

IoT-Based Automatic Parking Monitoring System Using Computer Vision Technology

MH. Ramdhani Ismar¹, Ruhmina Adristi Pertiwi¹, Angger Binuko Paksi^{1*}

¹Politeknik Negeri Madiun, Indonesia

*Corresponding author: Angger Binuko Paksi; email: angger@pnm.ac.id

Manuscript received June 26, 2026, revised September 10, 2026, accepted September 11, 2026

Abstract – State Polytechnic of Madiun faces challenges in managing parking areas, particularly during large events that lead to increased vehicle volume. The absence of information regarding parking space availability causes congestion and inconvenience for users. To address this issue, an Internet of Things (IoT)-based smart parking system integrated with computer vision technology using the YOLOv8n algorithm was developed. The system consists of an IP Camera and Raspberry Pi 4B to detect vehicles, supported by a website interface connected to Firebase Realtime Database for displaying parking slot status. This study employed the Research and Development (R&D) method with a prototyping approach, involving stages of requirement gathering, prototype development and evaluation, system development, system testing and evaluation, and system implementation. Testing was conducted using the black-box testing method to evaluate detection accuracy and overall system performance. The results indicate that the system successfully detected vehicles with satisfactory accuracy, although limitations were identified in Raspberry Pi performance, including delays and reduced frames per second (FPS) when detecting multiple objects simultaneously. Occupied parking slots are displayed in red, while available slots are shown in green. Detection results are stored in Firebase and through a website featuring parking information, an admin dashboard, parking and staff data management, and user profile functionality. The implementation of this system provides an effective solution for monitoring parking conditions and supports more structured parking area management at Campus 2 of the State Polytechnic of Madiun.

Keywords: Computer Vision, Internet of Things (IoT), Smart Parking System, YOLOv8n, Campus Parking Management

I. Introduction

Advancements in information technology have significantly influenced various sectors, including transportation and parking management. Parking refers to the temporary condition in which a vehicle is left stationary by its driver. In general, drivers tend to seek parking locations that are closest to their destinations. However, not all buildings or public facilities provide sufficient parking capacity, resulting in some users utilizing roadside areas as parking spaces [1].

Campus 2 of the State Polytechnic of Madiun is one of the locations that frequently experiences parking-related issues, particularly during large-scale events that increase the number of incoming vehicles. The lack of effective parking area management makes it difficult for users to identify the number and location of available parking spaces, resulting in irregular vehicle positioning and reduced accessibility within the campus area [2].

Furthermore, there is currently no parking information system available, either manually or digitally. The absence of a parking information system within the campus

environment causes users to experience difficulty in locating vacant parking spaces, leading to congestion and queuing during parking activities [3]. As a consequence, drivers spend more time searching for parking spaces, reducing visitor convenience and potentially increasing air pollution in parking areas [4].

Computer vision technology has emerged as one of the potential solutions to address these challenges, as it enables machines to imitate human visual perception. OpenCV (Open Computer Vision Library), an open-source library designed to simplify digital image processing and computer vision programming, enables the monitoring and classification of multiple parking slots using a single camera, thereby reducing implementation costs [5].

This study utilizes the YOLO (You Only Look Once) algorithm, an object detection method based on Convolutional Neural Networks (CNN) capable of performing real-time detection [6]. YOLOv8 represents the latest generation of the YOLO algorithm and introduces an anchor-free detection approach that offers

simpler architecture and improved processing speed compared to previous YOLO versions.

The proposed smart parking system integrates Internet of Things (IoT) technology to monitor parking availability. This study aims to improve parking area management and support automated vehicle access monitoring. The developed system is expected to provide drivers with real-time information regarding parking space availability by utilizing an IP Camera and Raspberry Pi integrated into a web-based platform connected to Firebase. Through this website, users can obtain real-time parking information, thereby improving convenience and efficiency in locating available parking spaces.

II. Research Method

This study employed the Research and Development (R&D) method to develop an IoT- and computer vision-based parking monitoring system. This method was selected because it supports a prototyping approach, enabling iterative system development according to user requirements.

The research process began with an initial investigation stage consisting of interviews with the campus facilities management unit to identify system requirements, direct observation to understand field conditions, and literature studies from various relevant sources. The subsequent stage involved system development using the prototyping method, aiming to obtain an initial representation of the system design and implementation stages before full-scale deployment.

The prototyping process consisted of the following stages:

1. Requirement Gathering

This stage focused on identifying hardware and software requirements, including the YOLOv8n algorithm trained using the COCO dataset, Raspberry Pi 4B, IP Camera, and Firebase Realtime Database.

2. Prototype Development

A prototype was developed through simulation using a miniature parking model and toy vehicles as testing objects. Detection results were integrated into a web-based interface.

3. Prototype Evaluation

Evaluation was conducted by collecting user feedback to assess detection accuracy and interface usability.

4. System Development

This stage included backend development using Python and OpenCV and frontend implementation through a website-based platform.

5. System Testing

Testing involved black-box testing and direct field testing to ensure all functionalities operated as expected.

6. System Evaluation

System evaluation was performed through interviews and evaluation forms to determine system suitability with user requirements.

7. System Implementation

After approval from users, the developed system was implemented in the parking area of the Business Administration Building, Campus 2, State Polytechnic of Madiun.

The research method was designed by integrating hardware and software components as the primary elements of the system. The hardware components functioned to acquire data directly from the physical environment, while the software components were responsible for processing, storing, and presenting data to users through a web application interface. The integration of both components created an automated, efficient, and responsive monitoring system.

TABLE I
HARDWARE SPECIFICATION

Hardware	Function
Ip Camera	Captures real-time image/video data from the parking area to be processed for object detection.
Raspberry Pi 4B	Processes image/video streams from the camera and executes the object detection algorithm before transmitting data to Firebase.
Adaptor Raspberry Pi 4 Type C 5V/3A	Supplies stable electrical power required to operate Raspberry Pi 4B

TABLE II
SOFTWARE REQUIREMENTS

Software	Function
Firebase	Serves as a real-time database for storing processed data from Raspberry Pi 4B..
Visual Studio Code	Code editor used during system development.
Laragon	Local server environment used to run Laravel applications.
Laravel	Local server environment used to run Laravel applications.
Google Chrome	Browser used to test and display the website interface during development and evaluation.



Fig. 1. Design System

Figure 1 illustrates the schematic architecture of the proposed IoT-based automatic parking monitoring system using computer vision technology.

The schematic diagram shows communication between the IP Camera and Raspberry Pi through a wireless connection. The IP Camera functions as the image and video acquisition device, while Raspberry Pi 4B serves as the central processing unit. Both devices are connected through a Wi-Fi network.

The process begins when the IP Camera transmits visual data to Raspberry Pi via wireless communication. The received data are processed and temporarily stored in Raspberry Pi before being transmitted to Firebase and displayed through the web application interface.

III. Results and Discussion

This section presents the implementation of the developed system, system performance analysis, and website interface evaluation.

In the prototype implementation stage, Raspberry Pi was utilized as the main processing unit, while a webcam functioned as a substitute for the IP Camera to capture visual data. Miniature cars and motorcycles were employed as representative objects of actual vehicles.

Prior to implementation, the detection model was trained using approximately 250 image samples, collected over a period of 3–4 hours. The training process generated an object detection model named best.pt.

The developed system successfully identified parked vehicles and determined parking slot availability. Occupied parking slots were represented using red indicators, whereas vacant slots were indicated using green indicators, as illustrated in the prototype visualization.



Fig. 2. Implementation in Prototype

The IoT-based automatic parking monitoring system integrated with Computer Vision technology was implemented using an IP Camera and Raspberry Pi 4B. The IP Camera operated through the TP-Link TAPO application, functioning as a visual data acquisition device before transmitting captured data to Raspberry Pi for processing. Raspberry Pi, accessed remotely using RealVNC, acted as the primary processing unit of the developed system.

Vehicle detection was performed using the YOLOv8n algorithm, selected due to its lightweight architecture, detection accuracy, and compatibility with OpenCV and Ultralytics libraries.

Although the system demonstrated satisfactory detection performance, implementation on Raspberry Pi

4B revealed several limitations, including processing delays, temporary freezing, and reduced frame rates (FPS) when detecting multiple objects simultaneously [7]. These limitations were primarily caused by hardware computational constraints. Nevertheless, the system remained capable of accurately detecting vehicles when objects appeared clearly within the defined bounding box regions.

Figure 3, detection results were visualized as parking slots:

Red → Occupied parking slot

Green → Available parking slot



Fig. 3. Implementation in Parking Area

The generated detection data were transmitted to Firebase Realtime Database and displayed on the website interface.

The performance limitations observed in Raspberry Pi represent an important consideration in evaluating system effectiveness in terms of speed, operational stability, and reliability for real-time parking monitoring.

TABLE III
SYSTEM TESTING RESULTS

No.	Test Scenario	Input	Output	Result	Frame Rate
1	Car detection at slot M4	Vehicle detection at slot M4	System marked slot as "Occupied"	Successful with average delay of 5 minutes	11 fps
2	Motorcycle detection at slot R4	Vehicle detection at slot R4	System marked slot as "Occupied"	Successful with average delay of 5 minutes	10 fps
3	Motorcycle detection at slot R2	Vehicle detection at slot R2	System marked slot as "Occupied"	Successful with average delay of 5 minutes	10 fps

4	Motorcycle detection at slot R1	Vehicle detection at slot R1	System marked slot as "Occupied"	Successful with average delay of 5 minutes	11 fps
5	Car detection at slot M2	Vehicle detection at slot M2	System marked slot as "Occupied"	Successful with average delay of 5 minutes	12 fps

The testing results indicate that the system successfully detected vehicle presence across all evaluated parking slots for both motorcycles and cars.

The generated output in the form of an "Occupied" indicator demonstrates that the sensing and processing mechanisms operated correctly.

However, testing also revealed that the average response time required by the system to update parking status was approximately five minutes, indicating an opportunity for further optimization to improve responsiveness.

IV. Conclusion

This study successfully developed and implemented an Internet of Things (IoT)-based parking monitoring system integrated with Computer Vision technology for deployment at Campus 2 of the State Polytechnic of Madiun.

The proposed system utilized an IP Camera as the visual input source, Raspberry Pi 4B as the processing unit, and the YOLOv8n algorithm to perform real-time detection of two-wheeled and four-wheeled vehicles.

Detection results were transmitted to Firebase Realtime Database and visualized through a website interface that also provides staff management and parking history functionalities.

The developed system demonstrated the capability to provide real-time parking availability information, enabling users to identify vacant parking spaces more efficiently while supporting more structured parking area management.

However, several limitations were identified during implementation. The primary limitation was a detection response delay averaging approximately five minutes, mainly caused by the computational limitations of Raspberry Pi 4B when processing multiple objects simultaneously.

To improve future system performance, the use of higher-performance edge computing devices such as NVIDIA Jetson Nano, Jetson Xavier, or Mini PCs capable of supporting deep learning-based visual computation is recommended.

References

- [1] R. Oktaviani, N. Nazwirman, D. Djamaludin, and V. S. Windyasari, "Aplikasi Sistem Parkir Kendaraan Bermotor Menggunakan Teknologi Radio Frequency Identification (RFID) Di Universitas Islam Syekh Yusuf Tangerang", *JIMTEK*, vol. 1, no. 2, pp. 96–103, Jul. 2020.
- [2] R. Alfian and V. Lusiana, "Penerapan OpenCV dengan Metode Haar Cascade untuk Mendeteksi Jumlah Kendaraan di Tempat Parkir", *J. Teknol. Inf. Komun.*, vol. 8, no. 1, pp. 104–106, Jan. 2024, doi: 10.35870/jtik.v8i1.1360.
- [3] H. Husain, A. Akhriana, H. Herlinda, A. Ahmad, N. Nurdiansah, and R. P. W. Tutang, "Engineering of Information Technology-Based Car Parking Systems," *PROtek: Jurnal Ilmiah Teknik Elektro*, vol. 9, no. 2, p. 86, Aug. 2022, doi: 10.33387/protk.v9i2.4203.
- [4] M. Safran, A. Alajmi, and S. Alfarhood, "Efficient Multistage License Plate Detection and Recognition Using YOLOv8 and CNN for Smart Parking Systems," *Journal of Sensors*, vol. 2024, 2024, doi: 10.1155/2024/4917097.
- [5] A. Muzaki, T. T. Mabruho, R. Ibrahim, and R. Wulaningrum, "Deteksi Ketersediaan Lahan Parkir Dengan Menggunakan OpenCV," in *Prosiding Seminar Nasional Teknologi & Sains (STAINS)*, vol. 3, no. 1, pp. 248–252, 2024.
- [6] A. S. Riyadi, I. P. Wardhani, M. S. Wulandari, and S. Widayati, "Comparison of ResNet, YOLOv3, and TinyYOLOv3 Methods for Image Detection Using Python Programming," *PETIR*, vol. 15, no. 1, pp. 135–144, Jan. 2022, doi: 10.33322/petir.v15i1.1302.
- [7] R. Parvadhavardhni, P. Santoshi, and A. M. Posonia, "Blind Navigation Support System using Raspberry Pi & YOLO," in *Proc. 2nd International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*, Salem, India, 2023, pp. 1323–1329, doi: 10.1109/ICAAIC56838.2023.10140484.