**Indonesian Journal of Electrical Engineering and Computer Science**
Vol. 36, No. 3, December 2024, pp. 1640∼1654
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v36.i3.pp1640-1654

# Fusion algorithms on identifying vacant parking spots using vision-based approach

**Ginanjar Suwasono Adi[1], Hertog Nugroho[2], Griffani Megiyanto Rahmatullah[2], Muhammad Yusuf Fadhlan[2], Dinan Mutamaddin[2]**
[1]Department of Electrical Engineering, Politeknik Negeri Malang, Malang, Indonesia
[2]Department of Electrical Engineering, Politeknik Negeri Bandung, Bandung, Indonesia

**Article Info**

***Article history:***

Received Jul 3, 2024
Revised Aug 7, 2024
Accepted Aug 19, 2024

***Keywords:***

Computer vision
Fusion detection method
Object detection
Occupancy state algorithms
Parking management system
YOLOv7

**ABSTRACT**

In densely populated cities, parking space scarcity results in issues like traffic congestion and difficulty finding parking spots. Recent advancements in computer vision have introduced methods to address parking lot management challenges. The availability of public image datasets and rapid growth in deep learning technology has led to vision-based parking management studies, offering advantages over sensor-based systems in comprehensive area coverage, cost reduction, and additional functionalities. This study presents an innovative fusion algorithm that integrates object detection with occupancy state algorithms to accurately identify vacant parking spaces. The employment of the YOLOv7 framework for vehicle instance segmentation, combined with three occupancy algorithms Euclidean distance (ED), intersection over reference (IoR), and intersection over union (IoU) are compared to determine the occupancy state of observed areas. The proposed method is evaluated using the CNRPark-EXT dataset, and its performance is compared with state-of-the-art methods. As a result, the proposed approach demonstrates robustness under varying conditions. It outperforms existing methods in terms of system evaluation performance, achieving accuracies of 98.88%, 97.99%, and 90.04% for ED, IoR, and IoU, respectively. This fusion detection method enhances adaptability and addresses occlusions, emphasizing YOLOv7's advantages and accurate shape approximation for slot annotation. This study contributes valuable insights for effective parking management systems and has potential usage in the real-world implementation of intelligent transportation systems.

***Corresponding Author:***

Hertog Nugroho
Department of Electrical Engineering, Politeknik Negeri Bandung
Bandung 40012, West Java, Indonesia
Email: hertog@polban.ac.id

## 1. INTRODUCTION

The scarcity of parking spaces in highly populated cities is causing difficulties for people to park their vehicles, leading to issues like traffic congestion, inconveniency finding suitable parking spots, and limited accessibility [1], [2]. This also impacts their overall vehicle trip experience [3]. Consequently, effective parking lot management necessitates real-time monitoring of vehicle parking occupancy [4], [5]. Being aware of available spots and providing this information to users can decrease bottlenecks, improve scalability, and quicker identification of vacant parking areas. Different approaches have been suggested to deal with the issue of parking management in urban areas. Solutions using wireless sensor networks (WSN) are well-regarded for their ease of implementation, affordability, and ability to work with different sensors that can monitor parked vehicles [6]. Sensors such as ultrasonic [7], [8] and inductive loop sensors [7], [9]–[12] are frequently employed in the WSN research for smart parking management systems [4]. On the other hand, the extended observation areas and the enormous number of parking slots might create real-world deployment challenges, leading to high maintenance expenses. Additionally, in WSN-based systems, the algorithm for determining the state of a parking spot is relatively straightforward. Each sensor is being assessed using thresholds to decide the occupancy state. Despite this simplicity, the sensors' limits in object identification and lack of intelligence degrade accuracy, leading to potential false detection.

In recent years, various computer vision methods have been proposed to tackle challenges associated with parking lot management [5]. These approaches focus on analyzing images captured from parking areas and address various goals, such as identifying the boundaries of each parking space [13]–[19]; classifying individual parking spaces to determine if they are occupied or unoccupied [6], [20]–[26]; and counting the number of vehicles in images [27]–[30]. Furthermore, the availability of public image datasets [23], [31], [32] and rapid growth of deep learning technology [33]–[41] have produced several state-of-the-arts (SoTAs) in vision-based parking management studies [6], [23], [25], [26], [31], [32], [42], [43]. Computer vision techniques have an edge over individual sensors in WSN-based systems. These methods can cover extensive parking areas using just one camera, avoiding the necessity of a separate sensor per spot. This minimizes setup and maintenance costs. Furthermore, cameras can aid in tasks like surveillance and identifying unusual behavior.

This study addresses the problem of recent occupancy detection methods in parking surveillance systems. Existing methods exhibit limitations, including vague slot definitions, high computational costs, and potential bias issues [6], [31], [32], [42]. Ke *et al.* [6] proposed a method relying on a matching algorithm and leaving it unclear how parking slots are defined. Another approach employs computationally complex detectors, incurring significant expenses for slot annotation [32]. Additionally, some methods proposed by Almeida *et al.* [31] exhibit bias, performing well only under specific conditions. Moreover, certain classification approaches utilize patches for slot assessment [23], [25], [26], [31], [43], introducing problems related to overlapping annotations and limited feature information for distant slots. These limitations highlight the necessity for a more accurate and adaptable approach to occupancy detection in parking surveillance systems.

To overcome these challenges, we developed a fusion detection method to identify the occupancy state of parking slots using the publicly available CNR-EXT dataset [23]. This method achieves a substantial breakthrough in system performance compared to state-of-the-art (SoTA) methods. We contribute to showing that our method has better dealt with accuracy, performance, and robustness. We have devised a pipeline that combines object detection through computer vision techniques with an occupancy state algorithm to enable detection under extreme lighting conditions and occlusions. Furthermore, our comprehensive experimental results and findings serve as a valuable resource for future research.

This paper is organized as follows. In section 2, we describe details of the complete system of the proposed method. Section 3 presents the experiments and results obtained by the system. This section presents and explains the evaluation framework (fine-tuning hyperparameters) used to achieve optimized results of the YOLOv7 best model, the system performance of occupancy detection methods, and the positioning of this research. Finally, section 4 describes the conclusions of the paper and the possible future research that could build upon these findings.

## 2. METHOD

The proposed framework's overview is presented as a flow diagram in Figure 1. It consists of two main components: (i) data preprocessing and training and (ii) data testing and occupancy detection. The system's architecture is tuned to balance computing efficiency, reliability, and scalability.

Assuming that each camera position is fixed to monitor the desired area in the parking lot, we begin by manually annotating and labeling each parking slot to obtain regions of interest (ROIs) for individual parking spots from 9 distinct camera positions, which were acquired from the CNRPark-EXT dataset [23]. Once all the desired slots have been annotated, the resulting coordinate vertices for these slots are saved as blobs in our system for future use.

As part of the vehicle detection method, we utilized renowned computer vision algorithms, YOLOv7 [44] due to their optimal accuracy and efficiency for object detection tasks. To improve the robustness and accuracy of our system's performance, we propose a combination of object detection and occupancy state algorithms that use intersection over union (IoU), intersection over reference (IoR), and Euclidean distance (ED), which will be explained in the following subsection.

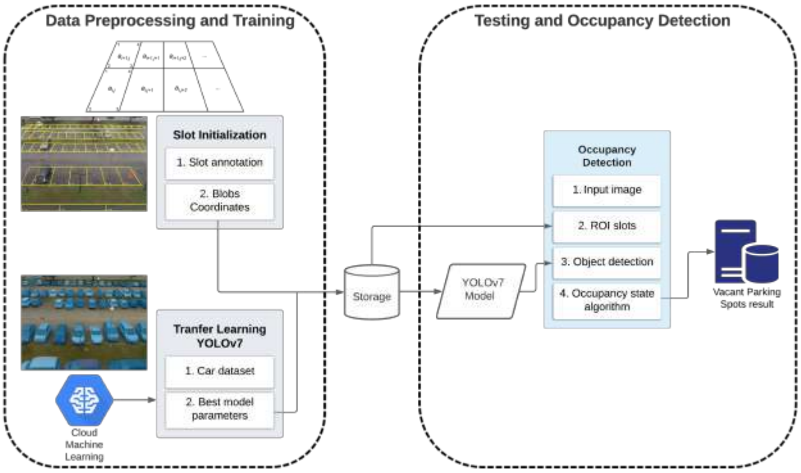


Figure 1. Overview of the end-to-end pipeline design and methodology. It is made up of two primary parts: (i) data preprocessing and training and (ii) data testing and occupancy detection

## 2.1. Slot initialization

Given an image of a parking area, each parking slot is manually annotated to give a close approximation of its true form based on four coordinates as shown in Figure 2; for $k$ = {1, 2, 3, 4}, where ($x_k$, $y_k$) represent the horizontal and vertical axis coordinates of each parking spot, respectively. Then, a unique ID is assigned to each parking space based on the $i^{th}$ row and $j^{th}$ column indices. We represent each parking space with its ID as a list,

$$a_{ij} \in \mathbb{N}, \begin{matrix} i = 1, 2, \ldots, I \\ j_m = 1, 2, \ldots, J; m = 1, 2, \ldots, M \end{matrix} \tag{1}$$

$$(x_k, y_k) \rightarrow \boldsymbol{a} := \{\{x_1, y_1\}, \{x_2, y_2\}, \{x_3, y_3\}, \{x_4, y_4\}, \{a_{ij}\}\} \tag{2}$$

where the components of list $a_{ij}$ denote the location of parking slot in row $i$ and column $j$. It should be emphasized that the value of $J$ for the $j^{th}$ column may vary across different rows, which is indicated by $M$. As a result, a tuple may represent the complete specified slot ID along with its coordinates as expressed in (2).

Figure 3 shows the comparison between a raw image from camera eight [23], represented in Figure 3(a), and the outcome of the slot initialization step, shown in Figure 3(b). In the latter, yellow lines designate each parking slot, and white text indicates the ID of each slot. The resulting tuple of $\boldsymbol{a}$ is then saved in our system for future procedures once the appropriate slots have been defined.

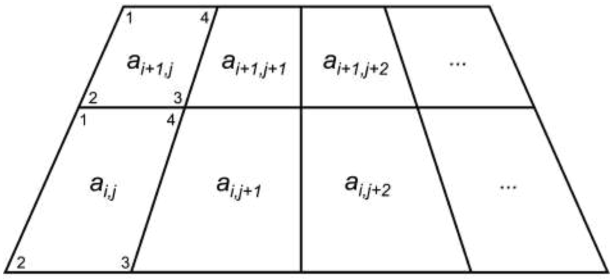
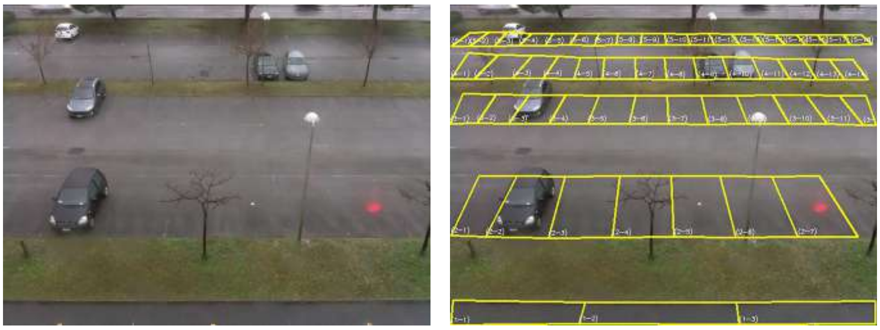


Figure 2. Annotation process to obtain slot coordinates



(a) (b)

Figure 3. The comparison (a) raw image from camera eight [23] and (b) slot initialization based on the same camera view

## 2.2. Transfer learning for car detection

### 2.2.1. Dataset preparation

CNRPark-EXT dataset used in this research consists of 810 parking lot images gathered from 9 cameras with different viewpoints and various environmental conditions: sunny, overcast, and rainy. This dataset contains information on the binary classification of 2 labeled classes: class 1: occupied slot, which has a car image; class 2: vacant slot, which is the annotated image of an empty slot. However, since we want to increase the robustness and accuracy in our system instead of using binary classification as in [23], we used instance segmentation to detect vehicles in the observed image and then evaluate their position in the slot area. Thus, we need to re-annotate our dataset for data preparation. Roboflow platform was employed to resize the input image into $640 \times 640$ pixels so that this input will be compatible with the YOLOv7 detector. Additionally, we annotated vehicles individually as polygon shapes for segmentation purposes. To vary our dataset, we used augmentation tools in the Roboflow platform by rotating our train set randomly between 15° clockwise and counter-clockwise so that we gathered 1,243 image frames with a total of 16,653 car instances from the CNRPark-EXT dataset. Then, we randomly divided all datasets from 1,243 images with the portion of 70%, 20%, and 10% for the train, validation, and test sets, respectively.

### 2.2.2. Transfer learning YOLOv7

In this research, we employ the YOLOv7 architecture [44] to perform instance segmentation for vehicle detection. YOLOv7 stands out in vehicle segmentation due to its innovative enhancements, as highlighted in several studies. The YOLOv7 algorithm introduces improvements such as lightweight models with high detection accuracy, low complexity, and reduced parameter count [45]. Additionally, the YOLOv7-based method [46] also excels in instance segmentation, resulting in superior segmentation accuracy and faster inference speeds compared to other popular algorithms like mask region-based convolutional neural network (R-CNN) [37] and YOLACT [41]. Moreover, the YOLOv7-based vehicle detection method addresses challenges such as low accuracy in detecting small and occluded targets by utilizing a pyramid pooling structure and parallel channel-spatial attention modules, achieving high average precision and processing speed [47]. Therefore, these collective advancements highlight YOLOv7's superiority in vehicle segmentation over other models, making it suitable for this study. After data preparation was completed, we processed the training set for transfer learning to achieve the best YOLOv7 model. We then validated the model using the validation set to examine the performance of our inference model. Finally, the best weights from the evaluation process were applied to our system for future use.

## 2.3. Occupancy state algorithms

We compared three occupancy state algorithms: IoU, IoR, and ED. In IoU was implemented to estimate the state of each slot. The key to this algorithm is to compare the intersection area of a specific slot $ROIS_{ij}$ and the area of a selected car $ROIC_n$. Since the shape of every parking space might vary, taking on the form of a polygon, we can measure the ROIs of each parking space by using the quadrilateral equation as,

$$ROIS_{ij} = \left| \frac{(x_1y_2 - y_1x_2) + \cdots + (x_4y_1 - y_4x_1)}{2} \right| \tag{3}$$

where $ROIS_{ij}$ is a list containing all the calculated areas of each parking space. Whereas $ROIC_n$ is a list of every area of detected cars that can be obtained from YOLOv7 instance segmentation inference. Hence, the IoU can be computed as,

$$IoU_{ij} = \max_{i=1,j=1}^{I,J_m} \left[ \frac{ROIS_{ij} \cap ROIC_n}{ROIS_{ij} \cup ROIC_n} \right] \tag{4}$$

The output of the slot state after performing the IoU algorithm can be expressed as,

$$Pred(a_{ij}) = \begin{cases} occupied, & IoU_{ij} \geq Th \\ vacant, & otherwise \end{cases} \tag{5}$$

where $Th$ is the ratio of $(ROIS_{ij} \cup ROIC_n)$. However, since this algorithm depends on the threshold value corresponding to the percentage of overlap between the evaluated $ROIS_{ij}$ and $ROIC_n$, the IoU algorithm may fail to detect occupied slots that are far away from the camera when higher threshold values are set. Therefore, we propose two other algorithms, ED and IoR, to minimize the error of false detection in occupancy state detection. The details of these algorithms are described in the following subsection.

### 2.3.1. Intersection over reference

In Algorithm 1, we proposed the IoR. This algorithm aims to maximize the occupancy area of a detected car onto an individual slot by modifying the divisor of IoU as a factor of reference area $ROIS_{ij}$. By evoking (3), The IoR formula can be expressed as,

$$IoR_{ij} = \max_{i=1,j=1}^{I,J_m} \left[ \frac{ROIS_{ij} \cap ROIC_n}{ROIS_{ij}} \right] \tag{6}$$

Algorithm 1. IoR to determine slot state

```
Input: parking slot image, slot coordinate, YOLOv7 best model
Output: $Pred(a_{ij})$
Initialisation:
compute YOLOv7 instance segmentation
compute $ROIS_{ij}$, $ROIC_n$
for all slot in row do
    set $Th$
    compute IoR
    sort max IoR
    if max IoR $\geq Th$ then
        set $Pred(a_{ij})$ occupied
    else
        set $Pred(a_{ij})$ vacant
    end if
    update $Pred(a_{ij})$
end for
return $Pred(a_{ij})$
```

$$Pred(a_{ij}) = \begin{cases} occupied, & IoR_{ij} \geq Th \\ vacant, & otherwise \end{cases} \tag{7}$$

In (7) can be applied to predict the slot state of this algorithm, where the threshold value $Th$ can be set as the proportion of $ROIS_{ij}$. Additionally, we also apply a sorting mechanism to prevent redundancy of a certain vehicle occupying more than one slot using $\operatorname{argmax}$ function for IoU and IoR algorithms in this study.

### 2.3.2. Euclidean distance

Algorithm 2 shows the implementation of ED to predict slot state in an observed image. To perform the ED algorithm, the centroid of every individual slot will be assessed for all detected centroid cars. Recalling (2), where each element represents the coordinates of every distinctive slot, the center of gravity of each slot can be calculated as,

$$C_{ij} = \left( \frac{x_1 + x_2 + x_3 + x_4}{4}, \frac{y_1 + y_2 + y_3 + y_4}{4} \right) \tag{8}$$

Algorithm 2. ED to determine slot state

```
Input: parking slot image, slot coordinate, YOLOv7 best model
Output: Pred($a_{ij}$)
Initialisation:
compute YOLOv7 instance segmentation
compute $C_{ij}$, $C_n$
for all slot in row do
    set $Th$
    compute ED
    sort min ED
    if min ED $\leq Th$ then
        set Pred($a_{ij}$) occupied
    else
        set Pred($a_{ij}$) vacant
    end if
    update Pred($a_{ij}$)
end for
return Pred($a_{ij}$)
```

Then, the first-order image moments are calculated to provide the centroid of each vehicle detected in the frame after running YOLOv7 inference. This requires adapting a greyscale image with pixel intensities represented as $I(x, y)$, which allows for the computation of image moments $M_{vw}$ as,

$$M_{vw} = \sum_x \sum_y x^v y^w I(x, y) \tag{9}$$

the sum of all pixel intensities in the image of the grey level is defined as the zeroth-order moment $M_{00}$. The first-order moments, $M_{10}$ and $M_{01}$, provide information about the spatial distribution of the pixel intensities along the $x$ and $y$ axes, respectively. These moments are used to calculate the centroid of the instance car in the directions $x$ and $y$. Thus, the centroid of all detected cars can be calculated as follows (10).

$$C_n(x, y) = \left( \frac{M_{10}}{M_{00}}, \frac{M_{01}}{M_{00}} \right) \tag{10}$$

After obtaining $C_{ij}$ and $C_n$ we perform Euclidean distance for all centroid slots to every centroid of detected cars by the following equation,

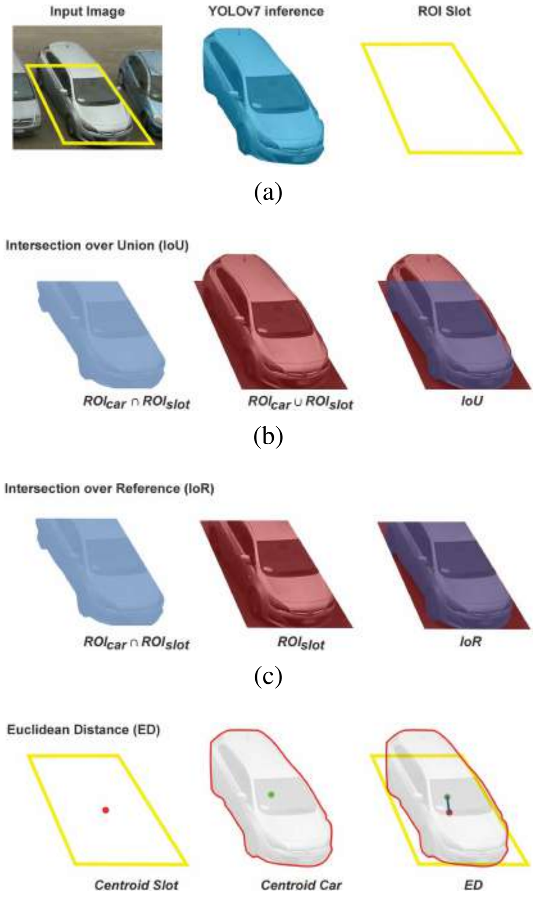
$$d(p, q) = \sum_{i=1}^{I} \sum_{j=1}^{J} \left( \sum_{n=1}^{N} (q_{i,j} - p_n)^2 \right)^{1/2} \tag{11}$$

where $p \in C_n$ centroid of every car and $q \in C_{ij}$ centroid of each individual slot. Let $\{Th \in min\{diag(a_{ij})\}\}$ be the shortest diagonal for all individual slots of slot $j$ in row $i$, where the components of vector $Th_{ij}$ represent the value of thresholds for every slot $j$ in row $i$. Thus, the output of the slot state whether the parking slot is vacant or occupied can be represented as,

$$Pred(a_{ij}) = \begin{cases} occupied, & \min\{d(p, q)\} \leq Th \\ vacant, & otherwise \end{cases} \tag{12}$$

Note that to avoid redundancy of a particular slot occupied by more than one vehicle, we implement a simple sorting procedure that will evaluate those specific slots only with the closest vehicle. To aid comprehension of these algorithms' functionality, Figure 4 offers a visual representation of the proposed methods. The top row of Figure 4(a) corresponds to the system input, which consists of the input image, YOLOv7 inference, and the slot area $ROIS_{ij}$. The second row Figure 4(b) depicts the IoU algorithm, where it shows a conventional method for evaluating the occupancy state of a specific slot by showing the overlap between the intersection of $(ROIS_{ij} \cap ROIC_n)$ and the union of $(ROIS_{ij} \cup ROIC_n)$. Following this, the third and fourth rows illustrate our proposed algorithms: Figure 4(c), the IoR algorithm, which focuses on the proportion of $(ROIS_{ij} \cap ROIC_n)$ to the desired $ROIS_{ij}$, and Figure 4(d), the ED algorithm, which measures the spatial accuracy of the detected vehicle to the designated slot by calculating the Euclidean distance between the centroid of the slot

$C_{ij}$ and the centroid of the car $C_n$. Together, these visual elements in Figure 4 provide a comprehensive guide to understanding the input system and the progression from the IoU algorithm to the proposed IoR and ED methods, highlighting their distinct approaches and contributions to fusion algorithms for identifying vacant parking spots.



(a)

(b)

(c)

(d)

Figure 4. Visual illustration of the proposed methods. Top row: (a) depicts the input system, which consists of the input image, YOLOv7 inference, and the ROI slot; (b) represents the IoU algorithm; (c) illustrates the IoR algorithm; and (d) shows the ED algorithm

## 3. RESULTS AND DISCUSSIONS

### 3.1. YOLOv7 performance

In this study, we utilized transfer learning instead of training our deep learning model from scratch. This approach allowed us to achieve improved model performance by leveraging our data distribution while reducing the computational cost. In Figure 5, the comparison between YOLOv7 performance with default hyperparameters and optimized hyperparameters during the training phase is portrayed. We employed the default hyperparameters setup on YOLOv7 for the initial training process using 50 epochs to our dataset. Figures 5(a) and 5(b) portrays the loss performance of the training versus validation process and the result of model performance for default hyperparameters YOLOv7, respectively. Based on the graph in Figure 5(a) we can analyze where at about $17^{th}$ epoch, the validation loss starts to appear saturated, while the training loss shows a slight downtrend until the end of the learning process. Even though the model performance of this setup produced good marks on the mean average precision (mAP@50) of 97.5%, precision of 96.9%, and recall of 94.9%, this model implies overfitting to our data distribution.

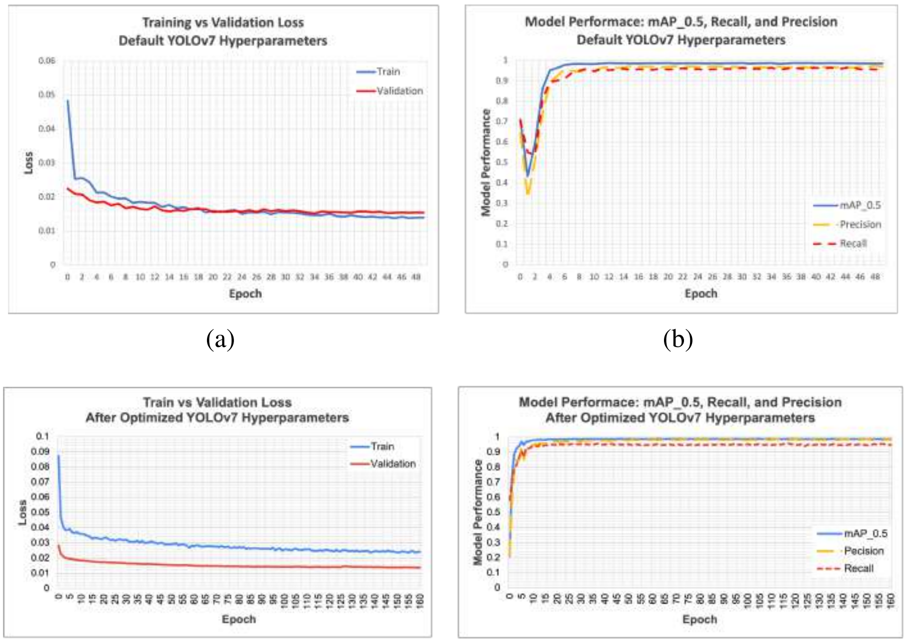
Given this circumstance, we tried to optimize the hyperparameters and the learning process of YOLOv7 to avoid overfitting conditions that may cause the model to only remember our data train distribution instead of having the ability to accomplish generalization. We discovered that the best setup for YOLOv7 hyperparameters to our data distribution, as stated in Table 1 is by using 500 epochs, early stopping of 20 patience, and applying the stochastic gradient descent (SGD) optimizer. It is essential to highlight that the unspecified hyperparameters listed in Table 1 are determined using the default configuration. An early stopping mechanism is intended to discontinue the learning process if there is no improvement in the training loss of as many as 20 epochs. Hence, we do not need to wait until the learning process of 500 epochs is finished. YOLOv7 allows us to use adaptive gradient algorithms (ADAM) and SGD optimizers. ADAM can adjust the learning rate for each parameter during training. It can progress faster in the training phase but may not always result in the best generalization performance. The geometry adaptation in ADAM, through adaptively scaling gradient coordinates, reduces anisotropic gradient noise and increases the Radon measure of a basin. As a result, ADAM tends to take longer to escape sharp minima with small Radon measures.

In contrast, SGD uses a fixed learning rate for all parameters that leads to improved model performance slowly but could achieve higher test performance. The SGD's local instability allows it to better navigate towards flatter minima with more considerable Radon measure. Flatter minima, commonly associated with generalized solutions, explain why ADAM usually suffers from worse generalization performance than SGD [48]–[51]. With the rapid advancement of parallel computing technologies, such as graphics processing units (GPUs) and tensor processing units (TPUs), the time and computational cost of training deep learning models are no longer significant obstacles. As a result, obtaining the parameters required for training data has become significantly faster and more affordable. This availability has also been supported by many cloud platforms so that many users can easily design and deploy their deep learning models in the cloud server without necessarily owning a local server. All the computations for the transfer learning stage of this research were performed using Google Colab and the Nvidia® Tesla T4 GPU accelerator.

Table 1. Optimized YOLOv7 hyperparameters

| Hyperparameters | Description | Value | |
|---|---|---|---|
| | | Default | Optimized |
| lr0 | Initial learning rate (lr) at the beginning of training process | 0.01 | 0.001 |
| lrf | Final lr (during training, lr is gradually decreased till reaching final value) | 0.01 | 10e-6 |
| momentum | Determining contribution of the previous gradient update to the current update. As the model learns to recognize cars, momentum helps accelerate convergence and prevents the model from getting stuck in suboptimal local minima. (a higher momentum value leading to more stable and consistent updates) | 0.937 | 0.9 |
| warmup_epoch | Number of epochs for which the learning rate is gradually increased from 0 to lr0. | 3.0 | 1.0 |
| mixup | The probability of applying the mixing images for each training batch augmentation technique during training. The mixup coefficient combines two images into a single image to provide new training examples. It helps to prevent overfitting and can lead to improved performance while enhancing the model to learn more robust and generalized features by training on diverse combinations of images. | 0.0 | 0.5 |

As depicted in Figure 5(c), the results of tuning hyperparameters for YOLOv7 show a noticeable trend in the validation loss, which closely follows the training loss. The validation loss is lower than the training loss because of YOLOv7's tendency to learn more effectively during the training process. This discrepancy can be attributed to the training set's data distribution varying more significantly than the validation set, owing to the data augmentation process. Furthermore, optimizing hyperparameters, particularly selecting a lower final learning rate (lrf), has enabled the SGD optimizer to enhance the model's performance, resulting in smoother and more generalized outcomes. The performance of the optimized YOLOv7 hyperparameters, as illustrated in Figure 5(d), yields comparable results to the default setup with the mAP@50 of 97.4%, precision of 98%, and recall of 93.9%. However, it is worth noting that the default model might still exhibit some overfitting to our specific data distribution. Despite the slight differences observed in mAP@50 and recall between the optimized and default setups, the optimized performance excels in precision, ensuring greater consistency in inferencing. This emphasizes the significance of fine-tuning hyperparameters to achieve more robust and reliable results.



(a)

(b)

(c)

(d)

Figure 5. Performance metrics comparison of: (a) loss value in default YOLOv7 hyperparameters; (b) loss value in optimized YOLOv7 hyperparameters; (c) model performance with default YOLOv7 hyperparameters; and (d) model performance with optimized YOLOv7 hyperparameters

### 3.2. Evaluation performance on occupancy state algorithms

To evaluate our proposed algorithms, we split 10% of a total dataset based on our data preparation stage. This results in 125 unique images from CNRPark-EXT of 9 different view cameras in 3 conditions (sunny, overcast, and rainy). Evaluation performance metrics of accuracy (13), precision (14), recall (15), and F1-score (16) were applied to predict how our proposed method performed in this study based on the ground truth label from CNRPark-EXT using the confusion matrix given in Table 2. Here, TP indicates true positive, TN refers to true negative, while FP and FN represent false positive and false negative, respectively.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{13}$$

$$Precision = \frac{TP}{TP + FP} \tag{14}$$

$$Recall = \frac{TP}{TP + FN} \tag{15}$$

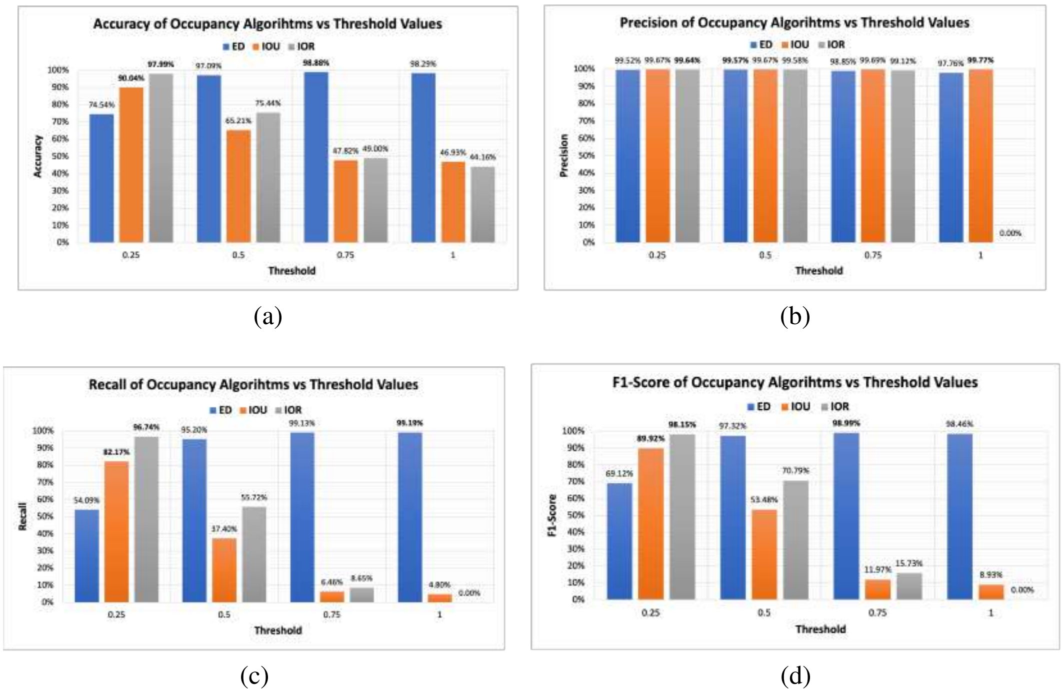
$$F1 - score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{16}$$

Table 2. Confusion matrix of occupancy detection algorithms

| Ground truth | Prediction | Evaluation |
|---|---|---|
| Occupied | Occupied | TP |
| Empty | Empty | TN |
| Empty | Occupied | FP |
| Occupied | Empty | FN |

In this study, it is essential to note that we employed various threshold values, denoted as $\{Th \in \{0.25, 0.5, 0.75, 1\}\}$, for each of the proposed algorithms in order to determine the optimal solution for each method. In the case of the ED algorithm, as outlined in (12), the threshold value is defined as the proportion of the shortest diagonal distance for each slot. Conversely, for the IoU algorithm, as described in (5), the threshold value corresponds to the percentage of overlap between the evaluated $ROIS_{ij}$ and $ROIC_n$. Lastly, the IoR algorithm determines the threshold value as the ratio of the evaluated $ROIS_{ij}$, as indicated in (7). As a result, Figure 6 illustrates the evaluation performance of this study's proposed methods. Generally, the ED algorithm demonstrates an upward trend as its threshold value increases with respect to accuracy, recall, and F1-score metrics. In contrast, the IoU and the IoR algorithms exhibit higher scores for these metrics at lower threshold values, and their performance tends to decline as the threshold value increases.

Specifically, Figure 6(a) illustrates that the ED algorithm attains its highest accuracy of 98.88% at a threshold value of 0.75. Figure 6(b) demonstrates that the ED algorithm achieves its optimal precision of 99.57% at a threshold of 0.5. Furthermore, Figure 6(c) shows that the ED algorithm reaches its highest recall of 99.19% at a threshold of 1. Figure 6(d) highlights that the ED algorithm achieves the highest F1-score of 98.99% at a threshold value of 0.75. The IoU algorithm's peak performance is observed at a threshold value of 0.25, achieving an accuracy of 90.04%, as depicted in Figure 6(a), a recall of 82.17%, as shown in Figure 6(c), and an F1-score of 89.92%, as indicated in Figure 6(d). Additionally, Figure 6(b) reveals that the IoU algorithm achieves a precision of 99.77% at a threshold value of 1. Lastly, there are interesting outcomes for the IoR algorithm. At a threshold value of 1, the results show 0% in precision shown in Figure 6(b), recall displayed in Figure 6(c), and F1-score represented in Figure 6(d) because there are no true positive detections. According to (6), this threshold requires the $(ROIS_{ij} \cap ROIC_n)$ to perfectly overlap 100% with the $ROIS_{ij}$. The 0% precision and the 0% recall signifies that no actual occupied slots were correctly detected. Consequently, the F1-score is also 0%, confirming the algorithm's failure to identify any true positives at this threshold. This strict requirement for perfect overlap is likely impractical, and adjusting the threshold to allow for less than 1 could improve the IoR algorithm's performance. the IoR algorithm demonstrates its best performance at a threshold value of 0.25, achieving an accuracy of 97.99%, as depicted in Figure 6(a), a precision of 99.64%, as shown in Figure 6(b), a recall of 96.74%, as illustrated in Figure 6(c), and an F1-score of 98.15%, as indicated in Figure 6(d). The sample results of occupancy detection from proposed methods using best threshold parameters are depicted in Figure 7. The regions in red are indicated as occupied slots, while the green areas are specified as vacant slots and the blue areas represent detected vehicles through instance segmentation using the YOLOv7.



(a)

(b)

(c)

(d)

Figure 6. The comparison of performance analysis for proposed methods using different metric: (a) accuracy, (b) precision, (c) recall, and (d) F1-score

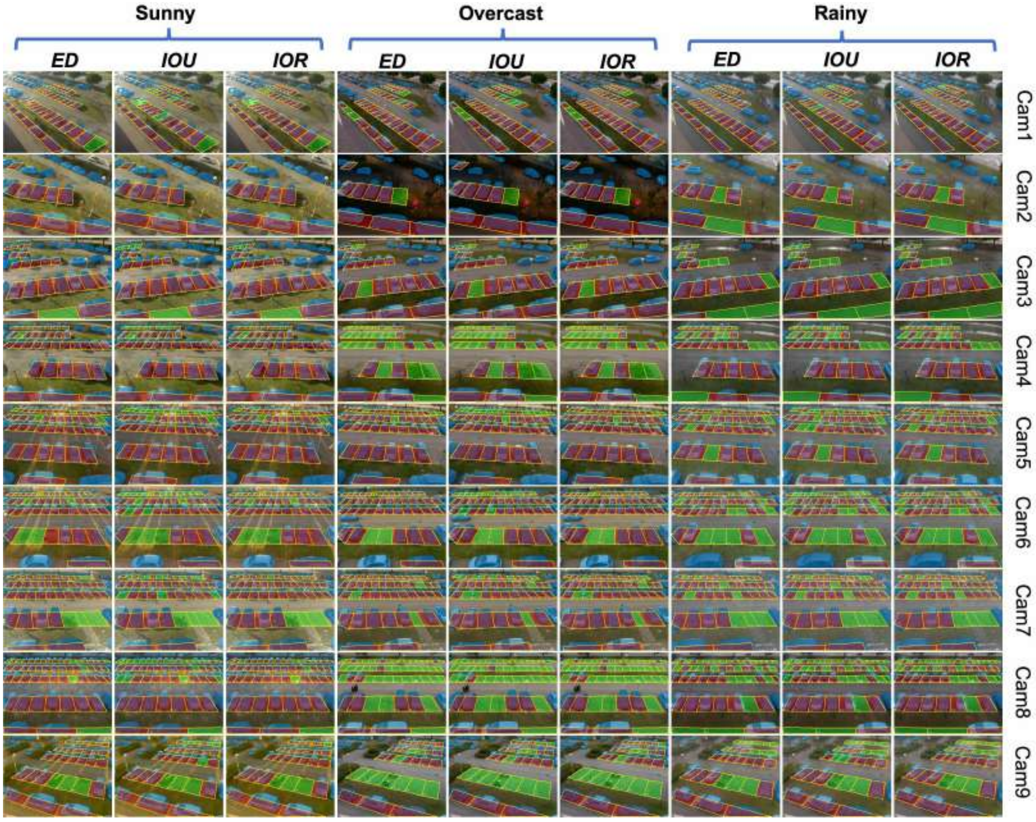


Figure 7. The visualization of occupancy detection from 9 distinct images in the CNRPark-EXT dataset under different weather conditions (sunny, overcast, and rainy)

### 3.3. Positioning of this research in comparison with SoTAs

We have summarized the comparative analysis between our research and the SoTAs for occupancy detection in parking surveillance systems in Table 3. It is essential to highlight that there can be variations in terms of system inputs and computational processes across different systems. Furthermore, we emphasize evaluating system performance and the primary algorithm procedures. In contrast to an alternative detection method [6], our proposed approach demonstrates better accuracy and incorporates a broader assortment of evaluation metrics for the validation process. While [6] has made commendable contributions by integrating its system into an edge device and validating its methodology using a proprietary dataset, the outlining of how they define the evaluated slot within the observed parking lot remains vague, as their approach hinges on a matching algorithm. Notably, their proposed algorithm cannot ascertain occupancy status when the SSD detector fails to localize a vehicle.

Conversely, our study involves a pre-initialization phase wherein every slot within each camera view is meticulously annotated and stored within our system. This particular slot annotation process facilitates accurate occupancy assessments within the designated areas. Another study [32] leveraged homographic transformation properties and adjusted correction factors to establish a uniform grid for directly mapping discrete parking spot numbers without necessitating manual supervision. However, these steps entail substantial computational expenses regarding the slot annotation process. Regarding the primary detection algorithm, Nieto *et al.* [32] employs Faster R-CNN detectors, known for their inherent computational complexity. In contrast, our system utilizes YOLOv7, balancing competitive accuracy and faster inference speeds, rendering it particularly well-suited for real-time applications.

In contrast to the classification methods outlined in [23], [25], [26], [31], [43], our approach employs a fusion detection method to discern the occupancy state of observed parking slots, thereby providing heightened adaptability and resilience in addressing occlusion scenarios. While the utmost accuracy achieved by [31] stands at 99.6%, their outcomes exhibit a particular bias, indicating that the model's performance remains robust primarily when the training subset closely resembles the test subset, resulting in marked disparities when exposed to diverse test sets. In terms of primary algorithms, [23] utilized a CNN-based approach employing the AlexNet architecture, while [43] introduced the quantized SimpleNet (Q-SimpleNet). Additionally, [25] applied EfficientParkingNet, and [26] employed a modified MobileNetV3 in their respective works. Similar to our study, they utilized the CNRPark EXT dataset. However, a notable difference from our approach is their use of patches for individual slots to determine their occupancy status. In contrast, our method employs detection algorithms for assessment. Notably, as they annotate each slot with a bounding box, overlapping annotations between adjacent slots could introduce interference capable of influencing the inference results of the applied detectors. Furthermore, the size of the defined patches on distant slots diminishes due to their greater distance from the camera. Consequently, employing minute bounding boxes may predispose the outcomes to bias or misclassification due to the limited feature information within the evaluated area.

In summary, our approach endeavours to preserve the information of each slot to the greatest extent feasible by emulating designated slots akin to real-shape conditions. As evidenced by system performance outcomes, the accuracy of the ED algorithm within our proposed method slightly surpasses that of [23], [25], [26], [43]. Additionally, our intersection over reference (IoR) algorithm demonstrates comparable performance accuracy to that reported in [23].

Table 3. Comparison between proposed method and state of the arts

| Research study | Input | Computation process | Pipeline logic | Primary algorithm | Dataset | Testing scenarios | System performance metrics |
|---|---|---|---|---|---|---|---|
| $^{*}$Almeida *et al.* 2015 [31] | Image | Desktop | Classification | SVM and LBP/LPQ | PKLot image dataset | Outdoor, clear, sky, overcast, rainy | Accuracy: 84.2%-99.6% AUC: 0.9194-0.9998 |
| Amato *et al.* 2017 [23] | Image | Edge device | Classification | CNN: AlexNet | CNRPark EXT image dataset | Outdoor, sunny, overcast, rainy | Accuracy: 98% AUC:0.9974 |
| Nieto *et al.* 2018 [32] | Video | Desktop | Detection | Homographic transformation, Faster R-CNN, fusion | PLds image dataset | Outdoor, clear, rainy | AUC:0.919 |
| $^{*,**}$Ke *et al.* 2020 [6] | Video | Edge device and server | Detection | SSD, BG, SORT, fusion | Pascal VOC, MIO-TCD, and private dataset | Outdoor, indoor, sunny, cloudy, rainy, foggy | Accuracy: 95.6% |
| Zhuang *et al.* 2022 [43] | Image | Desktop and Edge device | Classification | CNN: Q-SimpleNet | CNRPark and PKLot image datasets | Outdoor, sunny, overcast, rainy | Accuracy: 98.6% @CNRPark, 97.5% @PKLot |
| Rahman *et al.* 2022 [25] | Image | Desktop | Classification | Efficient-ParkingNet | CNRPark image datasets | Outdoor, sunny, overcast, rainy | Accuracy: 98.44% |
| Yuldashev *et al.* 2023 [26] | Image | Desktop | Classification | Modified MobileNetV3 | CNRPark and PKLot image datasets | Outdoor, sunny, overcast, rainy | Accuracy: 98.01% |
| **Proposed study** | **Image** | **Cloud based** | **Detection** | **YOLOv7, ED$^{\#}$, IOU$^{+}$, IOR$^{++}$, fusion** | **CNRPark EXT image dataset** | **Outdoor, sunny, overcast, rainy** | **Accuracy: 98.88%$^{\#}$, 90.04%$^{+}$, 97.99%$^{++}$ F1-score: 98.99%$^{\#}$, 99.2%$^{+}$, 98.15%$^{++}$** |

$^{*}$The test procedure may have led to biased results, or it is not clear. $^{**}$Authors included private dataset in the test. $^{\#}$The evaluation performance of Euclidean distance algorithm. $^{+}$The evaluation performance of intersection over union algorithm. $^{++}$The evaluation performance of intersection over reference algorithm.

## 4. CONCLUSION

This paper presents an approach that transcends current methods for accurately identifying the occupancy state of parking spaces using a public dataset. The proposed fusion algorithms combine computer vision techniques with occupancy state algorithms, resulting in enhanced accuracy and robustness in system evaluation performance. The developed pipeline integrates object detection through an optimized YOLOv7 framework with multiple occupancy state algorithms: IoU, IoR, and ED. Comprehensive experimental evaluations provide valuable insights into the performance of these methods, demonstrating the approach's superiority over SoTA techniques with accuracies of 98.88% for the ED algorithm, 90.04% for the IoU algorithm, and 97.99% for the IoR algorithm. This research advances the field by offering an innovative solution that surpasses existing methods and contributes to the development of more reliable parking surveillance systems.

Nevertheless, this study has a few important limitations that should be taken into consideration when designing future research investigations concerning occupancy detection in intelligent parking management systems. Since manual annotation and secondary datasets are still used in this study, investigating both indoor and outdoor environments with primary datasets could provide valuable insights into the robustness of the proposed methods. Additionally, addressing the significant challenge of manual labeling in parking spot recognition represents a merit contribution to the field. Furthermore, future research could explore alternative lightweight deep learning models and deploy this system within an artificial intelligence of things (AIoT) architecture to evaluate its efficiency and effectiveness in real-world applications.

## ACKNOWLEDGEMENTS

This work was supported in part by the Center for Research and Community Service Politeknik Negeri Bandung under Grant B/98.35/PL1.R7/PG.00.03/2023.

## REFERENCES

[1] S. Zoika, P. G. Tzouras, S. Tsigdinos, and K. Kepaptsoglou, "Causal analysis of illegal parking in urban roads: the case of Greece," *Case Studies on Transport Policy*, vol. 9, no. 3, pp. 1084–1096, Sep. 2021, doi: 10.1016/j.cstp.2021.05.009.

[2] A. Russo, J. Van Ommeren, and A. Dimitropoulos, "The environmental and welfare implications of parking policies," *OECD Environment Working Papers*, vol. 145, 2019, doi: 10.1787/16d610cc-en.

[3] D. C. Shoup, "Cruising for parking," *Transport Policy*, vol. 13, no. 6, pp. 479–486, Nov. 2006, doi: 10.1016/j.tranpol.2006.05.005.

[4] T. Lin, H. Rivano, and F. Le Mouel, "A survey of smart parking solutions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 12, pp. 3229–3253, Dec. 2017, doi: 10.1109/TITS.2017.2685143.

[5] P. R. L. de Almeida, J. H. Alves, R. S. Parpinelli, and J. P. Barddal, "A systematic review on computer vision-based parking loT management applied on public datasets," *Expert Systems with Applications*, vol. 198, p. 116731, Jul. 2022, doi: 10.1016/j.eswa.2022.116731.

[6] R. Ke, Y. Zhuang, Z. Pu, and Y. Wang, "A smart, efficient, and reliable parking surveillance system with edge artificial intelligence on IoT devices," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 8, pp. 4962–4974, Aug. 2020, doi: 10.1109/TITS.2020.2984197.

[7] S. Lee, D. Yoon, and A. Ghosh, "Intelligent parking lot application using wireless sensor networks," in *2008 International Symposium on Collaborative Technologies and Systems*, May 2008, pp. 48–57, doi: 10.1109/CTS.2008.4543911.

[8] W.-J. Park, B.-S. Kim, D.-E. Seo, D.-S. Kim, and K.-H. Lee, "Parking space detection using ultrasonic sensor in parking assistance system," in *2008 IEEE Intelligent Vehicles Symposium*, Jun. 2008, pp. 1039–1044, doi: 10.1109/IVS.2008.4621296.

[9] J.-H. Moon and T. K. Ha, "A car parking monitoring system using wireless sensor networks," *International Journal of Electrical and Computer Engineering*, vol. 7, no. 10, pp. 1317–1320, 2013, doi: 10.5281/zenodo.1335772.

[10] Z. Zhang, X. Li, H. Yuan, and F. Yu, "A street parking system using wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 9, no. 6, p. 107975, Jun. 2013, doi: 10.1155/2013/107975.

[11] Z. Zhang, M. Tao, and H. Yuan, "A parking occupancy detection algorithm based on AMR sensor," *IEEE Sensors Journal*, vol. 15, no. 2, pp. 1261–1269, Feb. 2015, doi: 10.1109/JSEN.2014.2362122.

[12] E. Sifuentes, O. Casas, and R. Pallas-Areny, "Wireless magnetic sensor node for vehicle detection with optical wake-up," *IEEE Sensors Journal*, vol. 11, no. 8, pp. 1669–1676, Aug. 2011, doi: 10.1109/JSEN.2010.2103937.

[13] K. Ali and G. Mohamed, "Roadside parking spaces image classification using deep learning," in *Lecture Notes in Networks and Systems*, 2021, vol. 199 LNNS, pp. 323–333, doi: 10.1007/978-3-030-69418-0_29.

[14] F. Dornaika, K. Hammoudi, M. Melkemi, and T. D. A. Phan, "An efficient pyramid multi-level image descriptor: application to image-based parking lot monitoring," *Signal, Image and Video Processing*, vol. 13, no. 8, pp. 1611–1617, Nov. 2019, doi: 10.1007/s11760-019-01512-6.

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

[16] Q. Li, C. Lin, and Y. Zhao, "Geometric features-based parking slot detection," *Sensors*, vol. 18, no. 9, p. 2821, Aug. 2018, doi: 10.3390/s18092821.

[17] Y. Gao, C. Lin, Y. Zhao, X. Wang, S. Wei, and Q. Huang, "3-D surround view for advanced driver assistance systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 320–328, Jan. 2018, doi: 10.1109/TITS.2017.2750087.

[18] Z. Wu, W. Sun, M. Wang, X. Wang, L. Ding, and F. Wang, "PSDet: efficient and universal parking slot detection," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, Oct. 2020, pp. 290–297, doi: 10.1109/IV47402.2020.9304776.

[19] 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*, vol. 20, no. 7, p. 2138, Apr. 2020, doi: 10.3390/s20072138.

[20] M. Gregor, R. Pirník, and D. Nemec, "Transfer learning for classification of parking spots using residual networks," *Transportation Research Procedia*, vol. 40, pp. 1327–1334, 2019, doi: 10.1016/j.trpro.2019.07.184.

[21] M. S. Farag, M. M. M. El Din, and H. A. Elshenbary, "Deep learning versus traditional methods for parking lots occupancy classification," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 19, no. 2, pp. 964–973, Aug. 2020, doi: 10.11591/ijeecs.v19.i2.pp964-973.

[22] G. Amato *et al.*, "A wireless smart camera network for parking monitoring," in *2018 IEEE Globecom Workshops (GC Wkshps)*, Dec. 2018, pp. 1–6, doi: 10.1109/GLOCOMW.2018.8644226.

[23] G. Amato, F. Carrara, F. Falchi, C. Gennaro, C. Meghini, and C. Vairo, "Deep learning for decentralized parking lot occupancy detection," *Expert Systems with Applications*, vol. 72, pp. 327–334, Apr. 2017, doi: 10.1016/j.eswa.2016.10.055.

[24] G. Amato, F. Carrara, F. Falchi, C. Gennaro, and C. Vairo, "Car parking occupancy detection using smart camera networks and Deep Learning," in *2016 IEEE Symposium on Computers and Communication (ISCC)*, Jun. 2016, pp. 1212–1217, doi: 10.1109/ISCC.2016.7543901.

[25] S. Rahman, M. Ramli, F. Arnia, R. Muharar, M. Ikhwan, and S. Munzir, "Enhancement of convolutional neural network for urban environment parking space classification," *Global Journal of Environmental Science and Management*, vol. 8, no. 3, pp. 315–326, 2022, doi: 10.22034/gjesm.2022.03.02.

[26] Y. Yuldashev, M. Mukhiddinov, A. B. Abdusalomov, R. Nasimov, and J. Cho, "Parking Lot occupancy detection with improved MobileNetV3," *Sensors*, vol. 23, no. 17, p. 7642, Sep. 2023, doi: 10.3390/s23177642.

[27] W. Li, H. Li, Q. Wu, X. Chen, and K. N. Ngan, "Simultaneously detecting and counting dense vehicles from drone images," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 12, pp. 9651–9662, Dec. 2019, doi: 10.1109/TIE.2019.2899548.

[28] L. Ciampi, G. Amato, F. Falchi, C. Gennaro, and F. Rabitti, "Counting vehicles with cameras," in *SEBD*, 2018, vol. 2161.

[29] P. Dobes, J. Spanhel, V. Bartl, R. Juranek, and A. Herout, "Density-based vehicle counting with unsupervised scale selection," in *2020 Digital Image Computing: Techniques and Applications (DICTA)*, Nov. 2020, pp. 1–8, doi: 10.1109/DICTA51227.2020.9363401.

[30] G. Amato, L. Ciampi, F. Falchi, and C. Gennaro, "Counting vehicles with deep learning in onboard UAV imagery," in *2019 IEEE Symposium on Computers and Communications (ISCC)*, Jun. 2019, pp. 1–6, doi: 10.1109/ISCC47284.2019.8969620.

[31] P. R. L. de Almeida, L. S. Oliveira, A. S. Britto, E. J. Silva, and A. L. Koerich, "PKLot – a robust dataset for parking lot classification," *Expert Systems with Applications*, vol. 42, no. 11, pp. 4937–4949, Jul. 2015, doi: 10.1016/j.eswa.2015.02.009.

[32] R. M. Nieto, A. Garcia-Martin, A. G. Hauptmann, and J. M. Martinez, "Automatic vacant parking places management system using multicamera vehicle detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 3, pp. 1069–1080, Mar. 2018, doi: 10.1109/TITS.2018.2838128.

[33] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, May 2010, pp. 253–256, doi: 10.1109/ISCAS.2010.5537907.

[34] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[35] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: 10.1109/TPAMI.2016.2577031.

[36] W. Liu *et al.*, "SSD: single shot multibox detector," in *Computer Vision–ECCV 2016: 14th European Conference*, 2016, pp. 21–37, doi: 10.1007/978-3-319-46448-0_2.

[37] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017, pp. 2980–2988, doi: 10.1109/ICCV.2017.322.

[38] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 6517–6525, doi: 10.1109/CVPR.2017.690.

[39] J. Redmon and A. Farhadi, "YOLOv3: an incremental improvement," *arXiv preprint*, Apr. 2018, doi: 10.48550/arXiv.1804.02767.

[40] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: optimal speed and accuracy of object detection," *arXiv preprint*, 2020, doi: 10.48550/arXiv.2004.10934.

[41] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "YOLACT: real-time instance segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2019-October, 2019, pp. 9156–9165, doi: 10.1109/ICCV.2019.00925.

[42] H. Nugroho, G. S. Adi, and M. K. Afandi, "Detection of empty/occupied states of parking slots in multicamera system using mask R-CNN classifier," *Current Journal: International Journal Applied Technology Research*, vol. 4, no. 1, pp. 53–68, Apr. 2023, doi: 10.35313/ijatr.v4i1.114.

[43] Y. Zhuang, Z. Pu, H. Yang, and Y. Wang, "Edge–artificial intelligence-powered parking surveillance with quantized neural networks," *IEEE Intelligent Transportation Systems Magazine*, vol. 14, no. 6, pp. 107–121, Nov. 2022, doi: 10.1109/MITS.2022.3182358.

[44] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2023, pp. 7464–7475, doi: 10.1109/CVPR52729.2023.00721.

[45] A. Li, C. Gong, D. Guo, G. Du, W. He, and V. Zyrianov, "Intelligent vehicle object detection based on improved yolov7 algorithm," *Research Square Preprint*, 2024, doi: 10.21203/rs.3.rs-4538660/v1.

[46] W. Xia, P. Li, Q. Li, T. Yang, and S. Zhang, "TTIS-YOLO: a traffic target instance segmentation paradigm for complex road scenarios," *Measurement Science and Technology*, vol. 35, no. 10, p. 105402, Oct. 2024, doi: 10.1088/1361-6501/ad5b10.

[47] J. Cui, D. Xing, X. Gao, L. Qi, and R. Yu, "Road vehicle detection based on deep learning," in *2024 IEEE 2nd International Conference on Control, Electronics and Computer Technology (ICCECT)*, Apr. 2024, pp. 1456–1460, doi: 10.1109/ICCECT60629.2024.10545867.

[48] C. Song, "The performance analysis of Adam and SGD in image classification and generation tasks," *Applied and Computational Engineering*, vol. 5, no. 1, pp. 757–763, May 2023, doi: 10.54254/2755-2721/5/20230697.

[49] A. Gupta, R. Ramanath, J. Shi, and S. S. Keerthi, “Adam vs. SGD: closing the generalization gap on image classification,” in *OPT2021: 13th Annual Workshop on Optimization for Machine Learning*, 2021, pp. 1–7.

[50] Z. Zhang, “Improved Adam optimizer for deep neural networks,” in *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*, Jun. 2018, pp. 1–2, doi: 10.1109/IWQoS.2018.8624183.

[51] P. Zhou, J. Feng, C. Ma, C. Xiong, S. Hoi, and E. Weinan, “Towards theoretically understanding why SGD generalizes better than ADAM in deep learning,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 21285–21296, 2020.

## BIOGRAPHIES OF AUTHORS

**Ginanjar Suwasono Adi** received his B.A.Sc (2013) in Telecommunication Engineering at Politeknik Elektronika Negeri Surabaya, Indonesia, and the M.Sc. (2017) degree in Electrical Engineering and Computer Science at National Taipei University of Technology, Taiwan. From 2017 to 2023, he served as a faculty member in the Department of Electrical Engineering at Politeknik Negeri Bandung. Currently, he holds the position of lead researcher at the Artificial Intelligence of Things (AIoT) research group within the Department of Electrical Engineering at Politeknik Negeri Malang. His research interests include computer vision, deep learning, and integrating IoT applications with artificial intelligence. He can be contacted at email: ginanjar.adi@polinema.ac.id.

**Hertog Nugroho** received B.S. degree in Electrical Engineering from Bandung Institute of Technology, Indonesia, in 1984, M.Sc. degree and Ph.D. degree in Electrical Engineering from Keio University, Yokohama, Japan, in 1995 and 1999 respectively. He is currently working as a Professor in Politeknik Negeri Bandung. His research interests include computer vision, pattern recognition, and signal processing. He can be contacted at email: hertog@polban.ac.id.

**Griffani Megiyanto Rahmatullah** received his Bachelor’s Degree in Telecommunication Engineering from Politeknik Negeri Bandung in 2015. He further advanced his expertise by obtaining a Master’s Degree in Computer Engineering from Institut Teknologi Bandung in 2017. Since 2018, he has been imparting knowledge as a lecturer in the Department of Electrical Engineering at Politeknik Negeri Bandung. His research interest cover the ever-evolving internet of things (IoT), intricate computer networks, and the groundbreaking field of computer vision. He can be contacted at email: griffani.megiyanto@polban.ac.id.

**Muhammad Yusuf Fadhlan** completed his four-year Telecommunication Engineering diploma at Politeknik Negeri Bandung, Indonesia, in 2014. He pursued a master’s degree in Electrical Engineering and Computer Science from National Taipei University of Technology, Taiwan, graduating in 2018. Since then, he has been dedicated to his role as a lecturer in the Department of Electrical Engineering at Politeknik Negeri Bandung. Presently, he leads the Computer Vision Laboratory (CVL) research group and holds the position of head of the Digital Signal Processing Laboratory. His research focuses on artificial intelligence and computer vision, with a particular emphasis on applying these technologies to sports analytics. He can be contacted at email: muhammad.yusuf.ttel410@polban.ac.id.

**Dinan Mutamaddin** received his B.A.Sc (2023) in Telecommunication Engineering at Politeknik Negeri Bandung, Indonesia. He previously undertook a computer vision project at the Internet of Things (IoT) open laboratory (X-CAMP) affiliated with XL Axiata Corporation, in Jakarta, Indonesia. His final project revolves around the integration of deep learning techniques with computer vision for the application of intelligent transportation system. He can be contacted at email: dinan.mutamaddin.tkom419@polban.ac.id.