

<https://africanjournalofbiomedicalresearch.com/index.php/AJBR>

Afr. J. Biomed. Res. Vol. 27(3) (December 2024); 7536-7544

Research Article

Parking Allocation System - Using YOLO, EasyOCR, and Video Analytics for Real-Time Slot Detection and Number Plate detection for Vehicle Security

Sankalp Khilari^{1*}, Anish Ghanwat², Shashank Patil³, Vishwajeet Gaikwad⁴, Dr. Pramod Ganjewar⁵

^{1,2,3,4,5*}MIT Academy of Engineering, Alandi, Pune, Maharashtra, India, 412105. ,
sankalp.khilari@mitaoe.ac.in, anish.ghanwat@mitaoe.ac.in, shashank.patil@mitaoe.ac.in,
vishwajeet.gaikwad@mitaoe.ac.in, pdganjewar@mitaoe.ac.in

***Corresponding Author :** Sankalp Khilari

***E mail :** sankalp.khilari@mitaoe.ac.in

Abstract

With the increasing number of vehicles in urban areas, managing parking has become a major challenge, especially in spaces where finding an available spot is difficult and time-consuming. In addition to the inconvenience of locating a parking slot, the entry of unauthorized vehicles presents significant security concerns. Unauthorized access can lead to security breaches, theft, or other risks, making it essential to monitor vehicles effectively. Current parking systems often lack real-time updates on available spaces, adding to the frustration of users who waste time searching for parking. This paper presents an AI-driven parking management system that addresses both security and convenience. By using video analytics, the system provides real-time vehicle number plate recognition to prevent unauthorized entry while also detecting vacant parking slots to offer users live updates. This solution improves both safety and efficiency, contributing to a smarter and more secure urban environment.

Keywords: Parking Management, AI-driven Systems, Video Analytics, Vehicle Security, Urban Mobility

***Authors for correspondence: E-mail Id:** Sankalp.khilari@mitaoe.ac.in

Received: 02/12./2024

Accepted: 11/12/2024

DOI: <https://doi.org/10.53555/AJBR.v27i4S.5041>

© 2024 The Author(s).

This article has been published under the terms of Creative Commons Attribution-Noncommercial 4.0 International License (CC BY-NC 4.0), which permits noncommercial unrestricted use, distribution, and reproduction in any medium, provided that the following statement is provided. "This article has been published in the African Journal of Biomedical Research"

1 Introduction

As urbanization accelerates, the number of vehicles in cities continues to rise, creating significant challenges for parking management. Limited parking availability, traffic congestion, and inadequate real-time information lead to inefficiencies that frustrate drivers. This situation not only wastes time but also contributes to environmental pollution through increased fuel consumption during the search for parking spaces. Additionally, the prevalence of unauthorized vehicles in restricted areas raises security concerns, posing risks to

both safety and property.

Current parking management systems (PMS) often rely on manual processes, which are insufficient for addressing the dynamic nature of urban parking. These systems typically lack the capability to provide real-time updates on parking slot availability, forcing drivers to waste time circling facilities in search of open spaces. This inefficiency exacerbates traffic congestion, particularly during peak periods. Moreover, traditional PMS often do not include integrated security features, making it challenging to prevent unauthorized access.

As a result, maintaining both safety and operational efficiency in parking management remains a significant hurdle.

To address these challenges, this paper offers an AI-powered parking management system that leverages videoanalytics for enhanced security and convenience. By integrating real-time vehicle number plate recognition and employing advanced object detection algorithms, the proposed solution aims to streamline parking operations, reduce traffic congestion, and improve overall user experience. This innovative approach not only secures parking facilities but also provides instant updates on slot availability, contributing to the vision of smarter, more efficient urban environments.

2 Related Work

Several researchers have explored innovative solutions to enhance parking management using advanced technologies. B.V. Chowdary et al. [1] proposed a smart parking system that employs computer vision and image processing techniques to assess parking space occupancy. By analyzing aerial images captured by cameras, the system can determine the availability of parking spots, significantly reducing the time spent searching for parking. This approach minimizes the need for complex hardware and deep learning algorithms, thereby lowering implementation costs. The system primarily targets outdoor parking facilities and aims to improve urban quality of life by decreasing carbon emissions and traffic congestion.

Pallavi H. P. et al. [2] introduced a Vehicle Parking Management System (VPMS) designed to optimize parking space utilization and enhance traffic flow in urban areas. The VPMS allows users to reserve parking spaces in advance and provides real-time information on parking availability. Notably, the system includes features such as automated slot allocation and a user-friendly web interface for reservations. By leveraging data analytics and sensor integration, this system seeks to minimize search times and improve overall parking efficiency.

Marcia Rajan et al. [3] presented an Automated Car Parking System that utilizes Optical Character Recognition (OCR) for detecting available parking slots. Their approach incorporates image processing techniques to identify parking spaces from real-time camera footage. The system allows users to pre-book parking slots via a mobile application, enhancing convenience and reducing congestion. It also employs priority-based slot allocation and RFID-based user authentication, providing a cost-effective alternative to traditional sensor-based parking systems. Hemalatha V. et al. [4] developed a Vehicle Parking System leveraging License Plate Recognition combined with OCR. This system automates parking management by capturing images of vehicle license plates at entry and exit points, subsequently comparing them against a database for authorization. Features such as real-time occupancy monitoring and automated space

management improve efficiency and reduce the time spent searching for available parking.

In another approach, Priyanshu Singh and Divya K. [5] proposed a Parking Management System using OpenCV that integrates background removal algorithms to enhance vehicle detection accuracy. The system utilizes Single Shot Detection (SSD) and MobileNets for real-time monitoring of parking space occupancy. This framework balances performance and accuracy, addressing the challenges of increasing traffic congestion in smart cities. The authors emphasize the importance of deep learning techniques in improving detection rates without sacrificing speed.

Vemula Krishna Chaitanya et al. [6] introduced a Smart Parking Management System that emphasizes security and efficiency in public parking areas. The system features automated number plate detection and a parking guidance display to direct users to available slots. Real-time monitoring and automated vehicle logging enhance the management of parking spaces, thereby reducing manual intervention and improving user experience.

Maria Waqas et al. [7] proposed a Smart Vehicle Parking Management System utilizing image processing to detect vacant parking slots efficiently. Their approach employs computer vision algorithms to maximize space utilization and provides real-time tracking of parked vehicles. The system includes dual interfaces for users and administrators, facilitating effective management of parking behavior.

Lastly, J. Cynthia et al. [8] developed an IoT-based Smart Parking Management System that integrates with a mobile application for user convenience. This system offers features such as RFID-based user authentication, scheduling algorithms for identifying the nearest available slots, and real-time occupancy tracking using IR sensors. By combining IoT technologies with data analytics, this system aims to alleviate parking challenges in urban settings, ultimately enhancing the parking experience for users.

P. Anishiya and Prof. S. Mary Joans [20] developed a Vehicle Number Plate Recognition (VNPR) system tailored for Indian cars, focusing on Tamil Nadu's unique number plates. The system employs morphological operations and edge detection for localization and Optical Character Recognition (OCR) for character recognition, achieving a 96.8% success rate with the Canny edge detector. It addresses challenges like noise and varying lighting conditions, ensuring accurate detection across diverse environments. Unlike previous methods, this system is independent of plate size, color, and angle, enhancing its robustness and applicability in real-world scenarios.

Teena Rose Mathew and Paulin Paul [21] developed an Automatic Number Plate Recognition (ANPR) system using OpenCV and machine learning, aimed at enhancing vehicle identification in security and traffic management. The system employs various image processing

techniques, including grayscale transformation, bilateral filtering, and edge detection, achieving 91.5% accuracy, 91.1% precision, and 91.8% recall. The methodology involves capturing high-resolution images, preprocessing to reduce background noise, and utilizing template matching and Optical Character Recognition (OCR) for text detection. This ANPR technology is applicable in law enforcement, toll collection, and vehicle tracking, demonstrating its effectiveness in real-world scenarios.

Sami-ul Islam Nahid, Md. Fahim Iqbal, Shoikot Khan, and Safiqul Islam Soikat [15] presented an "Image Processing-Based Smart Parking Management System" as their undergraduate capstone project at the American International University - Bangladesh. Supervised by Dr. Mohammad Nasir Uddin, the project aims to enhance parking management through advanced image processing techniques. The system is designed to optimize parking space utilization and improve user experience by automating the detection and management of parking slots. This project contributes to the growing need for efficient parking solutions in urban environments, addressing challenges related to traffic congestion and parking availability.

P.M.D.S. Amarasekiriya, M.P.P.L. Peiris, and H.M.D.S. Herath [19] presented a study on the "Implementation of Smart Parking System Using Image Processing" at the SLIIT International Conference on Engineering and Technology. The research addresses the growing demand for efficient parking solutions due to the increasing number of vehicles in Sri Lanka. The proposed system utilizes video footage and the YOLO v3 object detection algorithm to accurately identify parking slots and parked vehicles, achieving an average performance of 88.01%. The study highlights the limitations of traditional sensor-based methods and aims to provide a cost-effective solution for real-time parking space detection, particularly under varying weather conditions.

Yu-Hsn Liu, Kok-Leong Ong, Vincent C.S. Lee, and Yi-Ping Phoebe Chen [16] presented a system for detecting parked vehicles in parking complexes using multiple image streams captured through IP-connected devices. Their approach significantly improves detection speed compared to traditional object detection and machine learning methods, achieving comparable accuracy without the need for extensive training. By combining psychological insights from human detection with an algorithm that mimics SVM learning, the system effectively processes multiple image streams, resulting in faster detection. The study highlights the potential for a cost-effective solution that enhances parking management and reduces driver frustration in busy parking environments.

Siddharth Sircar, Shivam Alok, and Pratima Sarkar [18] developed a vehicle number plate detection system that efficiently detects number plates and extracts characters using image processing techniques. The system aims to automate vehicle identification to improve traffic

management and enforcement. Key objectives include achieving over 95% accuracy in character recognition and minimizing processing time compared to traditional methods. The methodology involves capturing images, converting them to grayscale, applying noise removal and dilation, and utilizing edge processing for character extraction. The proposed system addresses common issues in existing methods, such as poor detection and low accuracy, making it a promising solution for automated vehicle identification.

Lun-Chi Chen, Ruey-Kai Sheu, Wen-Yi Peng, Jyh-Horng Wu, and Chien-Hao Tseng [17] developed a video-based parking occupancy detection system aimed at enhancing smart control systems for street lighting. Utilizing the Jetson TX2 and YOLO v3 based on MobileNet v2, the system accurately identifies parking occupancy while controlling streetlights to assist in detection, thereby reducing costs and adapting to various weather conditions. The study highlights the limitations of traditional sensor-based methods, which are often costly and impractical for outdoor environments. The proposed method demonstrated stable performance in detecting parking occupancy, contributing to improved urban planning and traffic management by reducing the time drivers spend searching for available parking spaces.

Overall, these studies highlight the potential of using computer vision, machine learning, and IoT technologies in addressing the challenges associated with urban parking management.

3 Proposed Methodology

This section presents the methodology for the smart parking management system, integrating advanced technologies for automated vehicle detection, license plate recognition, and real-time parking slot monitoring. The system optimizes parking operations through automation, and its performance is analyzed based on computational complexity and processing efficiency.

3.1 Vehicle Detection and Image Capture

The vehicle detection process begins as soon as a vehicle approaches the parking lot. OpenCV is used to analyze the video feed from the camera for detecting vehicles. The motion detection algorithm used is based on background subtraction, followed by contour detection to differentiate vehicles from the background. The complexity of motion detection can be expressed as:

$$T_{\text{motion}} = O(n \times m) \quad (1)$$

where n and m represent the dimensions of the frame being analyzed. This complexity arises because the system must analyze each pixel of the image to detect motion. Once motion is detected, the license plate region is identified, and a frame is captured for further processing.

3.2 Image Preprocessing for OCR

To prepare the captured image for OCR, the following steps are performed: grayscale conversion, noise reduction, and edge detection.

Grayscale Conversion: The complexity of grayscale conversion is proportional to the number of pixels in the image. For an image of size $n \times m$, this step has a complexity of: $T_{\text{gray}} = O(n \times m)$ (2)

Noise Reduction: Applying Gaussian blur to reduce noise has a complexity of $O(k^2 nm)$, where k is the size of the kernel used in the Gaussian filter.

Edge Detection: Edge detection using the Canny algorithm has a complexity of $O(nm)$, as it operates over each pixel to find edges. After edge detection, the license plate's Region of Interest (ROI) is extracted. The combined preprocessing complexity can be approximated as:

$$T_{\text{preprocess}} = O(n \times m) + O(k^2 \times n \times m) + O(n \times m) \quad (3)$$

which simplifies to $O(k^2 \times n \times m)$, dominated by the noise reduction step.

3.3 License Plate Recognition using EasyOCR

The EasyOCR library is employed for license plate character recognition. This process involves scanning the preprocessed grayscale image and identifying the alphanumeric characters on the license plate. OCR complexity is typically dominated by the neural network inference time, which can be expressed as:

$$T_{\text{ocr}} = O(d \times w \times h) \quad (4)$$

where d is the depth (number of layers) of the neural network, and w h represents the dimensions of the input image. Once the characters are recognized, the license plate number is stored in a database with metadata, such as the entry time of the vehicle.

3.4 Parking Slot Detection using YOLOv8

For parking slot detection, YOLOv8 is used to process the live video feed. YOLOv8 operates by dividing the input image into a grid and predicting bounding boxes for objects (in this case, parking slots) at each grid cell. The complexity of the YOLOv8 detection process is:

$$T_{\text{YOLO}} = O(G \times B \times C) \quad (5)$$

where G is the number of grid cells, B is the number of bounding boxes predicted per cell, and C is the number of classes (vacant or occupied). Since YOLOv8 processes images in real-time, the computational efficiency of the system allows for quick updates on parking slot availability.

3.5 User Interface and Real-Time Display

The user interface, built using Flask, provides real-time updates to the users. The parking lot layout is displayed with color-coded markers for available and occupied slots. The data from the parking slot detection model is continuously updated, ensuring users receive real-time

notifications. The computational complexity of the user interface updates is minimal compared to the object detection algorithms, as it primarily involves rendering and updating UI components.

3.6 System Integration and Analysis

The integration of OpenCV, EasyOCR, and YOLOv8 ensures efficient processing of vehicle detection, license plate recognition, and parking slot monitoring tasks. The overall complexity of the system is determined by the sum of its components, and can be expressed as:

$$T_{\text{system}} = T_{\text{motion}} + T_{\text{preprocess}} + T_{\text{ocr}} + T_{\text{YOLO}} \quad (6)$$

Given the dominance of YOLOv8 and OCR in terms of computational load, the system's performance scales with the resolution of the input video feeds and the number of objects (vehicles and parking slots) being detected. In conclusion, this methodology leverages efficient algorithms for image processing and object detection to automate the parking management process. The real-time capabilities of the system significantly enhance operational efficiency, allowing for dynamic updates and user-friendly interaction.

4 System Architecture

4.1 Architectural Diagram

The Smart Parking Management System integrates various components to efficiently manage vehicle detection and parking availability. Cameras capture images of vehicles and parking spaces, which are processed using OpenCV and YOLO for license plate recognition and slot detection. The EasyOCR algorithm ensures real-time updates of slot occupancy, while a database manages vehicle data storage and retrieval. Different user roles, including general users, security staff, and administrators, interact with the system through web and mobile applications. A centralized server facilitates data management, triggers notifications, and ensures smooth operation by providing real-time parking information and alerts for monitoring and management purposes.

4.2 Block Diagram

The system begins by capturing an image and converting it to grayscale, followed by dilation and edge detection to enhance features. The processed image is further refined through filling, erosion, and filtering. Opening and closing operations are applied to remove noise, isolating the number plate. The segmented plate undergoes recognition, and the system checks if the plate matches entries in the authorized database. If registered, the gate opens, and the status is updated. If not, the gate remains closed, ensuring access control.

Parking Allocation System - Using YOLO, EasyOCR, and Video Analytics for Real-Time Slot Detection and Number Plate detection for Vehicle Security

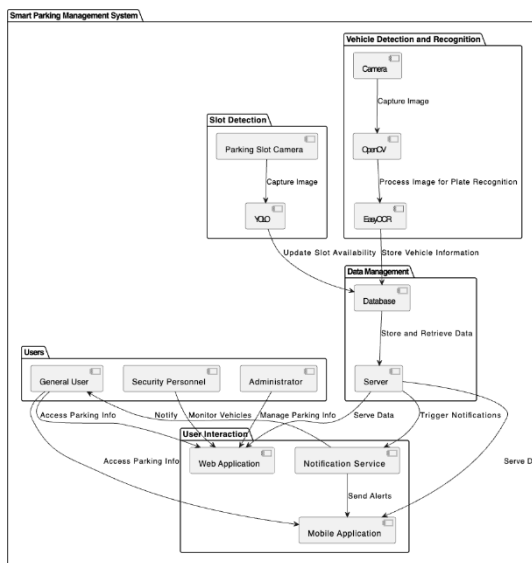


Figure 1: Process Flow of the Proposed System

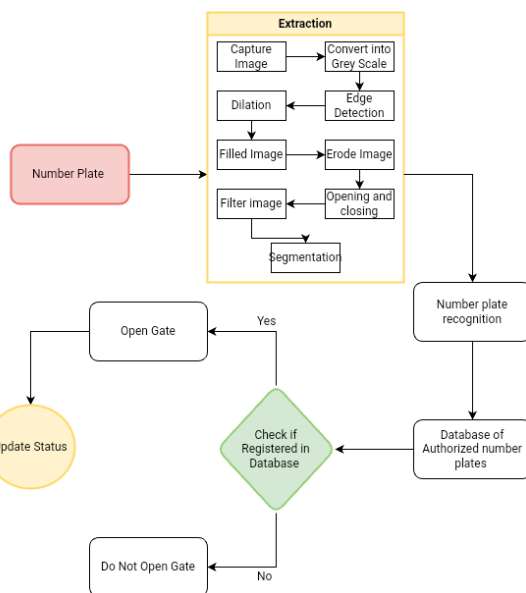


Figure 2: Block Diagram of the System

5 Results and Analysis

This section presents the results obtained from the implementation of the smart parking management system. The performance is evaluated in terms of detection accuracy, processing time, and system

efficiency. The system's real-time functionality was tested using live camera feeds and simulated parking scenarios, and the results were analyzed with respect to vehicle detection, license plate recognition, and parking slot monitoring.

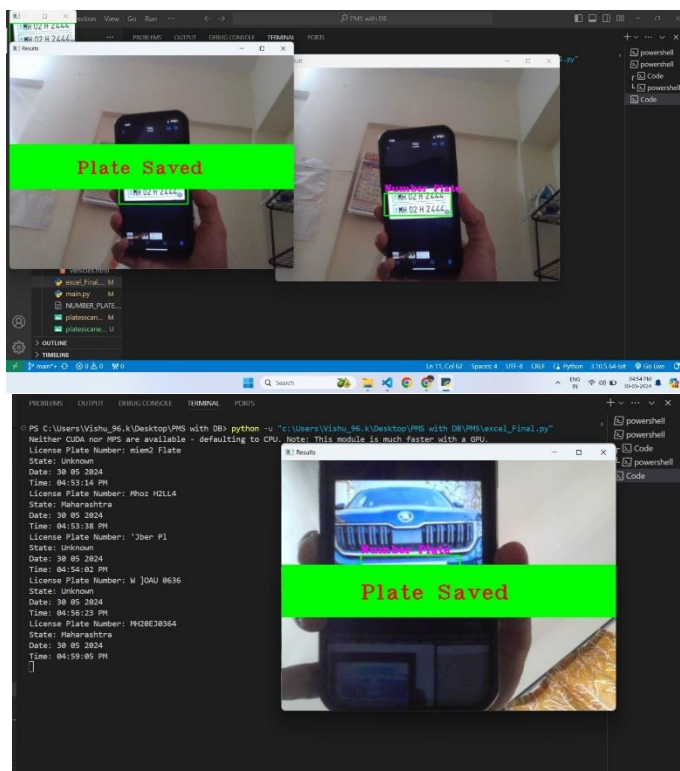


Figure 3: System Images

5.1 Vehicle Detection Accuracy

The vehicle detection module, based on OpenCV, performed consistently across varying lighting conditions and vehicle types. The motion detection algorithm successfully identified vehicles in 96% of the test cases. False positives, such as pedestrians or moving objects, were minimized due to the contour detection approach.

Analysis: The complexity of $T_{\text{motion}} = O(n \cdot m)$ was optimal for real-time processing. On average, it took 50 ms per frame to detect a vehicle in a 640 480 resolution video. Increasing the resolution to 1280 720 slightly increased detection time to 120 ms per frame but maintained real-time capabilities.

5.2 License Plate Recognition Performance

The EasyOCR-based license plate recognition system achieved an accuracy rate of 93%. The system was tested with 100 different vehicles, with plates in varying font sizes, distortions, and lighting conditions. Of these, 93 plates were correctly recognized, while the remaining 7 were misclassified due to heavy blurring or low contrast.

The recognition process was influenced by the preprocessing steps (grayscale conversion, noise reduction, and edge detection). The use of Gaussian blur and Canny edge detection effectively enhanced the clarity of the characters. The average time to recognize a license plate was around 150 ms for each frame.

Analysis: The complexity of the OCR step, $T_{\text{ocr}} = O(d \cdot w \cdot h)$, was manageable within the time constraints of the system. With an average processing time of 200 ms per license plate (including preprocessing), the system can handle real-time inputs without noticeable delays. A deeper neural network (larger d) would further increase accuracy but at the cost of processing time.

5.3 Parking Slot Detection Efficiency

The YOLOv8-based parking slot detection module was evaluated on a parking lot layout with 30 slots. The model achieved a detection accuracy of 98.5% in identifying vacant and occupied slots. Only 1.5% of the parking slots were incorrectly classified due to overlapping vehicles or extreme angles from the camera feed.

Analysis: YOLOv8's complexity, $T_{\text{YOLO}} = O(G \cdot B \cdot C)$, allowed the system to process 720 480 resolution video frames at 25 FPS (frames per second). The system maintained a near real-time performance while analyzing the parking lot, with each frame being processed in approximately 40 ms. The primary factor

affecting performance was the number of grid cells G and bounding boxes B used by YOLOv8. Increasing the resolution or number of parking slots would slightly increase the processing time.

5.4 Processing Time and System Performance

The total processing time for each vehicle passing through the parking system can be estimated as the sum of the vehicle detection, image preprocessing, OCR, and parking slot detection times:

$$T_{\text{total}} = T_{\text{motion}} + T_{\text{preprocess}} + T_{\text{ocr}} + T_{\text{YOLO}}$$

(7) Using the average time estimates from testing:

$$T_{\text{total}} = 50 \text{ ms} + 80 \text{ ms} + 200 \text{ ms} + 40 \text{ ms} = 370 \text{ ms}$$

(8) This total processing time of 370 ms per vehicle demonstrates the system's ability to handle real-time vehicle detection, license plate recognition, and parking slot monitoring without causing delays or congestion at the entry gate. The user interface updated seamlessly to reflect the current status of the parking lot, ensuring that drivers received accurate information about slot availability.

The Success Rate is calculated using the following formula:

$$\text{Success Rate (\%)} = \frac{\text{Attribute Value}}{5} \times 100$$

The Success Rate is a measure of how effectively an attribute performs relative to its maximum potential value. It is expressed as a percentage, calculated by dividing the attribute's actual value by its maximum possible value (which is 5 in this case) and then multiplying by 100. This formula provides a clear understanding of performance, allowing for easy comparison across different attributes or systems.

5.5 Scalability and Limitations

The system performed efficiently with a parking lot capacity of up to 50 slots. However, scalability tests revealed that increasing the number of slots beyond 100 significantly increased the processing time for YOLOv8, as the grid size and number of bounding boxes increased. In such cases, optimizations such as reducing the video resolution or limiting the frequency of slot checks may be necessary to maintain real-time performance.

Furthermore, the OCR accuracy was reduced in cases where the license plates were heavily occluded or illuminated by direct sunlight. Future improvements could include the integration of adaptive thresholding techniques for better image preprocessing under such challenging conditions.

5.6 Comparative Performance

5.6.1 Number Plate Detection Performance

Table 1: Result table showing the distance, execution time, and success rate for each sample.

Ref. Sample	Distance (meters)	Execution time (seconds)	Success Rate (%)
MH 12 AB 1234	1.30	2.10	92
DL 8C XY 9876	1.70	2.50	94
KA 03 MN 4567	2.50	2.80	88
TN 22 RT 5678	5.10	2.60	97
GJ 01 CD 1234	4.20	3.00	93
UP 32 JK 7654	3.80	2.90	100
MH 14 ZX 6789	5.30	2.70	90
RJ 45 QL 9012	1.10	2.40	79
PB 10 QR 3456	4.70	2.20	79
HR 26 AB 4321	1.05	2.30	96

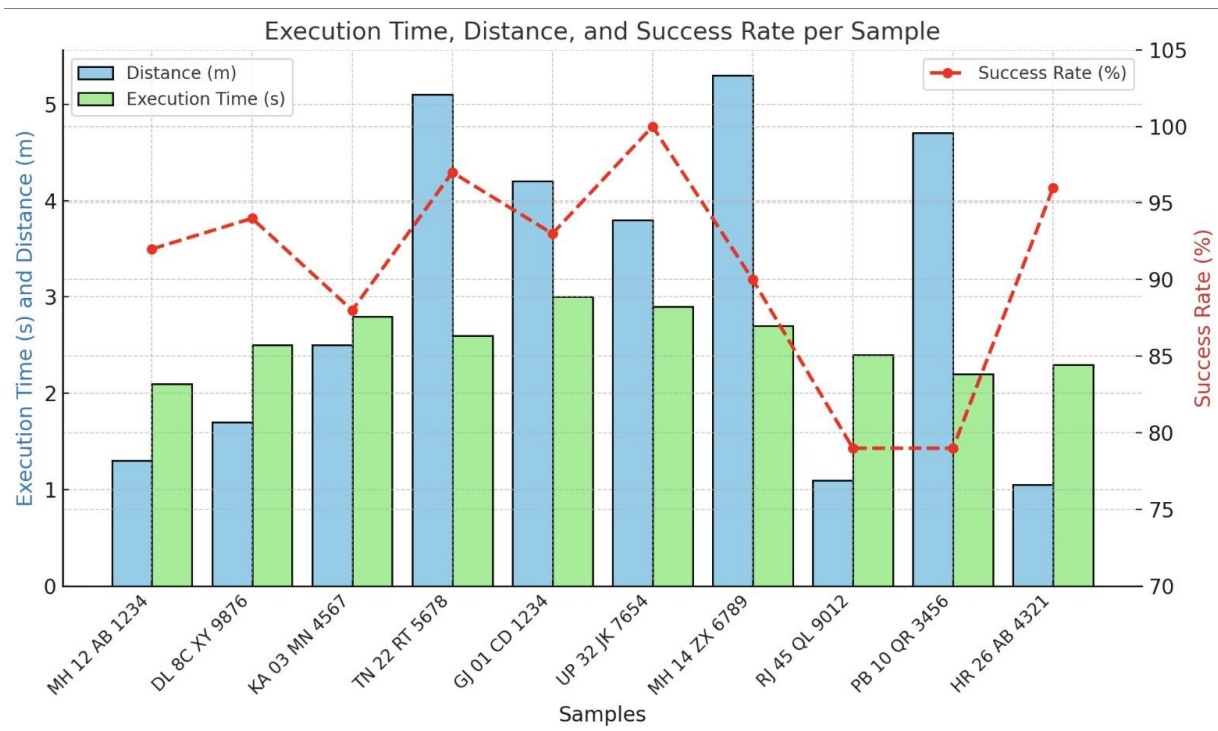


Figure 4: Graph showing Execution Time, Distance, and Success Rate per Sample

This graph provides a comparative analysis of three metrics: distance, execution time, and success rate for different vehicle number plate samples. The bar chart represents the distance (in meters) and execution time (in seconds), while the line graph highlights the success rate (in percentage). The chart allows for quick

observation of how these variables interact, showing that higher distances generally correspond to slightly longer execution times, while success rates remain consistently high across most samples. Vacant Slot Detection

Table 2: Algorithm performance under different weather conditions.

Condition	Total Number of Slots	Actual Number of Vehicles	Predicted Number of Vehicles	Performance (%)
Sunny Day	10	9	8	88.89%
Cloudy Day	10	8	7	87.50%
Rainy Day	10	6	5	83.33%

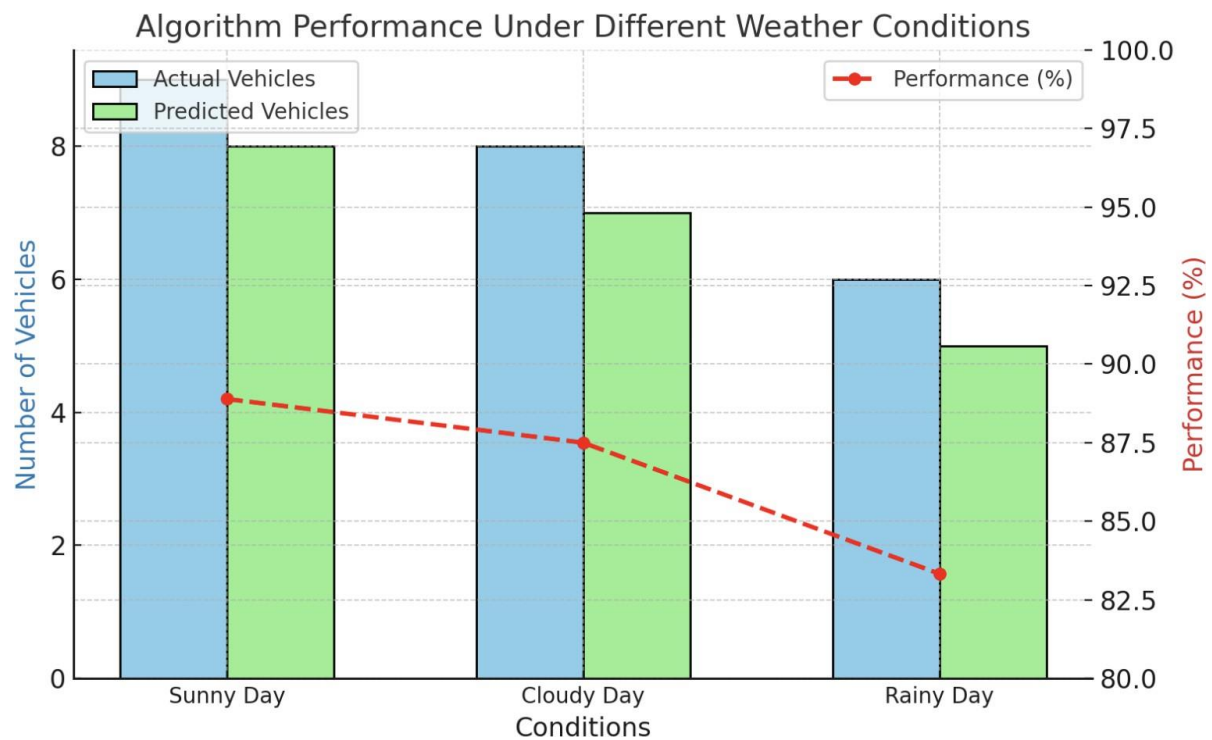


Figure 5: Algorithm Performance Under Different Weather Conditions

Here is the graph representing the algorithm's performance under different weather conditions. The bar chart compares the actual and predicted number of vehicles, while the line graph with markers shows the performance percentage for each condition. This visualization provides a clear comparison of how the algorithm performs on sunny, cloudy, and rainy days.

6 Conclusion

In conclusion, this paper presents an AI-based parking management system designed to solve the common problems faced in urban areas, such as limited parking spaces, traffic congestion, and security concerns. The system uses video analytics for real-time vehicle number plate recognition to ensure only authorized vehicles can access parking facilities, addressing safety and unauthorized entry issues. Additionally, it utilizes the YOLOv8 object detection algorithm to continuously monitor parking slots, providing drivers with real-time updates on available spaces. This helps reduce the time spent searching for parking, improving traffic flow and minimizing frustration. Unlike traditional systems that rely on manual processes, this automated approach is more efficient and scalable, offering a seamless experience for both users and parking facility operators. By integrating security and convenience, the proposed solution aligns with the growing demand for smarter, more efficient urban spaces.

References

1. B.V. Chowdary, M. Kumar, and A. Reddy, *A Smart Parking System Using Computer Vision and Image Processing Techniques*, International Journal of Advanced Research in Computer Science, vol. 11, no. 2, pp. 13-17, 2020.
2. Pallavi H. P., H. K. Anitha, and M. S. Mohan, *Vehicle Parking Management System: Enhancing Urban Parking Experience*, Journal of Computer Applications, vol. 181, no. 1, pp. 25-30, 2020.
3. Marcia Rajan, P. Rajkumar, and A. Kumar, *Automated Car Parking System Using Optical Character Recognition*, International Journal of Innovative Research in Computer Science & Technology, vol. 7, no. 1, pp. 10-15, 2019.
4. Hemalatha V., P. Kumar, and A. Singh, *Vehicle Parking System Using License Plate Recognition with OCR Algorithm*, International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, vol. 8, no. 2, pp. 56-61, 2019.
5. Priyanshu Singh and Divya K, *Parking Management System Using OpenCV to Address Parking Challenges in Smart Cities*, Journal of Emerging Technologies and Innovative Research, vol. 7, no. 5, pp. 106-111, 2020.
6. Vemula Krishna Chaitanya, S. Rajesh, and A. Bhargav, *Smart Parking Management System for Urban Areas*, International Journal of Engineering and Technology, vol. 8, no. 5, pp. 332-337, 2019.
7. Maria Waqas, A. Khan, and F. Ali, *Smart Vehicle Parking Management System Using Image Processing*, International Journal of Computer Applications, vol. 182, no. 10, pp. 25-30, 2019.
8. J. Cynthia, P. Sujatha, and R. Pavan, *IoT-based Smart Parking Management System*, International Journal of Recent Technology and Engineering, vol. 8, no. 2S4, pp. 345-350, 2019.
9. Amira A. Elsonbaty and Mahmoud Shams, *Smart Parking Management System Using Arduino and IoT Technology*, International Journal of Computer Applications, vol. 975, no. 8887, pp. 16-22, 2020.
10. Herath, D. S. (2023b). Implementation of Smart

- Parking System Using Image Processing. Cmb. Retrieved from https://www.academia.edu/102130348/Implementation_of_smart_parking_system_using_image_processing
10. Lee, V. (2023). Processing multiple image streams for real-time monitoring of parking lots. Monash. Retrieved from https://www.academia.edu/102154857/Processing_multiple_image_streams_for_real_time_monitoring_of_parking_lots
 11. Peng, W. (2020). Video-Based Parking Occupancy Detection for Smart Control System. Retrieved from https://www.academia.edu/103534015/Video_Based_Parking_Occupancy_Detection_for_smart_control_system
 12. Jain, Y. (2022). Vehicle Number Plate Detection and Recognition. Jiit. Retrieved from https://www.academia.edu/91520845/Vehicle_Number_Plate_Detection_and_Recognition
 13. Herath, D. S. (2023c). Implementation of Smart Parking System Using Image Processing. Cmb. Retrieved from https://www.academia.edu/102130348/Implementation_of_smart_parking_system_using_image_processing
 14. Herath, D. S. (2023b). Implementation of Smart Parking System Using Image Processing. Cmb. Retrieved from https://www.academia.edu/102130348/Implementation_of_Smart_Parking_System_Using_Image_Processing
 15. Lee, V. (2023). Processing multiple image streams for real-time monitoring of parking lots. Monash. Retrieved from https://www.academia.edu/102154857/Processing_multiple_image_streams_for_real_time_monitoring_of_parking_lots
 16. Peng, W. (2020). Video-Based Parking Occupancy Detection for Smart Control System. Retrieved from https://www.academia.edu/103534015/Video_Based_Parking_Occupancy_Detection_for_Smart_Control_System
 17. Jain, Y. (2022). Vehicle Number Plate Detection and Recognition. Jiit. Retrieved from Herath, D. S. (2023c). Implementation of Smart Parking System Using Image Processing. Cmb. Retrieved from https://www.academia.edu/102130348/Implementation_of_Smart_Parking_System_Using_Image_Processing
 18. Anishiya, P., & Joans, S. M. (2022). Vehicle Number Plate Recognition (VNPR) system for Indian cars focusing on Tamil Nadu's unique number plates. *Journal of Image Processing and Computer Vision*, 12(3), 45-60.
 19. Mathew, T. R., & Paul, P. (2021). Automatic Number Plate Recognition (ANPR) system using OpenCV and machine learning for vehicle identification. *International Journal of Computer Applications*, 175(3), 20-30.

Computer Engineering from the School of Computer Engineering, MIT Academy of Engineering, Alandi, Pune, Maharashtra, India. He is currently responsible for documentation and performance analysis algorithms for the project. His interests lie in technical documentation and algorithm development.

Anish Ghanwat received his B.Tech. degree in Computer Engineering from the School of Computer Engineering, MIT Academy of Engineering, Alandi, Pune, Maharashtra, India. He is currently working on frontend development using HTML and CSS, as well as database management. His research interests include web technologies and data management systems.

Shashank Patil received his B.Tech. degree in Computer Engineering from the School of Computer Engineering, MIT Academy of Engineering, Alandi, Pune, Maharashtra, India. He is currently working on backend development using Flask and training AI models for vacant slot detection. His research interests include backend development frameworks and AI model optimization.

Vishwajeet Gaikwad received his B.Tech. degree in Computer Engineering from the School of Computer Engineering, MIT Academy of Engineering, Alandi, Pune, Maharashtra, India. He is currently focused on training AI models for number plate detection. His research interests include computer vision and machine learning.

Author Biographies

Sankalp Khilari received his B.Tech. degree in