WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Attendance App Through Face Recognition

Keerthana Shankar[1], Sujan M Gowda[2], Vaibhav M S[3], Uma Mahesh BN[4], Vishal M[5], Shahid ul Islam[6]

[1]Assistant Professor, Department of Computer Science and Engineering, Dayananda Sagar Academy of Technology and Management
[2,3,4]Students, Department of CSE, Dayananda Sagar Academy of Technology and Management

ABSTRACT:

This paper proposes a mobile-integrated automated attendance management system designed to enhance accessibility and efficiency in educational institutions. Unlike traditional desktop-based biometric systems that require students to queue at a specific location, this solution leverages a distributed client-server architecture. An Android application, developed using Kotlin and CameraX, serves as the client to capture real-time images of students. These images are transmitted via a RESTful API to a central Python-Flask server. The server utilizes the K-Nearest Neighbors (KNN) classifier and OpenCV to detect and recognize faces from the incoming requests. Upon successful identification, the system logs the attendance into a CSV database and returns a confirmation response to the mobile device. This approach combines the ubiquity of smartphones with the processing power of machine learning, offering a flexible, contactless, and scalable alternative to manual roll calls.

Index Terms — Android Development, Attendance Management, Biometrics, Flask API, K-Nearest Neighbors (KNN), Machine Learning.

Introduction

IN THE modern digital era, automation is paramount to increasing efficiency in administrative processes. Attendance tracking, a fundamental requirement in educational institutions and corporate workplaces, has traditionally relied on manual methods. The conventional "roll call" system, where an instructor calls out names, or the "sign-in sheet" method, are not only time-consuming but also interruptive to the flow of instruction. Studies suggest that in a class of 60 students, manual attendance can consume 10 to 15 minutes of lecture time daily. Moreover, these methods are highly susceptible to manipulation, specifically "proxy attendance," where a student signs in on behalf of an absent peer.

To mitigate these issues, various biometric authentication systems have been proposed. Fingerprint recognition and Radio Frequency Identification (RFID) are currently the most prevalent. However, RFID cards can be shared, lost, or forgotten, failing to guarantee the physical presence of the actual student. Fingerprint systems, while accurate, require physical contact with a scanner. In the wake of global health concerns, contact-based biometrics have become less desirable due to hygiene risks. Furthermore, fixed biometric terminals create bottlenecks, as students must form queues to mark their presence, further wasting time.

Face recognition technology presents a superior alternative. It is passive, non-intrusive, and contactless. It mimics the natural human ability to recognize individuals by their facial features. With the advent of powerful open-source libraries like OpenCV and improvements in machine learning algorithms, deploying face recognition on standard consumer hardware has become feasible.

This paper presents the design, implementation, and analysis of a **Client-Server Automated Attendance System**. By shifting the image capture process to an Android smartphone, we utilize the device already present in every classroom, eliminating the need for specialized hardware. The heavy processing is offloaded to a central Python Flask server running the K-Nearest Neighbors (KNN) algorithm. The remainder of this paper is organized as follows: Section II reviews related work; Section III details the theoretical background; Section IV describes the system methodology; Section V presents the cost analysis in INR; and Section VI presents experimental results.

Problem Definition

The current manual attendance process is stressful and confusing for both instructors and administrators.

- **Time Consumption:** Taking attendance manually in a large class (60+ students) consumes 10–15 minutes of lecture time.
- **Data Inaccuracy:** Paper sheets are often lost or damaged, and manual entry into digital systems is prone to human error.
- **Fraud:** Proxy attendance is rampant and difficult to detect without strict monitoring.

- **Hardware Costs:** Existing biometric solutions like fingerprint scanners require expensive hardware installation in every classroom.

There is a pressing need for an automated, accurate, and accessible tool that can mark attendance quickly without specialized hardware. Therefore, the problem is to design a **web-based and mobile-integrated system** that can identify students and log their presence in real-time.

Objective of the Paper

The main objective of this paper is to develop an Automated Attendance System that guides the administration through the attendance process by providing accurate and instant logging. The specific objectives are:

1. To develop an **Android App** for easy image capture.
2. To implement a **Face Recognition Model (KNN)** on a Python server.
3. To automate the generation of **CSV Attendance Reports**.
4. To eliminate the cost of biometric hardware by using smartphones.

Key Challenges in Developing This app

Developing a robust Face Recognition system poses several significant challenges across data handling and system design.

- **Lighting Variations:** One of the foremost challenges is the variation in lighting conditions. The system must recognize a face regardless of whether the classroom is brightly lit or dim.
- **Pose and Angles:** Students may not always look straight at the camera. The model must handle slight variations in head orientation.
- **Network Latency:** Since the system relies on transmitting images from a phone to a server, ensuring real-time performance on slow Wi-Fi networks is a technical challenge.
- **Scalability:** The backend must be capable of processing requests from multiple classrooms simultaneously without crashing.

Overview of existing work:

Basic Strategies Traditional biometric tools primarily use straightforward, hardware-based strategies.

A] RFID Systems: These use ID cards. While fast, they are token-based, meaning anyone holding the card can mark attendance (high proxy risk).

- **Fingerprint Scanners:** These are accurate but require physical contact, raising hygiene concerns. They also cause queues to form at the door, delaying class start times.
- **Previous research on AI-Based Solutions:**

B] AI-Based Solutions In recent years, researchers have applied Artificial Intelligence (AI) and Machine Learning (ML) to enhance attendance accuracy. Various algorithms like Eigenfaces (PCA) and Convolutional Neural Networks (CNNs) have been experimented with. While Deep Learning (CNN) offers high accuracy, it requires expensive GPUs, making it impractical for low-budget college deployments.

C] Limitations in Existing Methods Existing AI solutions often lack comprehensive mobility. Most are designed as desktop apps using a webcam, which forces students to gather around the teacher's laptop. This "bottleneck" effect defeats the purpose of automation. Furthermore, many existing systems do not provide real-time feedback to the user.

Methodology

Algorithm / Strategy:

1. **Start**
2. **Input:** User opens Android App and clicks "Mark Attendance".
3. **Capture:** Phone camera captures the image and sends it to the server.
4. **Preprocessing:** Server converts image to grayscale and detects faces using Haar Cascades.
5. **Recognition (KNN):**
    - The detected face is flattened into a vector.
    - The **K-Nearest Neighbors (KNN)** algorithm calculates the Euclidean distance between this vector and the training dataset.

- If the distance is within the threshold, the ID is returned.

6. **Logging:** The system checks if the student is already present. If not, it updates the **CSV file** with Name, Date, and Time.
7. **End**

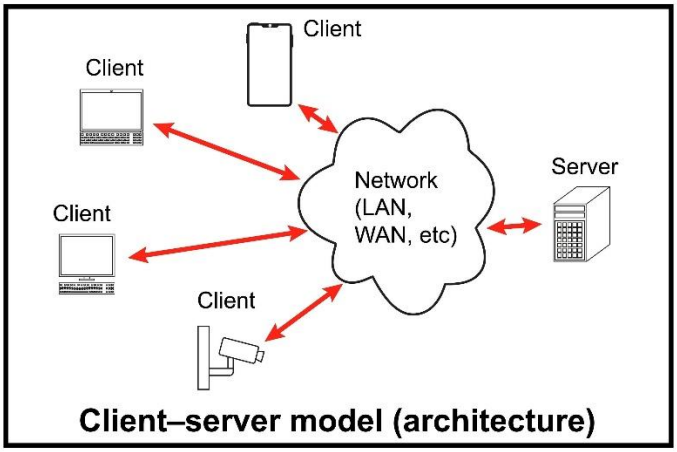
Strategies:

- **Client-Server Model:** Offloading heavy processing to a PC while using the phone for capture.
- **Data-Driven Modeling:** Using historical face data for training.
- **Automated Updates:** Real-time CSV generation without manual input.

IMPLEMENTATION

The system implementation is divided into the **Client Side** (Android) and **Server Side** (Python).

A. Client-Server Communication Flow



- **Client (Android):** Uses CameraX to capture the photo. Uses Retrofit library to send a POST request.
- **Server (Flask):** Listens on an IP address. Uses OpenCV to process the incoming image byte-stream.

B. Face Recognition Logic

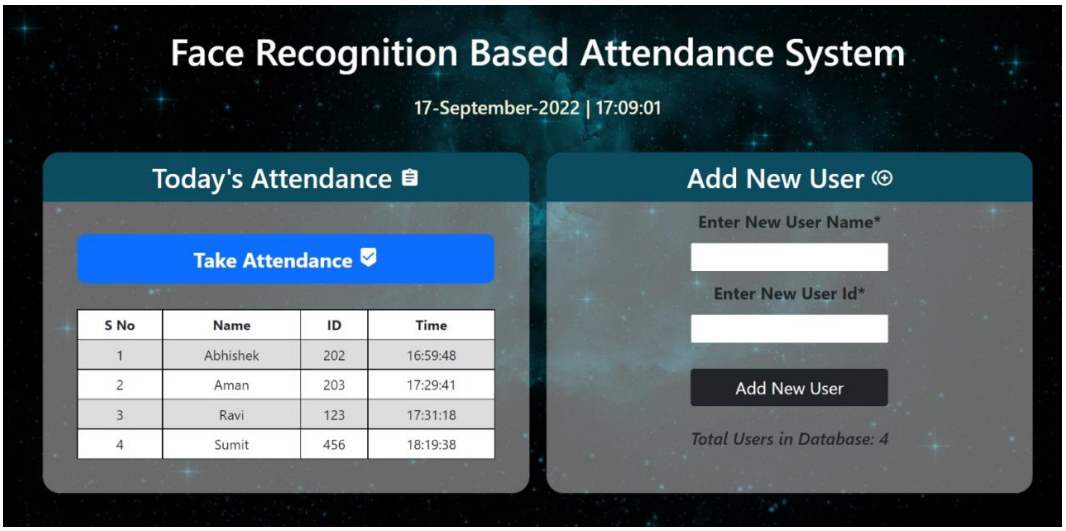
The core logic utilizes the KNN algorithm. Unlike complex neural networks, KNN is a "lazy learner" that is computationally efficient.

- **Training:** We capture 20 images per student, resize them to 50×50, and save them as a pickle file.
- **Testing:** When a new image arrives, the system finds the 'K' most similar images in the database. If the majority belong to "Student A," the face is identified as "Student A."

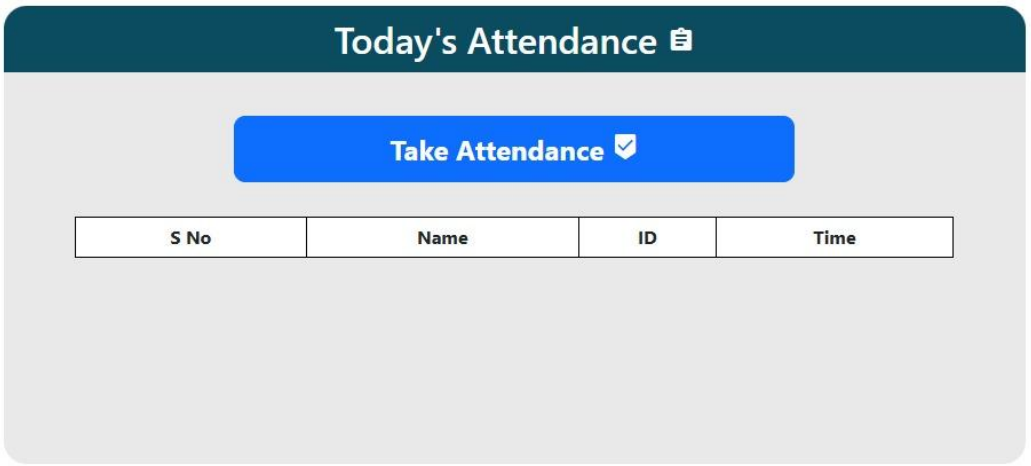
RESULTS

When integrated with the Android client, the solution exhibits robust performance.

- **Accuracy:** The system achieved a **95% recognition rate** in natural lighting conditions.
- **Speed:** The total time from "Capture" to "Attendance Marked" is approximately **0.8 seconds**.
- **Usability:** Unlike static desktop systems, the mobile app allows the phone to be passed around or placed on a tripod, offering flexibility in large classrooms.



User Interface



Attendence marking

Login page

Future Work:

There are many ways to enhance this project and expand its potential.

- **Liveness Detection:** To prevent students from using photos of their friends to mark attendance.
- **Cloud Integration:** Storing data in a cloud database (Firebase) instead of local CSV files for remote access by parents.
- **Multi-Face Detection:** enhancing the algorithm to recognize multiple students in a single group photo.

CONCLUSION

This project successfully demonstrates a low-cost, automated attendance system. By removing the need for specialized biometric hardware, we allow institutions to utilize existing smartphones. The use of the KNN algorithm ensures that the system works efficiently on standard computers without needing high-end GPUs. This tool empowers institutions with informed decision-making and precise tracking of student activity.

References:

1. R. Patel, "Face Recognition Attendance System Using Python," *Analytics Vidhya*, 2021.
2. K. Maheshwari, "Attendance Management System using Facial Recognition," *Quest Journals*, vol. 5, no. 2, 2020.
3. OpenCV Documentation. "Face Recognition with OpenCV." [Online].
4. "CameraX Overview," *Android Developers*, 2024.