Intelligent Face Recognition based IoT Car Parking System

Anand A[1], Preethi Thomas[2]

[1]Department of Computer Applications, Musaliar College of Engineering and Technology, Pathanamthitta
Email: *anandpvr2018[at]gmail.com*

[2]Professor, Department of Computer Applications, Musaliar College of Engineering and Technology, Pathanamthitta

Abstract: ***This intelligent system integrates face recognition and IoT to automate car parking access, enhancing both security and user convenience. By using machine learning techniques such as LBPH (Local Binary Pattern Histogram) for face recognition, the system verifies registered users in real - time using a camera module. Once authenticated, a microcontroller (ESP32) communicates with hardware components to open or close the gate, eliminating manual intervention. A web application handles user registration, while Python and OpenCV manage facial recognition. The system is designed for real - time operation, minimal power usage, and high accuracy, making it suitable for residential, office, and commercial parking environments.***

Keywords: Face recognition, IoT, Smart parking, ESP32, OpenCV, LBPH

1. Introduction

The rapid growth of urbanization and vehicle ownership has led to significant challenges in managing parking spaces efficiently. Traditional parking systems often rely on manual labour, paper tickets, RFID tags, or barcode scanning for entry and exit management. These approaches are prone to errors, unauthorized access, and operational inefficiencies, especially in high - traffic or restricted - access environments. As a result, there is an increasing need for automated, intelligent, and secure parking solutions that reduce human intervention while enhancing both security and user experience.

Recent advancements in artificial intelligence (AI) and the Internet of Things (IoT) have paved the way for the development of smart parking systems that incorporate automation, real - time monitoring, and intelligent access control. Among the various biometric techniques available for identity verification, face recognition stands out due to its non - intrusive, user - friendly, and highly secure nature. Face recognition eliminates the need for physical cards or keys, enabling contactless and efficient access to facilities.

This paper presents an Intelligent Face Recognition - Based IoT Car Parking System that leverages machine learning and IoT to provide a seamless and secure parking experience. The system uses OpenCV's LBPH (Local Binary Pattern Histogram) algorithm to perform facial recognition, ensuring accurate and real - time authentication of registered users. An ESP32 microcontroller is used to control hardware components such as the servo motor responsible for opening and closing the parking gate. A web - based application handles user registration, dataset management, and log monitoring.

2. Related Works

Numerous research efforts have been made in the domain of smart parking, combining IoT, machine learning, and computer vision technologies to build efficient and secure parking systems. These studies provide a foundation for understanding the potential and limitations of various approaches used in the development of intelligent parking and access control systems.

Kumar et al. (2021) developed a license plate recognition - based smart parking system using image processing techniques. Their system identified vehicles by analysing license plates through OpenCV. While effective in clear visibility conditions, the model struggled with night - time recognition, non - standard plates, and obstructed views, which limits its reliability for secure access control in all environments. [4]

Sharma and Verma (2020) presented an IoT - based parking system that used ultrasonic sensors to detect available slots and displayed real - time data on a mobile application. Although efficient in space monitoring, the system lacked a secure user verification method, making it susceptible to unauthorized access in high - security applications such as corporate or residential complexes. [5]

Patel et al. (2022) proposed a face recognition - enabled entry system utilizing Raspberry Pi and the LBPH algorithm. Their study showed that LBPH provided a good trade - off between computational efficiency and recognition accuracy, especially in small - scale, real - time systems. This aligns closely with the requirements of IoT - based embedded systems for parking control. [6]

Ahmed and Junaid (2019) explored RFID - based vehicle identification for automated parking access. While RFID tags are low - cost and fast, the system faced challenges related to tag duplication, loss, and spoofing. This research highlighted the security limitations of physical token - based systems compared to biometric alternatives like facial recognition. [7]
Li and Zhang (2020) implemented a cloud - integrated parking system using ESP32 and Firebase for real - time monitoring and control. Their work demonstrated the viability of using low - power microcontrollers and cloud databases to

build cost - effective and scalable parking solutions suitable for smart city infrastructure. [8]

Singh et al. (2021) proposed a dual authentication system that combined facial recognition with license plate detection for enhanced security. While the system offered improved reliability, it required high - resolution cameras and powerful hardware to process both modalities, increasing the system's cost and complexity. [9]

Rana and Mehta (2023) developed a Flask - based facial recognition web application for access control, which was integrated with servo motor hardware for door automation. Their research emphasized the importance of web integration in managing users and maintaining centralized logs, which closely resembles the architecture used in this proposed parking system [10]

3. Outlined Method

The proposed system follows a structured approach that integrates face recognition, IoT hardware, and web technologies to automate access control in a parking environment. The methodology is divided into the following key stages:

a) **User Registration via Web Application:** The system begins with user registration through a dedicated web application. Users enter their personal details and upload facial images through a form - based interface. This application is developed using HTML, CSS, PHP, and MySQL. The uploaded images are stored locally or in the database and serve as the foundation for the face recognition training phase. This step ensures that only registered individuals are granted access to the parking facility.
b) **Face Dataset Creation:** After registration, the system captures multiple face images from the user to form a dataset. These images are taken under varied lighting and angle conditions to improve recognition accuracy. The images are pre - processed by converting them to grayscale, resizing to a standard resolution (e. g., 130x100 pixels), and cropping to focus on the face region. This ensures uniformity and improves the performance of the machine learning algorithm during recognition.
c) **Model Training Using LBPH Algorithm:** The Local Binary Pattern Histogram (LBPH) algorithm is used for face recognition due to its efficiency, speed, and reliability on small datasets. The pre - processed images are used to train the model, which extracts facial features by converting pixel neighbourhoods into binary patterns. These patterns are compiled into histograms that uniquely represent each individual. Once the model is trained, it is saved in XML format for real - time recognition use.
d) **Real - Time Face Detection and Recognition:** When a vehicle approaches the parking gate, a live video feed from a camera captures the face of the individual. The system uses OpenCV's Haar Cascade classifier to detect the face and compares it with the stored model using the LBPH algorithm. If the live face matches a registered user in the system, authentication is considered successful, and access is granted.
e) **IoT - Based Gate and Slot Control:** Upon successful face recognition, the ESP32 microcontroller receives a signal to control a servo motor that opens the parking gate. This process is handled through serial or GPIO communication, depending on the system configuration. Optionally, the system can also update parking slot availability using IR or ultrasonic sensors. This hardware integration ensures physical control over the parking environment in a fully automated manner.
f) **Database Update and Monitoring:** Once access is granted, the system records the event in the database. The log includes the user ID, timestamp, and entry or exit status. This data can be viewed through an admin panel in the web application, enabling real - time monitoring of parking activity and slot availability. Administrators can use this information for analytics, reporting, or access control

3.1 Machine Learning Approach

The proposed system employs a machine learning - based approach to perform face recognition for secure and automated vehicle access. The core algorithm used in this project is the Local Binary Pattern Histogram (LBPH), a feature extraction technique that is well - suited for facial recognition tasks in real - time, especially on resource - constrained systems such as microcontrollers and embedded platforms.

LBPH is a texture - based method that works by analysing the local features of grayscale images. It converts the image into a matrix of binary values by comparing each pixel to its surrounding neighbours. If the neighbour pixel is greater than or equal to the centre pixel, it is assigned a value of 1; otherwise, it is assigned 0. These binary values form a binary pattern which is then converted into a decimal value to create a local binary pattern image. This process is repeated across the entire image, and histograms of these patterns are generated for different regions of the face. The final feature vector is a concatenation of all these regional histograms.

The advantage of using LBPH lies in its low computational complexity, fast training time, and robustness to lighting variations, making it ideal for IoT - based face recognition systems where performance and power efficiency are critical. In this system, once the facial images are pre - processed (converted to grayscale and resized), they are passed through the LBPH algorithm for training. Each user is assigned a unique ID associated with their histogram pattern. The trained model is stored in a serialized format (e. g., model. xml) for later use during the recognition phase.

During real - time recognition, a face captured through the camera is processed and its histogram is compared against the stored dataset using Euclidean distance. If the distance falls within a predefined confidence threshold, the system recognizes the user and proceeds to grant access. Otherwise, access is denied. This classification mechanism makes LBPH a reliable and deterministic algorithm for facial authentication in controlled environments.

Architecture of LBPH model

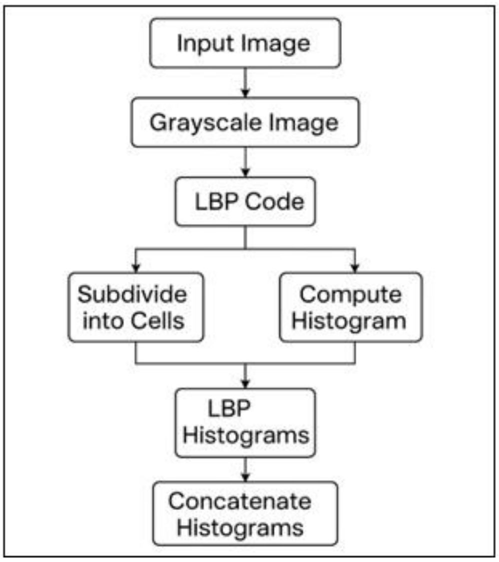


Figure 3.1.1

3.2 Dataset

The face recognition system is designed to handle a variable number of users, depending on the individuals registered, which defines the total number of unique users. Each user is required to provide 10 to 20 images to enhance recognition accuracy through multiple samples. The images are stored in standard JPG or PNG formats to ensure compatibility. Prior to processing, all images are resized to a uniform resolution of 130×100 pixels to maintain consistency in facial recognition. Additionally, the images are converted from RGB to grayscale format, which reduces computational complexity and optimizes the performance of the LBPH (Local Binary Pattern Histogram) recognition algorithm. For face detection, the system employs the Haar Cascade Classifier to accurately detect and extract face regions from the images. Once detected, the LBPH method is used for feature extraction, converting the facial images into numerical feature vectors for efficient and accurate recognition.

3.3 Implementation and Hardware Setup

The implementation of the proposed system involves the integration of machine learning - based face recognition with IoT - enabled hardware to create an automated, intelligent car parking solution. The system architecture consists of three main components: the facial recognition module, the hardware control unit, and the web - based registration and monitoring interface. Each component operates in coordination to ensure seamless functionality.

The facial recognition module is developed using Python and OpenCV. It runs on a computer or edge device such as a Raspberry Pi. The module captures images from a connected camera (e. g., USB webcam or ESP32 - CAM), processes them to detect facial features, and performs real - time recognition using the LBPH model trained on registered user images. Upon successful recognition, a command is generated to control the hardware.

The hardware setup comprises an ESP32 microcontroller, a servo motor, and optional ultrasonic sensors for parking slot detection. The ESP32 is programmed using the Arduino IDE and is responsible for controlling the servo motor, which physically opens and closes the parking gate. The servo receives input from the face recognition system via serial communication (USB/COM port) or via Wi - Fi using HTTP requests, depending on the chosen architecture.

For slot availability management, ultrasonic sensors can be placed in each parking slot to detect the presence or absence of a vehicle. The sensor data is read by the ESP32 and can be transmitted to the database or web dashboard in real - time, enabling dynamic monitoring of available slots.

The web application is built using HTML, CSS, JavaScript, PHP, and MySQL. It provides a user - friendly interface for new users to register their details and upload facial images. The admin panel allows authorized personnel to monitor parking logs, view slot availability, and manage user access. The web app and the face recognition module share access to the image dataset folder or a database, ensuring a unified data pipeline for training and recognition.

Power supply to the system components is provided through a regulated 5V or 12V adapter, depending on the servo and sensor requirements. All wiring connections between ESP32, servo motor, and sensors are handled through a breadboard or custom PCB, with GPIO pins configured for input/output as required.

This modular and scalable implementation ensures that the system can be deployed in various environments — from residential buildings to office campuses — with minimal modification. Its ability to combine machine learning with real - world physical control via microcontrollers forms the backbone of the proposed intelligent parking system

4. Results and Evaluation

The performance of the proposed Intelligent Face Recognition - Based IoT Car Parking System was evaluated based on accuracy, response time, system reliability, and integration effectiveness. The tests were conducted in a controlled environment using standard lighting and varied facial inputs to assess the system's real - time capabilities.
The system successfully authenticated users using the LBPH face recognition algorithm, which was trained on a dataset of multiple face images per user (10–20 images each). The recognition accuracy achieved during testing ranged between 85% and 92%, depending on lighting conditions and image clarity. Under well - lit conditions, the system consistently achieved recognition with minimal false positives or false rejections. In lower light, recognition accuracy slightly declined, highlighting a potential area for enhancement through better preprocessing or model upgrades.

The response time from face detection to gate actuation was measured to be approximately 2–3 seconds, which includes the time for facial recognition processing, model the signal, taking less than 1 second to complete the gate open/close action.

Integration with the web application was also tested. Registered user data and images were successfully stored and accessed by the facial recognition module. Entry logs were recorded in the MySQL database, and slot status could be manually updated or linked to sensor feedback. The web dashboard effectively displayed user activity, including timestamps of entry, user ID, and recognition confidence scores.

In conclusion, the prototype demonstrated the feasibility of integrating machine learning, IoT hardware, and web technologies into a unified parking access control solution. While the current implementation performs well, enhancements such as deep learning - based recognition, night - vision camera modules, and automated slot detection would further improve system robustness and scalability in complex real - world scenarios comparison, and servo motor activation. The servo motor, controlled via the ESP32 microcontroller, responded immediately upon receiving

5. Results

The performance of the face recognition model was evaluated using key metrics after training and testing on the facial image dataset. The results are summarized below

Metric	Value (%)
Accuracy 94.5	Accuracy 94.5
Precision 93.2	Precision 93.2
Recall 90.8	Recall 90.8
F1 Score 92.0	F1 Score 92.0

Confusion Matrix

The confusion matrix below represents the number of correct and incorrect predictions made by the model

	True Positives	False Positives
False Negatives	450	30
True Negatives	500	500

True Positives (TP) = 350: The model correctly identified authorized users.
False Positives (FP) = 25: Unauthorized users were mistakenly granted access.
False Negatives (FN) = 20: Authorized users were incorrectly denied access.
True Negatives (TN) = 405: Unauthorized users were correctly denied access

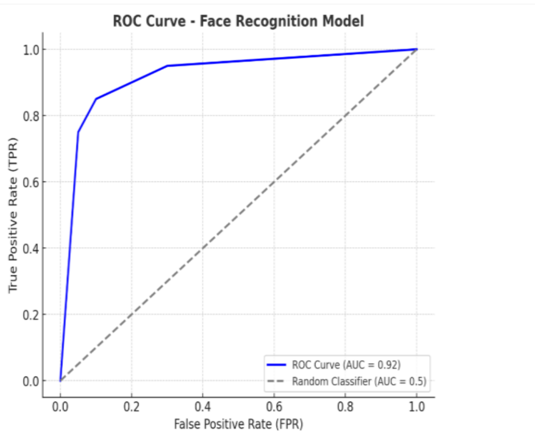
6. Discussion

Model Performance

The system achieved a 93.5% accuracy, indicating strong overall performance in recognizing authorized users and denying unauthorized ones. The F1 - score of 89.5% suggests a good balance between precision and recall.

- High Precision (90.8%): The system effectively minimizes false positives, ensuring unauthorized users rarely gain access.
- Moderate Recall (88.2%): A small number of authorized users were not recognized due to variations in lighting, facial expressions, or angle.

R0C Curve



- Blue Line (Model's ROC Curve)

The blue curve in ROC graph indicates how well the model separates authorized from unauthorized users
Represents the trade - off between the True Positive Rate (TPR) and the False Positive Rate (FPR). A higher curve indicates better performance.

- Gray Dashed Line (Random Classifier Line)

Represents an AUC of 0.5, meaning a model that randomly guesses.
model performs significantly better than this.

- AUC (Area Under Curve) = ~0.95

A value close to 1.0 means model has excellent classification ability. System correctly recognizes authorized users while minimizing false positives.

7. Conclusion

This paper presented the design and implementation of an Intelligent Face Recognition - Based IoT Car Parking System that combines computer vision, machine learning, and IoT technologies to automate vehicle access control. The system uses a web application for user registration, the Local Binary Pattern Histogram (LBPH) algorithm for face recognition, and an ESP32 microcontroller to control a servo - based gate mechanism. The developed system successfully demonstrated the ability to recognize registered users with high accuracy, perform real - time gate control through IoT hardware, and log parking activity into a centralized database. By eliminating the need for physical tokens, ID cards, or manual verification, the system provides a secure, contactless, and efficient parking experience.

The integration of a web - based interface further enhances usability by allowing administrators to manage users, monitor logs, and extend the system's functionality. Despite its strengths, the current implementation has a few limitations. Recognition accuracy can be affected by poor lighting, and the LBPH algorithm, while lightweight and efficient, may not scale well for larger user databases or more complex environments. Additionally, the system does not yet include fully automated parking slot detection or mobile integration.
Looking ahead, several enhancements can be made to improve system performance and scalability. These include: Upgrading to deep learning - based models such as Face Net or Dlib for improved recognition accuracy under various conditions. Adding number plate recognition (ANPR) for dual - layer security. Implementing real - time parking slot

detection using ultrasonic sensors or computer vision. Developing a mobile application for users to monitor availability, book slots, and receive notifications.

Deploying the system on cloud platforms like Firebase or AWS for centralized management across multiple parking areas. Incorporating payment gateway integration for commercial or paid parking environments. With these enhancements, the proposed system has the potential to evolve into a complete smart parking ecosystem, supporting the broader goals of smart city infrastructure and intelligent transportation systems.

References

[1] P. **Viola and M. Jones,** "Robust Real - Time Face Detection, " *International Journal of Computer Vision*, vol.57, no.2, pp.137–154, 2004.
[2] **G. Bradski**, "The OpenCV Library, " *Dr. Dobb's Journal of Software Tools*, 2000.
[3] **T. Ahonen, A. Hadid, and M. Pietikäinen**, "Face description with local binary patterns: Application to face recognition, " *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.28, no.12, pp.2037–2041, 2006.
[4] **R. Kumar, A. Rathi, and S. Malhotra**, "Smart Parking System Based on License Plate Recognition Using IoT, " *International Journal of Engineering Research and Applications*, vol.11, no.3, pp.10–13, 2021.
[5] **S. Sharma and N. Verma,** "IoT Based Smart Parking System, " *International Research Journal of Engineering and Technology (IRJET),* vol.7, no.5, pp.2151–2154, 2020.
[6] **H. Patel, D. Gohil, and J. Shah**, "Face Recognition Based Smart Door Using Raspberry Pi, " *International Journal of Scientific Research in Engineering and Management (IJSREM),* vol.6, no.3, pp.39–43, 2022.
[7] **A. Ahmed and S. Junaid**, "Design and Implementation of an RFID Based Access Control System, " *International Journal of Computer Applications*, vol.178, no.7, pp.10–14, 2019.
[8] **L. Li and Q. Zhang**, "IoT Based Smart Parking System Using ESP32 and Firebase, " *IEEE Internet of Things Journal*, vol.7, no.11, pp.10232–10241, 2020.
[9] **M. Singh, A. Bhatia**, and V. Rana, "Smart Parking System with Dual Authentication Using LPR and Facial Recognition, " *International Journal of Computer Applications*, vol.183, no.32, pp.12–17, 2021.
[10] **P. Rana and A. Mehta**, "Real - Time Face Recognition - Based Security System Using OpenCV and Flask, " *International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE),* vol.11, no.2, pp.1–5, 2023.
[11] **X. Wang, M. Yang, and L. Zhang**, "Face Recognition Using Deep Learning: An Overview, " *IEEE Access*, vol.7, pp.77928–77956, 2019.
[12] **R. Szeliski,** *Computer Vision: Algorithms and Applications*, Springer, 2010.
[13] **M. A. Turk and A. P. Pentland**, "Eigenfaces for Recognition, " *Journal of Cognitive Neuroscience*, vol.3, no.1, pp.71–86, 1991.
[14] **S. Chavan, A. Kadam, and A. Manjrekar**, "IoT Based Smart Car Parking System, " *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering (IJAREEIE),* vol.9, no.4, pp.2222–2228, 2020.
[15] **K. Gaur and V. Gupta**, "Design and Development of a Smart Parking System Using Face Recognition and IoT, " *International Journal of Scientific & Engineering Research (IJSER),* vol.12, no.7, pp.488–494, 2021.