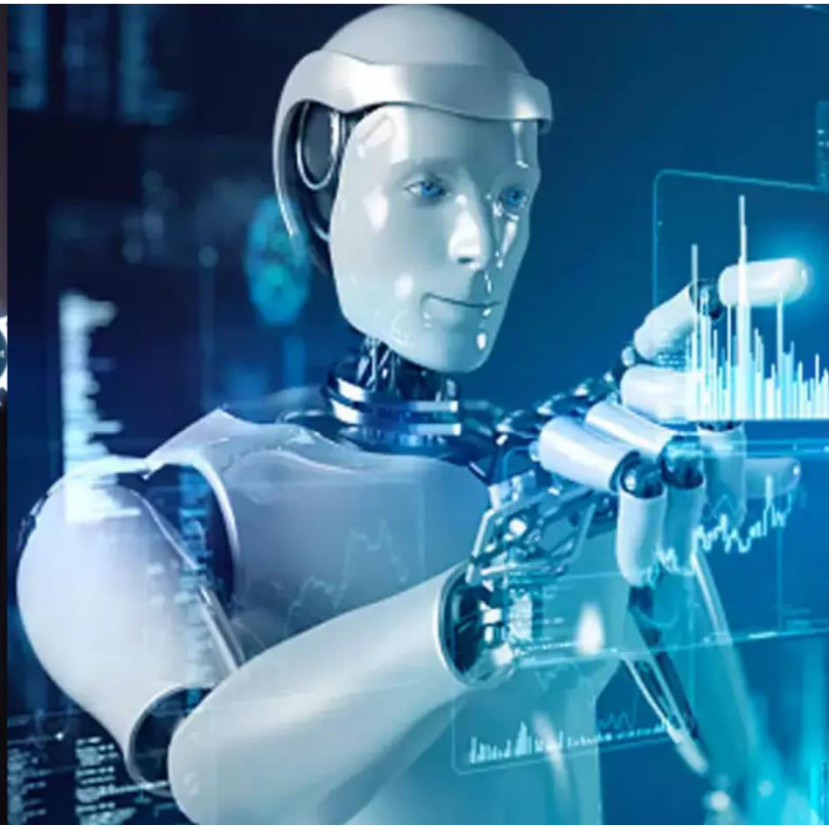
IJIRSET
International Journal of Innovative Research in Science | Engineering | Technology

ISSN(O): 2319-8753
ISSN(P): 2347-6710

International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.699

Volume 15, Issue 3, March 2026

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 3, March 2026

|DOI: 10.15680/IJIRSET.2026.1503134|

SmartPark: Automated Vehicle Parking Fare System

Bhavadharani V[1], Harithra R[2], Mercy Catherine D[3], Dr.R.Jothilakshmi[4]

U.G. Student, Department of Information Technology, R.M.D Engineering College, Kavaraipettai, Tamil Nadu, India[1]

U.G. Student, Department of Information Technology, R.M.D Engineering College, Kavaraipettai, Tamil Nadu, India[2]

U.G. Student, Department of Information Technology, R.M.D Engineering College, Kavaraipettai, Tamil Nadu, India[3]

Associate Professor, Department of Information Technology, R.M.D Engineering College, Kavaraipettai, Tamil Nadu, India[4]

ABSTRACT: Effective parking space management has become a major concern due to the exponential rise in urban car usage. In order to simplify and automate the parking procedure, this project offers a smart, automated car parking fare system that combines integrated technology, computer vision, and cloud-based data storage. The goal is to decrease operating overhead in parking lots, improve accuracy, and do away with manual intervention. The ESP32 microcontroller, which is linked to an ultrasonic sensor, a buzzer, and a 16x2 I2C LCD display, is the central component of this system. By calculating the distance to any item in front of it, the ultrasonic sensor continuously scans for approaching cars. The ESP32 uses serial communication to trigger a Python-based script on a host computer when it detects a car within 30 cm. This script activates a camera and uses EasyOCR, an optical character recognition tool based on deep learning, to recognize license plates in real time.

After a license plate is identified and verified, the system searches a MongoDB cloud database for prior entries. The system determines the parking period by comparing timestamps and computes the fare at a rate of ₹1 per minute if it finds a previous record for the car. The car is considered to be brand-new and there is no fare if no prior record is discovered. The ESP32 then receives the computed fare and license plate number and shows them on the LCD panel. When a vehicle is detected, a buzzer instantly gives audio feedback. To provide a smooth, affordable, and scalable system, this intelligent parking solution combines IoT (ESP32 and sensors), AI (OCR for number plate detection), and cloud computing (MongoDB for storage). It offers a major step toward smart city infrastructure and is perfect for usage in commercial or institutional parking spaces. Web dashboards, automatic barrier control for a completely autonomous parking system, and mobile app integration are examples of potential future improvements.

KEYWORDS: Smart Parking System, Automatic Number Plate Recognition, ESP32, Ultrasonic Sensor, EasyOCR, MongoDB.

I. INTRODUCTION

In today's fast-paced urban lifestyle, there is a high demand for clever and efficient automobile parking solutions. Manual parking methods not only take time but also necessitate substantial human intervention, which can result in errors, delays, and inefficiencies. With the rise of smart cities and advances in embedded systems, artificial intelligence (AI), and the Internet of Things (IoT), there is an increasing chance to automate and optimise traditional parking procedures.

This project introduces a Smart Vehicle Parking Fare System that makes use of an ESP32 microcontroller, an ultrasonic sensor, an LCD display, Python-based computer vision, and a cloud-based MongoDB database. The system detects incoming automobiles using a proximity sensor and reads the vehicle's number plate using OCR (Optical Character Recognition) algorithms. It calculates the parking period and fare using entry and exit timestamps entered in the database and displays the results to the user on an LCD. The system is intended to be user-friendly, cost-effective, and easily expandable for real-time applications in shopping malls, offices, colleges, and public parking lots.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 3, March 2026

|DOI: 10.15680/IJIRSET.2026.1503134|

II. PROBLEM STATEMENT

Traditional vehicular parking management systems mostly rely on human operations like ticket issuance and time recording. These methods frequently result in inaccuracies in tracking parking length, human error in fare calculation, and a lack of transparency in revenue collection. Furthermore, the absence of automated monitoring measures leads to inefficient use of parking spaces and increased operational complexity.

The lack of an intelligent system capable of automatically detecting vehicles, identifying license plate numbers, and precisely calculating parking length presents a significant barrier for modern parking facilities. As a result, there is a need for a smart and automated parking fare calculation system that can detect vehicle entry and exit, recognize number plates using image processing techniques, and calculate and display the parking fee with little human intervention.

III. OBJECTIVES

To create an automated car parking fare computation system that reduces human intervention and boosts operational effectiveness. Using an ultrasonic sensor, the system seeks to automatically identify the presence of automobiles, initiating the next processing steps. It also focuses on precisely identifying car license plates with computer vision methods combined with Python's EasyOCR.

The system uses timestamp data kept in MongoDB to determine the parking fee based on the identified vehicle entry and exit times. An LCD panel linked to the ESP32 microcontroller then shows the identified car number and the associated fare. Through a serial interface, the system also guarantees smooth real-time connection between the ESP32 and the Python-based processing unit for effective data interchange and system updates.

IV. LITERATURE SURVEY

Smart parking systems have become a growing area of research and innovation in recent years, especially with the development of Internet of Things (IoT), embedded systems, and computer vision technologies. Numerous efforts have been made to improve the efficiency of parking space monitoring and fare collection through automation.

In paper [1], Patrascu discussed the working principles of automated parking systems and explained how automation technologies can improve parking efficiency by reducing human intervention. The study highlighted the advantages of automated parking infrastructure in managing limited parking spaces and improving vehicle handling within parking facilities.

In paper [2], McDonald examined the relationship between parking systems and environmental sustainability. The study emphasized that efficient parking management can reduce traffic congestion and unnecessary fuel consumption caused by vehicles searching for parking spaces, thereby contributing to more sustainable urban transportation systems.

In paper [3], Hamelink presented a detailed guide on mechanical and automated parking systems, describing their design, structure, and operational benefits. The work focused on how automated parking solutions can optimize space utilization and minimize manual errors in traditional parking operations.

In paper [4], Du et al. provided a comprehensive review of Automatic License Plate Recognition (ALPR) techniques. The study analyzed different stages of ALPR systems, including license plate detection, character segmentation, and optical character recognition, highlighting the effectiveness of computer vision and machine learning methods in vehicle identification.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 3, March 2026

|DOI: 10.15680/IJIRSET.2026.1503134|

V. METHODOLOGY

CONCEPT OVERVIEW

The proposed system will automate vehicle parking management by combining IoT hardware, computer vision, and cloud-based data storage. To detect vehicles and recognize license plates, the system uses an ESP32 microcontroller, an ultrasonic sensor, and camera-based optical character recognition (OCR). The observed information is saved in a MongoDB database, allowing for real-time parking duration tracking and automated fare calculation. By merging embedded technologies with AI-based number plate recognition, the system improves the efficiency, accuracy, and automation of parking fee administration while reducing human involvement.

SYSTEM WORKFLOW

The system uses a systematic workflow to enable accurate vehicle detection, seamless data processing, and rapid parking fare computation.

Vehicle Detection Using Ultrasonic Sensor:

The ESP32 microcontroller continuously monitors the ultrasonic sensor located at the parking entry to detect incoming automobiles. When a vehicle approaches within a threshold distance of about 30 cm, the ultrasonic sensor identifies it and sends a trigger signal to the system. This signal starts the next processing stage by activating the camera module, which records an image of the vehicle's license plate for further recognition and processing.

License Plate Recognition (LPR) Using OCR:

When the device is activated, the camera records an image of the vehicle's license plate. The acquired image is then analyzed with EasyOCR, a deep learning-based optical character recognition model, to extract textual information from the license plate. The OCR system identifies and retrieves the plate's alphanumeric characters, as well as performing basic format checks. The recognized license plate number is then compared to the information in the MongoDB database to see if the car has previously entered the parking lot.

Parking Duration and Fare Calculation:

If a matched record is located in the database, the system obtains the prior timestamp for the vehicle. The parking duration is computed by subtracting the entry and exit timestamps. The technology automatically calculates the parking fee based on a predefined rate of ₹10 per hour. The LCD module linked to the ESP32 displays the calculated fare and vehicle number.

Data Communication and Storage:

The Python-based processing unit sends processed data, such as license plate number and fare, to the ESP32 via serial connection. The ESP32 refreshes the I2C-based LCD display with user-specific information. MongoDB stores all detection records, timestamps, and transaction details, allowing for quick retrieval and tracking.

Real-Time Alerts and Notifications:

The ESP32's associated buzzer alerts users when a vehicle is detected. The LCD module displays the car number and associated parking fee. The automated feedback mechanism improves user awareness and lowers the need for manual supervision.

IMPLEMENTATION FRAMEWORK

The system is built using embedded systems, IoT communication, and AI-based image processing approaches. The ESP32 microcontroller serves as the primary controller, handling sensor input, device connectivity, and display output. The number plate recognition is performed using a Python-based processing package that uses EasyOCR. Meanwhile, MongoDB serves as a cloud-based database, allowing for quick storing and retrieval of parking records.This integrated design enables scalability, real-time data processing, and consistent system performance, making the suggested solution appropriate for use in institutional, commercial, and smart city parking infrastructures.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 3, March 2026

|DOI: 10.15680/IJIRSET.2026.1503134|

VI. SYSTEM ARCHITECTURE

OBJECTIVE

The main objective of the proposed system is to provide a smart and automated car parking fare calculating system that reduces human interaction while increasing operational efficiency. The system uses both hardware and software technologies to identify vehicles, recognize license plates, track parking time, and display the estimated price automatically. An ultrasonic sensor attached to an ESP32 microcontroller detects the presence of a car at the parking entrance. A camera and optical character recognition (OCR) technology, notably EasyOCR, are used to capture and extract the vehicle's license plate number. The system stores and tracks vehicle entry and exit timestamps in a MongoDB database, which allows for automatic fare calculation. The calculated parking fee along with the vehicle number plate is then displayed on an LCD module connected to the ESP32, providing real-time feedback to the user.

WORKFLOW

The proposed method uses an automated workflow to ensure accurate vehicle detection, rapid data processing, and consistent fare calculation. The ultrasonic sensor continuously checks the parking entrance for the presence of a car. When the ESP32 detects an object inside the predefined distance threshold, it instructs the system to activate the camera for image capture. The camera captures an image of the approaching car, which is then processed using EasyOCR to retrieve the license plate number. The retrieved number plate is then compared to records stored in the MongoDB database.

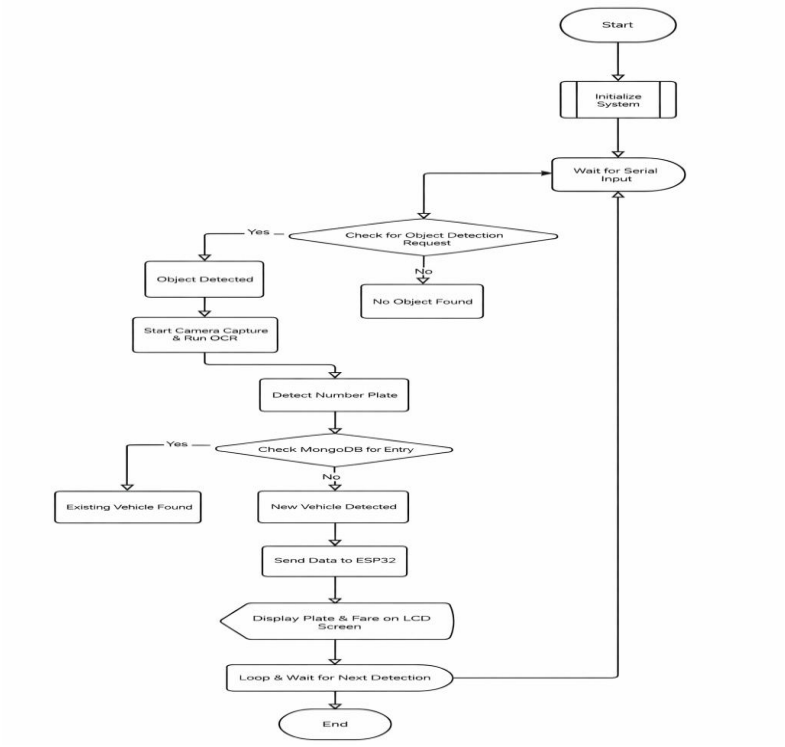


Fig 1.1 Workflow

If the plate number fails to be found in the database, the system records the vehicle as a new entry and saves the current timestamp. If the plate number already exists, the system obtains the previously stored timestamp and calculates the parking period using the difference between the entry and exit times. The system then computes the parking fare and transmits it to the ESP32. The LCD panel displays the vehicle number plate and the calculated fare. After finishing this operation, the system returns to monitoring mode and awaits the next vehicle detection.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 3, March 2026

|DOI: 10.15680/IJIRSET.2026.1503134|

MODELING

The system is designed using a mix of hardware components and software modules that work together to deliver the overall functionality of the parking automation system. The hardware components include the ESP32 microcontroller, ultrasonic sensor, camera module, buzzer, and a 16x2 I2C LCD display. These components manage vehicle detection, system control, and user input.

On the software side, Python is used for system control and data processing. EasyOCR extracts car number plates from collected photos, whereas OpenCV performs image capture and processing processes. MongoDB serves as the database for storing car records and timestamps. The Python application and the ESP32 communicate serially, ensuring that data is transferred smoothly between the processing unit and the embedded hardware components.

Hardware Components:

Component	Role
ESP32	Core controller, reads ultrasonic sensor & communicates with PC
Ultrasonic Sensor	Detects presence of vehicle
Camera	Captures image for OCR
LCD (I2C or SPI)	Displays plate number & fare

Fig 1.2 Hardware Components

Software Modules:

Module	Function
Python (PC)	Camera interface, OCR, MongoDB handling, fare calculation
EasyOCR / OpenCV	Plate detection and text extraction
MongoDB	Entry/exit record storage
Serial Comm	Python ↔ ESP32 data exchange
Arduino Code	Handle display & input from Python via Serial

Fig 1.3 Software Modules

ARCHITECTURE

To guarantee effective system operation, the system architecture is built in a tiered structure with layers for detection, processing, and presentation. The ESP32 microcontroller and the ultrasonic sensor, which continuously check the entrance area for the presence of vehicles, are part of the detecting layer. The ESP32 communicates with the Python-based processing system via the serial interface when it detects a car within the designated range. The processing layer is in charge of turning on the camera, taking a picture of the car, and using EasyOCR to recognize the license plate. To ascertain whether the car is entering or leaving the parking lot, the retrieved license plate number is compared to the MongoDB database.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 3, March 2026

|DOI: 10.15680/IJIRSET.2026.1503134|

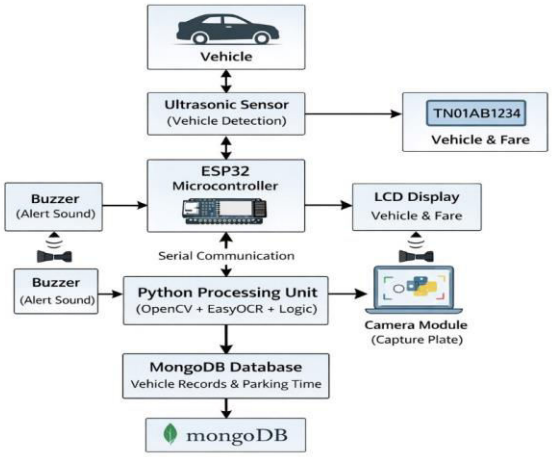


Fig 1.4 Block Diagram

If the vehicle is new, the system saves the license plate number and the current timestamp. If the vehicle is already in the database, the system calculates the parking length and charges the appropriate parking fee. Finally, the display layer, which comprises of the ESP32 and the LCD module, gets the processed data via serial communication. The ESP32 displays the vehicle number and calculated fare on the LCD panel, providing the user with real-time information while also completing the automatic parking fare calculation procedure.

VII. EXPERIMENTAL RESULT

SYSTEM OVERVIEW

The system includes an ESP32 microcontroller, an ultrasonic sensor, a buzzer, an LCD display, a camera, and a Python script that uses EasyOCR and MongoDB to automate vehicle identification, number plate recognition, and parking fare calculation. The ESP32 detects vehicle presence and connects with a Python script on a computer, which handles real-time number plate detection, fare computation, and database administration.

FUNCTIONAL RESULTS

1. Object Detection with Ultrasonic Sensor (ESP32):

The ultrasonic sensor detects the presence of things within 30 cm. When a vehicle is spotted, the system sounds a buzzer to indicate success and transmits a "OBJECT_DETECTED" signal to the Python-based processing module via serial connection. This signal initiates the next stage of the system, which begins the number plate identification procedure.

2. Camera Activation and Number Plate Recognition (Python with EasyOCR):

When the ESP32 microcontroller triggers the Python script, the system's camera is activated, and the image acquisition process begins. The collected frames are analyzed in real time using EasyOCR to detect and recognize vehicle number plates, with a confidence level set above 40% to assure accurate detection. The identified plate numbers are then checked using a regular expression (regex) pattern to ensure the format is proper, with the extracted text including roughly 5-10 alphanumeric characters.

3. MongoDB Integration for Parking History:

If the identified vehicle number plate already exists in the MongoDB database, the system obtains the previously recorded timestamp for that entry. The parking duration is calculated by comparing the current timestamp to the stored entry time. Parking fees are determined at a predefined rate of ₹10 per hour, which equates to ₹1 each 6 minutes of parking.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 3, March 2026

|DOI: 10.15680/IJIRSET.2026.1503134|

4. Fare Calculation and Display (ESP32 + LCD):
The calculated parking fee, together with the identified vehicle number plate, is relayed back to the ESP32 microcontroller. The ESP32 then parses the received message and processes the data, displaying the vehicle number and computed fare on the linked LCD screen. This allows the system to give the user rapid feedback on the vehicle identification and parking charge.

5. ILLUSTRATIVE OUTPUT
To demonstrate the working of the proposed system, consider a vehicle with the license plate number TN01AB1234. When the vehicle exits the parking area, the system retrieves the previously recorded entry time from the database, which is 2026-02-30 12:00:00, and compares it with the current exit time 2026-02-30 13:15:00. Based on these timestamps, the total parking duration is calculated as 75 minutes. According to the predefined parking rate of ₹10 per hour (₹1 per 6 minutes), the system computes the parking fee as ₹12.5. The recognized vehicle number along with the calculated fare is then transmitted to the ESP32 and displayed on the LCD screen, providing immediate feedback to the user.

LCD Output:

Fig 1.4 Sample Output

VIII. PERFORMANCE DISCUSSIONS

Accuracy: Easy OCR demonstrated good accuracy in recognizing clear, well-lit number plates with a confidence level > 0.40. However, accuracy drops in low-light or blurry frames. The regex filter helps reduce false positives.

Responsiveness: The system promptly responds to object detection with minimal delay (under 2 seconds), making it effective for real-time applications.

Efficiency: MongoDB efficiently stores and retrieves vehicle records, allowing quick fare computation even as the database grows.

Reliability: The ESP32's continuous loop ensures the system remains active and ready for new detections. The combination of hardware and software offers a reliable and automated fare management system.

IX. FUTURE SCOPE

The proposed automated vehicle parking fare system can be improved in a variety of ways, including performance, scalability, and usability. One potential improvement is to improve number plate recognition accuracy by incorporating advanced deep learning models like YOLO or OpenALPR, or by integrating hybrid approaches that combine Haar Cascade detection with EasyOCR to achieve better detection under varying lighting and angle conditions.

Another prospective extension is the integration of cloud-based platforms like AWS or Firebase, which would allow for remote monitoring, data storage, and analytics, as well as real-time notifications for system administrators. A mobile application or web-based dashboard might also be created to allow vehicle owners and parking administrators to view parking history, fare data, and vehicle tracking information in real time.

The system can easily be upgraded to include several cameras at various entry and departure points, allowing for centralized monitoring of big parking facilities. Furthermore, adding digital payment systems such as UPI or PayTM will enable users to make quick parking payments directly through the system, increasing convenience and efficiency.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 3, March 2026

|DOI: 10.15680/IJIRSET.2026.1503134|

Additional enhancements could incorporate RFID or QR-based entry mechanisms to accommodate vehicles without clearly visible number plates or to facilitate prepaid parking services. To improve system safety, security features such as tamper detection, intrusion warnings, and better surveillance can be introduced employing additional sensors and artificial intelligence. Finally, machine learning techniques might be used to examine past parking data and dynamically alter parking rates based on demand, length, or peak usage times.

X. CONCLUSION

This constructed smart parking system efficiently incorporates an ESP32 microcontroller, ultrasonic sensor, LCD display, Python-based OCR detection, and a MongoDB database to automate vehicle entrance logging and fare calculation based on license plate recognition. The device detects vehicle presence using an ultrasonic sensor and triggers the camera to collect the number plate, which is then processed with EasyOCR. The identified number plate is cross-referenced with previous entries in the MongoDB database to compute parking time and charge. The results are shown in real time on an LCD and recorded for future reference.

This solution greatly eliminates manual involvement, improves parking system automation, and offers an efficient and cost-effective method for managing vehicle entries and payment. It highlights the practical benefits of merging IoT and AI technology in smart infrastructure.

REFERENCES

[1] D. Patrascu, "How Automated Parking Systems Work," Autoevolution, 2010.
[2] S. McDonald, "Cars, Parking and Sustainability," Transportation Research Forum, 2012.
[3] L. J. Hamelink, The Mechanical Parking Guide, CreateSpace Independent Publishing Platform, 2011.
[4] S. Du, M. Ibrahim, M. Shehata, and W. Badawy, "Automatic license plate recognition (ALPR): A state-of-the-art review," IEEE Transactions on Circuits and Systems for Video Technology, vol. 23, no. 2, pp. 311–325, 2013.
[5] Z. Tian, H. Pan, and Y. Zhang, "Automatic license plate recognition using deep convolutional neural networks," IEEE Access, vol. 8, pp. 195000–195011, 2020.
[6] M. Silva and J. Jung, "Real-time automatic license plate recognition through deep learning techniques," Journal of Intelligent Transportation Systems, vol. 25, no. 3, pp. 1–12, 2021.
[7] A. K. Jain and B. Bhanu, "Automatic vehicle identification using license plate recognition," Pattern Recognition Letters, vol. 33, no. 12, pp. 1581–1589, 2018.
[8] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2001.
[9] A. Rosebrock, "Automatic license plate recognition with Python and OpenCV," PyImageSearch, 2020.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details