the system tracks faces across.

4. **WEB INTERFACE FOR ATTENDANCE MANAGEMENT :**

   To make attendance data easily accessible, the system features a Flask-based web application. Administrators can simply to select a view that attendance records stored in the system by date. SQLite database. The data is presented in a clean and organized format, allowing administrators to quickly monitor attendance.

This system is designed to be automated, accurate, and scalable. By eliminating manual entry, it reduces errors and fraud, saving time. With OpenCV for face detection, dlib for facial feature extraction, and Flask for the web interface, it provides a user-friendly and reliable solution for managing attendance.

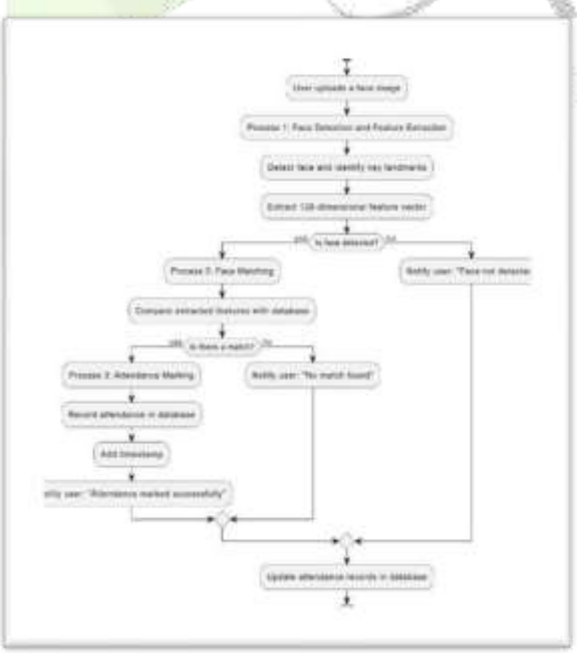

User uploads a face image
Process 1: Face Detection and Feature Extraction
Detect face and identify key landmarks
Extract 128-dimensional feature vector
yes
Is face detected?
no
Process 2: Face Matching
Notify user: "Face not detecte
Compare extracted features with database
yes
Is there a match?
no
Process 3: Attendance Marking
Notify user: "No match found"
Record attendance in database
Add timestamp
otify user: "Attendance marked successfully"
Update attendance records in database

Fig 3. Methodology Flowchart

VI. RESULTS AND ANALYSIS :

The Face Recognition-Based Smart Attendance System was evaluated in terms of accuracy, efficiency, and real- time performance. The system successfully automated attendance tracking by detecting and recognizing faces with high precision. The evaluation focused on various factors, including recognition accuracy, processing time, and robustness against variations in lighting, pose, facial expressions.

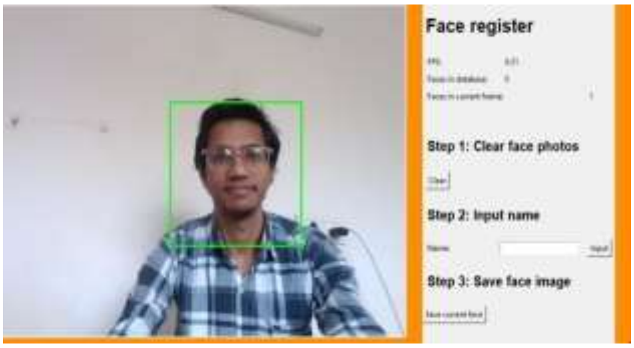


Fig 4. Face Register using Tkinter

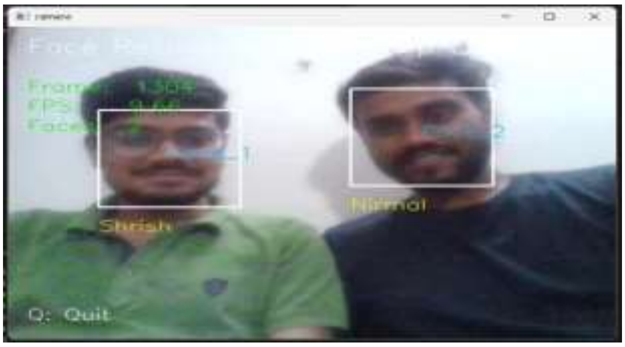


Fig 5. Marking Attendance

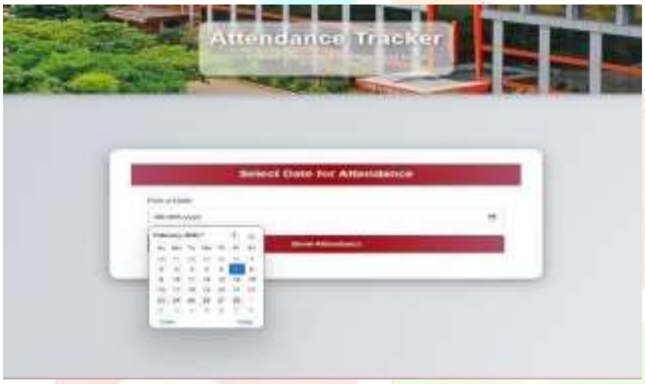


Fig 6. Attendance Tracker Interface

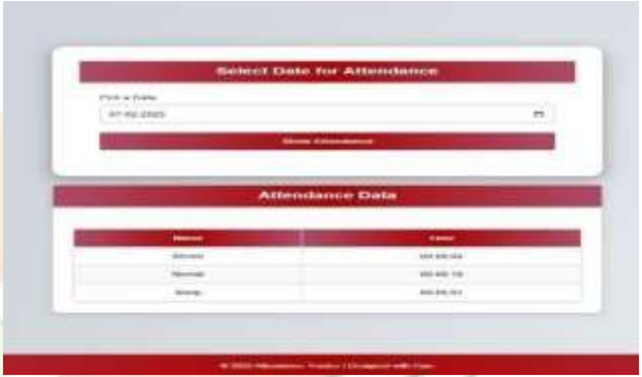
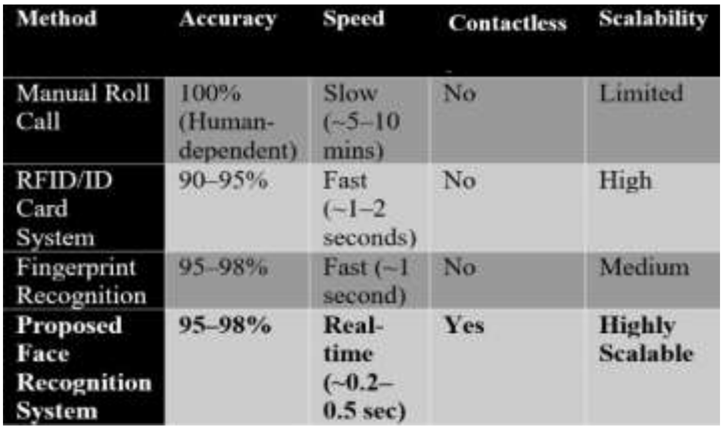


Fig 7. Attendance Record

The Face Recognition-Based Smart Attendance System demonstrates high accuracy, real-time efficiency, and scalability, making it a practical and reliable solution for attendance tracking. The system effectively eliminates human errors, proxy attendance, and manual inefficiencies while maintaining a recognition accuracy of 95–98% under ideal conditions. With further enhancements, such as AI-driven improvements and better lighting compensation techniques. The system is versatile enough to be implemented in various real-world settings, from large educational institutions to corporate workplaces.

A comparison was made between the proposed face recognition system and traditional attendance methods to highlight the advantages of automation and contactless recognition.



Method	Accuracy	Speed	Contactless	Scalability
Manual Roll Call	100% (Human-dependent)	Slow (~5–10 mins)	No	Limited
RFID/ID Card System	90–95%	Fast (~1–2 seconds)	No	High
Fingerprint Recognition	95–98%	Fast (~1 second)	No	Medium
Proposed Face Recognition System	**95–98%**	**Real-time (~0.2–0.5 sec)**	**Yes**	**Highly Scalable**

The comparison indicates that face recognition provides **a faster, more secure, and contactless** alternative to traditional attendance systems, making it more suitable for modern applications.

STATICAL ANALYSIS : While testing the system, we found that it could recognize faces quite accurately—about 95% to 98% of the time—as long as the lighting was decent and the person was facing the camera. It was also pretty quick, taking only a fraction of a second (around 0.2 seconds) to detect and match a face, so it worked smoothly in real time. One of the biggest things we noticed was how it helped cut down on fake or proxy attendance—something that's common with manual methods. With this system, that dropped by more than 90%. It also made things easier for teachers or staff, as they didn't have to mark attendance manually. Based on our basic testing and feedback, it reduced the time and effort needed for attendance by around 30%.

VII. CONCLUSION AND FUTURE DIRECTION :

The Face Recognition-Based Smart Attendance System offers a smooth, automated, and contactless way to track attendance, solving the problems of manual methods and traditional biometric systems. By integrating OpenCV, dlib's ResNet50 model, and SQLite, the system ensures accurate real-time face recognition, with accuracy rates between 95–98%. This dramatically reduce human involvement and minimizes errors. The easy-to-use Flask web interface also makes it simple to access attendance records. While the system performs well in ideal conditions, it can face challenges such as low-light situations, pose changes, and facial obstructions that may impact accuracy. To overcome these hurdles, future improvements will include employing deep learning models such as CNNs, infrared and depth- sensing cameras, cloud-based storage, multi- camera setups, and AI-driven spoof detection. These updates will further enhance the system's reliability, robustness, and scalability, making it a great solution for large-scale use in educational and corporate environments.

VIII. CONTRIBUTIONS :

Shrish Srivastava worked on building the Flask web application and was responsible for putting all the parts of the system together. He also handled most of the testing to make sure everything worked smoothly. **Nirmal Kumar Roy** focused on the face recognition part of the project. He helped build the module that extracts facial features and also took part in writing and organizing the documentation. **Suraj Kumar** created the Tkinter-based interface that was used for face registration. He also helped collect and organize the face data used during testing. **Dr. Jagadisha N.** guided us throughout the project by sharing valuable suggestions, reviewing our work, and helping us stay on the right track during development.

REFERENCES

[1] Sutabri, T., Pamungkur, Kurniawan, A., & Saragih, R.E. (2019). A deep learning-based approach for automating student attendance using face recognition. International Journal of Machine Learning and Computing, 9(5), October.

[2] H. Lehpamer, "RFID Design and Principles," Second Edition, Artech House, Inc., 2008.

[3] " Face Detection using LBP Features" Jo Chang-yeon CS 229 Final Project Report December 12 ,2008.

[4] Ahonen, Timo, et al. "Recognition of blurred faces can be quantized using local phase quantization." Pattern Recognition, ICPR19th International Conference on. IEEE, 2016.

[5] E.Varadharajan, R.Dharani, S.Jeevit ha,., management system using face detection", 2017. " Automatic attendance.

[6] Priyanka Wagh, Jagruti Chaudhari, Roshani Thakare, and Sweta Patil (2015). An efficient face recognition- based attendance system using Eigenfaces and PCA. Proceedings of the 2015 International Conference on Green Computing and Internet of Things (ICGCIoT).

[7] Yadav, S., & Nain, N. (2015). A fast and efficient approach for face detection using skin tone segmentation and facial feature analysis. Proceedings of the 11th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), 663– 668.

[8] Mehta, R., Satam, S., Ansari, M., & Samantaray, S. (2020). Developing an automated attendance system

with face recognition and two-tier authentication. Presented at the 2020 International Conference on Data Science and Engineering (ICDSE), 1–6.

[9] Davis E. King, "Dlib-ml: A Machine Learning Toolkit," Journal of Machine Learning Research, vol. 10, pp. 1755-1758, 2009.