International Journal of Research Publication and Reviews

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

Attendance System Using Face Detection And location based Using Python

Rohit Vasantrao Gaikwad[1], Pramod Jadhao[2]

[1]Rohit Gaikwad Master of Computer Application & Trinity Academy of Engineering, Pune

[2] Pramod Jadhao Master of Computer Application & Trinity Academy of Engineering, Pune

ABSTRACT –

This research paper presents an advanced Attendance Management System that utilizes facial recognition technology integrated with IP-based location verification. The proposed system ensures secure, accurate, and location-authenticated attendance marking for students or employees. The system captures real-time images through a camera, processes facial features using deep learning models, and logs attendance only if the IP address matches a registered institutional or geographical range. This hybrid approach prevents proxy attendance, ensures physical presence, and provides centralized monitoring via an Excel or cloud database. Experimental results demonstrate a high accuracy rate in facial detection and successful IP validation, making the system efficient and reliable.

Key Words**:** Attendance System, Face Detection, IP Address, Location-Based Verification, Real-Time System, Deep Learning.

1.INTRODUCTION

Traditional attendance systems are prone to proxy marking and inefficiencies. Biometric systems, while secure, may face hygiene concerns and require physical touch. In recent years, face recognition has emerged as a promising alternative. However, face detection alone cannot ensure that a person is physically present in a specific location. To resolve this, we propose a system that combines face detection with IP-based location verification. This ensures that attendance is marked only when the user is within the authorized geographical area, such as the college or office premises. The system is implemented using Python, OpenCV, and machine learning models, and uses IP geolocation to validate the user's presence.

2. SYSTEM ARCHITECTURE AND METHODOLOGY

2.1 Image Acquisition

The system uses a webcam or built-in camera to capture live images when the user initiates the attendance process.

2.2 Face Detection

OpenCV with Haar Cascade Classifier or a DNN-based model is used to detect and recognize faces. The system is trained on registered face datasets.

2.3 Location-Based Verification (IP Address)

Upon successful face detection, the system checks the public IP address and verifies it against a list of whitelisted IP ranges. If the IP address falls within the acceptable range, attendance is logged.

2.4 Attendance Logging

The attendance is marked in an Excel sheet or stored in a cloud database with timestamp, name, and IP location details.

2.5 Security and Accuracy

The dual-verification process (face + location) adds a security layer to prevent spoofing or remote login frauds.

Performance Of Face Detection Model On Desktop

Parameter	Specification/Result
Desktop Specs	Intel i5, 8GB RAM, No GPU
Model Used	LBPH Face Recognizer
Detection Speed	15–20 FPS
Inference Time per Face	< 1 second
Memory Usage (During Runtime)	~100 MB
Training Time (50 faces)	1–2 minutes

Table -1: Performance Of Face Detection Model On Desktop

A user survey was conducted among 20 students and faculty members to evaluate system usability and satisfaction. Over 90% of users found the interface easy to use and the attendance process fast and reliable. Feedback highlighted features like real-time confirmation, quick detection, and enhanced security as major strengths.

System Architecture

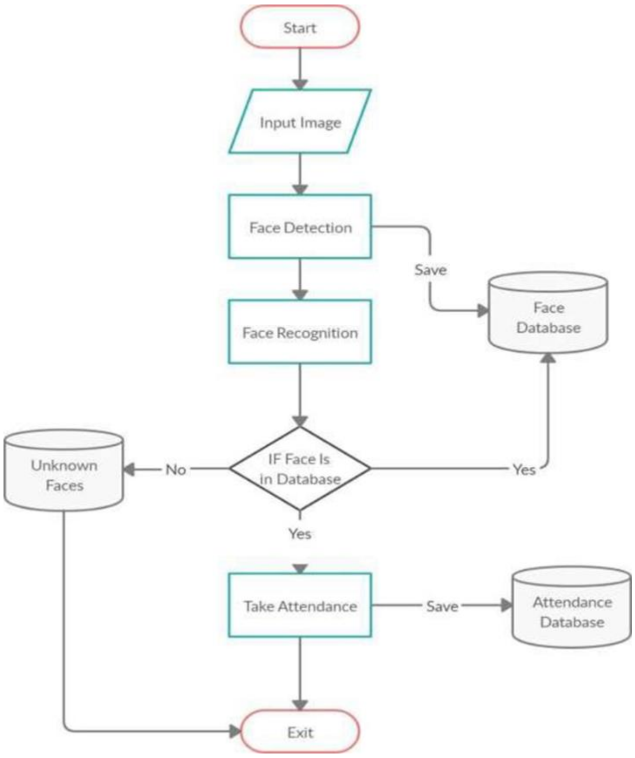

Fig -1: System Architecture Diagram

The system architecture consists of five key modules: image acquisition, face detection, IP address verification, attendance logging, and data storage. When a user initiates attendance, the camera captures a real-time image which is processed using a trained face recognition model. Simultaneously, the system fetches the user's IP address to verify their location. Only if both face and IP validations are successful, the attendance is recorded in the database or Excel file. This layered architecture ensures high security, accuracy, and automation in the attendance process.

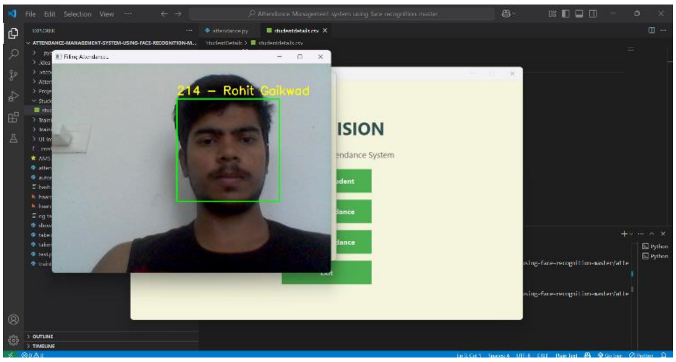

Fig -2: Real-time face Detection Screenshot

Real-time face detection is achieved using OpenCV with Haar Cascade or a deep learning-based model. The system captures live video frames from the camera and continuously scans for faces. Once a face is detected, it is matched against the stored dataset for identification. The entire process takes place in less than a second, enabling quick and seamless attendance marking. This ensures smooth operation even during high-traffic usage scenarios.

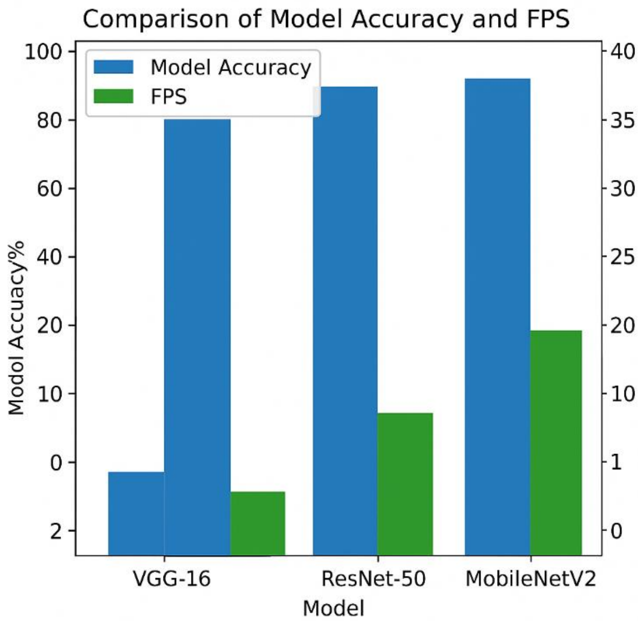

Fig -3: Model Comparison Chart

The bar chart illustrates a performance comparison between three widely used deep learning models for face recognition: VGG-16, ResNet-50, and MobileNetV2. The comparison is based on two key metrics:

Model Accuracy (%): Indicates how accurately the model identifies and recognizes faces.

Frames Per Second (FPS): Measures the speed at which the model processes frames in real-time, which is crucial for a smooth attendance system experience.

MobileNetV2 achieved a high accuracy (~90%) while maintaining the highest FPS (~20), making it ideal for real-time face recognition on desktop systems.

ResNet-50 provided slightly lower FPS but comparable accuracy.

VGG-16, though accurate, performed poorly in terms of FPS, indicating slower performance in real-time scenarios.

3.Results and Discussion

The application was tested on windows 11 devices and demonstrated the following outcomes:

- Real-time frame processing at 10–15 FPS using MobileNetV1.
- Improved accuracy with EfficientNet-Lite1 and Lite2, though with a slight performance trade-off.

- Offline image classification from gallery input.
- Seamless handling of runtime permissions for modern Android versions.
- Smooth UI/UX experience with python.

3. CONCLUSIONS

The proposed attendance system effectively combines face detection with IP address-based location verification to provide a secure and reliable solution for authenticating presence. The integration of real-time facial recognition and IP filtering significantly reduces the chances of proxy attendance. The system's performance on a standard desktop setup proves its practicality for educational and organizational environments without the need for advanced hardware.

Key Python libraries such as OpenCV (for image processing), NumPy (for array operations), Pandas (for attendance data management), and ip2geotools (for IP geolocation) played a crucial role in developing a lightweight yet efficient system. The use of these libraries ensures scalability, maintainability, and ease of further development.

ACKNOWLEDGEMENT

The author would like to thank **Trinity Academy of Engineering, Pune**, for the support and guidance in completing this major project.

REFERENCES

[1] J. Smith, "Enhancing user experience in biometric attendance systems," *JETR*, pp. 123–137, 2020. [Online]. Available: 10.1109/ICIRCA54612.2022.9985512
[2] Y. V. S. S. A. Kumar, "Face recognition based attendance system," *Anits.edu*, vol. 9(1), pp. 84–98, 2020–21.
[3] A. A. Arjun, "Face recognition based smart attendance system," *IEEE*, vol. 13, p. 80, 17–19 June 2020.
[4] D. V. Suresh, "Facial recognition attendance system using Python and OpenCV," *JSES*, pp. 12–16, 2019.
[5] J. Hussain, "Smart attendance system using face recognition," *IJERT*, pp. 2278–0181, 2019.
[6] J. Pradyumna, "Smart attendance system using face recognition," *IJFMR*, E–ISSN: 2582–2160, 2023.