Journal of Innovative Technology, Vol. 4, No. 2, pp. 75-82, September 2022
http://doi.org/10.29424/JIT.202209_4(2).0008

Parking lot management application using instance segmentation

Wing-Kwong Wong [1], Szu-Hong Wang [2*],Wei-Yu Zhu [3]

ABSTRAC

In urban areas, it is difficult to find an available parking space. Many detection-based approaches have been proposed for solving the problem. In general, a detection-based approach involves ultrasonic sensors, weight sensors, geomagnetic and infrared detectors. In spite of its effectiveness, such an approach should pay higher costs of construction and maintenance. In this paper, a deep learning-based intelligent parking guidance system (DLPG) is proposed, where the Raspberry Pi system is used to monitor the parking spaces. When a server receives the video stream, the instance segmentation algorithm "You Only Look At CoefficienTs (YOLACT) is used to obtain comprehensive information about parking spaces. By experiment, the accuracy rate of the camera set at a real height of 5.4m can achieve 90%, and the overall accuracy is as high as 98.17%. In addition, we discover the important factors in affecting detection results are the pitch angle and the base angle. The most important factors affecting the detection results are the pitch angle and the base angle.

Keywords: Vehicle occlusion, instance segmentation, deep learning, parking lot management, weather factor, daylight factor

1. INTRODUCTION

Since the Industrial Revolution (IR) in the 19th century, the engine and vehicle manufacturing technologies have advanced and become more sophisticated with time, while the invention of vehicles has improved the quality and convenience of life. The demand for vehicles increased dramatically, bringing significant growth in both economic and technological fields. According to a survey by (Patil 2017), the total number of vehicles is 1.2 billion and will reach 2 billion by 2035. But the number of parking spaces is not proportional to the growth in the number of vehicles. People will spend more time looking for parking spaces in urban areas, and this results in significant CO2 emissions and traffic risks. Based on INRIX survey, the average U.S. motorist spends 17 hours a year to find parking spaces and consumes $345 in fuel costs. Intelligent parking is applied with common detection methods such as license plate recognition (Anagnostopoulos *et al.* 2006) and wireless sensors (Tang *et al.* 2006). It solves the problem of utilizing parking spaces while reducing management costs (Melnyk *et al.* 2019).

The automotive navigation systems help the driver of a vehicle to locate a particular destination and identify the best route for getting to that destination. Most automotive navigation systems use Global Positioning System (GPS) signals and electronic maps to identify the vehicle's current position relative to the desired destination. Therefore, most drivers can easily use automotive navigation systems or Google Map of smartphone navigation software to quickly locate available parking lots. Most of the parking lots were nearly full in urban areas, either waiting for other vehicles to leave or finding other parking lots. It is difficult to find an empty parking space in an almost full parking lot as shown in Fig. 1. In urban areas, a deep learning intelligent parking guidance system (DLPG) is needed to help drivers find parking spaces quickly, which minimizes the looking time and vehicle emissions.

Therefore, the DLPG system is proposed in this paper. The deep learning instance segmentation is applied to learn the features of vehicles and parking lanes. The system is more efficient than traditional approach, it can detect vehicles and parking spaces. The remainder of this paper is organized as follows. The system architecture of the proposed the method is described in Section II. The real-time implementation and demonstration are shown in Section III. Finally, the conclusion is presented in Section IV.

Fig. 1 Many drivers spend more time finding parking spaces in an urban area.

Manuscript received July 25, 2022; revised September 1, 2022; accepted September 28, 2022.

[1] Associate Professor, Department of Electronic Engineering, National Yunlin University of Science and Technology, Yunlin, Taiwan.

[2*] Assistant Professor (corresponding author), Department of Electrical Engineering, Bachelor Program in Interdisciplinary Studies, National Yunlin University of Science and Technology, Yunlin, Taiwan. (email: wangsr@ncut.edu.tw)

[3] Master Student, Department of Electronic Engineering, National Yunlin University of Science and Technology, Yunlin, Taiwan.

2. THE PROPOSED PARKING LOT MANAGEMENT SYSTEM

The proposed architecture of the parking lot management system is shown in Fig. 2. The main hardware architecture consists of a parking lot monitoring system and a server. The parking lot monitoring system is composed of a Raspberry Pi 4 with a webcam, which is used to obtain images of the parking lot. The server receives parking lot images over the network, which identify the parking space and regard it as a web server. The user can browse the website to check the parking situation and find the available parking spaces quickly. The system environment of this study is shown in Fig. 3. The programming language is Python, where deep learning uses the YOLACT model. The web framework uses Flask, and the recognition results of the parking spaces are transmitted to the web server and displayed on the web page through the Socket communication protocol.

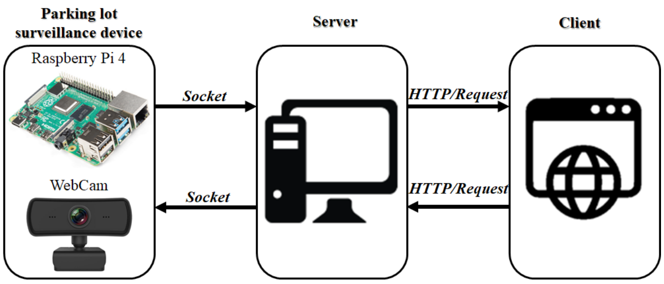


Fig. 2 The architecture of the intelligent parking lot management system

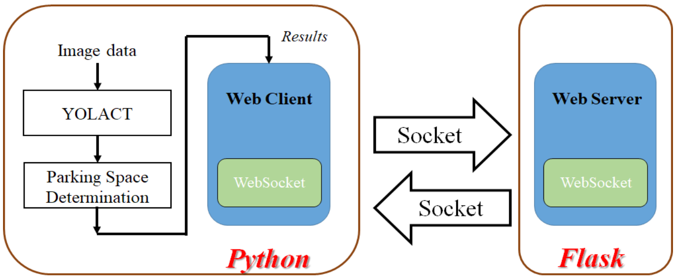


Fig. 3 The environment of intelligent parking lot management system

The overall algorithm is shown in Fig. 4, where the YOLACT model is used for vehicle detection. To obtain the coordinates of the parking grid of the image is the first step, where the instance segmentation is used to predict the parking grid paint and the Hough transform is used to detect the parking grid lines. Then the main lines and sub-lines of the parking space should be filtered to get the coordinates of the parking space. The above coordinates are not ordered, and a parking curve fitting scheme is proposed to solve this problem. The vehicles are detected according to the three-dimensional space with the YOLACT model, and display the results on the web page.

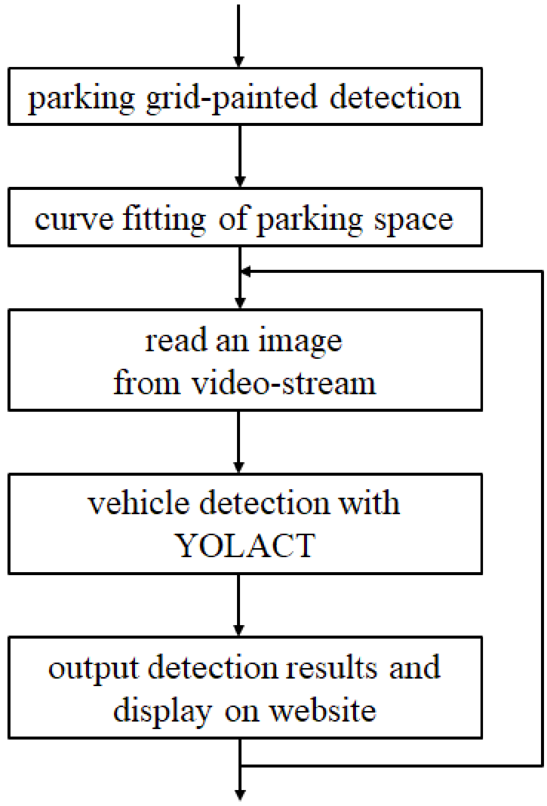
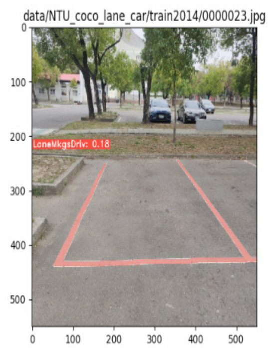


Fig. 4 The proposed algorithm.

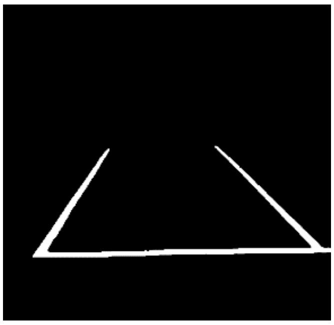
2.1 Parking grid line detection

In the study, the image read by the camera is used to detect the vehicles in parking spaces, and deep learning is used to fit the parking spaces. There are various types of grid lines in parking lots, some with clearly arranged grid lines, and complex parking grid lines, and some even without grid lines for drivers to reference. Therefore, a deep learning-based grid line detection is proposed.

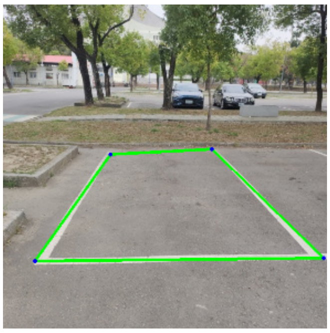
The original image captured by the camera is recognized to obtain a semantic segmentation result map, as shown in Fig. 5-(a). In addition to the existing image, the mask of the model prediction is generated. After image processing and ecological noise elimination, the mask of the parking frame can be obtained. Finally, through histogram statistics and hough transform (Aggarwal and Karl 2006) filtering out the main line, and get the parking coordinates, as shown in Fig. 5-(c).



(a)



(b)

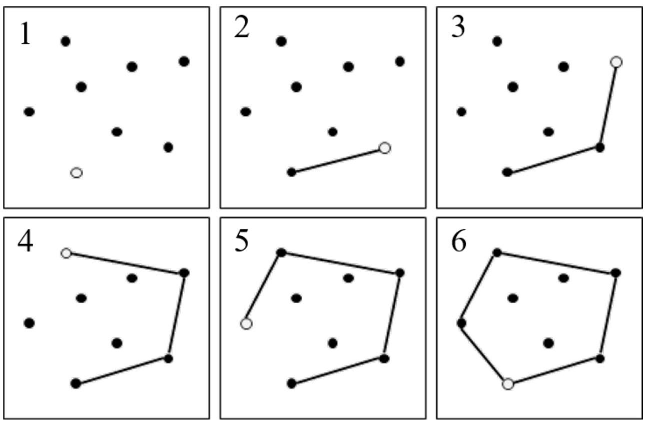


(c)

Fig. 5 Parking grid line detection

2.2 Curve fitting of parking spaces

The purpose of curve fitting of parking spaces is to obtain each parking space in the parking lot, where the parking grid line detection and curve fitting of parking spaces are pre-processed to understand the overall spatial layout of the parking lot. The parking spaces of each parking lot are continuous, and the parking space can be obtained correctly by perspective transformation. Before the perspective transformation, the four outermost corner points of all the coordinates need to be obtained, which is resolved with the Convex Hull algorithm. This algorithm takes the vertices of a convex packet on a two-dimensional plane, goes around the periphery and exhausts all points on the plane to find the outermost vertices to cover the outer periphery. The coordinates are approximated to quadrilateral by arc length after convex packet processing, as shown in Fig. 6.



(a) (b) (c)

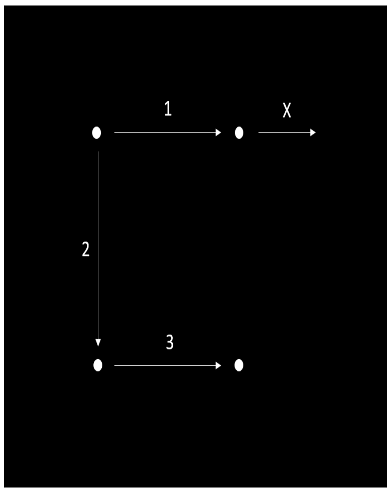
Fig. 6 The schematic diagram of convex packet processing

The different viewing angles of the real camera setup affect the distortion on the perspective projection, which can be prevented by converting to a bird's eye view. After obtaining the coordinates of the above four corners, the perspective transformation of the quadrilateral is performed such that the parking positions can be aligned continuously. The kernel of the perspective transformation is a 3x3 Homography matrix H as in Eq. (1), and the Homography matrix H can be obtained by four sets of corresponding points to Eq. (2).

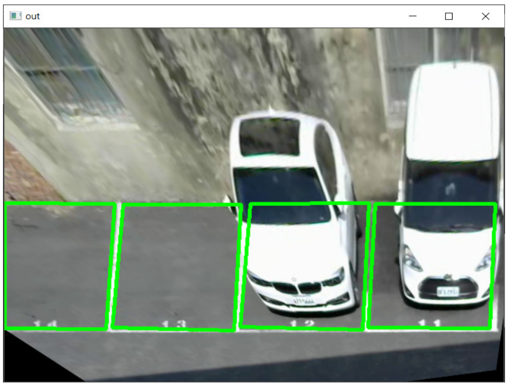
$$H = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \quad (1)$$

$$\begin{bmatrix} x1 \\ y1 \\ 1 \end{bmatrix} = H \begin{bmatrix} x2 \\ y2 \\ 1 \end{bmatrix} = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} x2 \\ y2 \\ 1 \end{bmatrix} \quad (2)$$

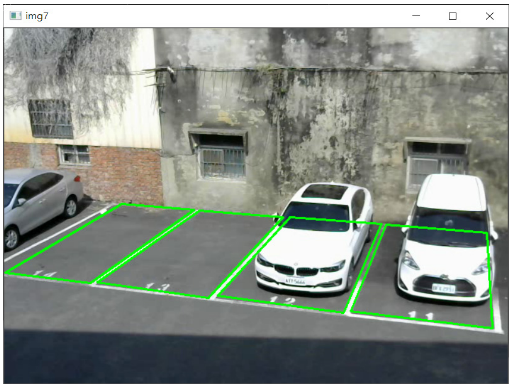
All pixels in the image are perspective-transformed to align the coordinates of the parking space with each other. The condition of creating a parking space is that it contains four coordinates, and the four coordinates must match the corresponding distance to form a parking space. Each coordinate will search for the closest coordinate to the right during the process of parking space matching. If the right coordinate matches the distance range, the search for the right coordinate will continue until four coordinates are found, then a parking space will be established. If the distance is too short or if no coordinates are found at the border, the process will be stopped and not saved, which result is shown in Fig. 7-(a). After searching all the coordinates as in Fig. 7-(b), four parking spaces were obtained, and the four coordinates of each parking space were arranged in counterclockwise order. The coordinates are finally converted back to the original image as shown in Fig. 7-(C), without any misjudgment due to environmental or grid lines.



(a)



(b)



(c)

Fig. 7 The schematic diagram of convex packet processing

2.3 The vehicle detection model

YOLACT (You Only Look At CoefficienTs) was proposed by Daniel Bolya, Chong Zhou, Fanyi Xiao, Yong Jae Lee in 2019 (Bolya et al. 2019). It is a convolutional model for real-time detection without loss of accuracy. YOLACT takes the concept of SSD and YOLO and improves the usual two-stage real-time segmentation model into one-stage for acceleration. Compared with other semantic segmentation models, e.g., Mask-RCNN, the authors of YOLACT pointed out that the steps of ROI Align are too time-consuming, so they removed the action of Feature Repooling and replaced it with Mask and Coefficients improvement. This is also the origin of the name. In this paper, an experiment to compare YOLACT and Mask-RCNN (He *et al*. 2020) was designed with the same training set, test set and hardware environment.

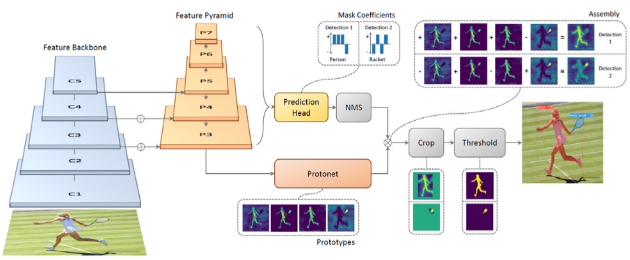
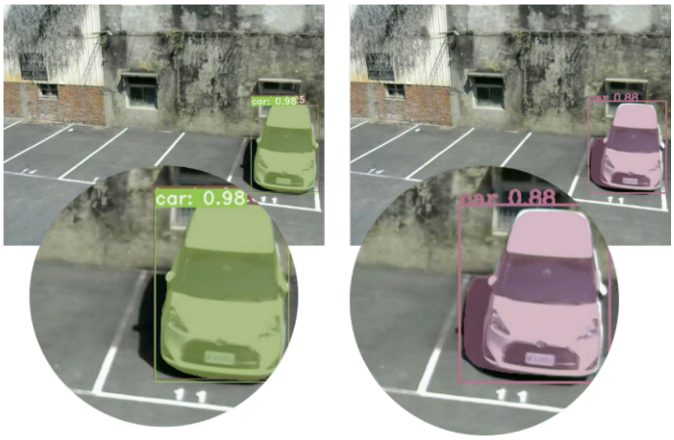


Fig. 8 The instance segmentation model - YOLACT

Compared with other models, the speed is much higher and the accuracy can be maintained. According to the paper, the ResNet50 model running on MS COCO dataset can reach 33.5FPS and the accuracy is 29.8mAP. In this paper, an experiment was run to compare YOLACT and Mask-RCNN with the same training set, test set and hardware environment. The results are shown in Table 1. The processing efficiency of YOLACT reaches 20 FPS, while that of Mask-RCNN is 0.24 FPS. In terms of mask, YOLACT also performs well, as shown in Fig. 9-(a), it can accurately cut the edges of the detected objects, while Mask RCNN will include the shadow as part of the vehicle (Fig. 9-(b)). There are many sets of semantic partitioning data. Most of them are self-driving views like cityscapes, but few of them are labeled with lane lines and vehicles. Therefore, this paper uses Camvid and MS COCO, and also collects 1000 images of parking lots nearby our university.

Table 1 Comparison of YOLACT and Mask-RCNN

Model	YOLACT	Mask-RCNN
Operating System	Ubuntu 18.04	
graphics card	RTX 2080Ti	
CUDA	11.3	
Dataset	MS COCO	
frame	14902	
Resolution	640 x 480	
efficiency	20 PFS	0.24 FPS



(a) (b)

Fig. 9 The instance segmentation results, (a) YOLACT, and (b) Mask-RCNN

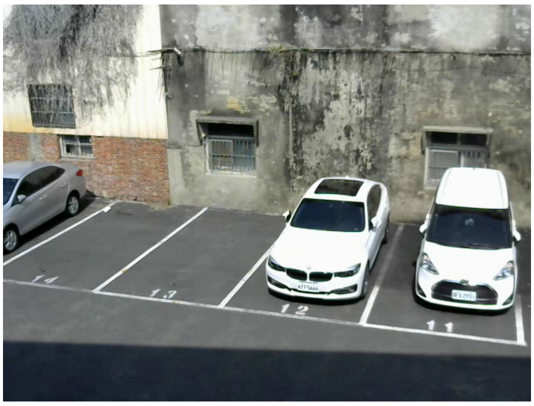
2.4 Determination of the parking space utilization

The result of vehicle pixel binarization is shown in Fig. 10 after the camera captures an image with a known parking space layout and performs an instance segmentation. After the accurate classification of vehicle pixels, the vehicle location information is clearly distinguished, which is an important criterion for determining whether there is a vehicle in the parking space. The way to decide whether the vehicle is in the parking space or not is the Intersection over Union (IoU) as the following equation:

$$D = \frac{ROI_{flat} \cap Mask_{car}}{ROI_{flat}} \quad (3)$$

where $ROI_{fla}t$ and $Mask_{car}$ are the area of the parking space and the area of the instance segmentation. If the result of D is higher than the threshold value, there is a vehicle on the parking space and vice versa, where the threshold value is set to 0.3.

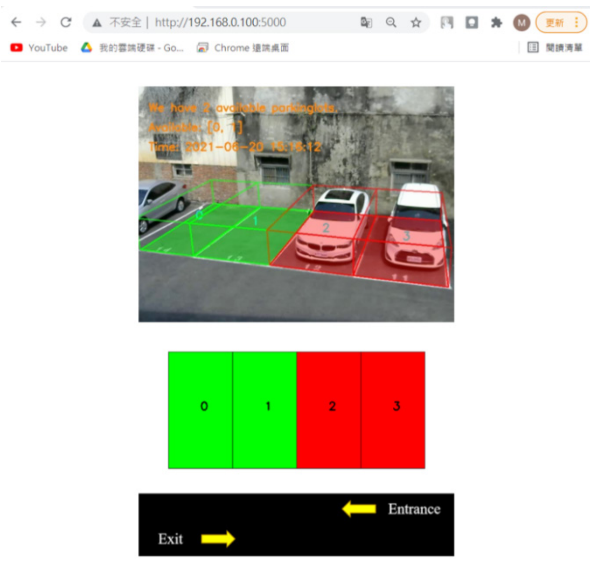
Finally, the result is sent to the web server, which shows the current time, the available parking space number and the parking space layout. The users can obtain information about the parking lot through the web page and find the available parking spaces in a short time as shown in Fig. 11.



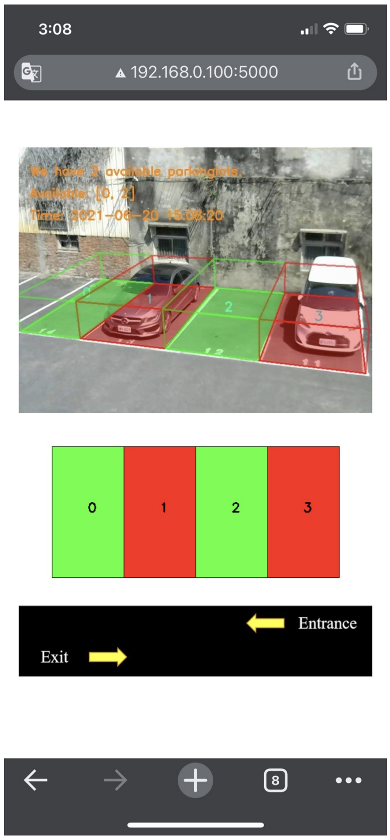
(a)

(b)

Fig. 10 The original image and the results of the instance segmentation of vehicle pixels



(a)



(b)

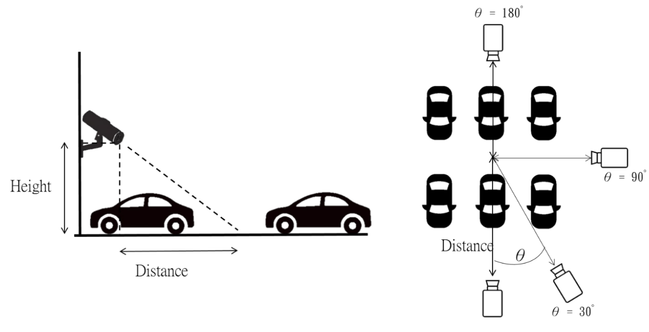
Fig. 11 Results on web and mobile

3. EXPERIMENTAL RESULTS

This study is dedicated to the application of visual parking space detection, which uses lower cost cameras or parking lot monitors to achieve parking management. In reality, the angle and height of the camera has been fixed for a long time, but the angle and height become important factors affecting this system. In order to increase the stability and practicality of the system, we simulated the experiment of various angle and height changes. Here there are three factors, where Distance is the distance from the vehicle to the camera plane, and Height is the height of the camera, and θ is the angle of change between the camera base and the centerline, as shown in Fig. 12.

The accuracy of parking space is used as the verification method, where D_TP is the number of parked spaces correctly determined, D_FP is the number of parked spaces incorrectly determined, D_TN is the number of empty parking spaces correctly determined, and D_FN is the number of empty parking spaces incorrectly determined. Then the accuracy rate is defined respectively by the following equations:

$$Accuracy = \frac{D_{FP} + D_{FN}}{D_{FP} + D_{FN} + D_{TP} + D_{TN}} \times 100\% \tag{4}$$



(a) (b)

Fig. 12 Experimental diagram for top view angle

To simulate the real environment, a 1:64 parking lot model was created with distances ranging from 20 cm to 40 cm (the real distance is 10 m to 25 m). The simulated camera height variables are 8.5 cm, 17 cm and 30 cm, respectively, which range from 5 m to 20 m relative to the real height. The angle θ is measured from 0º to 180º, with a unit change of 30º.

When the Distance is 20cm, the results of each angle and height change are shown in Fig. 13. The accuracy of the camera at 30cm (19.2m) height is 100%, and it is not affected by any angle change. When the height is lower than 30cm and θ is between 60º and 120º, the number of false vehicle errors is relatively higher, but the accuracy rate is still above 85%. The angle 90º of θ detection situation as shown in Fig. 14, the height of 8.5cm in the front vehicle is a sedan, the vehicle farther away from the camera, the detection of the car is easy to be interfered, resulting in false alarm. When increasing the height to 17cm, the error rate is greatly reduced, but if the front vehicle is a large car, misjudgment will result as shown in Fig. 14-(b).

The result is approximately the same as the distance of 20cm when the distance is 30cm as shown in Fig. 15. The occlusion problem between vehicles at angles of 60º and 120º is relatively serious, except for the height of 30cm where the accuracy is kept at 100%, the other heights are more seriously affected by the angle. The detection of angle 60º is shown in Fig. 16, the lower height between angle 60º and 120º will have a higher chance of false positive, the error rate decreases with the increase of height.

The distance of 40cm and the height below 30cm are still affected by the angle as in Fig. 17, however, the difference between 8.5cm and 17cm is not significant. The detection results of height 8.5cm and 17cm are similar as shown in Fig. 18, while the height of 30cm accuracy remains 100%.

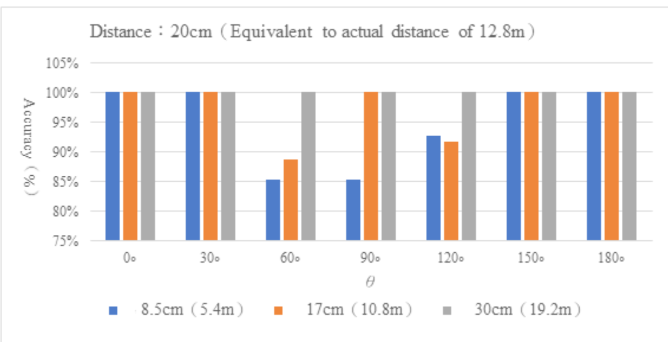
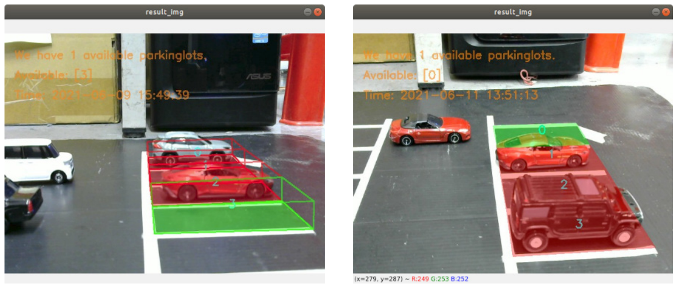


Fig. 13 Distance 20cm distribution of each angle



(a) Height 8.5cm (b) Height 17cm

Fig. 14 The results of distance 20cm and angle 90º at different heights

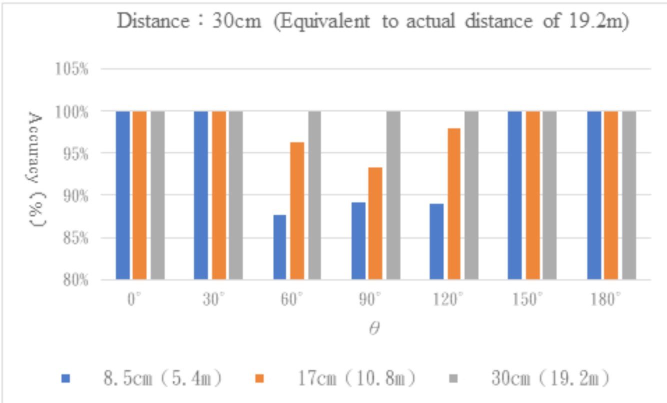
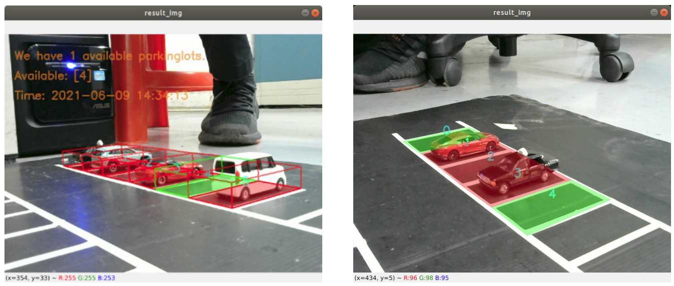


Fig. 15 Distance 30cm distribution of each angle



(a) Height 8.5cm (b) Height 17cm

Fig. 16 The results of distance 30cm and angle 60º at different heights

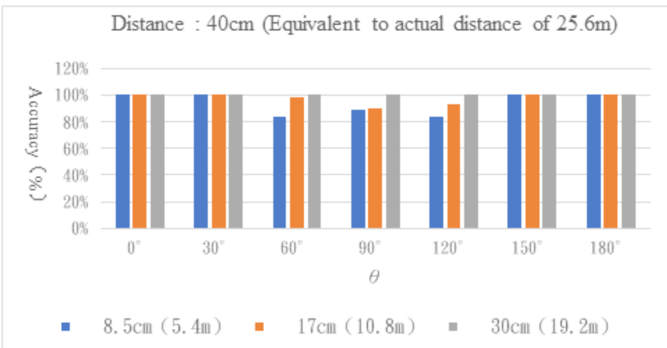
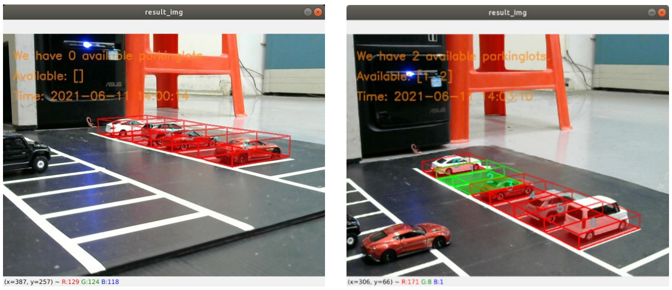


Fig. 17 Distance 40cm distribution of each angle



(a) Height 8.5cm (b) Height 17cm

Fig. 18 The results of distance 40cm and angle 60º at different heights

The CNRPark-EXT dataset is a collection of images from the CNRPark parking lot in Pisa, Italy, which has 164 parking spaces. It consists of 12,000 images collected from two cameras on different dates in July 2015. The CNRPark-EXT captures different levels of light every half hour, with shadows, trees, and streetlights partially shielding the cars. The dataset includes both sunny and rainy days to bring the experiment closer to reality.

There were 201 images of the experimental sunny conditions, each image contains 40-50 parking spaces, as shown in Fig. 19. The total number of detected parking spaces in 201 images was 8,643, which 3,741 parking spaces were parked and 4,902 parking spaces were vacant. The confusion matrix of the sunny day experiment is shown in Table 2. In the case of blurred or irregular parking lines, different light intensities, and complex backgrounds, most of the vehicles can be correctly identified with 86.71% accuracy. There were 116 images in the rain condition, each image contains 40-50 parking spaces, the total number of parking spaces is 4,988, which includes 2,402 occupied parking spaces and 2,586 vacant parking spaces. The results of the rain experiment, under the influence of trees in some of the parking spaces, the number of false detections was 416, with a correct rate of 91.65%, and the confusion matrix is shown in Table 3.

Fig. 19 Daytime detection of parking lot results

Table 2 Confusion matrix for sunny days

	Actual	
	Positive	Negative
Model Prediction True	3,802	2
Model Prediction False	1,146	3,693

Table 3 Confusion matrix for rainy days

	Actual	
	Positive	Negative
Model Prediction True	1,989	0
Model Prediction False	416	2,583

4. CONCLUSION

This paper proposes a new way of detecting parking spaces by cutting out the pixel location of vehicles by instance segmentation model, and then detecting whether the parking space is empty or not. The method is able to deal with the occlusion effect caused by most of the cameras installed in the parking lot, and the performance of the lower cameras is also good. Compared to traditional sensors, this method is low-cost and has advantages in future maintenance and repair. For most of the experiments in in this study (Zhang and Du 2020; Amato et al. 2017), the camera is set up at a higher height to enhance the accuracy rate. In our experiments, the accuracy rate of the camera set at a real height of 5.4m is more than 90%, and the overall accuracy is as high as 98.17%. The most important factors affecting the detection results are the pitch angle and the base angle. In addition, users can know the real-time status of parking spaces by browsing the website, which can reduce the time spent on searching for available parking spaces and reduce the emission of pollution. In the future, it would be ideal to reduce the computation workload and perform calculations directly on the edge platform. It is expected that the proposed method can be applied to parking lots in the future, so that drivers no longer have to spend a lot of time in parking lots to find parking spaces.

REFERENCES

Aggarwal, N., and Karl, W.C. (2006). "Line detection in images through regularized hough transform." *IEEE Transactions on Image Processing*, **15**(3) https://doi.org/10.1109/TIP.2005.863021

Amato, G., Carrara, F., Falchi, F., Gennaro, C., Meghini, C., and Vairo, C. (2017). "Deep learning for decentralized parking lot occupancy detection." *Expert Systems with Applications*, **72**, 327-334 https://doi.org/10.1016/j.eswa.2016.10.055

Anagnostopoulos, C.N.E., Anagnostopoulos, I.E., Loumosm, V., and Kayafas, E. (2006). "A License Plate-Recognition Algorithm for Intelligent Transportation System Applications." *IEEE Transactions on Intelligent Transportation Systems*, **7**(3), 377-392 https://doi.org/10.1109/TITS.2006.880641

Bolya, D., Zhou, C., Xiao, F., and Lee, Y.J. (2019). "YOLACT: Real-Time Instance Segmentation." *2019 IEEE/CVF International Conference on Computer Vision (ICCV 2019)*, 9157-9166 https://doi.org/10.1109/ICCV.2019.00925

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2020). "Mask R-CNN." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 386-397 https://doi.org/10.1109/TPAMI.2018.2844175

Melnyk, P., Djahel, S., and Nait-Abdesselam, F. (2019). "Towards a Smart Parking Management System for Smart Cities." *2019 IEEE International Smart Cities Conference (ISC2 2019)*, 542-546 https://doi.org/10.1109/ISC246665.2019.9071740

Patil, P. (2017). "Smart IoT based system for vehicle noise and pollution monitoring." *International Conference on Trends in Electronics and Informatics (ICEI 2017)*, 323-326 https://doi.org/10.1109/ICOEI.2017.8300941

Tang, V.W., Zheng, Y., and Cao, J. (2006). "An Intelligent Car Park Management System based on Wireless Sensor Networks." *2006 First International Symposium on Pervasive Computing and Applications*, 65-70 https://doi.org/10.1109/SPCA.2006.297498

Zhang, C., and Du, B. (2020). "Image-Based Approach for Parking-Spot Detection with Occlusion Handling." Journal of Transportation Engineering, **146**(9), 04020098 https://doi.org/10.3390/a14040114