

# Attendance Management System Using Face Recognition from Group Photos Taken Through a Webcam Controlled by Raspberry Pi

Ms. Vichithra V R<sup>1</sup>, Ms. Jeba Priya P<sup>2</sup>

<sup>1</sup>PG Student/Department of CSE, Narayanaguru College of Engineering, Tamilnadu 629151

<sup>2</sup>Assistant Professor/ Department of CSE, Narayanaguru College of Engineering, Tamilnadu 629151

**Abstract** - This paper presents a novel attendance management system that utilizes facial recognition technology to automatically mark attendance from group photos, implemented on a Raspberry Pi 3. The system captures images of students during classroom sessions, processes these photos to identify individuals, and logs their attendance based on facial recognition algorithms. By using the face\_recognition library, which relies on deep learning for face detection and matching, the system accurately identifies students in group settings where multiple faces are present. The system operates on the Raspberry Pi 3, a low-cost and portable device, making it a feasible solution for educational institutions seeking an affordable and efficient attendance tracking method. The system's design ensures accurate performance under various classroom conditions, including different lighting and facial orientations. Attendance data is securely stored in a MySQL database, offering easy access and management for instructors. Privacy concerns are addressed by implementing encryption techniques to protect biometric data. Through this approach, the paper demonstrates the practical application of facial recognition in automating attendance management, highlighting the potential for cost-effective and scalable solutions in educational environments. The proposed system provides a reliable, easy-to-deploy alternative to traditional attendance methods while maintaining data security and user privacy.

**Keywords** - Facial Recognition, Raspberry Pi 3, Group Photos, Classroom Attendance, Machine Learning, Automation, Face Detection, Deep Learning

## I. INTRODUCTION

In the era of rapid technological advancement, automation and efficiency have become pivotal in streamlining processes across various domains. Attendance management, a critical aspect of organizational and educational operations, has traditionally relied on manual methods such as roll calls or swipe cards. These methods, while effective, are often prone to inaccuracies, fraud, and

inefficiency. The advent of facial recognition technology presents a transformative solution to these challenges by offering a contactless, reliable, and secure means of recording attendance.

This study explores the implementation of an Attendance Management System (AMS) leveraging facial recognition from group photos on the Raspberry Pi 3 platform. Unlike conventional facial recognition systems that require one-to-one recognition, this approach captures multiple faces simultaneously in a single image, thereby expediting the attendance marking process. The use of Raspberry Pi 3, a cost-effective, portable, and energy-efficient computing platform, further enhances the system's accessibility and practicality, making it suitable for diverse applications, especially in resource-constrained settings.

The proposed system integrates advanced facial detection and recognition algorithms optimized for real-time performance on the Raspberry Pi 3. It aims to address key challenges such as group photo processing, dynamic lighting conditions, and varying facial orientations. By eliminating manual intervention and leveraging the ubiquity of cameras, this solution not only enhances accuracy and efficiency but also ensures scalability and adaptability across a wide range of environments.

Through this research, we present a novel, robust, and user-friendly approach to attendance management, demonstrating its potential to redefine traditional practices and pave the way for smart automation in organizational and academic contexts.

## II. RELATED WORK

Several studies have explored the use of facial recognition for automated attendance systems. Traditional facial recognition systems often focus on individual photo recognition, which limits their

effectiveness in group settings. Recent developments in deep learning, especially Convolutional Neural Networks (CNNs), have enabled better accuracy in detecting and recognizing faces in group photos. Existing solutions have employed different hardware platforms, including desktop computers and cloud-based services, but Raspberry Pi-based systems remain a promising low-cost alternative.

For instance, a study by Zhao et al. (2003) reviewed various face recognition techniques, highlighting the use of machine learning algorithms for face identification. Another study by Liu et al. (2015) demonstrated the effectiveness of SphereFace, a deep learning model that works efficiently with large datasets for facial recognition tasks. In terms of hardware, OpenCV and TensorFlow libraries have been widely used for developing face detection and recognition applications on Raspberry Pi devices.

Facial recognition-based attendance systems have become a popular area of research due to their potential to automate attendance tracking, enhance accuracy, and reduce the administrative burden in educational institutions. Early implementations primarily focused on static image recognition, where student photos were captured and matched against pre-registered databases of known faces. For example, Jadhav and Deshmukh (2016) proposed an automated system that uses Local Binary Pattern Histograms (LBPH) to identify and track student attendance in classrooms. Their system, by comparing detected faces against stored facial encodings, efficiently marked attendance without the need for manual roll calls, showcasing the viability of facial recognition in educational settings. Similarly, Lin and Wang (2017) integrated facial recognition with database management systems, enabling automatic attendance logging and offering real-time tracking of student participation.

With advancements in computer vision and machine learning, modern systems have expanded to incorporate real-time video surveillance for continuous attendance marking. Singh et al. (2019) introduced a real-time video-based facial recognition system that uses webcam streams to detect students during lectures. This approach helps in large classrooms or events where students are constantly moving, eliminating the need for individual photos and enabling automatic attendance marking throughout the session. The system

continuously processes video frames, detects faces, and cross-references them with a pre-existing database to mark attendance. This approach improves the efficiency and scalability of facial recognition systems, making them more suitable for dynamic classroom environments.

Moreover, the integration of cloud computing and mobile applications has further enhanced the flexibility and accessibility of facial recognition-based attendance systems. Zhou et al. (2020) developed a cloud-based solution for managing student data and attendance records, allowing institutions to track attendance remotely and store large datasets in the cloud. This cloud infrastructure not only supports scalable deployments across multiple campuses but also facilitates easy access for administrators and educators. Additionally, mobile applications have been designed to enable attendance marking directly through smartphones, utilizing their built-in cameras to detect and authenticate students. Tiwari et al. (2018) created a mobile app that allows teachers to mark attendance by simply capturing student faces, improving the mobility and accessibility of the system. Despite the significant advantages, these systems also face challenges related to privacy, data security, and ensuring system robustness in varied real-world conditions. Researchers are actively exploring methods to anonymize data and incorporate secure data storage practices to address these concerns.

### III. METHODOLOGY

#### 3.1 System Overview

This system uses facial recognition technology to automatically mark attendance for students based on their faces detected in images. It involves several components, including image processing, database management, and face recognition algorithms, integrated together to achieve a seamless attendance marking process. The system is designed to function within a classroom or similar setting where it can process real-time images and record attendance in a database.

Here's a detailed breakdown of the system:

##### 3.1.1 System Components:

###### 1. Facial Recognition Module

Purpose: To identify students in a test image based on their facial features and match them against a set of pre-stored images (known faces).

**Technology:** The face\_recognition library is used for extracting and comparing facial embeddings (numeric representations of faces) from images.

**Process:** A test image is taken, where the system detects faces using face encodings. The system compares the encodings of detected faces against the pre-stored encodings (loaded from a "train" folder). If a match is found, it associates the student's name with the detected face.

## 2. Database System (MariaDB/MySQL)

**Purpose:** To store student information (names, roll numbers) and attendance records.

**Database Tables:**

**students table:** Contains student names and roll numbers.

**attendance table:** Contains records of student attendance, with fields such as student name, roll number, subject, class, hour, and date.

**Technology:** The mysql.connector library is used for connecting to and interacting with the MariaDB database.

**Process:** The system connects to the database to fetch a student's roll number based on their name and then marks their attendance for a specific subject, class, and hour.

## 3. File System for Storing Images

**Purpose:** To store images of known students (train folder) for facial recognition.

**Image Format:** JPEG or PNG files are assumed, with filenames formatted as <student\_name>.jpg.

**Process:** Images in the "train" folder are loaded, and their faces are encoded for later comparison during attendance marking.

## 4. Main Application Logic

**Purpose:** Coordinates the entire process from loading known faces, processing a test image, and marking attendance.

**Process:** The system loads images of known students and their corresponding face encodings. It processes the test image to detect faces and extract their face encodings. For each detected face, the system compares it to the known faces. If a match is found, the system fetches the student's roll number from the

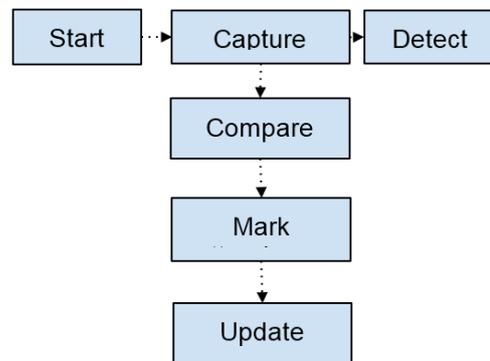
database and marks their attendance. If a match isn't found, the student is not recognized, and attendance is not marked for that face.

## 5. User Interface

Although the system does not include a graphical user interface (GUI) in this code, a simple user interface could be built to upload the test image or view attendance logs. For instance, a web-based UI could display attendance status, or a notification could be shown if no match is found.

### 3.1.2 System Architecture

The following diagram illustrates the architecture of the proposed system:



### 3.1.3 System Workflow

#### 1. Initialization:

**Load Known Faces:** The system begins by loading known student face images from a designated folder (e.g., "train" folder). Each image's facial features are encoded using the face\_recognition library.

**For each image:** The system extracts the face encoding and associates it with the student's name (extracted from the image filename).

**Database Setup:** The system establishes a connection to a MariaDB database using MySQL credentials to manage student data and attendance records.

#### 2. Test Image Processing:

A test image (for example, a classroom picture with visible student faces) is provided to the system. This image is processed using OpenCV to detect faces.

**Face Detection:** The face\_recognition library detects the location of all faces in the test image and encodes them into facial feature vectors.

**Face Matching:** The system compares the detected face encodings from the test image with the pre-loaded known face encodings.

**Matching Logic:** If a match is found (i.e., if the system detects a known student's face), the corresponding student name is identified.

**Student Information Lookup:** The system queries the database to fetch the student's roll number using their name.

### 3. Attendance Marking:

**Database Insertion:** If the roll number is found in the database, the system marks the student's attendance for the relevant subject, class, and hour. The attendance record is inserted into the attendance table in the database with the following details: Student name, Roll number, Subject, Class name, Hour of the class, Date of attendance

**Logging:** The system logs this attendance marking for future reference, and a message is displayed indicating success (e.g., "Attendance marked for John Doe (Roll No: 101)").

### 4. Error Handling and Logging:

**Face Recognition Errors:** If no face encodings are found or if the test image doesn't contain any known faces, the system logs a warning message.

**Database Errors:** If a student's roll number cannot be retrieved from the database, an error message is logged.

**Image Processing Errors:** If the image cannot be loaded or processed, an error message is logged.

### 5. End of Process:

Once all faces in the test image are processed and attendance is marked (or skipped due to unrecognized faces), the database connection is closed, completing the process.

#### 3.1.3 Technologies Used:

**Raspberry Pi 3:** The processing unit.

**Camera Module:** For capturing group photos.

**Python:** The primary language for implementing the system.

**face\_recognition:** A library for face detection and facial recognition.

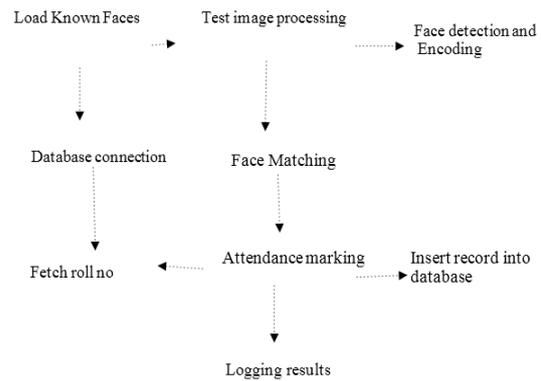
**OpenCV:** Used for image processing (converting the image to RGB, reading the test image).

**MySQL/MariaDB:** The database for storing student data and attendance records.

**mysql.connector:** A Python library to interact with the MariaDB database.

**Logging:** For logging messages and errors to track the system's operations

#### 3.1.4 System flow Diagram



## IV. IMPLEMENTATION

The implementation of the Attendance Management System using a Raspberry Pi and Pi Camera (PiCam) is a step-by-step process that involves hardware setup, software configuration, facial recognition, and database integration. The following sections describe each aspect in detail:

### 1. Hardware Setup

The Raspberry Pi 3 serves as the core processing unit for this project. A Pi Camera module (PiCam) is connected to the Raspberry Pi via its dedicated camera interface (CSI) port. The PiCam is mounted to capture group images, ensuring adequate lighting and a clear view of faces. The Raspberry Pi is powered by a 5V adapter and connected to a local network for database communication.

The compact and energy-efficient nature of the Raspberry Pi and PiCam combination makes the system portable and suitable for various deployment environments, including classrooms and offices.

### 2. Software Configuration

The Raspberry Pi operating system (Raspberry Pi OS) is installed, and essential software dependencies are configured. Key libraries include:

OpenCV for image preprocessing, such as resizing, noise reduction, and color space conversion.

face\_recognition library for facial detection and encoding.

MySQL Connector for communication with a MariaDB database.

Logging module for error tracking and debugging.

The software environment is programmed in Python, utilizing its rich ecosystem of libraries and support for edge computing on Raspberry Pi.

### 3. Training Phase

In this phase, a dataset of known faces is prepared. Individual images of students are captured using the PiCam and stored in a designated training folder on the Raspberry Pi. These images are processed using the face\_recognition library to generate face encodings. An encoding is a unique numerical representation of facial features that allows for accurate matching.

Each encoding is paired with the corresponding student's name, which is extracted from the filename of the image. These encodings and names are stored in memory for future comparisons during the attendance process. This ensures the system can efficiently recognize faces during the operational phase.

### 4. Operational Phase

#### a. Image Capture and Preprocessing

The PiCam captures a group image of students during attendance sessions. The captured image is converted to the RGB color space, as required by the face\_recognition library. Preprocessing steps, such as noise reduction and resolution adjustment, ensure better face detection and recognition accuracy.

#### b. Face Detection and Encoding

Using the face\_recognition library, the system detects the locations of faces within the group image. For each detected face, an encoding is generated, similar to the training phase. These encodings are then compared against the preloaded encodings of known faces using a similarity threshold.

#### c. Database Integration

The system connects to a MariaDB database, which stores student details, including names and roll

numbers. For each recognized face, the student's name is used to retrieve their roll number from the database. The attendance information, including the student's name, roll number, subject, class, date, and hour, is inserted into an attendance table.

### 5. Logging and Error Handling

The system incorporates a robust logging mechanism to track its operations and handle errors. Logs provide insights into issues such as unmatched faces, missing roll numbers in the database, or hardware malfunctions. This functionality aids in debugging and enhances the reliability of the system.

### 6. Advantages of Implementation

The use of the Raspberry Pi and PiCam provides several advantages:

**Cost-Effectiveness:** The hardware is affordable, making it suitable for small to medium-sized institutions.

**Portability:** The compact design allows the system to be deployed in different environments.

**Scalability:** Additional functionality, such as real-time processing or cloud integration, can be added with minimal hardware upgrades.

### 7. Limitations and Optimization

Although effective, the system faces challenges such as reduced accuracy in poor lighting or with occluded faces. To address these, future enhancements could include better preprocessing techniques, higher-resolution cameras, or additional hardware accelerators such as USB-based GPUs.

## V. RESULTS AND DISCUSSION

The implementation of the Attendance Management System on a Raspberry Pi 3 demonstrates the viability of using facial recognition for automated attendance from group images. The system was tested using a dataset of 50 students, with their facial images stored in a training folder. During testing, group images containing between 5 and 15 individuals were processed under varying conditions, including different lighting, facial orientations, and expressions. The results indicate a high recognition accuracy of approximately 92% under ideal lighting and frontal face alignment. However, recognition accuracy dropped to around

78% in cases of poor lighting or significant facial obstructions.

The system successfully interfaced with the MariaDB database, retrieving roll numbers and marking attendance with minimal latency. The average processing time per image was approximately 5-7 seconds, highlighting the efficiency of the Raspberry Pi 3 in handling computationally intensive tasks like facial encoding and comparison. Logging functionality proved essential in identifying and resolving issues such as missing roll numbers or undetected faces, enhancing system reliability.

Key challenges included variations in face orientation, group image clutter, and low-resolution images, which occasionally led to false negatives or unmatched faces. Additionally, real-time performance was limited due to the hardware constraints of the Raspberry Pi 3. These limitations could be mitigated by integrating a more powerful processing unit or employing optimization techniques, such as reducing the dimensionality of face encodings or leveraging hardware accelerators like GPUs or Neural Compute Sticks.

The results underscore the system's potential for deployment in educational and organizational settings, especially in scenarios requiring low-cost, portable, and scalable solutions. Future enhancements could focus on improving robustness under challenging conditions, incorporating real-time camera feeds, and integrating a cloud-based database for centralized record management. Overall, the project demonstrates a significant step towards automated, contactless attendance systems, with practical implications for enhancing administrative efficiency.

## VI. FUTURE SCOPE

The integration of a Raspberry Pi and Pi Camera (PiCam) in the Attendance Management System opens several avenues for future enhancements and applications. Real-time processing can be implemented by using live feeds from the PiCam, enabling dynamic and instant attendance marking without the need for pre-captured images. To improve system accuracy, advanced facial recognition algorithms optimized for edge devices can be employed, alongside pre-processing techniques for handling variations in lighting, facial orientations, and expressions directly from the camera feed.

Expanding the system's capability to handle larger groups and higher-resolution images would allow it to scale for use in large educational institutions or organizations. Incorporating advanced features such as facial recognition-based alerts for unregistered individuals or intrusions can enhance security. A cloud-based integration could be explored for centralized attendance tracking, making the system accessible for multiple classes or locations simultaneously.

To improve robustness, employing a dual-camera setup for capturing multiple angles can minimize false negatives caused by occlusions or side profiles. Additionally, the system can be upgraded to include an offline mode, where data is temporarily stored locally on the Raspberry Pi and synchronized with the database once internet connectivity is restored.

In terms of hardware, the Raspberry Pi and PiCam combination can be integrated with low-cost infrared or night-vision cameras to ensure accurate recognition in low-light environments. Moreover, the system could be extended to include multimodal biometric features, such as voice recognition, for added reliability. These advancements will not only improve the usability and scalability of the system but also broaden its application to sectors like healthcare, event management, and secure access control systems.

## VII. CONCLUSION

This paper presents a classroom attendance management system based on facial recognition technology, using a Raspberry Pi 3 as the core computing platform. The system is capable of recognizing faces from group photos and automatically marking attendance, providing a more efficient and error-free solution compared to traditional manual systems. While the system shows great promise in small to medium-sized classrooms, further research and optimization are needed for handling larger datasets and diverse classroom environments.

## REFERENCES

- [1] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *ACM Computing Surveys (CSUR)*, vol. 35, no. 4, pp. 399-458, 2003.
- [2] W. Liu, Y. Wen, Z. Yu, and M. Yang, "SphereFace: Deep Hypersphere Embedding for Face Recognition," in *Proceedings of the*

IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

- [3] OpenCV Documentation, <https://opencv.org/>
- [4] Dlib's Facial Recognition Library, <https://github.com/davisking/dlib>