ISSN 1816-6075 (Print), 1818-0523 (Online)
Journal of System and Management Sciences
Vol.12 (2022) No.1, pp.53-62
DOI:10.33168/JSMS.2022.0105

Advanced Smart Parking Management System Development Using AI

In Hwan Jung [1], Jae-Moon Lee [2] and Kitae Hwang [1]

School of Computer Engineering, Hansung University, Korea

ihjung@hansung.ac.kr

Abstract. This paper is aimed to introduce a smart parking lot management system using multiple cameras and artificial intelligence technique. When a vehicle enters a parking lot, it recognizes the vehicle number using embedded camera, tracks which parking space the vehicle is parked in, and updates parking space information. In addition, using a surveillance camera images, it has been also implemented to detect collision accidents that may occur while the vehicle is moving in the parking lot. Vehicle number recognition system uses OCR technique and is implemented on a Raspberry system. By managing the vehicle number recognized at the entrance of the parking lot as an Object ID, it was possible to effectively track the vehicle as a moving object inside the parking lot and finally identify the parking location. For accident detection, YOLO with CNN deep learning process is used. More than 500 possible collision images are trained in advance. Experimental results show that the detection accuracy of parking and accident detection increases as the number of training images increases. The accident detection needed more training images because it has more diversity. By using the smart parking system implemented in this paper, it is possible to effectively manage the vehicle's parking location, free space information and possible accidents. Using a cloud system, implemented system can provide drivers an integrated parking lot information over large areas.

Keywords: Smart Parking, YOLO, Deep learning, Vehicle Tracking, Accident Detection

1. Introduction

Artificial intelligence technology such as You Only Look Once (YOLO) has been applied and used in various industries. This paper introduces a smart parking system using artificial intelligence. In a complex urban environment where the number of vehicles continues to grow, the vehicle driver has a time to find a parking lot and the traffic congestion increases during Peak Hours (Acharya et al., 2018). After entering the parking lot, it takes time to find parking space. In order to alleviate these issues, a camera-based PGI(Parking Guidance Information) system has been studied (Chen & Chang, 2011). Camera-based PGI System has three advantages (Bong et al., 2008; Ichihashi et al., 2009; True, 2007). Firstly, special hardware infrastructure is not necessary because a CCTV camera can cover large parking spaces. Secondly, the system can provide an accurate location that is essential for finding the vacant parking position. Thirdly, camera-based system can also be applied to the parking lot in the street or residential area. Thus, existing studies have focused on identifying parking spaces. As the cost of human resources for parking management increases, the unmanned parking lot is increasing. It is very important for the unmanned parking management to check free parking spaces. Therefore, a camera-based system is advantageous which can be confirmed for accurate free parking space detection at a low cost. However, these unmanned parking is vulnerable to accidents that may occur. In this respect, the camera-based parking management system needs additional features to detect an accident.

In this paper, in addition to checking parking space using camera, vehicle number recognition and YOLO technology is applied to identify the parking location of the vehicle. In addition, in this paper, collision accidents can also be detected, which may occur while the vehicle is moving inside the parking lot.

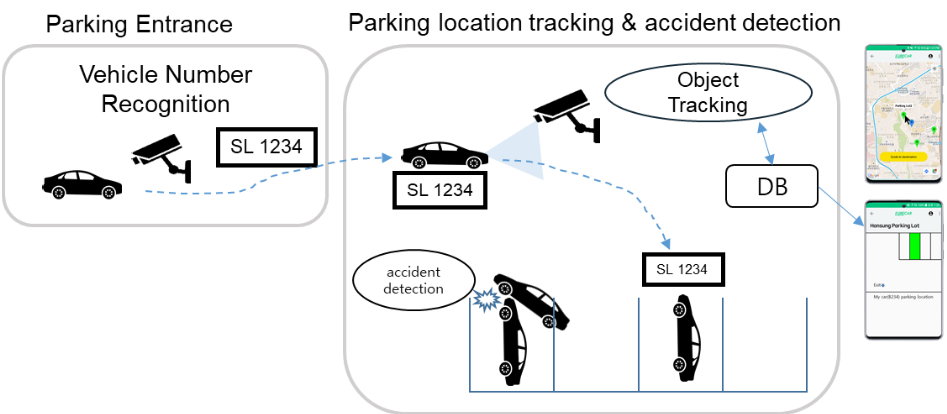

Fig. 1: Smart Parking Management System

2. Related Works

2.1. YOLO

YOLO, which has been proposed by Joseph Redmon and others in 2016 (Redmond et at., 2016), is a real-time object detection system based on Convolutional Neural Network (CNN). On the Conference on Computer Vision and Pattern Recognition (CVPR) in 2017, Joseph Redmon and Ali Farhadi released YOLO v2 which has improved the algorithm’s accuracy and speed (Redmond & Farhadi, 2017). In April this year, Joseph Redmon and Ali Farhadi proposed the latest YOLO v3, which has further improved the performance on object detection (Redmond & Farhadi, 2018). In order to track vehicles inside a parking lot, a YOLO v3 is used in this paper.

2.2. Vehicle Number Recognition

Vehicle number recognition technology has already been used widely and, in general, deep learning techniques such as Faster R-CNN or YOLO are used (Yu et al., 2020). In this paper, since license plate recognition is not the main topic, rather than using deep learning technology, a simple Optical Character Recognition (OCR) method for extracting characters from images was used (OpenCV, n.d.).

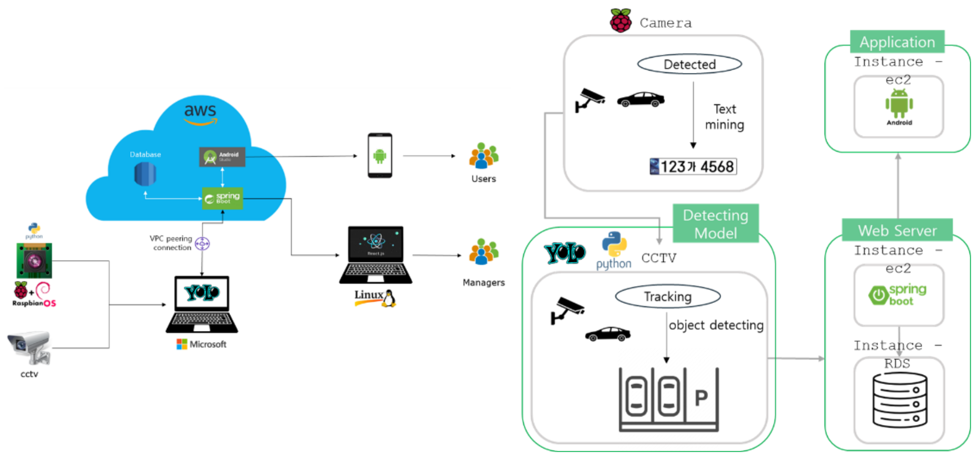

Fig. 2: System Architecture

3. Implementation of Vehicle Tracking and Parking Management

Figure 2 shows the overall system structure and Figure 3 depicts system flow from entrance to parking. At the entrance of a parking lot, a Raspberry Pi system with a camera is used to recognize vehicle number. Vehicle tracking and parking space management application exits for each parking lot. In order to provide parking lot information, such as exact location of parked car and number of free space, cloud system, AWS (Amazon, n.d.) is used. As a result, using cloud system, an integrated parking lot information provides a parking lot information for the drivers which is implemented as a smartphone application.

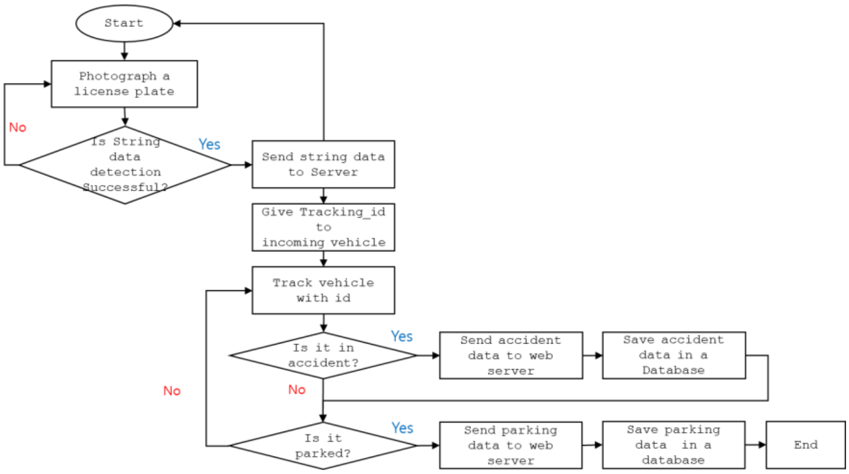

Fig. 3: System Flow

3.1. Vehicle Number Recognition

Figure 4 shows algorithm of vehicle number recognition. When a vehicle approaches an entrance, a Raspberry system detects it using ultrasonic sensors (Raspberrypia, n.d.). If a vehicle entrance or exit event is triggered by a sensor, using an attached camera, it captures plate image and processes for number recognition. First step is image processing for extracting text images. OpenCV library (OpenCV, n.d.) is used for border elimination and filtering. At second, a simple and convenient OCR library, Tesseract Open Source OCR engine (Tesseraact, n.d.), is used to convert images to numbers and characters. In this paper, a Python (Python, n.d.) is used for this vehicle number recognition which is implemented on Raspberry Pi system (Raspberrypib, n.d.).

```
Algorithm GetVehicleNumber() :
Input : CameraID, SensorID
Output : Vehicle Number as a String
-----
Distance = GetDistanceUsingUltrasonicSensor(SensorID)
If (Distance <= MinimumDistance)
Image Image1 = CapturePlate(CameraID);
Image Image2 = OpenCVImageProcessing(Image2); // Border Elimination and Filtering
String Number = pytesseract.image_to_string(Image2);
Return Number;
```

Fig. 4: Vehicle Number Recognition Algorithm

As show in Figure 3, if entrance management system successfully detects vehicle number, it passes the number to tracking application which is used as an object ID in YOLO tracking system. In order to send vehicle number to tracking system, a Message Queuing Telemetry Transport (MQTT) (MQTT, n.d.) publishing method is used.

3.2. Vehicle Tracking

Figure 5 shows the process of vehicle tracking using YOLO. The vehicle number is the ID of the object which is passed from entrance management system. In the Figure 5, vehicle A321 is parked in parking space 1. B234 number came in, parked in parking space 2, and it shows that the parking information database is updated.

seat_num	seat_condition	car_num
1	1	A321
2	0	None
3	0	None
4	0	None

seat_num	seat_condition	car_num
1	1	A321
2	1	B234
3	0	None
4	0	None

Fig. 5: Vehicle Tracking

In this paper, vehicle tracking was performed using YOLO v3. The vehicle number recognized at the entrance is transferred to the tracking system, and the tracking system uses the vehicle number as the object ID while applying the image captured by the CCTV camera to the YOLO system. After tracking vehicle while it is moving in the parking lot, if the vehicle stops at the individual parking space, the parking check process is executed. In order to confirm parking, it is necessary to learn the parking lot image in advance and store the parking space information in the YOLO system. To do this, it is necessary to learn images of empty and parked spaces for the entire parking space. In this study, a total of 100 parking images for 4 sample parking spaces were trained.

3.3. Accident Detection

In this paper, it is implemented to detect a collision accident that may occur while a vehicle is moving inside a parking lot. Collision detection was performed using a deep learning technique. In order to do this, more than 500 images corresponding to the case where the vehicle collided and the case where the vehicle did not collide were trained. In Figure 6, if the vehicle A321 is parked and B234 causes an accident while trying to park, the accident is detected using deep learning technique performed by tracking application.

As shown in Figure 3 system flow, when an accident occurs, the event is stored in the database and an alarm is sent to the smartphone app of the vehicle drivers. Figure 7 shows vehicle departure processing if there was an accident. If a vehicle, which was involved in an accident, is about to leave the parking lot, the entrance management system searches the database to see if the vehicle has been in an accident. If the vehicle is involved in an accident, the gate is not opened and the manager is notified.

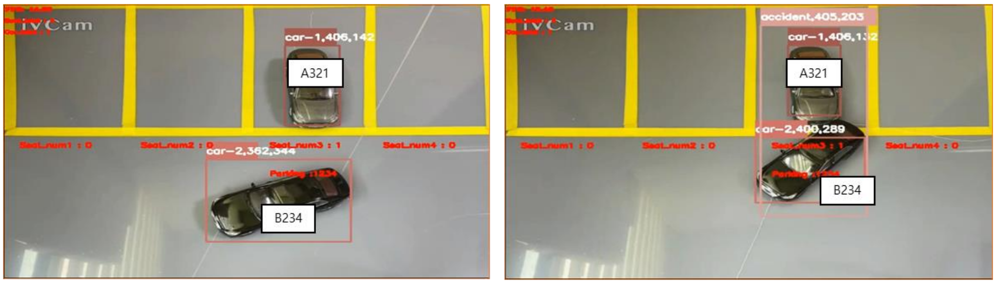

Fig. 6: Accident Detection

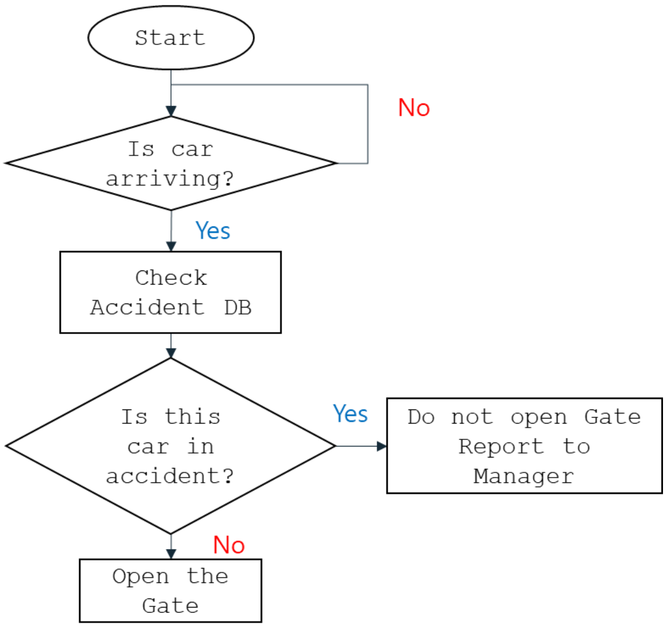

Fig. 7: Vehicle Departure Processing at the Gate

3.4. Parking Information App for Drivers

In this paper, a smart parking app for drivers is implemented for smartphones. The implemented app is shown in Figure 8. Figure 8(a) is the first screen. As shown in Figure 8(b), when the driver searches for the parking lot of the destination on the map, the parking lot information is displayed as shown in Figure 8(c). Figure 8(d) shows the result of parking completed. When a vehicle accident occurs, an alarm is displayed as shown in Figure 8(e). As shown in Figure 8(f), it is possible to check the parking fee.

Fig. 8: Smart Parking App

4. Experiment

A number of experiments were conducted to evaluate the accuracy of whether a vehicle was parked and whether an accident occurred in the implemented smart parking system. The main variable used in the experiment is the number of images for training.

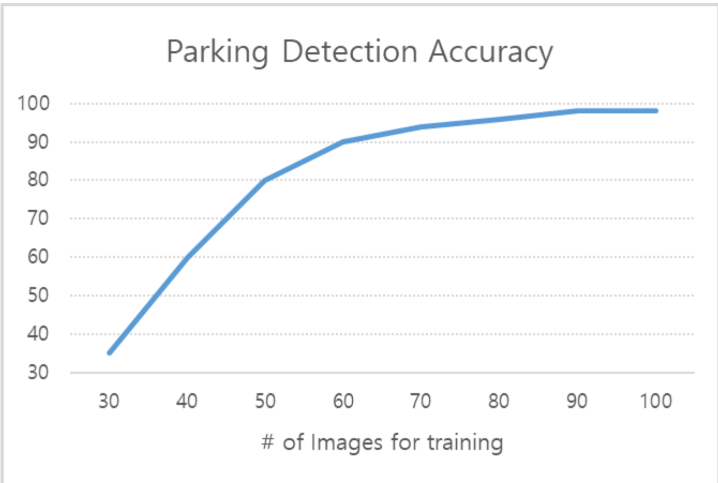

Fig. 9: Accuracy for Parking Detection

4.1. Experiment 1 – Parking Detection

Figure 9 shows the experimental results for parking detection. Assuming 4 parking spaces, the number of cases is a total of 4 factorials, that is, 24 cases. Therefore at least 24 image training is required. The experiment was conducted while increasing number of images from 30 to 100, step by 10. It shows that the accuracy increases as the number of training images increases. The accuracy rises steeply until the number of images before 50, but rises gently for more than 50, and there is no difference when the number of images is greater than 100. This experiment helps to find that the complexity of determining whether the vehicle is parked is not large. This is because the vehicle is rectangular and the parking space is also a rectangle, as it is the same principle as determining that a rectangular object is overlapping with multiple rectangular objects.

4.2. Experiment 2 – Accident Detection

Figure 10 shows the accuracy of accident detection. In this experiment, the image of the case where a new vehicle was being parked in a vacant space and the case where the parked vehicle was moving to leave. Vehicle accidents are very diverse, so in this study, the experiment was conducted while increasing number of images from 100 to 800, step by 100. As shown in the Figure 10, the accuracy increases steeply from 100 to 500, but when 500 images are trained, the accuracy is more than 90%, and learning more than that shows a gradual increase. This exerimental result shows that the complexity of determining the collision accident of the two vehicles is also not large. This is because it is possible to model both vehicles as rectangles, as in parking detection.

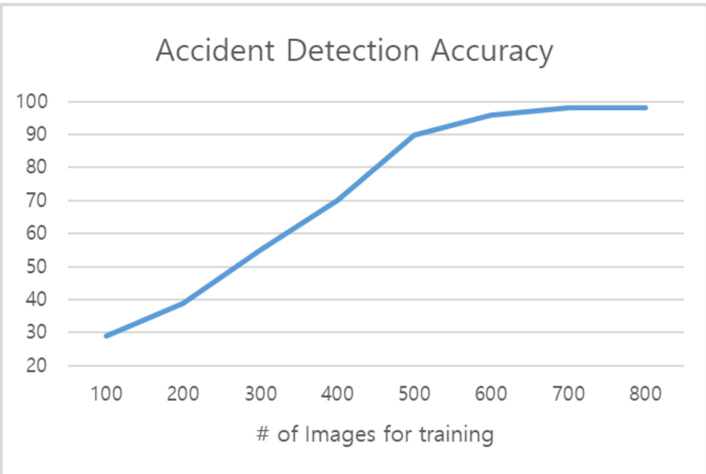

Fig. 10: Accuracy for Accident Detection

5. Conclusion

In this paper, a smart parking management system using AI technique was presented. The implemented system recognizes the vehicle number and uses it as an object ID, and tracks the vehicle by applying YOLO technology. Training and learning algorithms based on CNN deep learning algorithm were applied to detect whether a vehicle was parked or an accident occurred. A number of experiments was conducted to check the detection accuracy and it was confirmed that the deep learning algorithm works effectively after training reasonable number of images. Experimental results show that the detection accuracy of parking and accident detection increases as the number of training images increases. The accident detection needed more training images because it has more diversity. In both experiments, greater that 95% of detection accuracy was observed. The smart parking app allows drivers to easily check parking information and accident information. As a conclusion, the system implemented in this study can be utilized as an AI-based unmanned parking management system.

The implemented system used a simulated parking lot. Therefore, future research following this study is to implement a system for an actual parking lot. In addition, research to improve the detection accuracy and the processing speed will be performed.

Acknowledgements

This research was financially supported by Hansung University.

References

Acharya, D., Yan, W., & Khoshelham, K. (2018). Real-time image-based parking occupancy detection using deep learning. *In Research@ Locate*. 33-40.

Amazon. (n.d.). Amazon Web Services, https://aws.amazon.com

Bong, D., Ting, K., and Lai, K. (2008). Integrated approach in the design of car park occupancy information system (coins). *IAENG International Journal of Computer Science*, 35(1), 7-14.

Chen, M., & Chang, T. (2011). A parking guidance and information system based on wireless sensor network. *IEEE International Conference on Information and Automation*. 601-605.

Ichihashi, H., Notsu, A., Honda, K., Katada, T., & Fujiyoshi, M. (2009). Vacant parking space detector for outdoor parking lot by using surveillance camera and FCM classifier. *IEEE International Conference on Fuzzy Systems*. 127-134.

MQTT. (n.d.). MQTT: The Standard for IoT Messaging, https://mqtt.org/

OpenCV. (n.d.). Open Source Computer Vision, https://opencv.org/

Python. (n.d.), Python Programming Language, https://www.python.org/

Raspberrypia. (n.d.). Using a Raspberry Pi distance sensor, *RASPBERRY PI TUTORIALS,* https://tutorials-raspberrypi.com/raspberry-pi-ultrasonic-sensor-hc-sr04/

Raspberrypib. (n.d.). Raspberrypi Foundation, https://www.raspberrypi.org/

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE conference on computer vision and pattern recognition.* pp. 779-788.

Redmon, J., & Farhadi, A. (2017). YOLO9000: better, faster, stronger. *Proceedings of the IEEE conference on computer vision and pattern recognition.* 7263-7271.

Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. WEB: arXiv preprint arXiv:1804.02767.

Tesseraact. (n.d.). Tesseract Open Source OCR Engine, WEB: https://github.com/tesseract-ocr/tesseract

True, N. (2007). Vacant parking space detection in static images. *University of California, San Diego*, 17, 659-662.

Yu Y.J., Moon S.H., Sim S.J. & Park, S.H. (2020). Recognition of License Plate Number for Web Camera Input using Deep Learning Technique. *The Journal of Next-generation Convergence Technology Association.* 4, 565-572.