ICIC Express Letters
Part B: Applications
Volume **11,** Number **9**, September **2020**

ICIC International ⓒ2020 ISSN 2185-2766
pp. 1-8

IoT Based Parking Slot Availability using Canny Edge Detection

Rasyid Sabilillah[1], Khodijah Amiroh[1,*] and Moh. Hamim Zajuli Al Faroby[2]

[1]Department of Information Technology
Institut Teknologi Telkom Surabaya
Surabaya, Indonesia
rasyids@student.ittelkom-sby.ac.id; [*]Corresponding author: dijaamirah@ittelkom-sby.ac.id

[2]Department of Data Sains
Institut Teknologi Telkom Surabaya
Surabaya, Indonesia

Received November 2019; accepted March 2020

ABSTRACT. *As the population increases, the number of motorized vehicles used also increases. Motorized vehicles are parked haphazardly, which results in the availability of efficiently full parking slots. This parking problem often occurs at the ITTelkom Surabaya Campus, where students and lecturers need help finding available parking slots, and their time is wasted. Therefore, an information and monitoring system design was made regarding the availability of parking slots based on Images Processing and IoT which are expected to be implemented at the ITTelkom Surabaya Campus. The information system uses real-time image capture with the ESP-32 CAM microcontroller as the camera and the ESP8266 microcontroller as the 16x2 I2C LCD power. Image acquisition using the Canny Edge Detection method only and the Canny Edge Detection method followed by Closing morphology for comparison. Parking availability output is displayed on the Open-CV, Antares, and 16x2 I2C LCD. Image processing is based on four parameters: min threshold = 100, max threshold = 400, min pixels = 0, and max pixels = 225. System testing based on these parameters results in the Canny Edge Detection method followed by better Closing morphology than the application of the Canny Edge Detection method just. The test was carried out with 50 experiments of the wireless system using Black Box Testing, which produced results that were very consistent with the system. However, the delay value on the wireless system gets an average of 5.61 seconds, which is quite good to implement in real terms.*

Keywords: Internet of Things, Parking lot, Canny Edge Detection, Image Processing

1. **Introduction.** The increase in population in Indonesia is very rapid [1]. The number of motorized vehicles used increases as the population increases [2]. In urban areas, it is often found that the driver of a vehicle, like a car is only one person. If many drivers are like that, there will be a high increase in vehicle volume. This causes traffic jams, and on the other hand, the availability of parking in several parking lots will be reduced, even full.

Motorized vehicles are sometimes parked haphazardly [3], for example, car parks are used for motorbike parking, and vice versa, motorcycle parking is used for car parking, although this rarely happens. This resulted in the availability of efficiently full parking slots. In big cities, several smart parking systems have been implemented [4]. However, some of these parking systems still need to be more effective and efficient because motorized vehicle owners are still looking for parking availability even though they have seen the light sensors provided. It should make it easier for motorists to check and see available parking spaces in parking areas without causing traffic jams, wasting much time, and wasting fuel.

Several studies have been conducted to monitor the availability of parking slots efficiently. Della Sabila Azkarika used the template matching method with Raspberry Pi 3, Logitech C525 Webcam, and MySQLITE Database [5]. The time required for each test cycle is 14-16 seconds with a light intensity of 7.7 to 500 lux. Janak Trivedi, Mandalapu Sarada Devi, and Dave Dhara used a Raspberry Pi, USB camera, LCD, Telegram, and Python to manage real-time parking reservations. Consistent results are obtained with the application of this technology in real-time [6]. I Made Edy Listartha, Gede Indrawan, and I Gede Wahyu Pramartha used the Gaussian Blur and Canny Edge Detection methods with Raspberry Pi, Webcam, and Open-CV, which achieved 90.9 percent accuracy [7]. Ferry Pradana Putra and Indah Susilawati use the Canny Edge Detection method with Webcams and Open-CV, with an accuracy rate of up to 77 percent using optimal parameters such as a threshold of 510 at the height of 40 cm and light intensity of 100-115 lux [8]. However, Rizky Andyno Ramadhan's research used CCTV and Open-CV to display information on the number of available parking lots but got unfavorable results because of noise which affected the detection results [9].

At the ITTelkom Surabaya Campus, a parking lot is available and used for students and lecturers. However, information systems are only used to see the availability of parking slots for students and lecturers. Often the motorized vehicles used by students need to be parked neatly, which results in a whole parking lot. Several students also needed clarification about finding parking slots because of this problem, and their time was wasted. Therefore, a prototype design was made using the ESP8266 to make it easier to send data from the hardware to the database [10]. The design also uses the ESP32-CAM to facilitate the integration of hardware equipped with a camera with WiFi [11]. A module called LCD 16x2 I2C is used to minimize the use of legs on the LCD, which only contains four pins to be connected to the Arduino board [12]. Programming is then carried out using Open CV, which can be capable of algorithms with IoT [13]. As well as sending data using Antares, in sending to the Antares platform, the MQTT protocol is used because of its unique function in sending data with a small capacity [14]. With the establishment of this information and monitoring system, it is hoped to reduce the problems above.

2. **Research Method.**

2.1 **Design System**

In this design, the information system used for miniature monitoring of parking slot availability is an application with the Python OpenCV library. IoT devices, starting from microcontrollers and supporting sensors, will later be integrated into the Region of Interest Image Processing method, which then applies Canny Edge Detection followed by Closing morphology using the internet to connect and communicate with each other. The Region of Interest algorithm is a segmentation technique that can reduce the problem of image processing time [15]. The Canny algorithm is an edge detection algorithm that uses the Gaussian operator and the convolution approach for the image matrix function [16]. Closing morphology is a digital image processing technique that applies dilation morphology to the image, followed by erosion morphology [17].

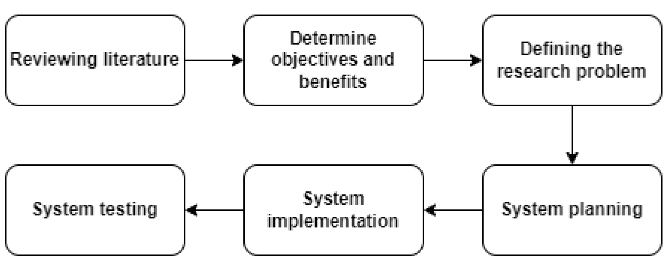


Figure 1. Research Flowchart

2.2 **Design Hardware**

In Figure 2, it can be seen how the hardware design is made. Each hardware device will be connected to the other using jumper cables connected to the breadboard. The function of the breadboard here is to connect several sensors to the microcontroller because the microcontroller itself cannot connect jumper cables directly with the number of sensors to be used.

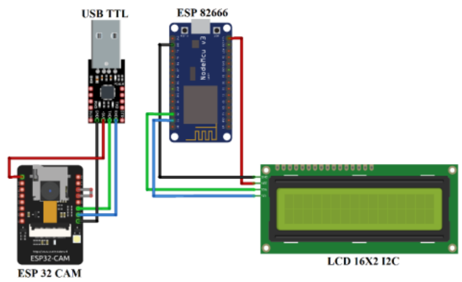


Figure 2. Hardware Design

3. **Results and Discussion.**

The test scenario uses Black Box Testing. Black Box Testing is a software testing method that focuses on testing the system's functionality as a whole, regardless of how the system is implemented [17]. This test is helpful in determining whether the system is functioning correctly. Whereas in testing the Digital Image Processing system using ROI (Region of Interest) is then applied with two other methods, namely the application of Canny Edge Detection only and the application of Canny Edge Detection followed by Closing morphology to compare the best method between the two application methods so that can proceed with the experiment. Image processing with these two methods uses four parameters, namely min threshold = 100, max threshold = 400 as the minimum and maximum values required for a pixel to be considered as an edge, min pixels = 0, and max pixels = 225 to return the minimum and maximum values, and coordinates of the position of the minimum and maximum values in the image. The threshold is used high enough to eliminate noise in the miniature parking area. At the same time, the pixels are adjusted according to the lighting conditions during system testing so that image objects can be appropriately detected.

3.1 **Prototype**

In the prototype, tools and materials are designed and put together to implement IoT system testing and Digital Image Processing shown in Figure 3.

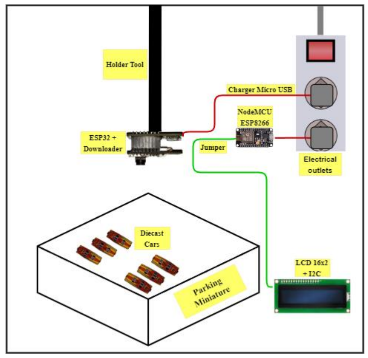


Figure 3. Design Prototype

3.2 **Implementation of Region of Interest**

In Figure 4, the application of the Region of Interest operator to manually determine the coordinates of parking slots that will be used in parking miniatures. After that, the coordinates are saved into a document (.csv) to be retrieved in the following implementation.

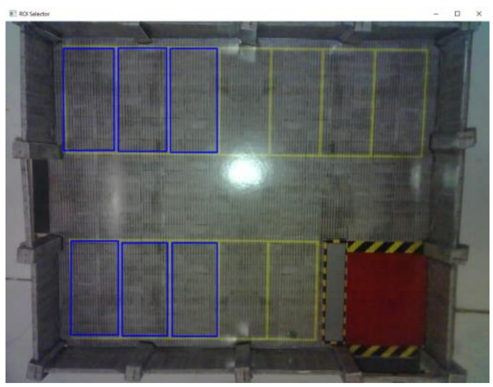


Figure 4. ROI Application Results

3.3 **Implementation of Canny Edge Detection**

Figure 5 tests the system using the Canny Edge Detection method without Closing morphology. In the Canny Edge Detection image on the right, the car object is not completely visible, and some noise is still visible. However, the information system shows Lecturer: 0 and Student: 0, which means there is no parking slots for students.

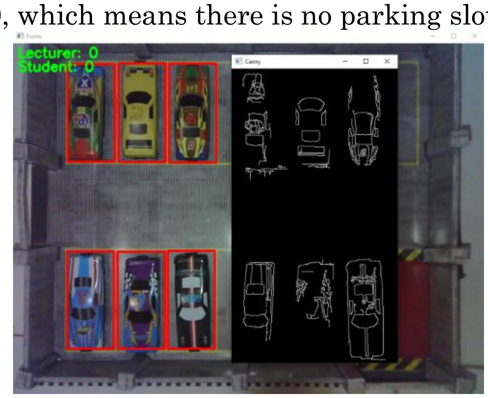


Figure 5. Canny Application Results

3.4 **Implementation of Canny Edge Detection and Closing Morphology**

Figure 6 tests the system by applying the Canny Edge Detection method followed by

Closing morphology. The car is almost evident in the Closing Image to the right, and some of the noise from the Canny Edge Detection image has been reduced. This is because morphological techniques can help eliminate noise in the image and improve edge detection, which is not good in some parts of the image. The information system shows Lecturer: 0 and Student: 0, which means there are no parking slots for students or students.

Figure 6. Canny and Clossing Morphology Monitoring Results

3.5 **Wireless System Testing**

Wireless system testing using Antares cloud service. Testing the system is to find out how high the delivery delay is, starting from the results of the information system to the LCD screen. In Table 1, the system was tested using the Canny Edge Detection method followed by the Closing 50 morphology experiment yielding observational results consistent with the system for calculating between the real total parking slots and the recommended total parking slots and has an average delay of 5.61 seconds. The delay could be better. In some of the experiments, there was a delay of more than 5 seconds because the network condition was unstable. The network is unstable because the network experiences high ping or pings of more than 150 ms. The ideal ping within a network for cloud services should be 1-50 milliseconds (ms). The lower the ping value, the faster the cloud server responds to user requests. This can improve the quality of cloud services and reduce the response time required for data processing. The ideal ping may vary depending on the user's location and distance from the cloud server. In addition, other factors, such as the quality of the user's internet network and traffic density, can also affect the excellent ping value in the network for cloud services.

Table 1. Wireless System Testing

Test	Lecturer parking slot available	Available student parking slots	Total parking slots	Suggested total parking slots	Observation result	*Delay* (s)
1	3	3	6	6	True	5.30401
2	3	2	5	5	True	5.235185
..	..	..	..	..	..	..
49	1	1	4	2	False	5.237797
50	1	1	4	2	False	5.369822
Delay Average					5.61653676	

In Table 2, five trials are shown from the system testing results. In the 12th experiment, it is very accurate with the information system scenario in the ESP-32 CAM column showing Lecturer: 1 and Student: 0; this indicates that there is one parking slot for lecturer parking while students are complete, which can be seen in the LCD column with Lec: 1, Stu: Full in line 1, while in line 2 displays the running text "Students, please park in another building" which is given a text speed delay of 500 ms so that the text runs neither too slow nor too fast.

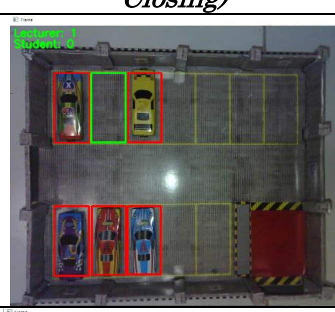
In this 16th experiment, it was very accurate with the information system scenario in the ESP-32 CAM column showing Lecturer: 0 and Student: 0; this indicates that the parking lot for lecturers and students is whole, which can be seen in the LCD column with Lec, Stu: Full on row 1, while line 2 displays the running text "Lecturers and Students, please park in another building" which is given a text speed delay of 500 ms.

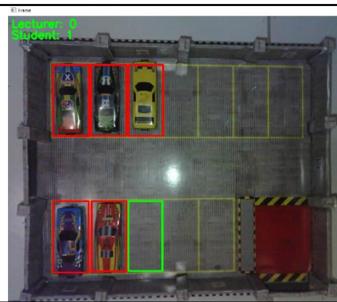
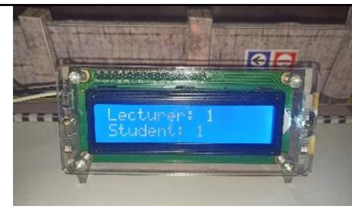
In the 31st experiment, it is very accurate with the information system scenario in the ESP-32 CAM column showing Lecturer: 0 and Student: 1; this indicates that the lecturer's parking lot is whole, while students have one slot available for parking which can be seen in the LCD column with Lec: Full, Stu: 1 in line 1, while in line 2 displays the running text "Lecturers, please park in another building" which is given a text speed delay of 500 ms.

In the 49th experiment with the information system scenario in the ESP-32 CAM column, it shows Lecturer: 1 and Student: 1; this indicates that there is one parking slot for lecturers parking, while students have one slot for parking which can be seen in the LCD column with Lecturer: 1 in row 1. In contrast, row 2 displays Student: 1. The 49th experiment was inaccurate because the miniature lecturer and student cars needed to be parked neatly or according to the parking line markings.

In the 50th experiment with the information system scenario in the ESP-32 CAM column, it shows Lecturer: 1 and Student: 1; this indicates that there is one parking slot for lecturers parking, while students have one slot for parking which can be seen in the LCD column with Lecturer: 1 in row 1. In contrast, row 2 displays Student: 1. The 50th experiment is inaccurate because the miniature lecturer and student cars need to be parked neatly or according to the parking line markings.

Table 2. System Testing Results

Test	*ESP-32 CAM + Open-CV (Canny + Closing)*	*LCD 16x2 I2C*
12		
16		

31		
49		
50		

5. **Conclusions.** Based on the results of testing the IoT and Digital Image Processing systems that have been carried out, it can be concluded that:

1. Image processing is based on four parameters: min threshold = 100, max threshold = 400, min pixels = 0, and max pixels = 225. The threshold is high enough to eliminate noise in miniature parking lots, while pixels are adjusted based on the lighting conditions during system testing so that image objects can be appropriately detected.
2. Application of the Canny Edge Detection method followed by Closing morphology based on parameters is better than the application of the Canny Edge Detection method alone because it can help remove noise in the image and can also improve edge detection that is not good in some parts of the image.
3. In testing the wireless system by applying the Canny Edge Detection method followed by Closing morphology, the results of observations are very consistent with the system based on Black Box Testing. The test gets a delay of 5.61 seconds. The delay could be better.
4. In some of the experiments that were carried out, they experienced delays of more than 5 seconds because the network conditions were unstable or experienced high pings or pings of more than 150 ms. The ideal ping within a network for cloud services should be 1-50 milliseconds (ms).
5. The 49th and 50th trials were inaccurate because the miniature lecturer and student cars were not parked neatly or according to the parking line markings, which resulted in the information system not matching the actual situation.

REFERENCES

[1] Akhirul, Y. Witra, I. Umar, and Erianjoni, "Dampak Negatif Pertumbuhan Penduduk Terhadap Lingkungan Dan Upaya Mengatasinya," 2020.

[2] L. Said, H. Maryam, and Nasruddin, *PENGARUH PERTUMBUHAN KENDARAAN DAN KAPASITAS JALAN TERHADAP KEMACETAN DI RUAS JALAN PERINTIS KEMERDEKAAN*. 2020. doi: 10.31219/osf.io/kpw6e.

[3] Listifadah and Hartono, "Studi Penataan Parkir di Wilayah Central Bussines District Kabupaten Pati," *J. Penelit. Transp. Darat*, vol. 21, no. 2, pp. 165–182, 2019, doi: 10.25104/jptd.v21i2.1384.

[4] M. Adnan and R. N. Rohmah, "Pemanfaatan Dua Mikrokontroller Platform IoT dalam Pengembangan Sistem Parkir," *Emit. J. Tek. Elektro*, vol. 20, no. 2, pp. 122–127, 2020, doi: 10.23917/emitor.v20i02.11023.

[5] D. S. Azkarika, "Monitoring Online Ketersediaan Slot Parkir Berbasis Kamera Via Website Slot Based on Camera Via Website Using," 2017.

[6] J. Trivedi, M. S. Devi, and D. Dhara, "Canny edge detection based real-time intelligent parking management system," *Sci. J. Silesian Univ. Technol. Ser. Transp.*, vol. 106, no. February, pp. 197–208, 2020, doi: 10.20858/sjsutst.2020.106.17.

[7] I. M. E. Listartha *et al.*, "IoT -Parking Lot Detection Based on Image Processing," *J. Sist. dan Inform.*, vol. 12, pp. 168–176, Jul. 2020.

[8] F. Pradana Putra and I. Susilawati, "Prototipe Sistem Deteksi Ketersediaan Lahan Parkir Menggunakan Metode Algoritma Canny Edge," pp. 1–6, 2021.

[9] R. A. Ramadhan, "Aplikasi Edge Detection untuk Sistem Informasi Parkir Mobil," 2019.

[10] Mariza Wijayanti, "Prototype Smart Home Dengan Nodemcu Esp8266 Berbasis Iot," *J. Ilm. Tek.*, vol. 1, no. 2, pp. 101–107, 2022, doi: 10.56127/juit.v1i2.169.

[11] Aprylia, "Smart House Berbasis Web Server Menggunakan Esp 32 Sebagai Door Lock Menggunakan Face Lock," *J. Pembang. Wil. Kota*, vol. 1, no. 3, pp. 82–91, 2021.

[12] M. Natsir, D. B. Rendra, and A. D. Y. Anggara, "Implementasi IOT Untuk Sistem Kendali AC Otomatis Pada Ruang Kelas di Universitas Serang Raya," *J. PROSISKO (Pengembangan Ris. dan Obs. Rekayasa Sist. Komputer)*, vol. 6, no. 1, pp. 69–72, 2019.

[13] A. Lazaro, J. L. Buliali, and B. Amaliah, "Deteksi Jenis Kendaraan di Jalan Menggunakan OpenCV," *J. Tek. ITS*, vol. 6, no. 2, 2017, doi: 10.12962/j23373539.v6i2.23175.

[14] M. A. AFANDI, F. FADHLAN, R. A. ROCHMANTO, and H. WIDYANTARA, "Perangkat Budidaya Microgreen berbasis Internet of Things," *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 10, no. 3, p. 581, 2022, doi: 10.26760/elkomika.v10i3.581.

[15] A. H. Pratomo, W. Kaswidjanti, and S. Mu'arifah, "Implementasi Algoritma Region of Interest (ROI) Untuk Meningkatkan Performa Algoritma Deteksi Dan Klasifikasi Kendaraan," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 7, no. 1, pp. 155–162, 2020, doi: 10.25126/jtiik.202071718.

[16] I. D. Kurniawati and A. Kusumawardhani, "Implementasi Algoritma Canny dalam Pengenalan Wajah menggunakan Antarmuka GUI Matlab," *J. Inst. Teknol. Sepuluh Nop.*, no. December, pp. 3–8, 2017.

[17] A. Susanto, "Penerapan Operasi Morfologi Matematika Citra Digital Untuk Ekstraksi Area Plat Nomor Kendaraan Bermotor," *Pseudocode*, vol. 6, no. 1, pp. 49–57, 2019, doi: 10.33369/pseudocode.6.1.49-57.