IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 2 August 2023, accepted 10 September 2023, date of publication 15 September 2023,
date of current version 28 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3315738

APPLIED RESEARCH

Transformer-Based Parking Slot Detection Using Fixed Anchor Points

QUANG HUY BUI AND JAE KYU SUHR, (Member, IEEE)
School of Intelligent Mechatronics Engineering, Sejong University, Seoul 05006, South Korea

Corresponding author: Jae Kyu Suhr (jksuhr@sejong.ac.kr)

This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korean Government (MSIT) under Grant 2022R1F1A1074708, and in part by the Basic Science Research Program through NRF funded by the Ministry of Education under Grant 2020R1A6A1A03038540.

ABSTRACT Transformer-based architectures have recently gained significant attention in various computer vision tasks. Their ability to capture non-local dependencies and intricate characteristics makes them a promising complement to CNNs. However, their application in parking slot detection tasks is still limited. Thus, this paper proposes an appropriate way to apply transformer-based architectures to parking slot detection tasks. The proposed method adopts the Detection Transformer (DETR) architecture, which employs a standard transformer encoder-decoder framework. Since this approach requires a long training time, this paper suggests utilizing fixed anchor points to replace object queries in the original DETR architecture. Each anchor point is assigned a known location and focuses only on a predefined area of the feature map, resulting in a considerable reduction in training time. In addition, this paper suggests using a more suitable and efficient two-point parking slot representation to improve detection performance. In experiments, the proposed method was evaluated with the public large-scale SNU dataset and showed comparable detection performance to the state-of-the-art CNN-based methods with 96.11% recall and 96.61% precision.

INDEX TERMS Automatic parking system, parking slot detection, deep learning, transformers, convolutional neural network (CNN), around view monitor (AVM).

I. INTRODUCTION

In the last decade, autonomous driving has emerged as a major area of interest for both the automotive industry and academic fields. As an important part of this trend, parking assistance and automatic parking systems have also gained widespread popularity and have been the subject of extensive research. Automatic parking systems typically encompass target parking position designation, path planning, path tracking, and user interface [1], [2]. Among these aspects, the precise and effective detection of available parking spaces stands as a key challenge. Various works in this domain have explored four primary approaches: intelligent parking management systems (IPMS)-based, user interface-based, free space-based, and parking slot marking-based [1]. Among them, the parking slot marking-based approach, which detects parking spaces by recognizing markings on the road surface, has witnessed significant growth in recent years. This tendency is attributed to the increased availability of in-vehicle camera systems, particularly around-view monitor (AVM) systems. An AVM system stitches multiple images captured by cameras mounted around the vehicle to generate bird-eye view images, providing drivers with a comprehensive view of their surroundings during the parking maneuver. The popularity of AVM systems as parking aids has prompted most car manufacturers to incorporate them into their vehicles [3], [4], [5]. This resulted in a surge of research focused on vision-based parking slot detection using AVM images.

The primary goal of vision-based parking slot detection using AVM images is to leverage the captured visual information in order to find the marking patterns of parking slots in various traffic scenarios. Traditionally, vision-based parking slot detection is based on manually designed hand-crafted

The associate editor coordinating the review of this manuscript and approving it for publication was Li Zhang.

features and geometric rules. Even though hand-crafted features remain a powerful tool when handling visual information about the parking slot, rigidly designed elements hinder their performance in volatile real-world scenarios. Ever since the advent of deep learning and convolutional neural networks (CNNs), they have dominated the field of general object detection [6]. CNNs have demonstrated their ability to efficiently learn and generalize visual patterns across numerous applications. Consequently, significant effort has been made to utilize CNN-based methods to detect parking slots [1]. CNN-based parking slot detection methods focus on detecting the representative features of a parking slot by exploiting its more robust deep features. Currently, CNN-based methods have achieved state-of-the-art performance in parking slot detection tasks [7], [8].

In recent years, transformers have emerged as a revolutionary neural network architecture, gaining considerable attention, particularly in the field of natural language processing (NLP) [9]. Leveraging the attention mechanism, transformers excel at capturing long-range dependencies, which has propelled them to the forefront of NLP research [10], [11], [12]. Recognizing the immense potential of transformers, researchers have ventured to apply them to computer vision tasks. Encouragingly, recent findings indicate that transformers can be a promising complement to convolutional neural networks (CNNs) in this domain. Unlike CNNs, which primarily rely on local operations and model relationships between neighboring pixels, transformers facilitate global operations, enabling the modeling of relationships among all pixels within a single transformer layer. This distinctive characteristic grants transformers a unique advantage in capturing long-range dependencies and contextual information. Consequently, transformer-based methods have demonstrated state-of-the-art performance across various computer vision tasks [13], [14].

Given the abovementioned advantages in handling visual information, the transformer-based approach presents immense potential for handling parking slot detection tasks. By utilizing transformer attention blocks, the network gains the capability to model non-local dependencies, enabling information aggregation from distant elements of the parking slot. As a result, the transformer-based model gains unprecedented insights into the complex and intricate characteristics of the parking environment, essential for precise and robust parking slot detection.

Despite the remarkable potential, the application of transformer-based architectures in parking slot detection remains relatively underexplored, with only one study having considered this approach [15]. It introduces the order-independent matching with shape similarity (OISS) method, which uses the detection transformer (DETR) architecture [16]. As the first parking slot detection method employing the transformer architecture, the OISS method confronts two significant challenges. First of all, predicting all four corners is inappropriate for parking slot detection because, typically, only two corners of the parking slot are visible in images during the detection stage. A tailored approach that accounts for this constraint is essential to improve accuracy and efficiency. Additionally, utilizing the original DETR architecture, the OISS method faces the drawback of prolonged training time, raising concerns over practical applicability and resource efficiency. In an era where rapid processing and real-time capabilities are essential, an optimized training process that preserves competency without compromising efficiency becomes a main objective in transformer-based research.

To overcome these drawbacks, this paper proposes an appropriate way to apply transformer-based architecture to parking slot detection tasks. For that purpose, this paper first reconsiders the representation of the parking slot using the location and orientation of its two entrance junctions rather than using its four corners. This is because all four corners of the parking slot are rarely visible in images during the detection stage, and only the slot entrance position is necessary for the vehicle to start parking. Even though the suggested parking slot representation performs well when combined with DETR architecture, the method still requires a long training time. The significant drawback of prolonged training time poses a substantial challenge for machine learning tasks, as it can lead to inefficient progress and hinder advancements in research. To solve this, this paper adapts another variation of DETR: Anchor DETR [17], which has not yet been used for parking slot detection tasks. Anchor DETR can shorten the training time by fixing each object query at a predefined location in the feature map called an anchor point. This paper suggests that each anchor point is responsible for an entrance center, and the location of two junctions will be predicted relative to the location of this anchor point. In experiments, the proposed method significantly outperforms the OISS method while requiring a much shorter training time. Moreover, compared to the state-of-the-art parking slot detection method, the proposed method shows comparable performance and has the advantages of faster processing time without the need for non-maximum suppression.

The contribution of this paper can be summarized as follows:

- It presents an appropriate way to apply transformer architecture to parking slot detection tasks that achieves comparable performance to state-of-the-art CNN-based methods.
- It proposes the two-point representation, which is more appropriate to present parking slots.
- It suggests using fixed anchor points to shorten the training time of the DETR approach.

The rest of this paper is organized as follows. Section II provides a comprehensive literature review of previous parking slot detection methods. Section III concisely overviews the detection transformer (DETR) architecture. Section IV introduces the proposed method. Section V presents the

experimental results and a comparative analysis with existing methods. Finally, Section VI concludes the paper, summarizing the findings and discussing potential future research.

II. RELATED WORKS

Since this paper is related to deep learning-based parking slot detection methods, this section does not include traditional hand-crafted feature-based methods. A literature review of the traditional methods can be found in [1].

In recent years, vision-based parking slot detection using deep learning has gained noticeable attention. Most of the current works focus on utilizing CNN-based architecture to find parking slots from input images. CNN-based parking slot detection methods can be categorized into multi-stage and one-stage approaches. The first multi-stage parking slot detection method was proposed in [18]. In the first stage, this method separately detects the locations of the two junctions of the parking slot entrance. In the second stage, the junctions are combined using their types and geometric rules. Finally, in the last stage, image patches containing the parking slot entrance defined by the two junctions are cropped out for further verification. Similarly, the methods in [19] and [20] also try to detect two entrance junctions in the first stage. However, by introducing more detailed and informative junction representations with corresponding combining rules, they avoid using the additional verification stage as in the method in [18]. The method in [21] detects all four corners of a parking slot and combines them in the second stage using geometric rules. This paper also proposes the auxiliary junction in case junctions are occluded. Instead of detecting detached junctions of a parking slot, the method in [22] proposes to directly locate the parking slot entrance using upright bounding boxes. This method then crops the bounding box detection result from the input image and forwards it to a separate network for occupancy classification. In another way, the methods in [23], [24], and [25] adopt the semantic segmentation approach to generate semantic masks for junctions and lines in the first stage. Precise positions of those parking slot elements are extracted from the masks and combined using geometric rules in the second stage. The aforementioned methods have successfully applied CNN-based architecture to handle the parking slot detection tasks. However, the reliance on geometric rules in those methods has remarkably restrained their performance and training process.

To handle this problem, end-to-end trainable parking slot detection methods were proposed. The method in [26] assumes that all parking slots appearing in one image have the same type and orientation. With this assumption, the first stage of this method predicts a common type and orientation for each image and uses that information as clues to generate rotated anchor boxes for location estimation in the second stage. Even though exhibiting good performance, this method suffers from high computational costs by using two separate networks. The method in [27] was the first to apply a two-stage general object detector to parking slot detection tasks. This method directly predicts four junctions of the parking slot in the first stage as region proposals. The second stage extracts feature from the region proposal for location refinement and occupancy classification. However, this method does not perform satisfactorily due to the lack of appropriate adjustments of the general object detector for parking slot detection tasks. The current state-of-the-art multi-stage parking slot detection method was proposed in [8]. This method has solved the problems of the method in [27] by efficiently modifying the two-stage general object detector. Its first stage also generates parking slot proposals. However, instead of considering the whole parking slot, this method generates region-specific proposals for different parking slot elements. In the second stage, features from the region-specific proposals are used for location, orientation refinement, and type and occupancy classification.

On the other hand, one-stage parking slot detection methods, in a similar manner to one-stage general object detectors, exclude the region proposal generation step and directly predict all information about the parking slot using a single network. The method in [28] detects the parking slot entrance by characterizing it using its center, length, and orientation. Because of the adequate description, this method achieves decent performance and a fast processing speed. The method in [29] is an improvement of the method in [28]. Also using the slot entrance characterization, this method adds the ability to classify occupancy by finding the midline of the parking slot. In addition, a self-calibrated neural network (SCNN) is applied for better detection performance. Instead of focusing on just the slot entrance, from every location inside a parking slot, the method in [30] directly predicts four vectors pointing to four corners of the parking slot at the same time. For a better junctions regression result, this method proposes using a centerness value to give higher scores to predictions coming from the center area of the parking slot. The current state-of-the-art one-stage parking slot detection method was proposed in [7]. This method proposes the use of the junction pair to represent the parking slot entrance. It achieves impressive performance by combining the entrance predictions with precise junction predictions.

With the significant development of deep learning in general object detection, besides the CNN-based approach, the transformer-based approach has gained more and more attention [13], [14]. Relying on the attention mechanism, transformer-based architecture shows a strong ability to model non-local dependencies of visual information. For vision-based parking slot detection using deep learning, the transformer-based approach is a potential research area. Based on our thorough literature review, there is only one method to apply transformer-based architecture [15]. This method utilized the DETR architecture [16] to directly predict a non-ordered set of four corners of the parking slot. In addition, it proposes an order-independent matching strategy for a more flexible detector that can handle more general slot shapes and positions. However, the performance of this

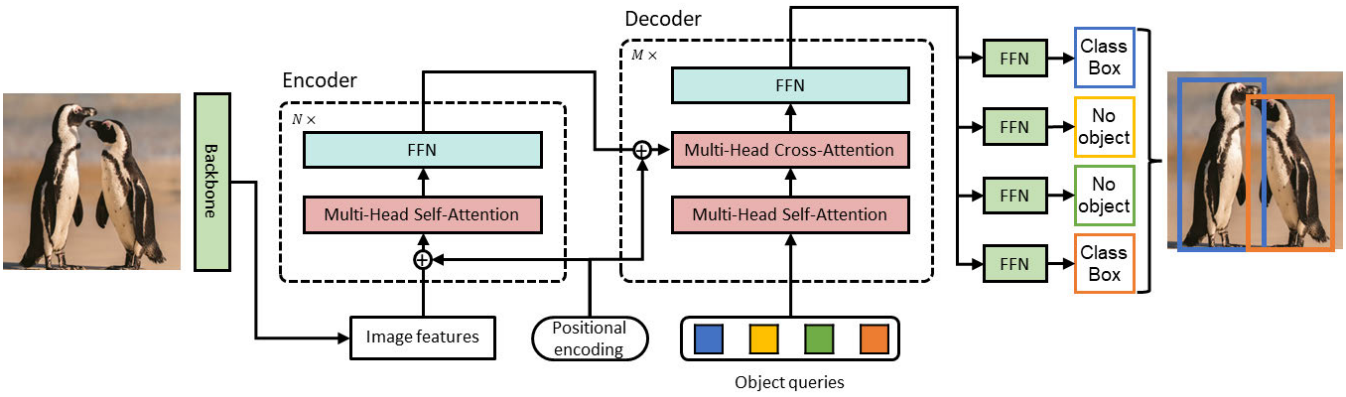


FIGURE 1. DETR architecture.

method is not satisfactory because the choice of parking slot representation is inappropriate. In most cases, only the parking slot entrance with two corners is visible in images. The locations of the other two corners are usually ambiguous, so they are not suitable for training. Moreover, because this method adopts the original architecture of DETR, it also inherits its disadvantage of long training time.

Considering the potential of the transformer-based approach, this paper proposes an appropriate way to apply the transformer-based approach to parking slot detection tasks by solving the disadvantages of the previous method suggested in [15].

III. REVISITING DETECTION TRANSFORMER (DETR)

DETR is a transformer-based architecture specialized for computer vision tasks [16]. DETR follows the encoder-decoder paradigm with consecutive stacks of encoder and decoder layers. The architecture of DETR is illustrated in Fig. 1.

Backbone – DETR employs a conventional CNN-based backbone to extract a feature map from the input image. As the transformer architecture is designed to handle sequential input, DETR flattens the obtained feature map from size $H \times W \times C$ to size $HW \times C$. The flattened feature map can be considered a sequence of HW embedded tokens, each has dimension C. To preserve the spatial relation of visual information, a positional encoding is subsequently added to the token sequence before going to the transformer encoder.

Transformer encoder – Each encoder layer in DETR consists of two sub-layers: the multi-head self-attention and the fully connected feed-forward network (FFN). The multi-head self-attention enhances each token by incorporating contextual information from the entire image. The self-attention calculates a score showing how much attention a particular token should pay to other tokens, while the multi-head mechanism allows multiple parallel attention operations to specialize in attending to different aspects of the input. The FFN, on the other hand, provides an additional non-linear transformation, enhancing the ability to capture complex feature representations and higher-order interactions among input elements. By utilizing these two sub-layers, the encoder allows the integration of both local and global contextual cues, thus strengthening the feature representation ability of the model.

Transformer decoder – DETR introduces an important modification to the conventional transformer decoder in order to adapt it for object detection tasks. The decoder in DETR takes as inputs the embedded tokens from the encoder and N object queries represented by learnable positional embeddings. These object queries serve as indicators that guide the network to attend to specific regions within the image. Each decoder layer in DETR consists of three sub-layers: multi-head self-attention, multi-head cross-attention, and FFN. The multi-head self-attention is similar to the one in the encoder, except for the input, which is now the object queries. Conversely, the multi-head cross-attention calculates a score showing how much attention a particular object query should pay to each position of the token sequence. This design enables the decoder in DETR to effectively incorporate the contextual information from both the embedded tokens and the object queries to generate accurate object detection results.

Prediction feed-forward network (FFN) – The decoder in DETR generates N output embeddings, which are independently passed through a feed-forward network for object detection. This results in N final predictions, where each prediction includes the predicted bounding box coordinates and corresponding object class. This approach allows DETR to avoid the need for post-processing techniques such as non-maximum suppression (NMS). DETR treats object detection as a set prediction problem rather than a per-region classification problem and utilizes the bipartite matching algorithm to assign predictions to ground truth objects, ensuring that each predicted box corresponds to a specific object instance. In addition, a special class of "no object" is also predicted by the feed-forward network, allowing the possibility of discarding the bounding boxes that do not contain any objects among the N predictions.

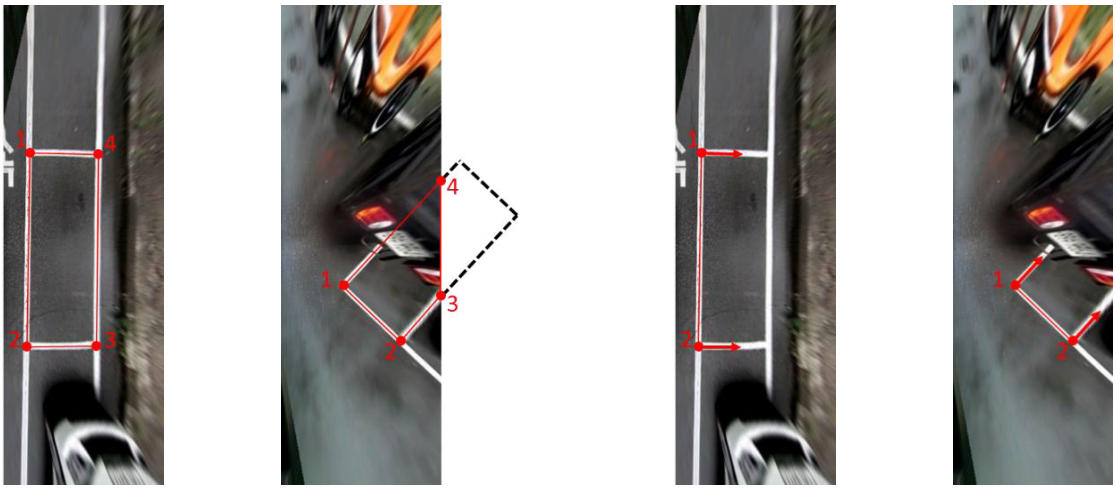


FIGURE 2. (a) Four-point representation when all points are visible, (b) Four-point representation when two points are visible, (c) Two-point representation when all points are visible, (d) Two-point representation when two points are visible.

IV. PROPOSED METHOD

A. DETECTION PERFORMANCE ENHANCEMENT WITH TWO-POINT REPRESENTATION

In order to effectively apply transformer-based architecture to detect parking slots, this paper first selects an appropriate parking slot representation. The OISS method [15] proposes to use four points to represent parking slots, as shown in Fig. 2(a). However, this representation is unsuitable since the whole parking slot is usually not fully captured in the AVM images, as shown in Fig. 2(b). In this case, the four-point representation will generate quadrilateral annotation with different shapes, which affects the detection performance. For this reason, this paper considers using two points and their directions at the parking slot entrance to depict the parking slot as shown in Figs. 2(c)-(d). In this paper, these two points are called junctions. This is more suitable than the four-point representation because the slot entrance is almost always visible in AVM images and information about the slot entrance alone is enough for the vehicle to start parking. Therefore, in this paper, a parking slot is denoted as $\{j_1, j_2, t, o\}$ where $j_i = (x_i, y_i, \theta_{xi}, \theta_{yi})$ is the location and orientation of the i-th junction, t is the type, and o is the occupancy of the parking slot. Compared with the four-point representation in the OISS method, this two-point representation shows remarkable performance enhancement in terms of detection and positioning accuracies. Detailed results will be presented in the experimental section.

B. TRAINING TIME REDUCTION WITH FIXED ANCHOR POINTS

The OISS method directly applies the original DETR architecture, which utilizes learnable object queries as input for the transformer decoder. As the object queries are processed through the decoder layers, they gradually acquire the topological information of parking slots, which possibly exist at any location within the input image. This positional ambiguity makes it difficult to train the network. The object queries require a significant amount of time to be optimized because they have to oversee a large area that varies depending on the input image. Consequently, the training process is excessively prolonged. To address this problem, the proposed method draws inspiration from Anchor DETR and fixes each object query at a predefined location, called an anchor point. This approach is similar to CNN-based detectors, where each position on the feature map serves as a rigid anchor point and only predicts the objects in its proximity. With its location known, each object query processes features in a much smaller area, resulting in a significant reduction in training time. In this paper, the anchor points are selected as uniform grid points in the image, as shown in Fig. 3. As opposed to general object detection, parking slots can never overlap; thus, each anchor point will represent at most one parking slot. The locations of

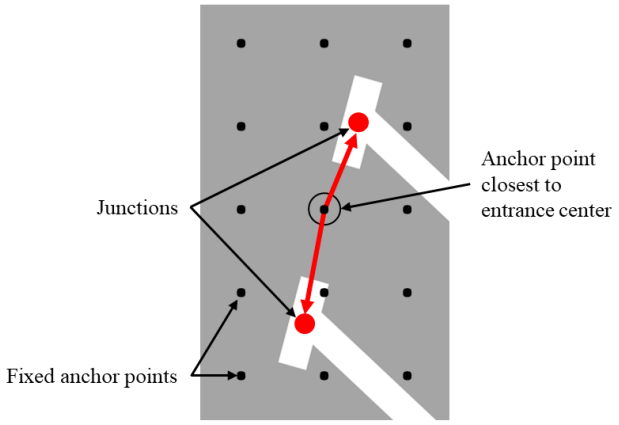


FIGURE 3. Anchor points selection.

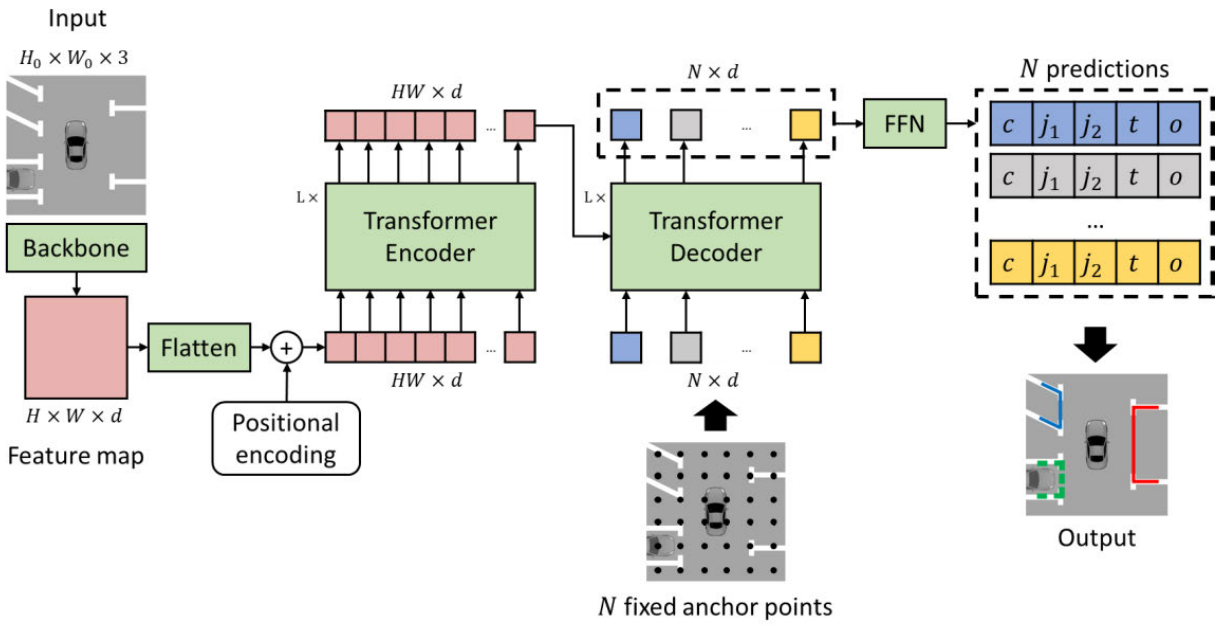


FIGURE 4. Architecture of the proposed method.

the two junctions are calculated relative to the location of the anchor point closest to the entrance center of the parking slot.

C. ARCHITECTURE OF THE PROPOSED METHOD

This paper proposes a novel method for detecting parking slots using a transformer-based architecture. The architecture of the proposed method is illustrated in Fig. 4. An AVM image with shape $H_0 \times W_0 \times 3$ is inputted into a CNN-based backbone network for feature extraction, resulting in a feature map with dimension $H \times W \times C$, where typically, $H = \frac{H_0}{32}$, $W = \frac{W_0}{32}$, and $C = 1024$. Subsequently, the obtained feature map is projected to a smaller channel dimension d using a 1×1 convolutional layer.

The feature map with shape $H \times W \times d$ is then forwarded to the encoder. This paper utilizes the standard transformer encoder with L encoder layers (as described in Section III). Since the transformer encoder layers processes sequential input, the feature map is flattened into a shape of $HW \times d$. In order to maintain the spatial relation of the visual information, a sinusoidal positional embedding is added to the feature map before it is passed to the encoder layers. The output of the encoder is a sequence of contextualized tokens with a shape of $HW \times d$.

The standard transformer decoder with L decoder layers (as described in Section III) then receives as input the output tokens of the encoder and a set of embeddings of size $N \times d$. Different from the DETR architecture, the N embedding now corresponds to N anchor points, each with a fixed location in the feature map. The decoder produces N output embeddings, which are then processed independently by a feed-forward network (FFN), resulting in N parking slot predictions. In this paper, a parking slot is represented by two junctions of its entrance. Therefore, each prediction includes the location and orientation of the two junctions (j_1, j_2), type (t), and occupancy (o) of the parking slot. The utilization of a self-attention sublayer in decoder layers and the bipartite matching algorithm ensures that the N predictions are distinct and non-overlapped. As a result, the need for post-processing steps such as non-maximum suppression (NMS) is eliminated. Since N is noticeably larger than the possible number of parking slots inside one image, a confidence value c is also predicted to filter out incorrect predictions. The results obtained from the proposed method are shown in the output of Fig. 4, with line color and type indicating parking slot type and occupancy, respectively.

D. LOSSES

In the original DETR architecture, to calculate the loss values, the information about which prediction corresponds to which ground truth has to be defined. The bipartite matching between the prediction and ground truth sets is typically calculated using the Hungarian algorithm. However, in this paper, because the location of each anchor point is fixed, the loss computing step can be simplified by comparing the predicted values at each anchor point with the ground-truth label at that exact position. The loss value can be calculated as

$$\begin{aligned} loss = {} & w_{conf} loss_{conf} + w_{loc} loss_{loc} + w_{ori} loss_{ori} \\ & + w_{type} loss_{type} + w_{occ} loss_{occ} \end{aligned} \tag{1}$$

where w_{conf}, w_{loc}, w_{ori}, w_{type}, and w_{occ} are the weights for the five losses and are experimentally set.

The loss for the confidence that an anchor point is close to an entrance center, $loss_{conf}$ is calculated as

$$loss_{conf} = \sum_{i=1}^{N} \left[\mathbb{I}_i \left(\hat{c}_i - c_i \right)^2 + \lambda_e \left(1 - \mathbb{I}_i \right) \left(\hat{c}_i - c_i \right)^2 \right] \quad (2)$$

where N is the number of anchor points. c_i is the ground truth for the confidence that the i-th anchor point is close to any parking slot entrance center. $\hat{c}_i$ is the prediction of the network for c_i. $\mathbb{I}_i$ indicates whether the i-th anchor point is close to any entrance center and is set to 1 if it is or 0 if it is not. Because the number of anchor points is much larger than the number of parking slots in an image, λ_e is multiplied to compensate for this imbalance.

The loss for the location of the two entrance junctions, $loss_{loc}$ is calculated as

$$loss_{loc} = \sum_{j=1}^{2} \sum_{i=1}^{N} \mathbb{I}_i \left[\left(\hat{x}_{i,j} - \frac{x_{i,j}}{L_{max}/2} \right)^2 + \left(\hat{y}_{i,j} - \frac{y_{i,j}}{L_{max}/2} \right)^2 \right] \quad (3)$$

where $\left(x_{i,j}, y_{i,j} \right)$ is the ground truth for the relative location from the i-th anchor point to the corresponding j-th junction. These values are divided by $L_{max}/2$ to be normalized to the range of $[-1, 1]$. L_{max} is the maximum length of the parking slot entrance. In the proposed method, the junction locations are calculated relative to the corresponding anchor point. This has led to a more precise location prediction compared to the OISS method, whose junction locations are calculated relative to the size of the entire image. The detailed results will be shown in the experiments section. $\left(\hat{x}_{i,j}, \hat{y}_{i,j} \right)$ is the prediction of the network for $\left(x_{i,j}, y_{i,j} \right)$ and has the same value range of $[-1, 1]$ because of the tanh activation function.

The loss for the orientation of the two junctions, $loss_{ori}$ is calculated as

$$loss_{ori} = \sum_{j=1}^{2} \sum_{i=1}^{N} \mathbb{I}_i \left[\left(\hat{\theta}_{x,i,j} - \theta_{x,i,j} \right)^2 + \left(\hat{\theta}_{y,i,j} - \theta_{y,i,j} \right)^2 \right] \quad (4)$$

where $(\theta_{x,i,j}, \theta_{y,i,j})$ is a unit vector representing the ground truth for the orientation of the j-th entrance junction, whose entrance center is close to the i-th anchor point. $(\hat{\theta}_{x,i,j}, \hat{\theta}_{y,i,j})$ is the prediction of the network for $(\theta_{x,i,j}, \theta_{y,i,j})$ and has values in the range of $[-1, 1]$ because of the tanh activation function.

The loss for the parking slot type, $loss_{type}$ is calculated as

$$loss_{type} = \sum_{i=1}^{N} \mathbb{I}_i \left[- \sum_{c=1}^{3} \left\{ \lambda_{t,c} t_{i,c} \log \left(\hat{t}_{i,c} \right) \right\} \right] \quad (5)$$

where $t_{i,c}$ is the ground truth for the probability that the type of parking slot whose entrance center is close to the i-th anchor point is c. $t_{i,c}$ is represented in one-hot encoding as (1,0,0), (0,1,0), and (0,0,1) for perpendicular, parallel, and slanted type, respectively. $\hat{t}_{i,c}$ is the prediction of the network for $t_{i,c}$. $\lambda_{t,c}$ is the parameter that compensates for the imbalance of the numbers of different parking slot types in the training dataset.

TABLE 1. Summary of the SNU dataset.

		SNU dataset
Parking situations		571
Image resolution (pixels)		768×256
Corresponding area (m)		14.4×4.8
No. of images	Training	18299
	Test	4518
	Total	22817
No. of slots in train set	Perpendicular	39743
	Parallel	5867
	Slanted	3276
	Total	48886
No. of slots in test set	Perpendicular	888
	Parallel	11653
	Slanted	1004
	Total	13545

The loss for the occupancy of the parking slot, $loss_{occ}$ is calculated as

$$loss_{occ} = \sum_{i=1}^{N} \left[\mathbb{I}_{i,occ} \left(\hat{o}_i - o_i \right)^2 + \lambda_{vac} \mathbb{I}_{i,vac} \left(\hat{o}_i - o_i \right)^2 \right] \quad (6)$$

where o_i is the ground truth for the occupancy of the parking slot whose entrance center is close to the i-th anchor point. $\hat{o}_i$ is the prediction of the network for o_i. $\mathbb{I}_{i,occ}$ indicates whether the i-th anchor point is close to the entrance center of an occupied parking slot and is set to 1 if it is or 0 if it is not. $\mathbb{I}_{i,vac}$ indicates whether the i-th anchor point is close to the entrance center of a vacant parking slot and is set to 1 if it is or 0 if it is not. λ_{vac} is the parameter that compensates for the imbalance of the numbers of occupied and vacant parking slots in the training dataset.

V. EXPERIMENTS

A. DATASET AND EVALUATION METRICS

The experiments in this study were conducted using the large-scale public SNU dataset [26] with the new set of labels provided in our previous work. A detailed description of the dataset is provided in Table 1. The SNU dataset consists of 22871 half AVM images, with 18299 images used for training and 4518 images used for testing. The original resolution of the images is 768 × 256 pixels, which corresponds to a real area of 14.4 × 4.8 m to the left or right side of the vehicle. This dataset is collected from various parking scenarios, including indoor and outdoor, daytime and nighttime, etc., with different illumination conditions. Annotations are available for three types of parking slots appearing in the dataset: parallel, perpendicular, and slanted. For each parking slot, the annotation includes the coordinates of its four corners in counterclockwise order.

The performance of the proposed method was evaluated using the criteria provided in [18], which is widely used by most parking slot detection applications. According to the

criteria, a parking slot is considered a true positive if the location and orientation predictions of its two junctions are within M pixels and N degrees from their ground truths. Otherwise, it is considered a false positive. In this paper, M and N are set to 12 pixels and 10 degrees for loose criteria and 6 pixels and 5 degrees for tight criteria. Recall and precision are calculated as

$$Recall = \frac{\#True\ Positive}{\#Ground\ Truth} \tag{7}$$

$$Precision = \frac{\#True\ Positive}{\#True\ Positive + \#False\ Positive} \tag{8}$$

B. EXPERIMENTAL SETTING

The input images from the SNU dataset were resized to 576×192 pixels before being fed into the backbone network. The anchor points are selected as a uniform grid of points in the feature map with a shape of 18×6 (different anchor point configurations are tested but did not improve detection performance, experimental results are shown in next section). The proposed method adopts DenseNet121 [31] as the backbone network because of its favorable performance in prior parking slot detection applications [8]. The weights of the backbone network were initialized with pre-trained weights from ImageNet. A batch size of 32 was utilized during the training process, and data augmentation techniques such as random horizontal and vertical flips were applied to augment the training data. The proposed method was optimized using the Adam optimizer whose learning rate, β_1, β_2, and ϵ were set to 10^{-4}, 0.9, 0.999, and 10^{-8}, respectively. All experiments were conducted using TensorFlow and a Nvidia GeForce RTX 3090 GPU.

C. PERFORMANCE EVALUATION

As mentioned above, this paper has considered several configurations and selected anchor points as a uniform grid with a shape of 18×6, totaling 108 points. Table 2 presents the detection performance and inference time of the proposed method with three different anchor point grid sizes: 12×4 (48 points), 18×6 (108 points), and 24×8 (192 points). According to this table, the proposed method shows the best detection performance when using the configuration of 18×6 anchor points. In addition, the options with 18×6 and 12×4 anchor points have similar inference times, while the 24×8 option has a slower inference time. The inference time does not increase proportionally with the number of anchor points due to the processing capability of the utilized GPU. Given the transcendent performance of the 18×6 anchor point configuration, we have employed it to obtain the experimental results of the proposed method throughout the remaining sections of this paper.

Table 3 presents the detection performances of the proposed method and three other parking slot detection methods on the SNU dataset. Among the three reference methods, the OISS method is the first method to apply transformer-based architecture for parking slot detection. The methods in [7] and [8] are currently state-of-the-art methods among two-stage and one-stage parking slot detection methods, respectively. According to Table 3, the proposed method noticeably outperforms the OISS method by approximately 10% recall and 7% precision with the loose criteria. This performance gap mainly comes from the differences in parking slot representation. The two-point approach is proven to be more beneficial to parking slot detection than the four-point approach. In addition, Table 3 also shows that the proposed method is capable of reaching a comparable performance to the state-of-the-art CNN-based methods. Under the loose criteria, the proposed method demonstrates an approximately 2% performance improvement compared to the one-stage CNN method. With the tight criteria, the proposed method still outperforms the one-stage CNN method by roughly 1% for both recall and precision. Moreover, the performance of the proposed method is just slightly lower than the state-of-the-art two-stage CNN method, which has undergone extensive customizations specifically tailored to parking slots. This result proves that the transformer-based architectures can also provide satisfactory performance in parking slot detection tasks. Table 4 presents the detailed positioning errors of the four methods. These errors were calculated from correctly detected parking slots only. As seen from the table, the proposed method has much better positioning errors than the OISS method. Compared to the CNN-based methods, even though the location error of the proposed method is not as good, the gap is less than one pixel. The reason for the inferior location accuracy is that, in the proposed method,

TABLE 2. Comparison of detection performance and inference time with different anchor point configurations.

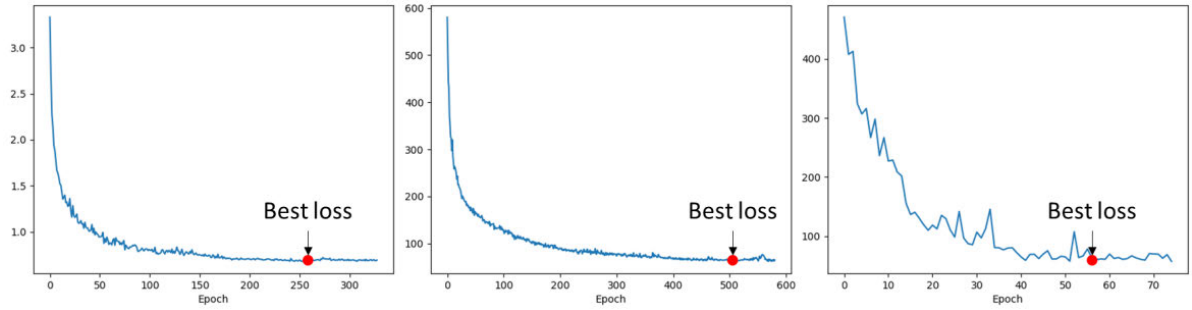
Number of anchor points	Recall	Precision	Inference time (ms)
12×4	95.67%	95.63%	12.00
18×6	96.11%	96.61%	12.72
24×8	94.85%	95.13%	22.35

TABLE 3. Comparison of parking slot detection performances on the SNU dataset.

Method	Loose criteria (12 pixels, 10 degrees)		Tight criteria (6 pixels, 5 degrees)	
	Recall	Precision	Recall	Precision
Proposed method	96.11%	96.61%	90.50%	90.97%
OISS [15]	86.58%	89.09%	57.68%	59.35%
One-stage CNN [7]	94.42%	94.56%	89.24%	89.37%
Two-stage CNN [8]	96.65%	96.96%	92.49%	92.78%

TABLE 4. Comparison of parking slot positioning errors on the SNU dataset.

Method	Location error (pixel)		Orientation error (degree)	
	Mean	Std	Mean	Std
Proposed method	1.58	1.09	1.14	1.04
OISS [15]	2.57	1.85	1.85	1.80
One-stage CNN [7]	1.18	0.87	1.41	1.27
Two-stage CNN [8]	1.05	0.79	1.26	1.24



(a) Case I (OISS) (b) Case II (DETR) (c) Case III (Proposed method)

FIGURE 5. Validation losses of the three transformer-based variations during the training process.

TABLE 5. Comparison of type and occupancy classification performances on the SNU dataset.

Method	Type classification rate	Occupancy classification rate
Proposed method	98.60%	99.02%
OISS [15]	N/A	N/A
One-stage CNN [7]	99.85%	98.34%
Two-stage CNN [8]	99.89%	98.92%

TABLE 6. Comparison of inference time on the SNU dataset.

Method	Time (ms)	Frame per second
Proposed method	12.72	78
OISS [15]	12.48	80
One-stage CNN [7]	9.84	102
Two-stage CNN [8]	31.51	32

TABLE 7. Performance comparison of different transformer-based variations.

Case	Method			Detection performance (Loose criteria)		Training time (epochs)
				Recall	Precision	
I	OISS	Two-point repr.		86.58%	89.09%	253
		Fixed anchor				
II	DETR	Two-point repr.	✓	95.82%	95.06%	508
		Fixed anchor				
III	Proposed method	Two-point repr.	✓	96.11%	96.61%	57
		Fixed anchor	✓			

✓ indicates included

junction locations are predicted from the area of the entrance center, while in CNN-based methods, they are directly predicted from the surrounding area of the junctions, which contains more information.

Table 5 presents the type and occupancy classification rates of the four methods. Classification rates are calculated from the correctly detected parking slots only. The row for the OISS method is marked as N/A because this method does not predict type and occupancy information. This table shows that all methods have good classification rates over 98%. Table 6 presents the inference time of the four methods using Nvidia GeForce RTX 3090. According to the table, the proposed method shows faster processing times than the state-of-the-art two-stage method while maintaining similar performance.

The main contribution of this paper is improving parking slot detectors by using a transformer-based architecture with fixed anchor points as object queries and an appropriate two-point parking slot representation. Table 7 presents the effectiveness of each modification by comparing the performance of different transformer-based variations. Case I is the OISS method, which uses the four-point parking slot representation and DETR-based architecture. Case II uses the two-point representation and the DETR-based architecture. Case III is the proposed method, which uses the fixed anchor points to replace object queries and two-point representation. According to Table 7, case II significantly outperforms case I by roughly 9% recall and 6% precision thanks to the appropriate two-point representation for the parking slot. Table 7 also presents the training time of each method. This training time shows the epoch where the model reaches the lowest validation loss. As mentioned in Section IV, the use of DETR architecture leads to a lengthy training time. Case II requires over 500 epochs to reach the optimal performance. By applying the fixed anchor points, Case III has remarkably shortened the training time with ten times fewer training epochs. This comparison clearly indicates that two-point parking slot representation can adequately improve the detection performance while the fixed anchor points can effectively shorten the training time. Figs. 5(a), (b), and (c) show the validation losses during the training processes of the OISS, DETR, and proposed methods, respectively. This figure illustrates the epoch where the model produces the lowest validation loss with a red dot.

Fig. 6 presents several parking slot detection results of the proposed method. The green, red, and blue lines indicate perpendicular, parallel, and slanted parking slots, respectively; the solid and dashed lines indicate vacant and occupied parking slots, respectively. It can be seen from these results that the proposed method can correctly detect the vacant and occupied parking slots of three types: perpendicular, parallel, and slanted. In addition, the proposed method can perform well in various parking environments with various illumination conditions.

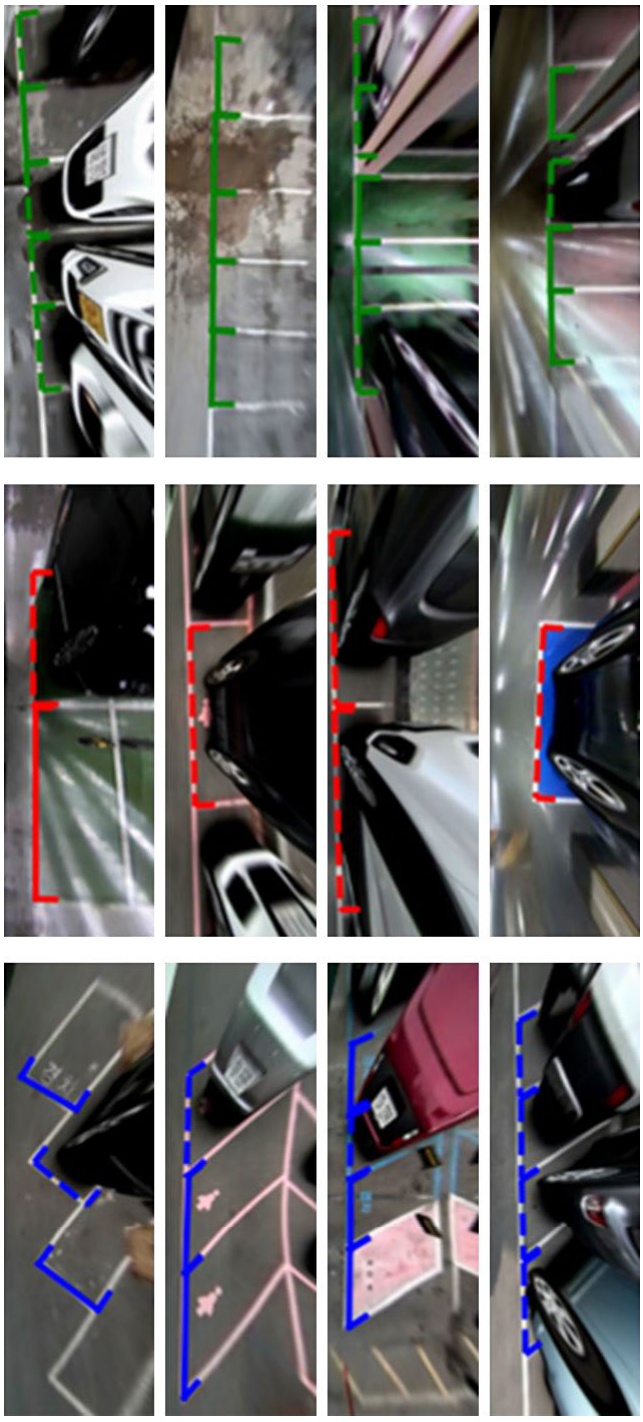


FIGURE 6. Parking slot detection results of the proposed method in various parking scenarios in the test images of the SNU dataset. The first, second, and third rows show the detection results for perpendicular, parallel, and slanted parking slots, respectively. Green, red, and blue lines indicate perpendicular, parallel, and slanted parking slots, respectively; solid and dashed lines indicate vacant and occupied parking slots, respectively.

Fig. 7 illustrates failure cases of the proposed method. In Fig. 7(a), the lower junction of the detected parking slot does not satisfy the location criterion due to another marking on the ground. In Fig. 7(b), the network mistakenly detects

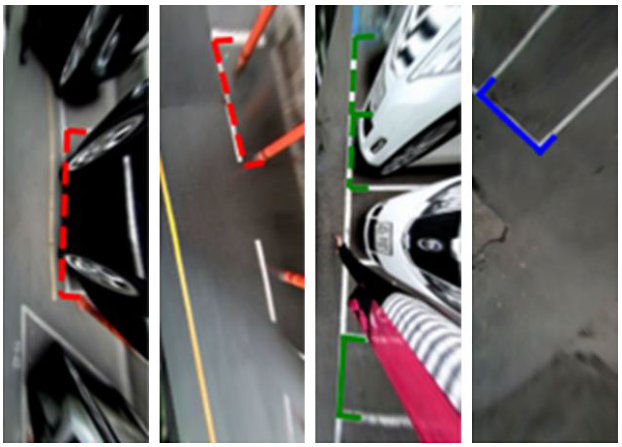


FIGURE 7. Failure cases of the proposed method in the test images of the SNU dataset. (a) and (b) show false positive cases, (c) shows a false negative case and (d) shows an incorrect type classification.

a lane marking and some poles as an occupied parking slot. In Fig. 7(c), a pedestrian occludes a junction which makes the network fail to detect two parking slots. In Fig. 7(d), a perpendicular parking slot is misclassified into a slanted parking slot due to the extreme orientation.

VI. CONCLUSION

This paper presents an appropriate way to apply transformer-based architectures to parking slot detection tasks. The proposed method adopts the standard transformer encoder-decoder architecture and utilizes fixed anchor points as a replacement for object queries. By combining this architecture with the appropriate two-point parking slot representation, the proposed method achieves not only shorter training time compared to the original transformer-based approach but also comparable detection performance to state-of-the-art CNN-based parking slot detection methods. These results highlight the potential of transformers to effectively address parking slot detection tasks. The mainstream approach in the current automotive industry relies on CNNs, which benefit from the widespread support from neural processing units (NPUs). However, with further specialized modifications, transformers have the potential to replace CNNs and become the preferred option for automotive applications. Our research represents a foundational step in preparing for the practical implementation of transformers in real-world automotive scenarios. Our future research will focus on investigating the viability of diverse transformer-based architectures and incorporating extensive customizations specifically tailored to the unique characteristics of parking slots in order to foster advancements in the domain of parking slot detection.

REFERENCES

[1] J. K. Suhr and H. G. Jung, "Survey of target parking position designation for automatic parking systems," *Int. J. Automot. Technol.*, vol. 24, no. 1, pp. 287–303, Feb. 2023, doi: 10.1007/s12239-023-0025-6.

[2] W. Wang, Y. Song, J. Zhang, and H. Deng, "Automatic parking of vehicles: A review of literatures," *Int. J. Automot. Technol.*, vol. 15, no. 6, pp. 967–978, Oct. 2014, doi: 10.1007/s12239-014-0102-y.
[3] *Intelligent Around View Monitor*. Nissan. Accessed: Jul. 2023. [Online]. Available: https://www.nissan-global.com/EN/INNOVATION/TECHNOLOGY/ARCHIVE/IAVM/
[4] *Multi-Camera System*. Bosch. Accessed: Jul. 2023. [Online]. Available: https://www.bosch-mobility.com/en/solutions/assistance-systems/multi-camera-system/
[5] *Surround View Camera*. Masserati. Accessed: Jul. 2023. [Online]. Available: https://www.maserati.com/global/en/ownership/maserati-manuals/safety/surround-view-camera
[6] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, "Deep learning for generic object detection: A survey," *Int. J. Comput. Vis.*, vol. 128, no. 2, pp. 261–318, Feb. 2020, doi: 10.1007/s11263-019-01247-4.
[7] J. K. Suhr and H. G. Jung, "End-to-end trainable one-stage parking slot detection integrating global and local information," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 5, pp. 4570–4582, May 2022, doi: 10.1109/TITS.2020.3046039.
[8] Q. H. Bui and J. K. Suhr, "CNN-based two-stage parking slot detection using region-specific multi-scale feature extraction," *IEEE Access*, vol. 11, pp. 58491–58505, 2023, doi: 10.1109/ACCESS.2023.3284973.
[9] A. Vaswani, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
[10] N. Patwardhan, S. Marrone, and C. Sansone, "Transformers in the real world: A survey on NLP applications," *Information*, vol. 14, no. 4, p. 242, Apr. 2023, doi: 10.3390/info14040242.
[11] S. Singh and A. Mahmood, "The NLP cookbook: Modern recipes for transformer based deep learning architectures," *IEEE Access*, vol. 9, pp. 68675–68702, 2021, doi: 10.1109/ACCESS.2021.3077350.
[12] F. Yvon, "Transformers in natural language processing," in *Human-Centered Artificial Intelligence: Advanced Lectures*. Cham, Switzerland: Springer, pp. 81–105, doi: 10.1007/978-3-031-24349-3_6.
[13] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu, Z. Yang, Y. Zhang, and D. Tao, "A survey on vision transformer," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 87–110, Jan. 2023, doi: 10.1109/TPAMI.2022.3152247.
[14] J. Bi, Z. Zhu, and Q. Meng, "Transformer in computer vision," in *Proc. IEEE Int. Conf. Comput. Sci., Electron. Inf. Eng. Intell. Control Technol. (CEI)*, Sep. 2021, pp. 178–188, doi: 10.1109/CEI52496.2021.9574462.
[15] Z. Yin, R. Liu, Z. Yuan, and Z. Xiong, "Order-independent matching with shape similarity for parking slot detection," in *Proc. BMVC*, 2021, p. 313
[16] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 213–229, doi: 10.1007/978-3-030-58452-8_13.
[17] Y. Wang, X. Zhang, T. Yang, and J. Sun, "Anchor DETR: Query design for transformer-based detector," in *Proc. AAAI Conf. Artif. Intell.*, vol. 36, no. 3, Jun. 2022, pp. 2567–2575, doi: 10.1609/aaai.v36i3.20158.
[18] L. Zhang, J. Huang, X. Li, and L. Xiong, "Vision-based parking-slot detection: A DCNN-based approach and a large-scale benchmark dataset," *IEEE Trans. Image Process.*, vol. 27, no. 11, pp. 5350–5364, Nov. 2018, doi: 10.1109/TIP.2018.2857407.
[19] J. Huang, L. Zhang, Y. Shen, H. Zhang, S. Zhao, and Y. Yang, "DMPR-PS: A novel approach for parking-slot detection using directional marking-point regression," in *Proc. IEEE Int. Conf. Multimedia Expo. (ICME)*, Jul. 2019, pp. 212–217, doi: 10.1109/ICME.2019.00045.
[20] Z. Wu, W. Sun, M. Wang, X. Wang, L. Ding, and F. Wang, "PSDet: Efficient and universal parking slot detection," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Oct. 2020, pp. 290–297, doi: 10.1109/IV47402.2020.9304776.
[21] Y. Park, J. Ahn, and J. Park, "Deep learning based parking slot detection and tracking: PSDT-Net," in *Proc. Int. Conf. Robot Intell. Technol. Appl.*, 2021, pp. 291–302, doi: 10.1007/978-3-030-97672-9_26.
[22] 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.
[23] Y. Wu, T. Yang, J. Zhao, L. Guan, and W. Jiang, "VH-HFCN based parking slot and lane markings segmentation on panoramic surround view," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 1767–1772, doi: 10.1109/IVS.2018.8500553.
[24] C. Jang and M. Sunwoo, "Semantic segmentation-based parking space detection with standalone around view monitoring system," *Mach. Vis. Appl.*, vol. 30, no. 2, pp. 309–319, Mar. 2019, doi: 10.1007/s00138-018-0986-z.
[25] S. Jiang, H. Jiang, S. Ma, and Z. Jiang, "Detection of parking slots based on mask R-CNN," *Appl. Sci.*, vol. 10, no. 12, p. 4295, Jun. 2020, doi: 10.3390/app10124295.
[26] H. Do and J. Y. Choi, "Context-based parking slot detection with a realistic dataset," *IEEE Access*, vol. 8, pp. 171551–171559, 2020, doi: 10.1109/ACCESS.2020.3024668.
[27] A. Zinelli, L. Musto, and F. Pizzati, "A deep-learning approach for parking slot detection on surround-view images," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2019, pp. 683–688, doi: 10.1109/IVS.2019.8813777.
[28] W. Li, H. Cao, J. Liao, J. Xia, L. Cao, and A. Knoll, "Parking slot detection on around-view images using DCNN," *Frontiers Neurorobot.*, vol. 14, p. 46, Jul. 2020, doi: 10.3389/fnbot.2020.00046.
[29] R. Zheng, S. Lian, W. Liang, Y. Tang, and W. Meng, "Center keypoint for parking slot detection with self-calibrated convolutions network," in *Proc. 17th Int. Conf. Control, Autom., Robot. Vis. (ICARCV)*, Dec. 2022, pp. 305–310, doi: 10.1109/ICARCV57592.2022.10004223.
[30] Y. Wang, Y. Guan, and R. Cao, "DetPS: A fully convolutional end-to-end parking slot detector," in *Proc. IEEE 17th Conf. Ind. Electron. Appl. (ICIEA)*, Dec. 2022, pp. 1051–1056, doi: 10.1109/ICIEA54703.2022.10005941.
[31] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708, doi: 10.1109/CVPR.2017.243.

QUANG HUY BUI received the B.S. degree in mechatronics engineering from the Ho Chi Minh City University of Technology, Ho Chi Minh City, Vietnam, in 2019. He is currently pursuing the Ph.D. degree with the Department of Intelligent Mechatronics Engineering, Sejong University, Seoul, South Korea. His research interests include computer vision and deep learning with a focus on applications for autonomous vehicles.

JAE KYU SUHR (Member, IEEE) received the B.S. degree in electronic engineering from Inha University, Incheon, South Korea, in 2005, and the M.S. and Ph.D. degrees in electrical and electronic engineering from Yonsei University, Seoul, South Korea, in 2007 and 2011, respectively.

From 2011 to 2016, he was with the Automotive Research Center, Hanyang University, Seoul. From 2016 to 2017, he was with the Korea National University of Transportation, Chungju, South Korea. He is currently an Associate Professor with the Department of Intelligent Mechatronics Engineering, Sejong University, Seoul. His research interests include computer vision, image analysis, pattern recognition, and sensor fusion for intelligent and autonomous vehicles.