INTERNATIONAL JOURNAL OF APPLIED TECHNOLOGY RESEARCH
2023, VOL. 4, NO. 1, pp. 53-68
https://ijatr.polban.ac.id/

POLBAN

Detection of Empty/Occupied States of Parking Slots in Multicamera System Using Mask R-CNN Classifier

[a]Hertog Nugroho, [a]Ginanjar Suwasono Adi*, [b]Muhammad Khoer Afandi

[a]Department of Electrical Engineering, Politeknik Negeri Bandung, Bandung 40559, Indonesia
[b]Macra Teknologi Indonesia, Jakarta 14460, Indonesia

Received January 15, 2022; Revised March 10, 2023; Accepted March 11, 2023; Published April 1, 2023

ABSTRACT

A fast growth of vehicles in big cities has an impact of arising road loads and difficulty of finding empty parking spaces. One solution to cope with the problem is to develop a parking management system which can provide useful information of available parking spaces to the potential users. This paper discusses about a new multicamera arrangement and the function to evaluate the empty/occupied states of the parking slots, as an alternative solution to the existing single camera system. The system adopted Mask R-CNN for its classifier, because of its capability to provide the polygon outputs for its detected objects, compared with the existing bounding box outputs provided by other classifiers. The proposed function has optimized the available information from all cameras, by considering the relative position of each camera to the parking spaces, and also capable of overcoming occlusion problem occurs in some cameras. The experiment shows that the capability of overcoming the occlusion problem has been validated, and its performance to evaluate the empty/occupied states of the parking slots was better than the single camera system to a certain threshold.

KEYWORDS

Parking management system
Mask R-CNN
Multicamera
Occlusion

INTRODUCTION

The expansion of urban centers is often accompanied by the growth of infrastructure and economic activity. One effect of this trend is an increase in the ability of citizens to purchase vehicles, as evidenced by data from the Central Statistics Agency [1]. The table shows the number of vehicles in four categories namely: cars, buses, trucks, and motorcycles in DKI Jakarta in the years 2019, 2020, and 2021. While motorcycles continue to rank as the most common form of transportation, there has been a significant surge in private car ownership between 2020 and 2021. In general, the data suggests that there has been a significant increase in vehicle ownership in DKI Jakarta, particularly in the car and motorcycle categories.

This vehicle growth has an impact on increasing road loads and the need for parking lots, especially when the mode of transportation is still heavily rely on private vehicles, such as motorcycles and cars. However, finding a parking space in big cities is a big challenge for drivers

*Corresponding Author: ginanjar.adi@polban.ac.id

nowadays, because the growth in the number of vehicles is not accompanied by the provision of sufficient parking spaces. Therefore, one solution to overcome this problem is to efficiently use the existing parking spaces by providing information on the availability of parking spaces for drivers.

Table 1. The number of vehicles in DKI Jakarta provinces (Source: Ditlantas Polda Metro Jaya) [1]

Category	2019	2020	2021
Car	3.310.426	3.365.467	4.111.231
Bus	34.905	35.266	342.667
Truck	669.734	679.708	785.600
Motorcycle	15.868.191	16.141.380	16.519.197
Total	19.883.246	20.221.821	21.758.695

This study proposes a smart parking management system that utilizes multiple cameras and computer vision algorithms to identify the occupancy status of available parking spots. The main contribution of this work is the novel approach of using multiple cameras to evaluate the state of parking spaces. The use of multiple cameras in this research provides an advantage in addressing occlusion problems that may arise in the observation area, while an object can be occluded from one camera view, but clearly visible from other camera view. This approach improves the system's efficiency and reliability through specialized designs in both the system architecture and algorithms.

LITERATURE REVIEW

Efforts to develop parking management system has been proposed. K. Gkolias and Eleni I. Vlahogianni [2] proposed a system of empty on-street parking spaces detection in urban road networks based on data provided by in-vehicle cameras. Assuming that there are a number of vehicles with installed cameras moving around a city and collecting pictures of the roadside on both sides with distance interval about 5-7 m for each take. Examples of two takes representing occupied and empty spaces are depicted in Figures 1.(a) and (b), respectively. It is easily understood that the scene illustrated in Figure 1.(a) doesn’t have available parking space for another car, while the empty space illustrated in Figure 1.(b) can be used, although the size of the candidate car should be measured. The collected pictures can be used to train the classifier for identifying the empty parking spaces and tell the potential driver about the available space. The system proposed CNN-based classifier to locate the empty space. However, it will be extremely difficult to deploy the system since it requires a network of connected surveillance cars to update the parking spaces condition, process the pictures in real time and provides information to the users.

(a) (b)

Figure 1. Examples of occupied (a) and empty (b) spaces (Courtesy of K. Gkolias et. al. [2])

Li et al. [3] proposed a parking slot detection system, using Around View Monitoring (AVM) system [4]. The system utilized 4 fisheye cameras which were installed at front, rear, left and right side of the car, and generate a 'bird's eye view' image through a view transform and image mosaicking. Figure 2 illustrates examples of the generated bird's eye view images. The car is seen lecated in the center of the images and the surrounding area is a part of potential parking slots with their markers, such as lines, T-shaped, and L-shaped corners. Based on this image, they developed parking slot detection method by identifying parking slot markers on image. They also adopted learning-based classifier to identify slot occupancy from the same image. Similar approaches were proposed by Wu et al. [5], Li et al. [6], Yu et al. [7], Zinelli et al. [8]. Their approaches were characterized by generation of vertical-view images surrounding ego-car and developed detectors to identify and localize parking slot markers which appear in the images, while the occupied slots were classified by analyzing features 'inside' each identified parking slots. In another work, while still using the same AVM system, Lee et al. [9], used motion information as additional information to help identify parking slot occupation. From a series of AVM images, they proposed a Bayesian filtering scheme to estimate parking slot markers probability. All of the above works depends solely on installed cameras around the vehicle. The limitation of these approaches is that the observation area is very limited and requires special vehicles which are not commonly available to ordinary people.

Figure 2. Examples of vertical images generated by ego-vehicles (Courtesy of Wu et. al. [5])

A different approach was proposed by Andrew Regester and Vamsi Paruchuri [10]. To take aerial parking images, they used a flying Unmanned Aerial Vehicle (UAV) which was equipped by camera. The UAV was then controlled to fly above a wide parking area. As is shown in Figure 3,

the scene covers a wide parking area with a large numbers of occupied and empty parking slots. The slots in such images were characterized by white lines bordering each parking slot and clearly visible. The picture of the cars also clearly visible without any occlusion, except in the border parking area where the cars park under the tree. Based on this observation, they developed parking slots detection method using line markers as the main features and developed line analysis to exploit the line features. However, the approach depends highly on the availability of UAV and should continuously running the UAV to obtain updated state of parking slots.

Figure 3. An example of a test image collected using a UAV (Courtesy of Vamsi Paruchuri [10])

An easily implementable approach without requiring special vehicles were proposed by Adi et al. [11]. They deployed a stationary camera facing a wide parking area as is shown in Figure 4. However, the position of the camera is on side of the parking area, so that in some situation, the car is occluded by other object. The example of this situation is seen in Figure 4 with the car parks on the opposite road, where the car is occluded by two poles. The camera setting is similar with the surveillance cameras usually found in some parking areas. Therefore, in the deployment, if the surveillance camera has already in place, it can be used without requiring any modification. To classify occupied and empty parking slots, they prepared reference images representing empty parking slots (green boxes in Figure 4), and then, using the reference images, they identified empty slots based on similarity metric. Similar deployment was also proposed by Farley et al. [12]. However, to classify occupied and empty parking slots, they tried several deep-learning machines, and they found out that AlexNet [13] provided the best performance.

Figure 4. An example of camera deployment facing a parking area (Courtesy of Adi et. al. [11])

An interesting approach was proposed by Postigo et al. [14]. With the similar environment as [11], but instead of using parking slot features to identify parking slots, they proposed a method to track vehicles entering the parking area. The tracker keeps track each vehicle since it enters a camera view until it goes out of the view, using background subtraction and transience map analysis. If during the time, it stops (characterized by a long sequence of absence of motion of that vehicle), its state was labeled as 'parking', and the location is recorded as a parking slot. Otherwise, it was 'moving'. The method didn't require training procedure. Similar approach was also proposed by Patel and Meduri [15]. However, they adopted deep-learning architectures to identify the vehicle, and then track the vehicle. While the approach can reduce labeling initialization work, the location of the parking slots is determined by the states of the car, and it doesn't precisely reflect the real situation.

While the above methods used only one camera, another work used more than 1 camera to cover a wider parking area or to help improve the performance. Amato et al. [16], discussed a deployment of several smart cameras facing the same parking lot with different view angles. Each smart camera was equipped by Raspberry Pi 2 processor so that the process can be run locally. The aim was to distribute the processing load on each camera to avoid bottleneck on the server. In the experiment, they prepared 2 datasets, and divided each dataset into 2 sub-sets: training and testing. Then, one camera was trained with one dataset and tested with another dataset. Similar arrangement was also applied on the 2nd camera. This inter-dataset scenario is implemented to generalize the result. For the classifier, they tried two deep-learning architectures: LeNet [17], and AlexNet [13] and found out that AlexNet achieve the best performance. While the scenario can give high level of generality, the rich information provided by multiple cameras wasn't maximally exploited. Meanwhile Nieto et al. [18] used the availability of multiple cameras to boost the performance. Their method consisted of several steps: vehicle detector using Faster R-CNN [19], homographic transformation, and perspective correction, which were done in each camera and information fusion collected from all active cameras. Figure 5 illustrates the arrangement of the active cameras. Each camera generated a map of parking slots with their empty/occupied status, and the system synchronized the indexes of the parking slots to build a map of overall parking slots with their statuses. However, although they claimed that

the system can be implemented using many cameras, the experiment was done using only 2 cameras.

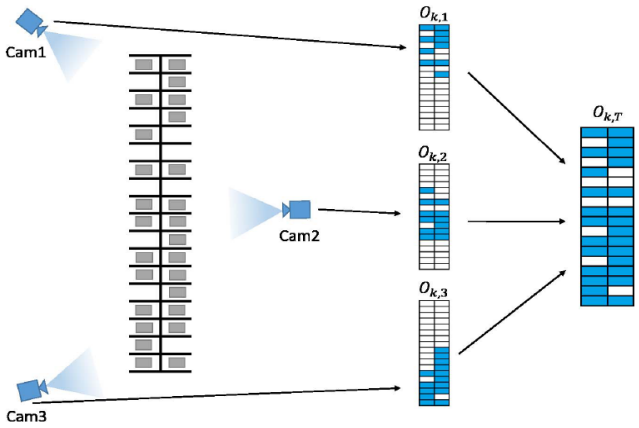


Figure 5. Information fusion example: parking and monitoring cameras (left), single-camera extracted occupation information (center) and result of the information combination (right). Courtesy of Nieto et. al. [18]

Our approach is different from Nieto et al. [18]. We consider that the information collected from more than 1 camera will be more accurate than that collected from a single camera. Therefore, in our deployment, the cameras were arranged such that the view of all cameras covers the same parking area as is illustrated in Figure 6. From the arrangement, we developed a new parking slot state determination utilizing all active cameras which is our contribution in this work. We describe the method in the following section.

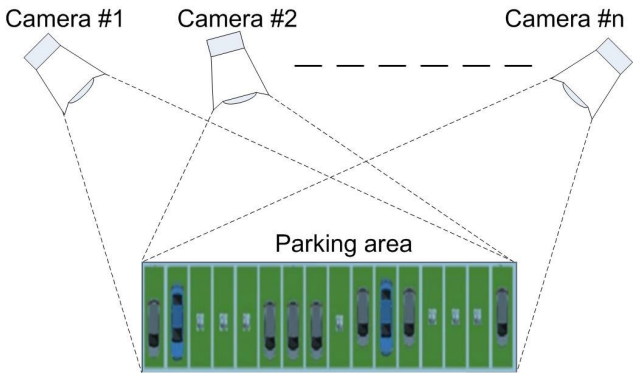


Figure 6. Arrangement of surveillance cameras in our scenario

PARKING SLOT STATE DETERMINATION

Since the occupation state of a parking slot is determined based on information extracted from a set of cameras, we have to decide the importance level of the information from each camera.

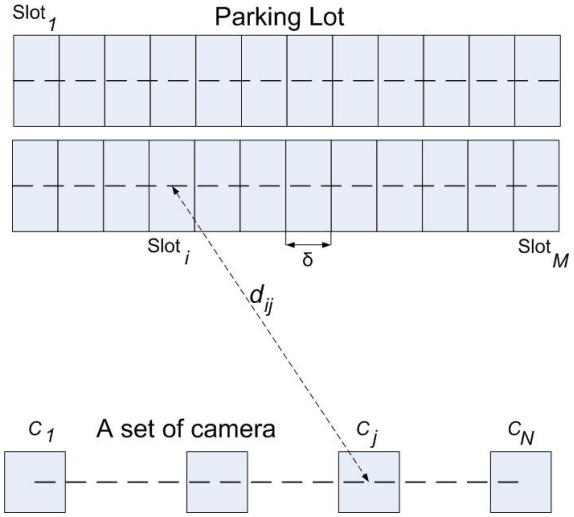


Figure 7. Parameters definition used in this work

Referring to Figure 7, let C_j, j = 1, 2, …N, is the active cameras, $Slot_i$, i = 1, 2, ..M, is the individual parking slot, and $S(i)$ is the state of the parking slot which is expressed in Equation (1),

$$S_i = \begin{cases} 1 & P_i \geq th_1 \\ 0 & else \end{cases} \tag{1}$$

where th_1 is threshold value of probability for each slot; and P_i as the occupied probability of slot i is expressed in Equation (2),

$$P_i = \sum_{j=1}^{N} \frac{(D - d_{ij})}{\sum_{j=1}^{N}(D - d_{ij})} s_{ij} \tag{2}$$

where d_{ij} is distance between $Slot_i$ and camera j (C_j), and D is a constant, determined by Equation (3),

$$D = \max_{\substack{1 \leq i \leq M, \\ 1 \leq j \leq N}} [d_{ij}] + \delta \tag{3}$$

where δ is a positive real number, and s_{ij} is the state of slot i which is evaluated from camera j. Its formula is expressed in Equation (4).

$$s_{ij} = \begin{cases} 1 & IoR_{ij} \geq th_2 \\ 0 & else \end{cases} \tag{4}$$

where th_2 is threshold value of each region and IoR_{ij} is a part of car_{jk}, k = 1,2, …K (K is a number of cars viewed from camera C_j) overlapped with area of the $Slot_i$, which is expressed in Equation

(5). Determination of threshold values th_1 and th_2 are a part of the analysis process in the experiment.

$$IoR_{ij} = \max_{1 \le k \le K} \left[\frac{ROI(car_{jk}) \cap ROI(Slot_i)}{ROI(car_{jk})} \right] \quad (4)$$

The area of the slot i, $ROI(Slot_i)$ is determined manually as an initial step and can be used as long as the arrangement of parking lot and camera deployment doesn't change. The area of car k viewed from camera j $ROI(car_{jk})$ is determined by a car classifier. From Equation (1), it can be shown that the parking slot closest to a certain camera position will get highest weight, since the camera will give the best information about the state of the slot. There is a possibility that a car(s) is(are) occluded from one camera view. However, there are still possibilities that the car(s) is(are) visible from other cameras, so that the car(s) can still be evaluated. This robust characteristic can only be obtained by a multi-camera system. In case that one car occupies two neighbouring slots, the system will decide that the two slots are occupied. The validity of this decision depends of the determination of threshold values th_1 and th_2 and is discussed in the experiment.

RESULTS AND DISCUSSION

Data Preparation

As far as we know, there is still no other works which the camera arrangement is similar with ours. Consequently, there is still no available public dataset for our case. Therefore, we train the system with public dataset and inherited the model to evaluate our system. This approach is to generalize the model so that the system can be applied to any environment. To train our system, we selected CNRPark-EXT dataset from http://cnrpark.it/. It consists of entire parking lot images gathered from several cameras with different viewpoints and various environment conditions (sunny, overcast, and rainy). The dataset contains information of binary classification of 2 labelled classes (class 1: occupied slot which has car image in it; class 2: vacant slot which is the annotated image of empty slot). Since we intended to train our system to be able to detect cars on the parking lot images, we adopted the class of occupied slot as labelled car. To balance the inference ability of our system, we also used empty slot class from in the process of training data. An example of test image is shown in Figure 8. It is taken from CNRPark-EXT dataset. The image illustrates a scene of a section of parking area with most of the parking slots were occupied. Most of the cars and empty parking slots are clearly visible. Only one car (in front) is occluded by a lamp pole and another one is occluded by a tree (in the middle section).

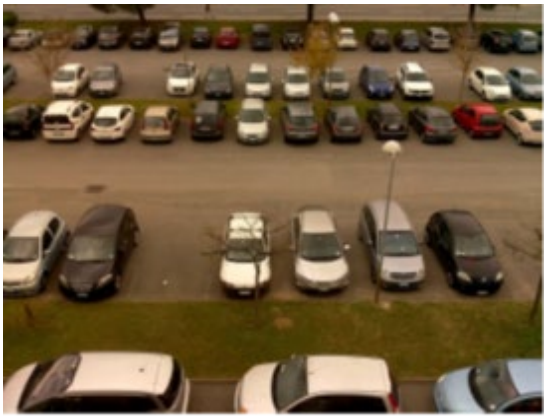


Figure 8. An example of test image from CNRPark-EXT dataset

We also built our own local dataset and chose one of the parking lots in our campus (http://www.polban.ac.id). We chose the location so that we can set up our system with only requiring simple permission procedure and can control the recording process without any disturbance, since the campus activities can't be interrupted. The cameras were deployed parallel with the parking slots as is illustrated in figure 7, with camera C_1 was positioned on the leftmost side of the arrangement and camera C_N was on rightmost side, with N equals 4 in our implementation. The video was recorded using Hikvision IP Camera 2MP DS-2CD1023G0E-I with 1920 x 1080 maximum. During the recording period, we managed to record 936 sets of images which 1 set of images consists of 4 time-synchronized images taken from 4 cameras as is shown in Figure 9. Please note that the viewing angle of the cameras were not aligned to the horizon of the landmark. We let them uncorrected to see their effects on the evaluation. The dataset consists of 2384 empty parking slots and 425 occupied parking slots under 3 weather conditions: sunny, rainy, and overcast. To make the ground-truth data, the vehicles of all images have been manually annotated using VGG Image Annotator [20]. The training images have been annotated for its use in the generation of the parked vehicle model, and the test images for the evaluation of our system. The parking slots have also been manually annotated to be used for occupied/empty slot evaluation. Figure 10 shows the result of parking slots annotation (red marked) which was done through our application.

(a)

(b)

(c) (d)

Figure 9. An example of synchronized images in our local dataset taken from 4 cameras: (a) The leftmost camera (C_1), (b) The inner left camera (C_2), (c) The inner right camera (C_3), and (d) The rightmost camera (C_4)

Figure 10. Parking slots annotation

Vehicle Detector

For vehicle detector, we tried two deep-learning machines: YOLOv4 [21] and Mask R-CNN [22]. YOLOv4 was claimed to be fast and accurate. It produces a bounding box to cover the recognized object (See Figure 11.a). Its performance surpassed its predecessor, YOLOv3 [23]. Meanwhile, Mask R-CNN was an extension of Faster R-CNN [19] (used by Nieto et. al. [18]). While the Faster R-CNN produced 2 outputs for each recognized object: a class label and a bounding box offset, the Mask R-CNN produces 1 more output, the object mask. A mask encodes the object's spatial shape feature, and thus, similar with segmentation procedure (Figure 11.b). This feature motivated us to adopt the detector, since intuitively, the shape is more precise to represent object's boundary than a bounding box. This is the reason behind selecting the two detectors: bounding box vs shape.

(a) (b)

Figure 11. Examples of object detection by (a) YOLOv4 in a rectangular shape (from mathworrk [24]) and (b) Mask R-CNN (courtesy of He et. al [22])

Performance Metrics

As has been mentioned in Section 3, there are 3 parameters to be decided, th_1 in Eq. (1), δ in Eq. (3), and th_2 in Eq. (4). For parameter δ, since its unit is the same as real world unit (in m), we decided to pick the width of the parking slot to be its value (See Figure 7). That way, its value is always calibrated to its real-world environment, independent from the camera position. Parameter th_1 and th_2 was determined through the experiment.

To quantify the performance of our system, we adopted the most commonly used performance metrics in classification field, such as *accuracy, precision, recall*, and *F1-score*, defined in Equations (6) through (9) while the confusion matrix for Eqs. (6) through (9) is given in Table 2.

Table 2. Confusion matrix for Occupied/Empty Parking slot

Ground Truth	System's output	Evaluation
Occupied	Occupied	True Positive (TP)
Empty	Empty	True Negative (TN)
Empty	Occupied	False Positive (FP)
Occupied	Empty	False Negative (FN)

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

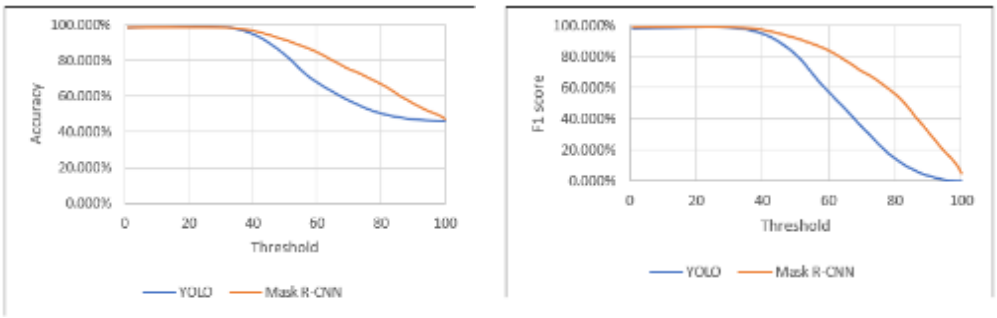
$$Recall = \frac{TP}{TP + FN} \tag{8}$$

$$F1\ score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{9}$$

Analysis

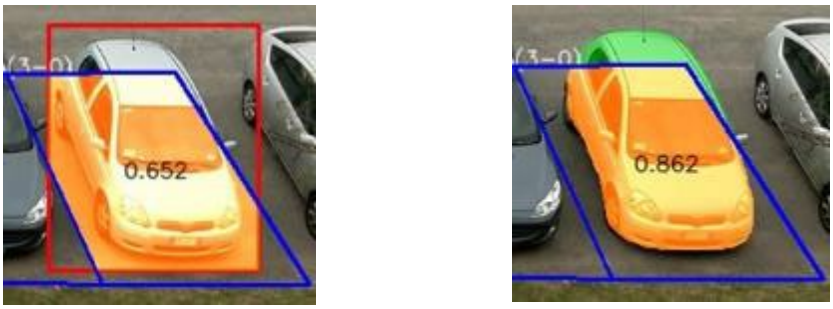
Performance comparison of YOLOv4 vs Mask R-CNN on CNRPark-EXT dataset

Figure 12 depicts (a) *accuracies* (Eq, (6)) and (b) *F1 scores* (Eq. (9)) of YOLOv4 (blue curve) and Mask R-CNN (red curve), as the range of threshold th_2 (Eq. 4) was set from 0 to 100%. We selected the metrics since accuracy can only be used in balanced data, while *F1 score* can be used in unbalanced data condition. From the graph, it can be concluded that Mask R-CNN outperform YOLOv4 in both metrics. This is predictable since the output of YOLOv4 is a bounding box covering not only the car, but also surrounding background, while the output of Mask R-CNN is a polygon which covers only the car's boundary. An example of *IoR* result (Eq. (5)) is shown in Figure 13, while running on one image of CNRPark-EXT dataset under YOLOv4 achieved the *IoR* score of 0.652 while under Mask R-CNN achieved the *IoR* score of 0.862. In this case, if we set the threshold between 0.652 and 0.862, the output of YOLOv4 would miss the car, while we still have correct result from Mask R-CNN.



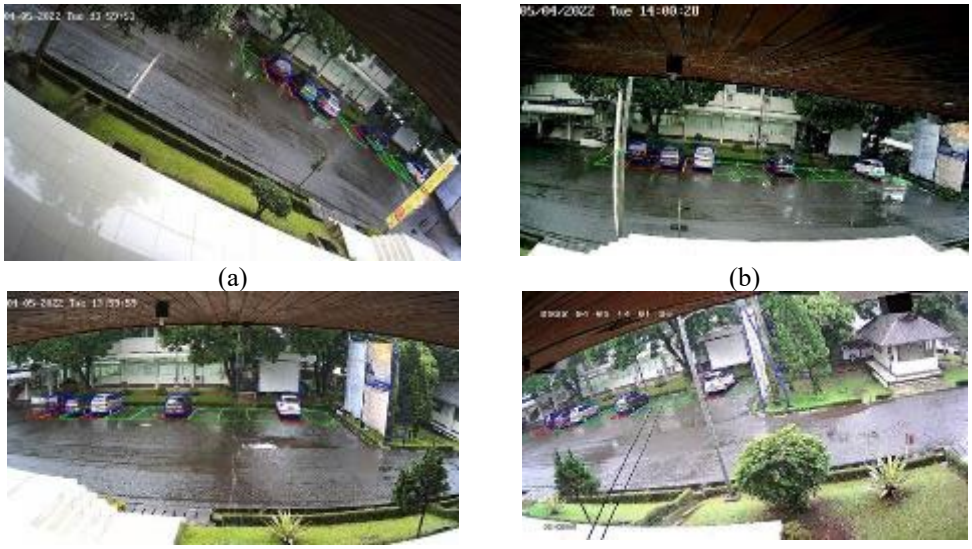
(a) (b)

Figure 12. Performance analysis of YOLOv4 and Mask R-CNN on CNRPark-EXT dataset (a) *accuracy* and (b) *F1 score*



(a) (b)

Figure 13. An example of IoR result by implementing Eq. (5) on output of YOLOv4 (a) and Mask R-CNN (b)



(a) (b)

(c) (d)

Figure 14. Evaluation result on each camera (a) The leftmost camera(C_1) (b) The inner left camera(C_2) (c) The inner right camera(C_3) and (d) the rightmost camera(C_4) The red lines indicates occupied slot and the green lines indicates empty slots

Performance comparison of using 1 camera vs multi-camera on local dataset

As a result of the above experiment, the Mask R-CNN has been decided to be used for the next experiment. The model for Mask R-CNN was inherited from the training process during the previous experiment. In this case, we used local dataset as has been discussed in Section 4.A. An example of evaluation result is shown in Figure 14, while the red lines indicate occupied slots and the green lines indicates empty slots. Careful observation on the lines indicates that the states of the parking slots on each image were not consistent. Some red lines on one image were green on others. This is reason why we put weights on the evaluation result of each camera (as is expressed in Eq. (2)), to give a priority to the cameras which contribute more accurate information. By this way, the system can also cope with occlusion problem.

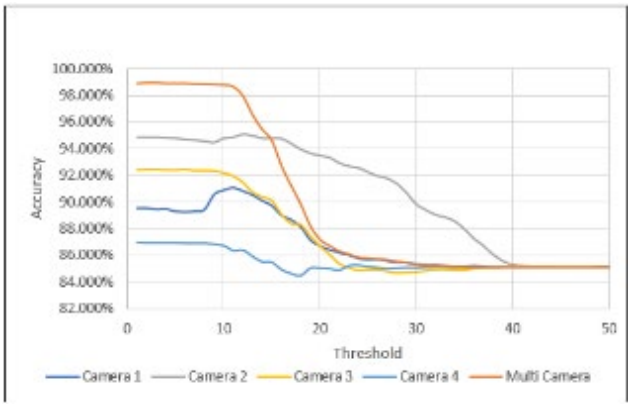


Figure 15. Evaluation result on accuracy of the multi-camera system

Figure 15 depicts evaluation results on accuracy of our system. Please note that until threshold th_2 15%, the multi-camera system outperformed the single camera system. However, when the threshold was increased, camera no 2 took over. Please note that camera no 2 is the closest camera to the parking slots. So, from its position, all the cars and the slots will appear large, including their intersection. In other words, the resolution of the objects is more detail, compared with the objects from a far side (which have low resolution). We will work with this issue for our next work.

CONCLUSION

This paper presents a multicamera system for the recognition of the empty/occupied state of the parking slots which can be used in parking management system for the benefit of both user and the management. The system adopted Mask R-CNN to detect vehicles, and based on its result, decided the state of the parking slot. Since in this work a new multi camera arrangement was proposed, a new dataset relating to this arrangement has been recorded and synchronized. Observation on the output of the system indicated that the problem of occlusion has been elegantly overcome, due to the proposed arrangement of the cameras. Experiments on the threshold indicated that the proposed multicamera system was effective until a certain value. After that value, a camera which was installed closest to the parking lot took over.

For the future work, we recommend studying the different formula to decide empty/occupied state of the parking slots in order to optimize the rich information obtained from more than one camera. Additionally, another dataset from different places can be useful to evaluate the performance of the system.

REFERENCES

[1] C. S. A. (biro P. S. (ind)), "Number of vehicles in Jakarta province 2019-2021," 2021. https://jakarta.bps.go.id/indicator/17/786/1/jumlah-kendaraan-bermotor-menurut-jenis-kendaraan-unit-di-provinsi-dki-jakarta.html (accessed Sep. 24, 2023).

[2] K. Gkolias and E. I. Vlahogianni, "Convolutional Neural Networks for On-Street Parking Space Detection in Urban Networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 12, pp. 4318–4327, 2019, doi: 10.1109/TITS.2018.2882439.

[3] Q. Li, C. Lin, and Y. Zhao, "Geometric features-based parking slot detection," *Sensors (Switzerland)*, vol. 18, no. 9, 2018, doi: 10.3390/s18092821.

[4] Y. Gao, C. Lin, Y. Zhao, X. Wang, S. Wei, and Q. Huang, "3-D Surround View for Advanced Driver Assistance Systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 1, pp. 320–328, 2018, doi: 10.1109/TITS.2017.2750087.

[5] Z. Wu, W. Sun, M. Wang, X. Wang, L. Ding, and F. Wang, "PSDet: Efficient and Universal Parking Slot Detection," in *IEEE Intelligent Vehicles Symposium, Proceedings*, 2020, pp. 290–297. doi: 10.1109/IV47402.2020.9304776.

[6] W. Li, L. Cao, L. Yan, C. Li, X. Feng, and P. Zhao, "Vacant parking slot detection in the

around view image based on deep learning," *Sensors (Switzerland)*, vol. 20, no. 7, 2020, doi: 10.3390/s20072138.

[7] Z. Yu, Z. Gao, H. Chen, and Y. Huang, "SPFCN: Select and Prune the Fully Convolutional Networks for Real-time Parking Slot Detection," in *IEEE Intelligent Vehicles Symposium, Proceedings*, 2020, pp. 445–450. doi: 10.1109/IV47402.2020.9304688.

[8] A. Zinelli, L. Musto, and F. Pizzati, "A deep-learning approach for parking slot detection on surround-view images," *IEEE Intell. Veh. Symp. Proc.*, vol. 2019-June, pp. 683–688, 2019, doi: 10.1109/IVS.2019.8813777.

[9] M. Lee, S. Kim, W. Lim, and M. Sunwoo, "Probabilistic Occupancy Filter for Parking Slot Marker Detection in an Autonomous Parking System Using AVM," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 6, pp. 2389–2394, 2019, doi: 10.1109/TITS.2018.2855183.

[10] A. Regester and V. Paruchuri, "Using Computer Vision Techniques for Parking Space Detection in Aerial Imagery," in *Advances in Intelligent Systems and Computing*, vol. 944, 2020, pp. 190–204. doi: 10.1007/978-3-030-17798-0_17.

[11] G. S. Adi, M. Y. Fadhlan, S. Slameta, G. M. Rahmatullah, and A. Faturahman, "Parking slot detection system based on structural similarity index," in *IOP Conference Series: Materials Science and Engineering*, 2020, vol. 830, no. 3. doi: 10.1088/1757-899X/830/3/032005.

[12] A. Farley, H. Ham, and Hendra, "Real Time IP Camera Parking Occupancy Detection using Deep Learning," in *Procedia Computer Science*, 2021, vol. 179, pp. 606–614. doi: 10.1016/j.procs.2021.01.046.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012, vol. 2, pp. 1097–1105.

[14] C. G. Del Postigo, J. Torres, and J. M. Menández, "Vacant parking area estimation through background subtraction and transience map analysis," *IET Intell. Transp. Syst.*, vol. 9, no. 9, pp. 835–841, 2015, doi: 10.1049/iet-its.2014.0090.

[15] R. Patel and P. Meduri, "Car Detection Based Algorithm for Automatic Parking Space Detection," in *Proceedings - 19th IEEE International Conference on Machine Learning and Applications, ICMLA 2020*, 2020, pp. 1418–1423. doi: 10.1109/ICMLA51294.2020.00220.

[16] G. Amato, F. Carrara, F. Falchi, C. Gennaro, and C. Vairo, "Car parking occupancy detection using smart camera networks and Deep Learning," in *Proceedings - IEEE Symposium on Computers and Communications*, 2016, vol. 2016-August, pp. 1212–1217. doi: 10.1109/ISCC.2016.7543901.

[17] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[18] R. M. Nieto, A. Garcia-Martin, A. G. Hauptmann, and J. M. Martinez, "Automatic Vacant Parking Places Management System Using Multicamera Vehicle Detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 3, pp. 1069–1080, 2019, doi: 10.1109/TITS.2018.2838128.

[19] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object

detection with region proposal networks," *Adv. Neural Inf. Process. Syst.*, vol. 2015-January, pp. 91–99, 2015.

[20] A. Dutta and A. Zisserman, "The VIA annotation software for images, audio and video," in *MM 2019 - Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 2276–2279. doi: 10.1145/3343031.3350535.

[21] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," 2020. [Online]. Available: http://arxiv.org/abs/2004.10934

[22] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2980–2988.

[23] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," 2018. [Online]. Available: http://arxiv.org/abs/1804.02767

[24] Mathworks, "Object Detection Using YOLO v3 Deep Learning," *MathWorks, Inc.*, 2020. https://www.mathworks.com/help/vision/ug/object-detection-using-yolov4-deep-learning.html (accessed Dec. 12, 2022).