International Journal of Information Engineering and Science
Volume. 1 Nomor. 4 Tahun 2024
e-ISSN :3048-195; p-ISSN :3048-1902, Hal 64-72
DOI: https://doi.org/10.62951/ijies.v1i4.122
Available online at: https://international.arteii.or.id/index.php/IJIES

Parking Slot Scanning for Maximum Efficiency Using Python

Algyon Faras[1*], Gany Andisa[2], Nakula Bintang Nashwandra[3], Jauza Nadhifah[4], David Reza Widhiwipati[5] , Endang Purnama Giri[6], Gema Parasti Mindara[7]
[1-5] Departemen Teknologi Rekayasa Peragkat Lunak, IPB University, Indonesia
[6] Departemen Ilmu Komputer, IPB University, Indonesia
[7] Departemen Teknologi Rekayasa Komputer, IPB University, Indonesia

Jl. Kumbang No.14, RT.02/RW.06, Bogor - Jawa Barat 16128
**Corresponding author: algyon_faras@apps.ipb.ac.id*

Abstract: *The growing number of vehicles in major cities has posed significant challenges in parking lot management. Motorists often have difficulty finding empty parking slots quickly, which not only wastes time but also aggravates traffic congestion and increases air pollution. This research develops a Python-based smart parking system by utilizing the OpenCV library to detect the status of parking slots in real-time. The system uses a camera as the main sensor and processes the image using techniques such as grayscale, Gaussian blur, and adaptive threshold to identify the parking slot status, whether empty or occupied, with good accuracy. The parking slot coordinate data is stored in CSV format to ensure efficient data management. Experimental results with video recordings show that the system is able to operate well in various parking conditions. The system proved to be cost-effective and easy to implement, making it an ideal solution for parking managers who want to improve management efficiency without being burdened with high costs. This research offers a practical solution to help motorists and parking managers optimize parking space usage, reduce search time, and minimize negative impacts such as congestion and carbon emissions.*

Keywords: *Smart Parking System, Python, OpenCV, Image Processing, Parking Slot Detection.*

1. INTRODUCTION

The growing number of vehicles in big cities has led to significant challenges in parking lot management (Mahardita et al., 2024). Motorists often face difficulties in finding empty parking slots quickly, which not only wastes time but also exacerbates traffic congestion and increases air pollution. In some cases, the long search for a parking space can be a source of frustration for motorists and lead to decreased operational efficiency in parking management (Haposan Yoga Pradika Napitupulu & I Gde Dharma Nugraha, 2024). Therefore, there is a need for parking systems that are able to offer efficient, practical, and affordable solutions to address these issues (Nadeak et al., 2024).

Currently, many smart parking systems use technologies such as ultrasonic, infrared, and RFID sensors to detect empty parking slots. However, these technologies have limitations, mainly in terms of high implementation costs and complex infrastructure, making them less suitable for implementation in areas with limited resources or tight budgets (Ardina et al., 2024). In addition, long-term maintenance for this technology can also be an additional burden for parking managers (Yonil et al., 2024).

Received: September 05, 2024; Revised: October 19, 2024; Accepted: November 17, 2024; Published: November 23, 2024

As an alternative, this research developed a Python-based smart parking system utilizing the OpenCV library to detect the status of parking slots in real-time (Dasari et al., 2023). The system uses a camera as the main sensor, which processes images through techniques such as grayscale, Gaussian blur, and adaptive threshold (Wulandari et al., 2022). The results of this processing allow the system to identify the status of the parking slot, whether empty or occupied, with fairly good accuracy (Adnan & Rohmah, 2020). Parking slot coordinate data is stored in CSV files to ensure efficient and flexible data management (Haposan Yoga Pradika Napitupulu & I Gde Dharma Nugraha, 2024).

Through testing with video footage, the system is proven to work in various parking scenarios (Siregar et al., 2024). This approach is not only cost-effective, but also easy to implement, making it an ideal solution for parking managers who want to improve parking lot management efficiency without having to face the burden of high costs (Ardina et al., 2024). This research aims to provide a practical solution that can assist motorists and parking managers in optimizing parking space usage, reducing search time, and minimizing negative impacts such as congestion and carbon emissions (Siregar et al., 2024);(Putra, 2020).

2. LITERATURE REVIEW

a. Smart Parking System

Smart Parking System is a system designed to provide parking availability information to users. The system uses various sensors to detect the status of parking slots and provide relevant information to users through an application. It aims to reduce congestion when searching for parking slots and improve parking management efficiency (Hernoko et al., 2021).

b. Python

One programming language that is increasingly popular and often used to develop various types of applications is Python. The popularity of Python makes it a language that is widely studied by students, especially in universities (Romzi & Kurniawan, 2020). Python is interactive, object-based, and compatible with various platforms such as Linux, Windows, Mac, and others. Python is known to be easy to learn because of its simple and elegant syntax. This ease is supported by built-in modules with an efficient and ready-to-use structure (Ratna, 2020).

This language is often used to simplify the management of large complex data. One of its advantages is its flexibility. This allows the development of data models, organization of data sets, creation of machine learning-based algorithms, web services,

and the application of data mining techniques to solve various tasks efficiently. Python comes with various libraries such as Pandas, Plotly, Numpy, scikit-learn, and XGBoost, which are very helpful in data processing (Pandika Pinata et al., 2020).

c. OpenCV

OpenCV is one of the libraries often used in image processing, utilizing the Application Programming Interface (API). Through this technique, OpenCV allows computers to see and recognize people, objects, and even detect faces. OpenCV consists of five main libraries, namely, Computer Vision (CV), Machine Learning (ML), HighGUI, Image and Video I/O, CXCORE, XML, and CvAux (Maryati & Tryatmojo, 2019).

d. Numpy

Numpy is a library that serves as a reference for matrix computation-based methods. Numpy is designed to facilitate users in various numerical calculations efficiently. Numpy is known as a fundamental package for Scientific Computing that provides a multidimensional array, equipped with various derived objects such as Masked Array and Matrix that can facilitate scientific data processing (Peling et al., 2024).

e. CSV

A Comma Separated Values (CSV) file is a plain text file that can store lists of data used to exchange information between applications, such as databases or contact managers. The data in it is typically separated by comma characters, although other characters are sometimes used, such as semicolons. CSV files make it possible to export complex data into a simple format that can be imported into other applications (Khairina & Rahman, 2022).

3. METHODS

This study employs a computer vision-based approach to scan and monitor parking slots in real-time using Python libraries such as OpenCV and NumPy. The development and implementation of this method involve the following steps:

a. Parking Area Identification

Each parking slot is identified manually by specifying coordinates through user interaction via mouse clicks. These coordinates are stored in a CSV file to ensure flexibility and ease of modification.

b. Image Preprocessing

Images captured from the camera are processed using techniques such as grayscale conversion, Gaussian Blur for smoothing, and adaptive thresholding for segmentation. This step is designed to enhance contrast between empty and occupied parking slots.

c. Slot Scanning and Validation

Identified areas are scanned using a non-black pixel detection method to determine the status of each slot (empty or occupied). Empty slots are highlighted in green, while occupied slots are displayed in red.

d. Data Reporting and Visualization

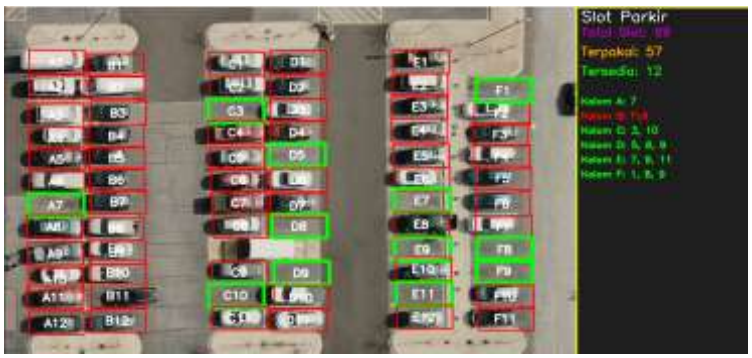
Information on available slots is organized by rows and columns and displayed in a visual interface. An additional information panel is designed to provide a summary of the total, occupied, and available slots.

e. Testing and Validation

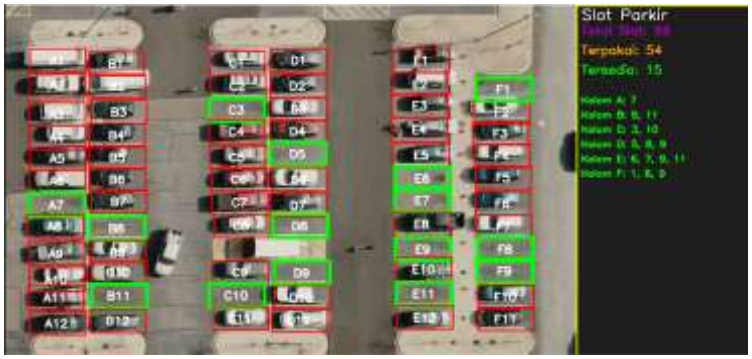
The system was tested with various parking slot configurations to ensure detection accuracy and data processing efficiency.

This approach was chosen for its ability to efficiently detect parking slot statuses without requiring additional hardware. The method can also be replicated by adjusting parameters for different scenarios.

4. RESULTS

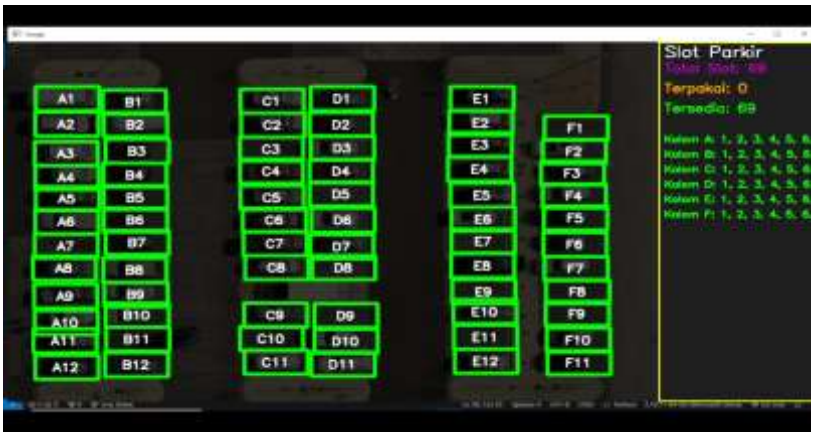


First Test Normal Light

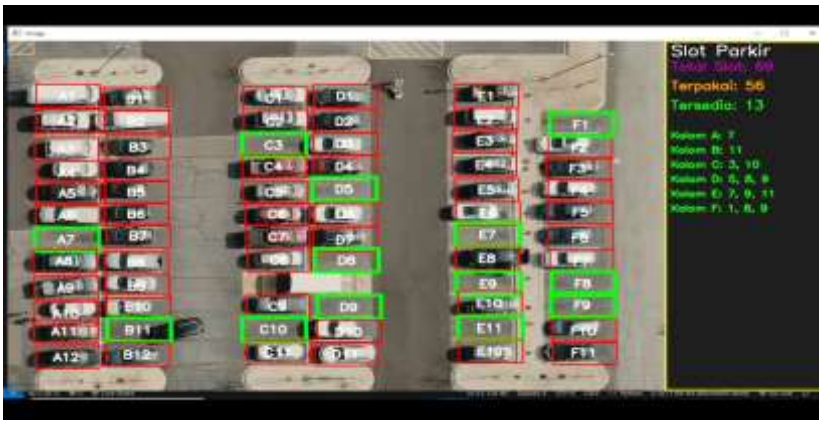


Second Test Normal Light

The testing results indicate that out of a total of 69 parking slots, the system successfully detected 57 occupied slots and 12 vacant slots in the first test, and 54 occupied slots and 15 vacant slots in the second test. The analysis was performed by calculating the number of white pixels within each parking slot based on predefined coordinates. Slots with white pixels below the threshold of 900 pixels were categorized as vacant, while those exceeding this threshold were categorized as occupied. Additionally, the information panel displayed detailed data on the number of vacant and occupied slots, including per-row distribution, to help users better understand the overall parking conditions.



Low-Light Test



Overexposure Test

Tests were conducted under conditions of both low lighting and overexposure. Under low-light conditions, the system struggled to detect parking slots due to poor visibility, leading to inaccuracies in slot detection. On the other hand, under excessively bright conditions, the system had difficulty detecting white vehicles, as their color blended with the bright background. These conditions caused the system to misclassify certain parking slots as vacant. This demonstrates the sensitivity of the system to environmental lighting conditions and highlights its reliance on sufficient and balanced illumination for accurate detection.

Black and White Test

Furthermore, additional limitations were identified during tests using black-and-white video formats. In these cases, the system experienced increased detection errors, as non-vehicle objects (e.g., humans or items) were frequently classified as vehicles occupying parking slots. This occurred because the detection method relied exclusively on analyzing white pixel changes without accounting for object features such as shape or size, leading to false detections of slot statuses.

5. DISCUSSION

This research aims to develop an intelligent parking system capable of automatically detecting parking slot statuses using video analysis. While the system performed well in identifying vacant and occupied slots with good accuracy, several limitations need to be addressed to improve its performance.

The system's sensitivity to lighting conditions was evident during the tests. Under low-light conditions, poor visibility made it difficult to accurately detect parking slots, while overexposure caused white vehicles to blend into the background, resulting in misclassification. These findings suggest that while the system is functional, its reliance on lighting balance limits its performance under extreme environmental conditions. Future iterations could incorporate advanced image processing techniques or adaptive lighting correction algorithms to improve accuracy across diverse lighting scenarios.

Another key issue identified was the false detection of objects in black-and-white video formats. The system was unable to differentiate between vehicles and non-vehicle objects, leading to frequent misclassification. This limitation arises because the detection process is based solely on white pixel analysis without considering object characteristics. Addressing this issue may require the integration of machine learning models or advanced object detection frameworks capable of distinguishing between vehicle and non-vehicle objects.

Moreover, testing so far has been limited to recorded videos, and real-time performance using live camera feeds has yet to be evaluated. Similarly, the system depends on manual input to define parking slot coordinates, which could be challenging in dynamic or large-scale parking areas. Implementing an automated slot detection feature could enhance the system's usability and scalability in such scenarios.

In conclusion, while the system demonstrates strong potential as an efficient smart parking solution, there are several areas for improvement. Future development should focus on mitigating the impacts of environmental lighting conditions, integrating advanced object recognition capabilities, and conducting real-time tests to ensure reliability in real-world applications.

6. CONCLUSION

Based on the results obtained from testing the Python and OpenCV-based smart parking system, it can be concluded that this system can successfully detect the status of parking slots in real-time with a fairly good level of accuracy. Through the use of image processing techniques such as grayscale, Gaussian blur, and adaptive threshold, the system can identify whether the parking slot is empty or occupied. The system proved to be more efficient and cost-effective compared to other smart parking technologies that rely on physical sensors. Other advantages found are the ease of implementation and adaptability of the system to various parking conditions, including changes in lighting and weather. Therefore, this system offers a practical solution that can be implemented in various parking locations to help motorists find parking spaces faster, reduce congestion, and reduce air pollution. The results of this study show that OpenCV-based smart parking systems can be a very useful alternative in optimizing the use of parking spaces in crowded big cities.

LIMITATION

Although the developed system shows good results, there are some limitations that need to be noted. One of them is the sensitivity of the system to non-ideal lighting conditions. At low or overexposure, the detection accuracy may decrease, which affects the result of classifying parking slots as empty or occupied. In addition, this image processing-based detection method is less able to distinguish vehicles from other objects, such as people or goods, which can interfere with detection results. This research is also limited to trials using video recordings and has not been tested on the use of live camera feeds, which may present additional challenges in real-time deployment of the system. The use of manual coordinates to

determine the position of parking slots also limits the flexibility of the system, especially in large parking areas or those with a variety of parking lot shapes. Therefore, future research can develop the system with more advanced object detection technologies, as well as improvements in image processing and automatic object recognition to overcome these limitations.

REFERENCES

Adnan, M., & Rohmah, R. N. (2020). *Pemanfaatan Dua Mikrokontroler Platform IoT Dalam Pengembangan Sistem Parkir*. *20*(02).

Ardina, E. N., Sulistyo, I. A., & Idris, L. O. M. (2024). Teknologi Smart Parking Untuk Sistem Keamanan Pada Kendaraan Berbasis Sensor Putar. *Smart Comp: Jurnalnya Orang Pintar Komputer*, *13*(1). https://doi.org/10.30591/smartcomp.v13i1.5644

Dasari, J., Vodnala, S., & Enugala, S. (2023). Smart Parking System using Python and OpenCV. *International Journal for Research in Applied Science and Engineering Technology*, *11*(8), 1976–1985. https://doi.org/10.22214/ijraset.2023.55497

Haposan Yoga Pradika Napitupulu & I Gde Dharma Nugraha. (2024). Sistem Berbasis Komputasi Kabut Untuk Sistem Parkir Pintar Terdesentralisasi Menggunakan Firebase. *Jurnal Nasional Teknik Elektro dan Teknologi Informasi*, *13*(1), 44–52. https://doi.org/10.22146/jnteti.v13i1.10095

Hernoko, M. G., Adi Wibowo, S., & Vendyansyah, N. (2021). PENERAPAN IoT (Internet of Things) SMART PARKING SYSTEM DAN PENDETEKSI KEBAKARAN DENGAN FITUR MONITORING. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *5*(1), 261–267. https://doi.org/10.36040/jati.v5i1.3281

Khairina, S. N., & Rahman, B. (2022). *SISTEM INFORMASI PENGOLAHAN DATA PAJAK KELUARAN PADA PT. IDAMAN*. *7*(1).

Mahardita, P. A., Fitriyah - -, & Setiyono, B.-. (2024). PENGELOLAAN PRAKTIK PARKIR ILEGAL DI KECAMATAN TEMBALANG (STUDI KASUS KAWASAN UNIVERSITAS DIPONEGORO). *Journal of Politic and Government Studies*, *13*(4), Article 4.

Maryati, R. I. S., & Tryatmojo, B. (2019). *Akurasi Sistem Face Recognition OpenCV Menggunakan Raspberry Pi Dengan Metode Haar Cascade*. *07*(02).

Nadeak, Y. V., Ho, P., & Masriah, M. (2024). Real-Time Monitoring System and IOT Smart Parking Booking (Case study: One Hotel in Tangerang). *Circuit: Jurnal Ilmiah Pendidikan Teknik Elektro*, *8*(2), 178. https://doi.org/10.22373/crc.v8i2.18020

Pandika Pinata, N. N., Sukarsa, I. M., & Dwi Rusjayanthi, N. K. (2020). Prediksi Kecelakaan Lalu Lintas di Bali dengan XGBoost pada Python. *Jurnal Ilmiah Merpati (Menara Penelitian Akademika Teknologi Informasi)*, 188. https://doi.org/10.24843/JIM.2020.v08.i03.p04

Peling, I. B. A., Ariawan, I. M. P. A., & Subiksa, G. B. (2024). Deteksi Bahasa Isyarat Menggunakan Tensorflow Lite dan American Sign Language (ASL). *Jurnal Krisnadana, 3*(2), 90–100. https://doi.org/10.58982/krisnadana.v3i2.534

Putra, A. S. (2020). Analisa Dan Perancangan Sistem Tata Kelola Parkir Cerdas Di Kota Pintar Jakarta. *Jurnal IKRA-ITH Informatika*.

Ratna, S. (2020). PENGOLAHAN CITRA DIGITAL DAN HISTOGRAM DENGAN PHYTON DAN TEXT EDITOR PHYCHARM. *Technologia: Jurnal Ilmiah*, *11*(3), 181. https://doi.org/10.31602/tji.v11i3.3294

Romzi, M., & Kurniawan, B. (2020). *Implementasi Pemrograman Python Menggunakan Visual Studio Code*.

Siregar, M. S. A., Usman, U. K., & Astuti, S. (2024). *Pengembangan Sistem Parkir Pintar Berbasis IoT Menggunakan ESP32, Sensor Ultrasonik HC-SR04, dan OLED Display*.

Wulandari, C., Wijayanti, D., Lestari, N., & Daulay, N. K. (2022). *PROTOTYPE SISTEM MONITORING PARKIR PINTAR BERBASIS IOT (INTERNET OF THINGS)*. *7*(2).

Yonil, R. D., Hastuti, H., Dewi, C., & Sardi, J. (2024). Sistem Smart parking Berbasis Arduino Uno Untuk Optimalisasi Penggunaan Ruang Parkir di Gedung Parkir Kota Bukittinggi. *JTEV (Jurnal Teknik Elektro dan Vokasional)*, *10*(2), 113. https://doi.org/10.24036/jtev.v10i2.130952