EPI International Journal of Engineering
Volume 4, Number 1, February 2021, pp. 14-20
DOI: 10.25042/epi-ije.022021.03

*p*ISSN 2615-5109
*e*ISSN 2621-0541

A Low-Cost Vehicle Counting System based on the Internet of Things

Lanny Sitanayah[a], Apriandy Angdresey[b,*], Jeri Wahyu Utama[c]

[a]Department of Informatics Engineering, Faculty of Engineering, Universitas Katolik De La Salle, Manado - Indonesia.
Email: lsitanayah@unikadelasalle.ac.id
[b]Department of Informatics Engineering, Faculty of Engineering, Universitas Katolik De La Salle, Manado - Indonesia.
Email: aangdresey@unikadelasalle.ac.id
[c]Department of Informatics Engineering, Faculty of Engineering, Universitas Katolik De La Salle, Manado - Indonesia.
Email: 15013071@unikadelasalle.ac.id

Abstract

In urban areas where land for parking is very limited, drivers often waste time, fuel, and emissions circling around without information if unoccupied parking spaces are available or not. In this paper, we design and implement a low-cost wireless system to count the number of cars and motorcycles in a parking lot. The system consists of two sensor devices, which are installed at an entrance gate and an exit gate of a parking lot. Each device has a NodeMCU ESP8266, an HC-SR04 ultrasonic sensor, and an MPU-92/65 accelerometer. REST API is used as the web service to connect sensor devices and users, that will access the parking information using a web browser. The C4.5 algorithm is utilized to construct a decision tree to classify detected objects like cars, motorcycles, or people. We show by experiment that our sensor devices and the wireless monitoring system work correctly.

Keywords: C4.5 algorithm; data mining; decision tree; sensor device

1. Introduction

Population growth increases the need for land transportation, especially for the four-wheeled vehicles (cars) and the two-wheeled ones (motorcycles). This happens as people have to commute every day. In urban areas where land for parking is very limited, it is difficult to find parking spaces, especially during busy hours. Parking lots owned by shopping centers or business premises are either attended by operators or equipped with vehicle parking systems. However, some parking lots which are dedicated to public places – for example, sports arenas and parks – are usually unattended and do not have parking systems. In this case, drivers often waste time, fuel, and emissions circling around such parking lots without information on whether unoccupied parking spaces are available or not.

To solve this problem, an inexpensive sensor-based solution can be used, thanks to the Internet of Things (IoT) technology. In IoT, a device with one or more sensors, batteryless [1] or with batteries, can be used to sense physical phenomena, such as in [2 - 5] and send data wirelessly for further processing. Some works in the literature that detect the presence of vehicles have been done. In [6], they use the background subtraction method from Open-CV to detect, count and classify objects. To detect moving objects, they use motion detection functions based on video recordings from cameras. In [7], a car detection system for an IoT-based toll payment is built. With an objective to reduce congestion, the government has implemented the Radio Frequency Identification (RFID)-based electronic transaction payments on several tolls as an alternative. In this research, every vehicle is installed with a receiver that is connected to one user data. This is used to detect the position of the vehicle when it enters the toll gate without obstructing other vehicles. Another research to find unoccupied parking spaces in off-street parking lots is presented in [8]. They utilize ultrasonic sensors and IoT technology using ESP-8266 to provide information regarding the position and status of the parking lots.

Different from the previous works, in this paper, we propose a vehicle counting system based on IoT. Our contributions in this paper are summarized as follows.

1) We design and implement a low-cost wireless system to count the number of cars and motorcycles in a parking lot. The system consists of two sensor devices – one device is deployed at an entrance of a parking lot and the other one is deployed at an exit. Each device has a NodeMCU ESP8266, an HC-SR04 ultrasonic sensor, and an MPU-92/65 accelerometer.

*Corresponding author. Tel.: +62-813-9807-8886
Universitas Katolik De La Salle
Manado, Indonesia

2) We use the C4.5 algorithm to construct a decision tree, which is used to classify detected objects as cars, motorcycles, or people.

Part of this work has been published as a poster abstract in [9]. The rest of this paper is organized as follows. Section 2 reviews some works in the literature relevant to ours. Section 3 shows our system design and implementation, including hardware and software. Section 4 describes the object classification method used in this paper. Section 5 shows our performance evaluation, and finally Section 6 concludes the paper.

2. Related Works

A system to count the number of vehicles in an open parking area in real-time using the Open-CV background subtraction method is proposed in [6]. This motion detection function is used to detect moving objects based on video recordings from a camera. It is made using functions available in the Open-CV library through the C++ Application Programming Interface. This study uses several parameters, one of which is *learningRate*. The value of the *learningRate* parameter is based on environmental conditions and determined from the test results of the *learningRate* parameter variation. In this study, they also classify vehicles, *i.e.* cars or motorcycles, using the coordinate data of the height and width to determine the size of the detected vehicles.

In [7], a car detection system for toll payment based on the Internet of Things is proposed. The purpose of this research is to reduce congestion at the toll entrance gate. The Internet of Things technology is utilized since the Radio Frequency Identification (RFID)-based electronic transaction payment system cannot reduce the congestion problem significantly. On the Internet of Things-based system, each vehicle is equipped with a receiver connected to one user data. Hence, the system is able to detect the position of the vehicle when entering the toll gate without obstructing other vehicles. This transaction process can be carried out at a radius of 10 meters from the toll gate without stopping vehicles with an average speed of 10 km/hour. In the implementation, the time required is around 10 seconds for the toll entrance gate and 15 seconds for the toll exit gate.

In [8], an application to track or find empty parking spaces in off-road parking lots by utilizing ultrasonic sensors and the Internet of Things is developed. To track or find an empty parking space, they use an ultrasonic sensor and an ESP-8266 chip, which can provide information about the position and status of a parking space, whether it is filled or empty.

Data mining is a process of dredging or gathering important information from a large amount of data, resulting in several models to identify patterns between attributes that exist in the data set. Classification is a process in data mining that aims to find valuable patterns from a relatively large amount of data, build a model that describes and differentiates the classes or concepts for future prediction. For example, in [10], the authors use large amounts of ad-hoc vehicle's real-time traffic data to predict traffic congestion in a city by using classification and clustering methods. This study shows that traffic congestion can be predicted more accurately and the percentage of accuracy can be increased by increasing the size of the training data.

In [2], the authors created a wireless system based on the Internet of Things to monitor and predict water quality in public swimming pools. They utilize a data mining prediction model, namely the Iterative Dichotomiser 3 decision tree algorithm. In [11], data mining is used to identify the sex of a tarantula by classifying the training data obtained for almost a year.

The C4.5 algorithm is one of the algorithms used for classification or segmentation and is predictive. This algorithm is the development of the Iterative Dichotomiser 3 (ID3) algorithm and hence they have the same basic principles. C4.5 is used in [12] to determine a model for predicting soil quality by using training data on soil composition. The experimental results obtained show that the expression of knowledge using the C4.5 algorithm is easy to understand, improves forecasting accuracy, and provides a reliable theoretical basis for precision fertilization. Another example of C4.5 utilization is to minimize the workload of student admissions in planning school promotions and improving planning strategies by classifying demographics [13]. In addition, the C4.5 algorithm is also used to identify diseases from existing symptoms [14].

In this work, the authors identify the possibility of indications of diabetes insipidus apart from gestational diabetes insipidus at productive age (students). They use the *k*-means algorithm to classify the symptoms, and the C4.5 algorithm to identify the type of diabetes. Furthermore, in [15], C4.5 is used to analyze catfish and carp diseases to determine the history of fish disease relationships, which will be used to identify fish diseases.

3. System Design and Implementation

The wireless vehicle counting system is illustrated in Fig. 1. We attach two sensors – an HC-SR04 ultrasonic sensor and an MPU-92/65 accelerometer – to each sensor device. We use the ultrasonic sensor to measure the physical distance to an object in front of it, while the accelerometer is used to measure any vibration resulted from the moving object. The distance and vibration data from the sensors is sent to a web server wirelessly using the NodeMCU ESP8266's ESP-12E Wi-Fi module. This data is received by REST API and forwarded to the application.

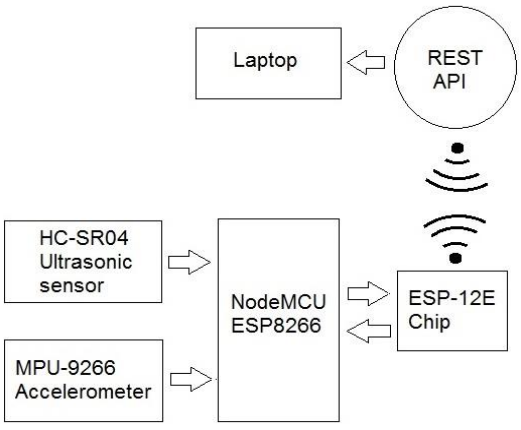

Figure 1. The wireless vehicle counting system

Table 1. Hardware and software

Hardware/Software	Purpose
NodeMCU ESP8266	Gather and send data
HC-SR04	Ultrasonic sensor
MPU-92/65	Accelerometer
Laptop/Mobile Device	Access data from the web server
Arduino Program	Gather and send data
PHP Program	Process data

In this work, we use two sensor devices, where we deploy one device at an entrance of a parking lot and the other one is at an exit. Table 1 lists the hardware and software components used in the system.

The total cost of the main components, i.e. a NodeMCU ESP8266, an HC-SR04 ultrasonic sensor and an MPU-92/65 accelerometer for one sensor device in our system is approximately US$ 8. Compared to [6] that uses a webcam and a Raspberry Pi 3 model B, the total cost is around US$ 43. The system in [7] requires that each vehicle must be equipped with a receiver, while the system in [8] requires that each parking space must be monitored by one sensor device. Even though the total cost for a receiver in [7] that consists of a NodeMCU ESP8266, a buzzer, and an LCD 16x2 is only around US$ 4, they must take into account the number of cars that use the system. Similarly, even though the total cost for a sensor device in [8] that consists of an ESP8266 D1 mini, an HC-SR04 ultrasonic sensor, and an LCD 16x2 is around US$ 4, they must provide as many devices as the number of parking spaces. Our system, on the other hand, only requires two devices and hence its total cost is the lowest. The prices reported in this paper are calculated based on prices from www.tokopedia.com.

3.1. Hardware

We show the hardware design of our sensor device in Fig. 2. Each sensor device has a NodeMCU ESP8266, an HC-SR04 ultrasonic sensor, and an MPU-92/65 accelerometer. NodeMCU and the ultrasonic sensor are attached to a breadboard, while the accelerometer is tied to a rope as a physical medium to carry vibrations. Both NodeMCU and the ultrasonic sensor are powered by external batteries, while the accelerometer gets power from NodeMCU. To connect NodeMCU and the accelerometer, we connect NodeMCU's D5 pin to the accelerometer's SDA pin, D6 pin to SCL pin, GND pin to GND pin, and 3V pin to VCC pin. Then, to connect NodeMCU and the ultrasonic sensor, we connect NodeMCU's D7 pin to the ultrasonic sensor's echo pin, while D8 pin is connected to the trigger pin. The pins configuration is summarized in Table 2 and the hardware implementation is shown in Fig. 3.

Table 2. Pins configuration

NodeMCU	Accelerometer	Ultrasonic Sensor
D5 pin	SDA pin	-
D6 pin	SCL pin	-
D7 pin	-	Echo pin
D8 pin	-	Trigger pin
GND	GND	-
3V	VCC	-

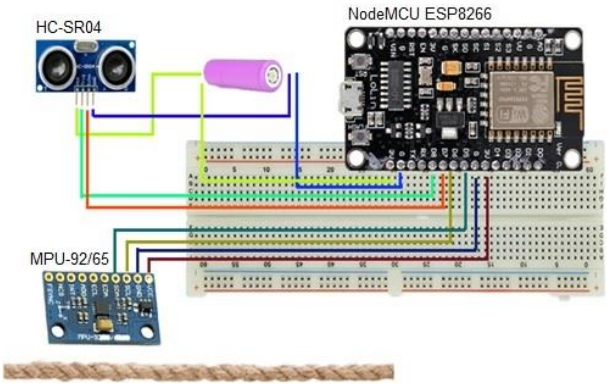

Figure 2. Hardware design

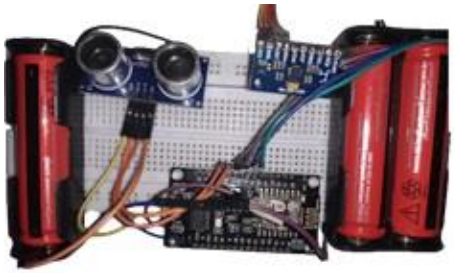

Figure 3. Hardware implementation

3.2. Software

The wireless vehicle counting system is divided into the client-side and the server-side. At the client-side, we have an Arduino program that is uploaded to NodeMCU of both sensor devices. The client flowchart is shown in Fig. 4. First, the Arduino program checks if all sensors are connected to NodeMCU and whether the Internet connection is available. Then, the Arduino program reads the sensors' data and sends it wirelessly to the webserver.

At the server-side, we have REST API as the back end and a PHP (Hypertext Preprocessor) program as the front end. The server flowchart is depicted in Fig. 5.

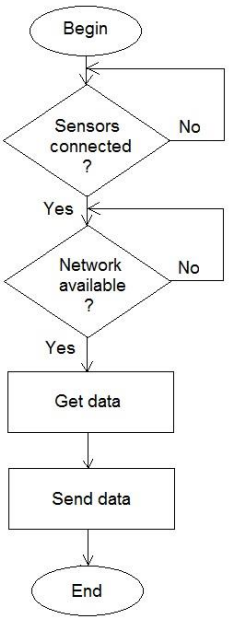

Figure 4. Client flowchart

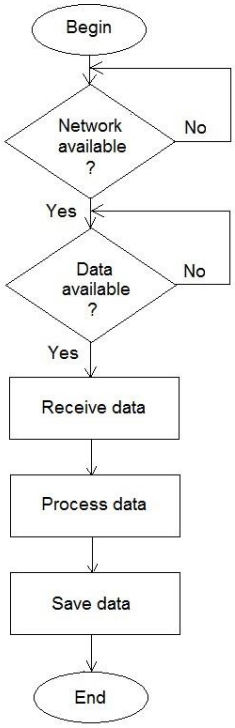

Figure 5. Server flowchart

First, if the Internet connection is available, we check if data is successfully received. If the data is available, it is processed and then stored in the database. To access the parking information from the web server, users can use a web browser. We utilize the C4.5 algorithm to construct a decision tree in order to classify objects as cars, motorcycles, or people. Then, we count how many cars and motorcycles that enter and exit the parking lot. The calculation result is displayed to users as the number of free parking spaces.

4. Object Classification Method

In this section, we present some classified objects based on data obtained from the sensor devices. We utilize the C4.5 algorithm for object classification. C4.5 is a method used for classification problems in machine learning and data mining. Its objective is for grouping and it is predictive for finding patterns with relatively large to very large values. In general, C4.5 has the following steps to build a decision tree: (1) find an attribute to be the root, (2) create a branch for each value and divide cases into branches, and (3) repeat the process for each branch until all cases on the branch have the same class. To choose a variable as the root, we calculate and select the highest gain using the following formula:

$$Gain(S,A) = Entropy(S) - \sum_{i=1}^{n} \frac{|S_i|}{|S|} * Entropy(S_i) \quad (1)$$

$$Entropy(S) = \sum_{i=1}^{n} P_i * Log_2(P_i) \quad (2)$$

where S is a set of cases, A is the attribute, and n is the number of partitions of attribute A. Furthermore, the number of cases in the i-th partition is $|S_i|$ and the number of cases in S is $|S|$. Before we calculate gain, we need to calculate entropy (see Equation 2) to determine how informative an input variable is, where P_i is the proposition of S_i to S.

Table 3 shows some of the raw data taken using the ultrasonic sensor and the accelerometer. Distances acquired by the ultrasonic sensor are measured in centimeters. The three-axis accelerometer data on the X-axis is denoted by A_x, on the Y-axis is denoted by A_y, and on the Z-axis is denoted by A_z. We also record the time when the data is captured and the class of objects to be classified. The class is divided into car, motorcycle, and person. Then, from the raw data that we obtained, we further clean and transform the data into categories in order to simplify the algorithm's calculations.

The following range formula is used to create categories for each variable:

$$Range(S) = \frac{X_{\max} - X_{\min}}{n} \quad (3)$$

Table 3. Raw data

Distance	A_x	A_y	A_z	Time	Class
92	-1.80	3.39	3.11	16:03:36	Person
72	-1.66	3.47	3.13	16:03:37	
88	-1.24	3.42	3.17	16:06:29	Car
77	-1.46	3.31	3.27	16:06:30	
79	-1.58	3.27	3.09	16:06:30	
73	-1.63	3.66	3.15	16:09:29	Car
72	-1.59	3.43	2.97	16:09:30	
72	-1.58	3.31	3.02	16:09:30	
94	-1.40	3.44	3.06	16:11:33	Person
95	-1.79	3.43	3.39	16:13:38	Motorcycle
85	-1.59	3.29	2.89	16:13:38	
74	-1.63	3.54	3.08	16:13:39	
95	-1.62	3.20	3.09	16:14:11	Motorcycle
80	-1.71	3.47	3.02	16:14:11	
90	-1.57	3.71	3.02	16:15:40	Car
80	-1.53	3.23	2.79	16:15:40	
78	-1.53	3.27	3.12	16:15:41	
76	-1.64	3.82	3.25	16:15:41	
76	-1.73	3.31	3.17	16:15:41	
79	-1.65	3.25	3.11	16:17:23	Motorcycle
89	-1.73	3.32	3.06	16:17:24	
95	-1.72	3.19	3.21	16:17:24	
80	-1.69	3.25	3.09	16:18:11	Person
81	-1.73	3.26	2.93	16:18:12	

Table 4. Clean data

Distance	A_x	A_y	A_z	Time	Class
Medium	Low	Medium	Medium	Low	Person
Medium	High	Medium	High	Medium	Car
Near	Medium	High	High	High	Car
Far	Low	Medium	Low	Low	Person
Medium	Low	High	Medium	Medium	Motorcycle
Far	Medium	Medium	Low	Low	Motorcycle
Medium	Medium	High	High	High	Car
Far	Medium	Low	High	Medium	Motorcycle
Near	Medium	Low	Low	Low	Person

Table 5. Entropy and gain

Variable	Category	Total (S)	Person (S_i)	Motorcycle (S_i)	Car (S_i)	Entropy	Gain
Distance	Near	2	1	0	1	0	0.91829
	Medium	4	1	1	2	1.5	
	Far	3	1	2	0	0	
A_x	Low	3	2	1	0	0	0.91829
	Medium	4	1	2	2	1.5	
	High	2	0	0	1	0	
A_y	Low	2	1	1	0	0	0.91829
	Medium	4	2	1	2	1.5	
	High	3	0	1	1	0	
A_z	Low	3	2	1	0	0	1.36274
	Medium	2	1	1	2	1	
	High	4	0	1	1	0	
Time	Low	4	3	1	0	0	1.5849
	Medium	3	0	2	1	0	
	High	2	0	0	2	0	
Total		9	3	3	3	1.584962	

where X_{max} is the highest reading of a variable, X_{min} is the lowest reading of that variable, and n is the number of categories that we want. In this paper, we use n = 3. For example, the categories for the distance variable are near (70.23 cm - 78.153 cm), medium (78.153 cm - 86.076 cm), and far (86.076 cm - 94.00 cm). After that, for each variable, we calculate the average value for each available data to decide in which categories the data belongs. Table 4 shows the data that has been cleaned.

A decision tree is then created after calculating the total entropy, entropy of each variable, the gain, and the highest gain. In this case, the total entropy = 1.5849625 and gain = 0.918295834. The calculation results of the entropy and gain are shown in Table 5.

5. Performance Evaluation

To evaluate our proposed system, including both hardware and software, we conduct a series of experiments, which are described in the following. The parameters used in the evaluation are given in Table 6.

Table 6. Evaluation parameters

Parameter	Value
Time interval between objects	≥ 1 sec
Object's speed	≤ 10 km/hour
Sensor device's height	0.5 m
Sensor device's distance	≤ 4 m

5.1. Experimental setup

In this research, we do not detect the direction of vehicles. Therefore, we assume that cars and motorcycles enter the parking lot through the entrance gate and leave through the exit gate. To count the number of cars and motorcycles, we utilize two sensor devices – one at an entrance to sense them entering the parking lot and one at an exit to sense them leaving. Each sensor device is mounted at a certain height above the ground as shown in Fig. 6. The height is carefully chosen, *i.e.* 0.5 m, so that the ultrasonic sensor can sense moving objects in front of it. As for the sensor device's distance to the moving object, we use the HC-SR04 ultrasonic sensor detection range, *i.e.* from 2 centimeters to 4 meters.

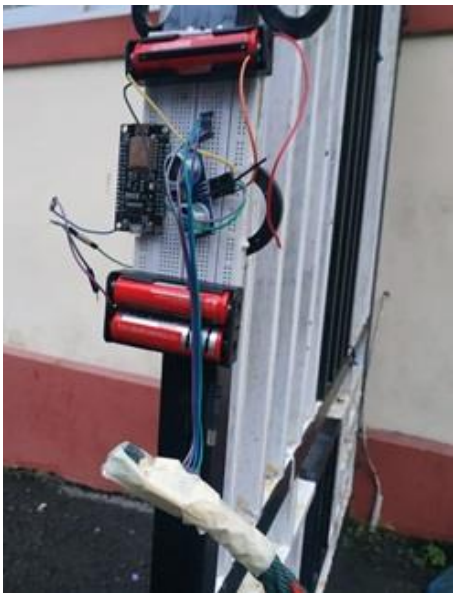

Figure 6. Sensor device installation

The accelerometer is attached to a rope that is put on the ground across the road. Therefore, when vehicles' wheels touch the rope, the accelerometer can sense the vibration.

Furthermore, we also assume that two moving objects are at least one second apart. Otherwise, they are detected as one object. The moving speed of an object is at most 10 km/hour as it is the maximum speed allowed for vehicles in the parking area. An object is detected when the readings of the ultrasonic sensor and the accelerometer change. When this happens, the sensor readings are sent to the webserver. To classify the detected objects, we use a decision tree which is constructed using the C4.5 algorithm. Our system classifies the objects as either cars, or motorcycles, or people. When the system detects a car or a motorcycle, it updates its counter and users can get the parking information using a web browser.

5.2. Experimental Results and Discussion

First of all, we test the hardware that has been built, whether all sensors can function properly and the device can send data to the webserver. We also test if the webserver can receive the data. The results of our experiment show that all sensors work properly, the device can send data and the web server can receive the data as shown in Fig. 7.

```
gx : -4.64, gy : 3.26, gz : 2.67 us : 45
gx : -1.91, gy : 2.09, gz : 2.65 us : 75
gx : -1.36, gy : 2.11, gz : 2.89 us : 76
gx : -1.96, gy : 1.76, gz : 1.96 us : 51
gx : -2.31, gy : 0.73, gz : 3.23 us : 40
gx : -1.69, gy : 3.47, gz : 1.33 us : 74
gx : -0.64, gy : 1.45, gz : 2.76 us : 69
gx : -2.23, gy : 2.97, gz : 1.73 us : 75
gx : -1.13, gy : 2.40, gz : 3.01 us : 68
gx : -0.05, gy : 2.84, gz : 3.45 us : 76
```

Figure 7. Sensor data received at the webserver

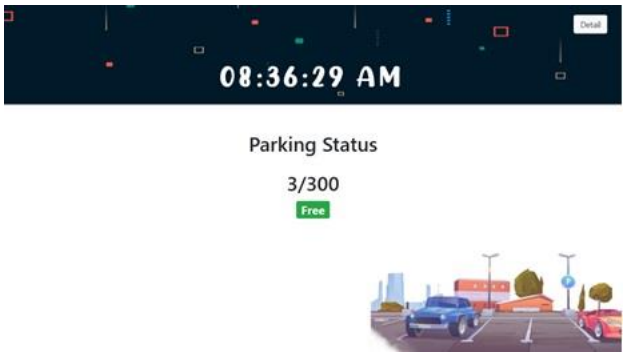

Figure 8. Application's main page in a web browser

Figure 9. Application's parking details in a web browser

To access the parking information from the web server, users can use a web browser. We use three web browsers for testing purposes. They are Google Chrome Version 84.0.4147.89, Mozilla Firefox version 76.0.1, and Microsoft Edges version 84.0.522.40. The application's main page in a web browser is depicted in Fig. 8 and the parking details are shown in Fig. 9. Figure 8 is the main page of the application. On this page, there is a digital clock at the top of the application and the parking information in the middle, which indicates the status of the parking lot. Free means there are unoccupied parking spaces, whereas full means all parking spaces are occupied. Furthermore, Figure 9 shows parking details that contain detailed information from the acquired data. On this page, there is a bar chart that informs us how many vehicles (cars and motorcycles) enter and leave the parking lot.

6. Conclusion and Future Work

This paper proposes a low-cost Internet of Things-based vehicle counting system to count the number of cars and motorcycles in a parking lot. Two sensor devices are required in this system – one at an entrance gate and one at an exit gate of a parking lot. Each sensor device consists of a NodeMCU ESP8266, an HC-SR04 ultrasonic sensor and an MPU-92/65 accelerometer. We use REST API as the web service to connect sensor devices and users. We use C4.5 to build a decision tree to classify detected objects like cars, motorcycles, or people. We show by experiment that the sensor devices and the system work correctly.

Due to the COVID-19 pandemic and social restrictions that has been in place when we carried out the evaluation, we were unable to conduct a thorough experiment. We will continue to acquire more data over a

longer period of time once we are able to do so. We also plan to deploy more sensor devices at more entrance and exit gates.

References

[1] J. Hester, N. Tobias, A. Rahmati, L. Sitanayah, D. Holcomb, K. Fu, W. Burleson, and J. Sorber. Persistent Clocks for Batteryless Sensing Devices. *ACM Transactions on Embedded Computing Systems*, 15:1–28, Aug. 2016.

[2] A. Angdresey, L. Sitanayah, and V.J.A. Sampul. Monitoring and Predicting Water Quality in Swimming Pools. *EPI International Journal of Engineering*, 3(2):119–125, 2020.

[3] L. Sitanayah, A. Angdresey, and E. Kristalino. Poster Abstract: A Sensor-based Application for Road Conditions Detection. In *The 18th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, pages 651–652, 2020.

[4] J.B. Sanger, L. Sitanayah, and V.D. Kumenap. Detection System for Cigarette Smoke. In *The 4th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, pages 145–149, 2019.

[5] J.B. Sanger, L. Sitanayah, and I. Ahmad. A Sensor-based Garbage Gas Detection System. In *The 11th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 1359–1364, 2021.

[6] M. Hariyanto, A. Sofwan, and A. Hidayatno. Perancangan Sistem Penghitung Jumlah Kendaraan Pada Area Parkir Dengan Metode Background Subtraction Berbasis Internet of Things. *Transient: Jurnal Ilmiah Teknik Elektro*, 7(3), 2019.

[7] A.R. Muzakhir, M. Imroba, and A.G. Putrada. Sistem Deteksi Mobil pada Aplikasi Pembayaran Gerbang Tol Berbasis Internet of Things. In *SMARTICS Journal*, volume 5, pages 97–103, 2019.

[8] A.D. Limantara, Y.C.S. Purnomo, and S.W. Mudjanarko. Pemodelan Sistem Pelacakan Lot Parkir Kosong Berbasis Sensor Ultrasonik dan Internet of Things (IoT) Pada lahan Parkir Di Luar Jalan. In *Prosiding Semnastek*, 2017.

[9] L. Sitanayah, A. Angdresey, and J.W. Utama. Poster: An Internet of Things-based Vehicle Counting System. In *The International Conference on Embedded Wireless Systems and Networks (EWSN)*, 2021.

[10] A. Angdresey and P.K. Sahoo. Analysis of VANET Big Data for Predicting Traffic Congestion in a City. In *The 12th Workshop on Wireless Ad Hoc and Sensor Networks (WASN)*, 2016.

[11] A. Angdresey and M. Wongkar. Identification of The Reproductive Apparatus of Tarantula Genus Brachypelma Using Linear Discriminant Analysis Method. In *International Conference on Electrical Engineering and Computer Science (ICECOS)*, pages 409–414, 2018.

[12] L. Dongming, L. Yan, Y. Chao, L. Chaoran, L. Huan, and Z. Lijuan. The Application of Decision Tree C4.5 Algorithm to Soil Quality Grade Forecasting Model. In *The 1st IEEE International Conference on Computer Communication and the Internet (ICCCI)*, pages 552–555, 2016.

[13] S.A. Abaya and B.D. Gerardo. An Education Data Mining Tool for Marketing Based on C4.5 Classification Technique. In *The 2nd International Conference on E-Learning and E-Technologies in Education (ICEEE)*, pages 289–293, 2013.

[14] A. Angdresey and J. Matrutty. Implementasi Algoritma *k*-Means dan C4.5 Dalam Mengidentifikasi Diabetes Insipidus Pada Usia Produktif. Studi Kasus: Mahasiswa Unika De La Salle Manado. In *SeminarNasional Riset dan Teknologi Terapan Ke-8 (RITEKTRA VIII)*, pages 515–520, 2018.

[15] Kusrini Sucipto and E.L. Taufiq. Classification Method of Multi-Class on C4.5 Algorithm for Fish Diseases. In *The 2nd International Conference on Science in Information Technology (ICSITech)*, pages 5–9, 2016.