IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 28 December 2022, accepted 15 January 2023, date of publication 20 January 2023, date of current version 30 January 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3238414

RESEARCH ARTICLE

Depth and Pixel-Distance Based Attention for Outdoor Semantic Segmentation

MYUNG-WOO WOO, (Graduate Student Member, IEEE), AND SEUNG-WOO SEO, (Member, IEEE)
Department of Electrical and Computer Engineering, Seoul National University, Seoul 08826, South Korea
Corresponding author: Seung-Woo Seo (sseo@snu.ac.kr)

ABSTRACT Semantic segmentation has been a crucial technology for various practical applications such as autonomous driving. Recently, attempts have been made to improve the performance of semantic segmentation using depth information. However, most attempts have been focused on the indoor environment for the following reasons. First, it is relatively more difficult to obtain accurate and dense depth information outdoors. Second, a network with a new structure is required to use depth information because processing depth as an input demands an additional encoder. To overcome aforementioned difficulties, we propose a novel **D**epth and **P**ixel-distance based **A**ttention (**DPA**) module, which utilizes depth information to compute the similarity between pixels. The similarity of pixels is computed using the fact that pixels belonging to the same object have similar depth values. Because only the relative difference in depth is considered, it is relatively robust despite the accuracy of the provided depth information. Furthermore, **DPA** is a simple plug-in module that can be applied to existing RGB-based segmentation backbones. Since no encoder is added, it is much more efficient in terms of computation. We conduct extensive experiments on the Cityscapes dataset using various baseline architectures. Regardless of the baseline models, **DPA** yields meaningful performance improvements in semantic segmentation tasks. It is also computationally more efficient compared to the methods that take depth information as input.

INDEX TERMS Attention mechanism, deep learning for visual perception, RGBD semantic segmentation, semantic scene understanding.

I. INTRODUCTION

Semantic segmentation is a task that assigns class labels to all pixels in an image. Significant progress has been made with the development of deep convolutional networks, and many successes [1], [2] have been achieved with an FCN [3] based approach. Semantic segmentation is a fundamental and essential field of computer vision and is used in various fields such as autonomous driving, robotics, and medical image processing. In situations where autonomous driving or robots are operating, depth information is often provided or produced for various purposes, such as 3D reconstruction, localization, and odometry. Depth information is additional geometric information omitted from the image. An image is an information projected in 2D from a 3D real environment, and information loss occurs in that process. The depth information can compensate for this information loss in an image. However, despite these advantages, there have been few studies on semantic segmentation using depth information in an outdoor environment for the following reasons.

1)Limitations of RGBD sensors. The RGBD sensor has a relatively short detection distance of less than 10m, and because the IR sensor is used, interference occurs when it is used outdoors in the sun. Therefore, most RGBD datasets are limited to indoor environments. Recently, stereo cameras and deep learning-based mono cameras have been used to acquire depth information at low cost outdoors. The depth information obtained in this manner is less dense and less accurate than the RGBD sensor, but if used well, it can help semantic segmentation sufficiently. 2)A new network structure is required for using a new modality. Networks such

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

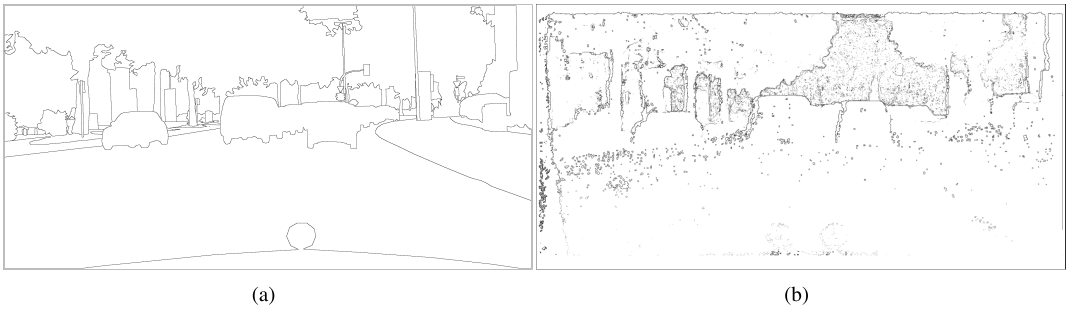


FIGURE 1. **Gradient map (a) Result of applying Laplacian filter on the label. (b) Result of applying Laplacian filter on the depth map. we visualized the large change in the value obtained by passing the Laplacian filter.**

as FuseNet [4], RedNet [5], ACNet [6], and ESANet [7] accept the depth as the input. These networks use parallel encoders for the depth input. This strategy generates better features by fusing the features from the RGB and depth encoders. Although this method is intuitive, it cannot use existing RGB-based networks and requires the design of a new network structure. Designing a new network structure is a challenging task. Also, because the encoder is used in duplicate, the increase in computation is significant.

In this paper, we focus on utilizing outdoor depth information that has not been used well in the past. For this, we propose a "**D**epth and **P**ixel-distance based **A**ttention(**DPA**) module" that can efficiently utilize depth information in an existing RGB-based network. It does not take depth information as input, but as information to find correlations between pixels. This is based on the assumption that pixels belonging to the same label belong to similar depth values. A person can infer the label of an object by looking at the depth map because the approximate contour of the object can be seen from the depth map. The contour of the depth map indicated discontinuous points in the depth value. In other words, the depth value changes rapidly. The part other than the contour changes the depth value smoothly, and the part that changes smoothly is the part within the same object. **Fig. 1** shows the result of extracting the part with a large change in value from the label and depth map through the Laplacian filter. Although there is a lot of noise in the depth map, it can be observed that the contours are similar. This shows that pixels belonging to the same label have similar depth values compared to pixels belonging to other labels. Using this characteristic, the similarity of the depth values can be defined as the similarity between the pixels. However, when the field of view is large outdoors, similar depths do not always have the same labels. Objects with similar depth values may also exist. If the similarity is computed only by depth as shown in **Fig. 2(b)**, the similarity is high for several other objects at the same depth. Objects that are far from the query point but have similar depth values have a high degree of similarity. A constraint is required to solve this problem. Even if they have similar depth values, the points that are too far apart do not belong to the same object. People can recognize objects by looking at the depth map because they observe not only the depth value, but also the position to which the value belongs. In other words, the depth-based similarity is satisfied when a point is within a certain distance. When the pixel-distance is added to the similarity computation, the area corresponding to an object, such as a query point is more concentrated, as shown in **Fig. 2(c)**. In this operation, vertical and horizontal positions are added to each point on the depth map to generate a 3D Euclidean space. The distance in this 3D Euclidean space is defined as the similarity between pixels.

Using this definition, **DPA** performs attention and aggregation on features from the backbone network. When the depth is input to the DPA module, pixel-coordinates are added to form a 3D Euclidean space, and the similarity between pixels is determined based on their adjacency in this 3D space. Based on this similarity, a high attention weight is assigned to the pixel that is the query and pixel with high similarity. As a result, the attention weight has a higher weight as the depth values are similar and the distance is closer. We call this the depth pixel-distance attention weight. Feature aggregation is performed using the depth pixel-distance attention weight. The feature aggregated in this manner implicitly contains the context information of the depth map. Because this method computes the attention weights through a raw depth map, there is no need for a new encoder to generate the depth feature. Depth information can be used for semantic segmentation by adding a **DPA** module to an existing RGB-based network without designing a new network. In addition, since it does not require additional encoder, the increase in computation is not significant and more efficient than using depth as input.

The main contributions of this paper are summarized as follows:

1) We propose a novel Depth and Pixel-distance based Attention(**DPA**) module that determines the similarity between pixels from the depth similarity and pixel position adjacency.
2) The proposed **DPA** can improve the semantic segmentation performance by plugging itself into an existing RGB-based network rather than designing a new network. It is also more efficient in computation than using depth as input.
3) Through extensive experiments, we prove that the correlation between pixels can be inferred through similarity computations based on depth and pixel position. The performance of semantic segmentation is

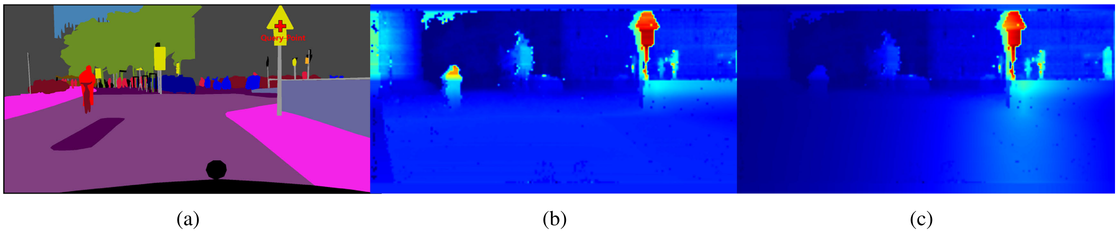


FIGURE 2. Depth similartiy map (a) Ground truth and query point. (b) Similarity map based on only depth values. (c) Similarity map based on depth values with pixel coordinate. In (b), other parts with the same depth also showed high similarity, whereas in (c), the distance between pixels in the image was given as a constraint, and similarity was calculated mainly for the object corresponding to the part to be queried.

enhanced by the **DPA** module regardless of the type of the backbone model.

II. RELATED WORK

A. SEMANTIC SEGMENTATION

Semantic segmentation is a high-level problem that requires understanding the image pixel by pixel, rather than classifying the entire image as one. This is the task of classifying in the semantic unit we defined, and it can be seen as pixel-level classification. Semantic segmentation has also made a lot of progress with the development of Convolutional Neural Networks(CNN) like other computer vision fields. However, unlike image classification, semantic segmentation requires the preservation of the location information of pixels. Due to the nature of CNN, features are extracted while reducing the resolution of the image and increasing the number of channels. Since positional information is greatly lost in this process, it is difficult to use the extracted features directly. Therefore, the process of up-sampling to the same resolution as the input image is required.

As mentioned above, there is an encoder part that extracts features while lowering the resolution(down-sampling) and a decoder part that estimates classes while increasing the resolution(up-sampling) based on the extracted features. This structure is called an encoder-decoder structure, and most adopt this structure today. FCN [3] is the most representative semantic segmentation of the encoder-decoder structure. It is a network that changed the Fully Connected (FC) layer in the classification task to a 1×1 convolution layer. Various semantic segmentation methods [8], [9], [10], [11], [12], [13] as well as U-Net [14], which have been proposed in the field of medical imaging and are being used in various fields, maintain this structure.

Since the structure of most semantic segmentation networks is the encoder-decoder structure, we try to propose a module that can be applied without changing this structure. The **DPA** module we propose is a plug-in module that does not change the encoder-decoder structure. Therefore, it is an efficient module that can be easily applied to semantic segmentation networks having the encoder-decoder structure.

B. RGBD SEMANTIC SEGMENTATION

There have been various attempts to use depth in different ways, but most have used it as an additional input. FuseNet [4], as the name suggests, fuses depth and RGB features. FuseNet [4], through parallel encoders, extracts and fuses depth and RGB to produce better features. RedNet [5], similar to FuseNet [4], fuses features produced by the depth encoder with RGB features. ACNet [6] uses the same basic strategy as the previous methods, except that the module and encoder for fusion are used separately. ESANet [7] also uses a parallel encoder and proposes a real-time structure for a mobile robot.

Another method is to change the CNN operation using depth information. In the case of depth-aware CNN [15], the weights of the CNN and pooling operations are adjusted using pixels of the same label having similar depth values. 2.5D convolution [16], Malleable 2.5D convolution [17] or 3D neighborhood convolution [18] also extend the CNN operation to more than 2D based on the above idea. Previous methods have focused on the indoor environment. In addition, the design of a new network is required to utilize depth information, and existing RGB-based networks cannot be used. The proposed **DPA** uses depth information to infer correlations between pixels and generate features with depth information. This method does not require an additional encoder or network to extract features from the depth information. Therefore, the **DPA** module can improve the semantic segmentation performance of existing RGB-based networks that perform well without major structural changes because no depth information is used as input.

C. CONTEXT AGGREGATION IN SEMANTIC SEGMENTATION

Semantic segmentation is the operation of labeling each pixel; however, to determine one pixel, it must also consider the surrounding pixels. Therefore, aggregating the contextual information of surrounding pixels is important for achieving accurate semantic segmentation. Although FCN [3] has achieved considerable success in semantic segmentation, it is difficult to extract sufficient contextual information from a small receptive field. To solve this problem, attempts have been made to apply it to receptive fields of various sizes or scales. PSPNet [2] extracts contextual information at various scales from different pooling layers using spatial pyramid pooling. DeepLabv2 [1] expands the receptive field while minimizing the increase in computational cost through ASPP, and fuses features from various receptive fields. In the case of DenseASPP [19], information of various scales is fused by adding a dense connection, that is, the idea of DenseNet [20], to the ASPP.

Recently, several attempts have been made to extract contextual information based on attention mechanisms. The non-local neural network [21] generates dense attention weights by computing the correlations between all pixels. It aggregates contextual information by using attention weights. DANet [22] infers not only the correlation between each pixel, but also the correlation between channels. In the case of CCNet [23], criss-cross path attention is performed to reduce the computational cost of non-local neural networks. References [24], [25], and [26] also augmented with features with richer contextual information by inferring the correlation between pixels for the feature map in various ways. In this paper, we also infer the correlation between pixels and perform aggregation of contextual information. Previous methods infer correlation by using the feature from the backbone as a query, but our method infers the correlation of pixels using depth information as a query. Through this process, a feature with only RGB information is transformed into a feature with depth contextual information.

III. METHOD

A. ARCHITECTURE

In this section **Fig. 4**, we explain how to apply the depth attention module to recent semantic segmentation models: BiSeNetV2 [27], STDC [28] and HRNet [29]. The baseline segmentation model is divided into two parts: The encoder that performs feature embedding and a segmentation head that makes predictions based on the embedded features. The **DPA** module is a process of augmenting RGB features to depth contextual features. Therefore, before entering the segmentation head, Depth and Pixel-distance based Attention is performed on the features from the encoder. As demonstrated in **Fig. 3**, we embed the RGB feature to **V**(value) through 1×1 convolutional layer. Depth information is concatenated with positional embedding to create **Q** (depth query), which is used as a query and key to compute the similarity between pixels. The computed similarity is normalized through softmax, converted into attention weights, and applied to **V**. Finally, a residual connection that adds the initial RGB features is applied here. We pass the depth contextual features from the Depth and Pixel-distance based Attention module through the segmentation head. For the segmentation head, 3×3 convolutional layer, batch norm, and ReLU are applied. The loss function applies cross-entropy to the final output. We apply OHEM [30] as we do for the baseline network(BiSeNetV2 [27], STDC [28]).

B. DEPTH AND PIXEL-DISTANCE BASED ATTENTION MODULE

In the Depth and Pixel-distance based Attention(**DPA**) module shown in **Fig. 3**, the similarity between pixels is inferred based on the depth information, and the features are aggregated through this. Two inputs are required in this process: the feature map from RGB based segmentation encoder $\mathbf{x} \in \mathbb{R}^{c_i \times h \times w}$ and the depth map $\mathbf{D} \in \mathbb{R}^{1 \times h \times w}$, where c_i is the channel of the feature map, h is the height and w is the width. The depth map for computing the similarity is downsampled according to the size of the feature map. Here, the positional embedding $\mathbf{P} \in \mathbb{R}^{2 \times h \times w}$ is concatenated such that the position between pixels can be identified. In the positional embedding, each channel consists of the horizontal embedding $\mathbf{P_X} \in \mathbb{R}^{1 \times h \times w}$ and vertical embedding $\mathbf{P_Y} \in \mathbb{R}^{1 \times h \times w}$. The horizontal embedding represents the horizontal position by incrementing by 1 starting from 0 at the leftmost part of the image. The vertical embedding represents the vertical position by incrementing by 1 starting from 0 at the top region of the image. Then, the positional embedding is normalized to [-1,1]. The depth query $\mathbf{Q} \in \mathbb{R}^{3 \times h \times w}$ has a total of three channels: depth, horizontal position, and vertical position maps. The **Q** is a 3D Euclidean space containing the 2D positions and depth on the feature map. Because the similarity is computed through the distance from the generated **Q**, the depth key **K** is the same as **Q**. In this space, the scale of each axis is not well aligned to compute similarity. That is, it did not consider how high weight each dimension would be. The scale parameter is applied to fit each coordinate axis. This scale parameter is scalar multiplied by each axis as a 3D learnable parameter. When the position on an arbitrary depth query is p_i, the 3D vector of the corresponding **Q** is:

$$Q(p_i) = (\alpha_d d, \beta_x x, \gamma_y y) \tag{1}$$

where d is the depth, x is the horizontal coordinate, and y is the vertical coordinate. α, β, and γ are scale parameters multiplied by each axis. This scale parameter is optimized end-to-end during training. The **Q** generated through this process becomes a 3D Euclidian space. Compute the similarity between pixels based on **Q**. The similarity is adjacency in the 3D Euclidean space generated by the **Q**. After computing the similarity between each pixel, normalize it with softmax to make it an attention weight. The attention weight between an arbitrary feature point p_0 and another point p_j is defined as:

$$W_d(p_i, p_j) = softmax(-\left\| Q(p_i) - Q(p_j) \right\|_2) \tag{2}$$

The smaller the distance between **Q**, the higher the attention weight.

Computing similarity is performed only on the area of the criss-cross [23] path. There are two advantages to a criss-cross [23] path. The first is an advantage in the terms of computation because the number of points participating in the similarity computation is reduced. This is a basic advantage of criss-cross [23] networks. Another reason is that because the space occupied by one object in the image is relatively small, it is inefficient to go through all the pixels to find the pixel points belonging to one object. It is impossible to find all correlated pixels through a criss-cross [23] path, however, sufficiently meaningful pixels can be found.

A 1×1 convolutional layer is applied to the feature map **x** that enters the module to generate the feature map

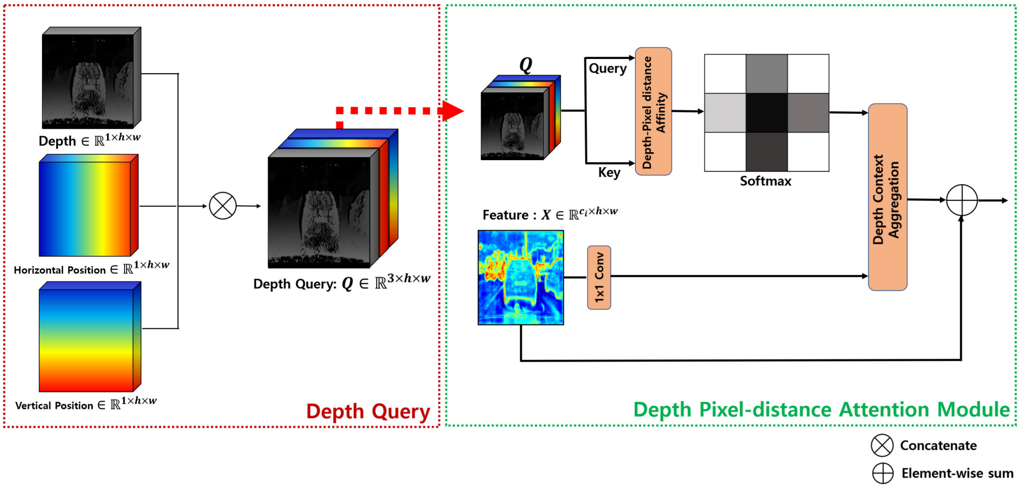

Depth ∈ $\mathbb{R}^{1\times h\times w}$
Horizontal Position ∈ $\mathbb{R}^{1\times h\times w}$
Vertical Position ∈ $\mathbb{R}^{1\times h\times w}$
Depth Query: $Q \in \mathbb{R}^{3\times h\times w}$
Depth Query
Q
Query
Key
Depth-Pixel distance Affinity
Softmax
Feature : $X \in \mathbb{R}^{c_i\times h\times w}$
1x1 Conv
Depth Context Aggregation
Depth Pixel-distance Attention Module
Concatenate
Element-wise sum

FIGURE 3. **Depth and Pixel-distance based Attention module The red box on the left is the process of generating a Q (depth query) which concatenates horizontal and vertical positional embeddings of the depth map. The green box on the right is the process of attention through Q. Using Q as key and query, we compute the similarity between pixels through Euclidean distance computation, and normalize it with softmax to generate attention weights.**

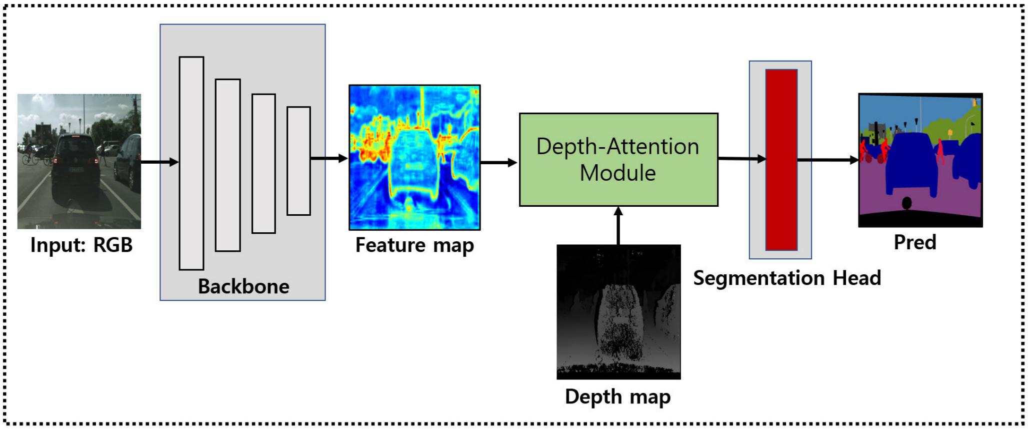


FIGURE 4. **Overall framework for our proposed depth based attention segmentation.**

V(value). We aggregate the feature map **V** by the attention weight W_d computed by the **Q**. The initial feature **x** is summed element-wise on the aggregated features. Through this process, the RGB feature **x** is augmented to the depth contextual feature **y**. The depth contextual feature is defined as:

$$\mathbf{y}_i = \sum\nolimits_{\forall j} W_d(p_i, p_j)\mathbf{V_j} + \mathbf{x_i} \tag{3}$$

In summary, the **DPA** module receives RGB feature **x** and raw depth **D** as inputs. Concatenate the depth map **D** and positional embedding **P** to generate depth query **Q**. The depth attention weight W_d is generated through the depth query **Q**, and the RGB feature **x** is aggregated with weight W_d. Finally, the segmentation prediction result is inferred by passing the depth contextual feature through the segmentation head.

IV. EXPERIMENTS

In this section, we first introduce the experimental setup and show the experimental results on the Cityscapes dataset [31]. We apply the **DPA** module to the three RGB semantic segmentation models. Two are STDC [28] and BiSeNetV2 [27], which are real-time segmentation models, and HRNet [29], which is the more complex and stronger backbone. The **DPA** module has improved the performance of semantic segmentation regardless of the model.

A. EXPERIMENTAL SETUP

1) MODELS

To confirm the effectiveness of **DPA** module, three networks are used as baselines. BiSeNetV2 [27] and STDC [28] are recently proposed as lightweight and high-performance semantic segmentation models. HRNet [29] is a complex and stronger segmentation network with more focus on performance. These models apply the **DPA** module to the feature maps before they are processed through the final output, that is, the segmentation head. We keep the auxiliary loss functions used in the models retained. *e.g.* boundary loss in STDC [28] or boost loss in BiSeNetV2 [27]. STDC [28] and BiSeNetV2 [27] applied OHEM [30] as the baseline, but

TABLE 1. Performance on Cityscapes [31] validation set according to the depth query configuration.

Depth	Horizontal	Vertical	mIoU(%)
-	-	-	74.80 ±0.22
✓			76.06 ±0.02
✓	✓		75.83 ±0.11
✓		✓	75.59 ±0.04
✓	✓	✓	**76.56** ±0.08

TABLE 2. Value of scale parameter according to the depth query configuration.

Depth query	α_d	β_x	γ_y
Depth	40.95	-	-
Depth + horizontal	30.05	7.07	-
Depth + vertical	41.69	-	0.33
Depth + horizontal + vertical	31.63	7.04	0.38

not HRNet [29]. For fair comparison, all experiments were conducted under the same conditions.

2) DATASETS

The Cityscapes dataset [31] is currently the most comprehensive outdoor scene understanding benchmark with disparity data for semantic segmentation. The Cityscapes dataset provides 5,000 fine annotations. The total number of segmentation classes is 19. There are 2,975 data for training, 500 for validation, and 1525 for testing. It provides a high-resolution RGB image of 2048 × 1024 and a coarse disparity map acquired from a stereo camera. A total of 20k coarsely annotated data are provided, but we do not use it for training. Based on the disparity map, a depth map is computed and used for training.

3) IMPLEMENTATION DETAILS

We train our model based on Pytorch [32] with a single GPU Tesla A100, CUDA 11.1, CUDNN 8.5.0, Pytorch 1.8.0. We adopt a batch size of 16 for BiSeNetV2 [27], 48 for STDC [28] and 12 for HRNet [29]. STDC [28] and HRNet [29] are initialized with the corresponding pretrained models, and BiSeNetV2 [27] is trained from scratch. We train our model utilizing the stochastic gradient descent (SGD) algorithm with 0.9 momentum. The weight decay is $5e^{-4}$. We utilize "poly" learning rate strategy in which the initial rate is multiplied by $(1-\frac{iter}{iter_{max}})^{power}$. The power is 0.9 and the initial learning rate is set as 0.01 for STDC [28], HRNet [29], 0.005 for BiSeNetV2 [27]. STDC [28] trains $60K$ iterations and BiSeNetV2 [27] trains $150K$ iterations. HRNet [29] trains $120K$ iterations. For augmentation, the input image is randomly flipped horizontally, scaled, and cropped to a fixed size for training. The input resolution is cropped to 1024 × 512.

B. ABLATION STUDY

To verify the performance of the training decision and Depth and Pixel-distance based Attention(**DPA**) module, we conduct experiments on various types of depth queries **Q**. Experimental results are obtained on the Cityscapes [31] validation set in STDC1-Seg75 as a baseline and the batch size is reduced from 48 to 24 to quickly confirm the experimental results.

1) DEPTH QUERY CONFIGURATION

The first row in **Table 1** shows the performance of vanilla STDC1-Seg75 without the **DPA** module applied. When only the depth value is used for depth query **Q**, there is a performance improvement of 1.26%. This result confirms that the depth value is useful information for computing the similarity between pixels. When the depth is concatenated with horizontal positional embedding and vertical positional embedding, there is an additional performance improvement of 0.50%. It can be observed that the positional embedding information, that is, the pixel distance, acts as a constraint, effectively handles values that have a similar depth value but are far away, and improves performance. However, concatenating only one of the horizontal or vertical position embeddings degrades performance. It has been confirmed that if only some positional information is provided as a constraint, it adversely affects the use of depth information.

2) SCALE PARAMETER

The scale parameter is learnable and represents the weight for each channel of the depth query **Q**. **Table 2** shows the scale parameter values after training is finished. As the scale parameter multiplied by each channel increases, even a small difference in the corresponding channel is computed as a large difference. Consequently, the larger the parameter value, the more attention is paid to the local area. α_d, β_x, and γ_y, are the values applied to the depth, horizontal, and vertical, respectively. The convergence value differs depending on the **Q** configuration, but β_x converges to approximately 7 and γ_y converges to approximately 0.3. This implies that, for the horizontal area, attention is limited to the local area, and focused on the global area for the vertical area. This is a natural result considering the characteristics of an image. In a horizontal position, because there is a high possibility that there are other objects with a similar depth, the attention area is limited locally. In a vertical position, the global area is considered because it is difficult for an object with a similar depth to exist. In addition, when horizontal positional embedding is concatenated with depth, it can be observed that α_d converges to a relatively small value. This is because horizontal positional embedding locally restricts the attention area; therefore, even if viewed more globally in terms of depth value, it is less affected by other objects with similar depths. However, when vertical positional embedding is concatenated with depth, the parameter α_d for the depth value converges to a larger value. This converges to a rather large value to reduce the influence of other objects occurring in the horizontal area because there is no constraint on the horizontal area. From this result, it can be observed that the scale parameter adjusts the scale between the channels of **Q** appropriately according to the intention.

TABLE 3. Per class IoU(%) results on the validation set of Cityscapes [31].

Method	road	sidewalk	building	wall	fence	pole	traffic light	traffic sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motorcycle	bicycle	mIoU
BiSeNetV2 [27]	97.9	83.0	91.8	46.5	56.1	61.7	68.7	77.3	92.1	63.0	**94.6**	80.8	57.9	94.5	**68.9**	78.8	**75.0**	58.8	76.8	75.0
BiSeNetV2 [27] + DPA	**98.1**	**84.3**	**92.3**	**54.3**	**60.1**	**64.6**	**70.3**	**79.1**	**92.3**	**64.2**	94.4	**82.0**	**60.0**	**94.7**	66.2	**80.6**	73.5	**61.0**	**77.0**	**76.3**
STDC1-Seg75 [28]	**98.1**	**84.2**	91.7	49.8	58.0	58.1	66.0	75.4	91.6	61.2	94.3	79.3	58.9	94.4	**74.9**	81.4	65.1	58.1	75.2	74.5
STDC1-Seg75 [28] + DPA	98.0	83.9	**92.0**	**55.0**	**58.4**	**60.8**	**68.7**	**77.1**	**91.8**	**61.4**	**94.6**	**79.7**	**60.1**	**94.5**	74.3	**83.8**	**71.4**	**62.0**	**75.3**	**75.9**
STDC2-Seg75 [28]	98.2	**85.3**	92.3	56.0	59.1	60.7	69.3	**78.0**	91.9	62.3	94.6	80.3	60.3	95.0	**81.2**	**87.8**	75.1	60.3	76.0	77.0
STDC2-Seg75 [28] + DPA	98.2	85.1	**92.5**	**60.0**	**60.3**	**61.9**	**70.4**	77.8	**92.0**	**64.1**	94.6	**80.5**	**60.5**	95.0	81.1	85.0	**76.5**	**64.3**	**76.4**	**77.7**
HRNet [29]	**98.5**	**87.1**	93.5	58.6	64.2	**71.2**	**75.1**	82.0	**93.2**	65.5	**95.2**	84.8	**66.4**	**95.7**	79.5	91.1	83.3	**70.0**	**80.4**	80.8
HRNet [29] + DPA	98.4	86.9	**93.6**	**62.5**	**66.7**	71.1	74.9	**83.0**	93.1	**65.5**	94.9	84.7	66.6	95.7	**84.4**	**92.0**	**85.7**	68.8	79.1	**81.4**

TABLE 4. Per class IoU(%) results on the test set of Cityscapes [31].

Method	road	sidewalk	building	wall	fence	pole	traffic light	traffic sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motorcycle	bicycle	mIoU
BiSeNetV2 [27]	98.3	83.6	91.7	44.8	52.5	59.8	70.7	74.4	92.8	69.3	94.7	84.0	65.2	95.1	56.3	73.3	60.1	58.1	73.2	73.6
BiSeNetV2 [27] + DPA	98.3	**84.5**	**92.1**	**49.2**	**55.4**	**61.9**	**71.6**	**75.6**	**93.0**	**71.1**	**94.8**	**84.8**	**67.7**	**95.2**	**62.9**	**74.3**	**69.5**	**61.8**	**74.5**	**75.7**
STDC1-Seg75 [28]	98.5	85.2	91.8	51.1	51.7	58.3	68.2	**73.3**	92.7	70.6	94.8	82.6	66.5	95.1	68.0	79.3	70.4	60.5	71.2	75.3
STDC1-Seg75 [28] + DPA	98.5	**85.4**	**92.2**	**54.0**	**53.8**	**59.4**	70.2	74.1	92.7	**71.0**	**94.9**	**83.4**	**67.3**	**95.3**	**70.7**	**83.8**	**80.1**	**62.7**	**72.4**	**76.9**
STDC2-Seg75 [28]	98.5	85.4	92.3	**54.6**	56.0	60.0	70.3	74.2	92.8	70.6	94.7	83.6	67.7	95.5	**71.4**	81.1	74.9	63.7	72.8	76.8
STDC2-Seg75 [28] + DPA	**98.6**	**85.8**	**92.4**	50.7	**56.5**	**61.1**	**71.5**	**74.8**	92.8	**70.7**	**95.0**	**84.1**	**69.2**	**95.6**	71.2	**83.3**	**79.5**	**65.8**	**72.8**	**77.5**
HRNet [29]	98.7	86.9	93.2	49.1	61.6	**71.1**	**78.4**	**81.4**	93.8	71.3	95.7	**87.9**	**73.7**	96.0	71.4	80.1	74.3	**72.3**	78.0	79.7
HRNet [29] + DPA	98.7	**87.1**	**93.4**	**54.7**	**61.9**	70.4	77.9	80.6	**93.9**	**72.9**	**95.8**	87.6	73.2	**96.2**	71.4	**86.3**	**80.1**	71.6	**78.3**	**80.6**

C. QUANTITATIVE RESULT

As in BiSeNetV2 [27], STDC [28] and HRNet [29], we train on the training set to check the validation results, and train using both the training and validation set for the test set. We do not use any evaluation techniques to verify the effectiveness of **DPA** module. *e.g.* multi-scale testing, flipping. First, as shown in **Table 3, 4**, when "**baseline + DPA** " is applied, the segmentation performance is consistently improved regardless of the model type. For BiSeNetV2 [27], there is an improvement in the mIOU of 1.3% on the validation set. In addition, the performance is improved for most classes. In particular, it can be observed that the performance improvement of classes such as **wall**, **fence**, and **pole** is noticeable. This shows that objects that cannot be detected well owing to the similarity of patterns with other objects during conventional RGB-based segmentation can be effectively detected through depth information. These characteristics show a similar trend in the test set. Consequently, the performance is improved by 2.1% on the test set.

STDC [28] includes STDC1, a small model, and STDC2, a large model. There is a performance improvement regardless of the size of the model. Performances are improved by 1.4% and 0.7% on STDC1 and 2 on the validation set, respectively. Similar to the previous BiSeNetV2 [27] case, there is a performance improvement in most of the classes. On the test set, there is a performance improvement of 1.6% and 0.7%, similar to the validation set.

For HRNet [29], the results show a performance improvement of 0.6% on the validation set and 0.9% on the test set. As the complexity and size of the model increase, the degree of performance improvement decrease. We assume that the degree of performance improvement decreases as the baseline network performance increases owing to the limitation of depth accuracy based on stereo cameras.

D. EFFICIENCY ANALYSIS

To verify whether the proposed DPA module is helpful in performance and efficiency, we compare in STDC [28]. The STDC [28] provides two types of encoders depending on the complexity of the network. STDC2 is used to improve performance with a deeper network compared to STDC1. In **Table 5**, it can be seen that STDC2 greatly increases both parameters and FLOP. When **DPA** is applied to each network, FLOPs and parameters increase relatively little. When comparing the baseline of STDC2 with STDC1+DPA, it can be seen that the increase in parameters and FLOPs by DPA is significantly less than the increase by STDC2. However, STDC1 + DPA is 0.1% higher in performance. This shows that the use of the **DPA** module not only improves performance by utilizing the depth information, but also has advantages in efficiency. **Table 6** shows the difference in efficiency between the method of utilizing the depth information through the parallel encoder and **DPA** module. ESA [7] uses RGB and depth information through each encoder. Due to the use of two encoders, FLOPs and parameters increase significantly with the use of the encoder for depth, despite utilizing the shallow network of ResNet34 [33]. Parameters increase by 46% and FLOPs increase by 60%. On the other hand, when **DPA** is applied

TABLE 5. Comparison of efficiency according to the complexity of the Network on Cityscapes test set.

	Params/M	FLOPs/G	mUoU(%)
STDC1	12.1	57.7	75.3
STDC2	16.1(33%↑)	87.5(52%↑)	76.8
STDC1+ DPA	12.3(2%↑)	59.7(4%↑)	76.9
STDC2 + DPA	16.3(35%↑)	89.5(55%↑)	77.5

TABLE 6. Comparison of efficiency with use of parallel encoder on Cityscapes test set.

	Params/M	FLOPs/G	mUoU(%)
ESA(RGB)	32.1	54.2	72.9
ESA(RGBD)	46.9(46%↑)	87.3(60%↑)	**75.7**
STDC2	16.1	87.5	76.8
STDC2 + DPA	16.3(1.2%↑)	89.5(2.3%↑)	**77.5**

TABLE 7. Comparison with depth-based convolution methods on Cityscapes val set.

Method	mIoU(%)	FLOPs/G	Params/M
Baseline [10]	79.94	1385.8	39.76
Depth-aware [15]	79.01	1385.8	39.76
2.5D [16]	78.63	1599.3	46.03
Malleable2.5D [17]			
Kernel=1	80.26	1385.8	39.76
Kernel=3	80.81	1599.3	46.03
Baseline + DPA	80.56	1440.8	39.95
HRNet [29] + DPA	81.45	1019.4	68.25

to STDC2, it increases by 1.2% and 2.3%, respectively. This shows that using **DPA** is more efficient than using two encoders.

E. COMPARISON WITH DEPTH-BASED CONVOLUTION METHOD

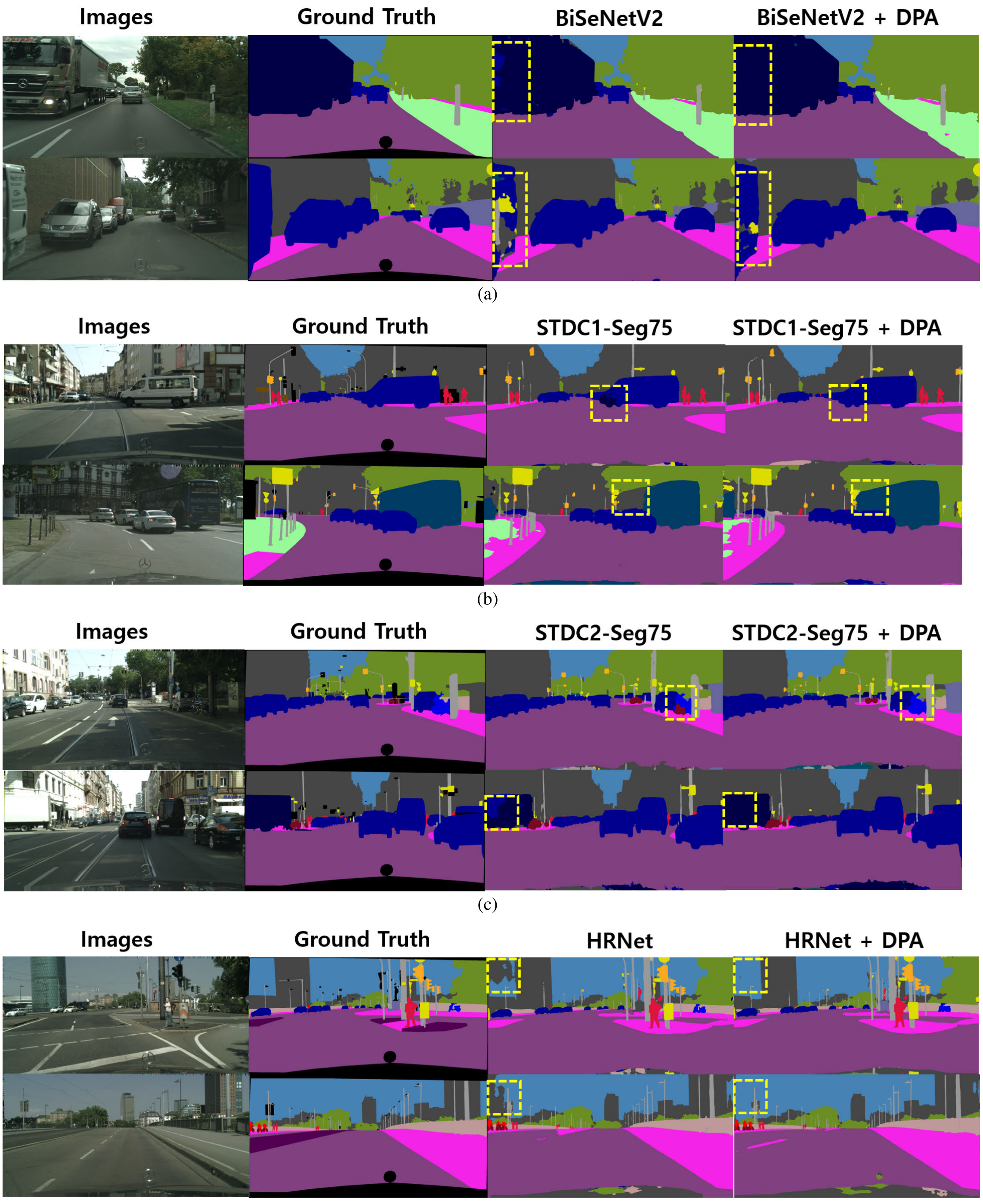
The depth-based convolution models [15], [16], [17] perform a convolution operation by increasing the weight of pixels having similar depth values or by extending the convolution operation using depth values. This is consistent with **DPA**'s assumption that the closer the depth values, the higher the correlation. Since depth-based convolution methods [15], [16], [17] are mainly studied indoors, the excellence of the **DPA** module for outdoor environments is confirmed through performance comparison with depth-based convolution methods [15], [16], [17] in outdoor environments. To fairly compare the effect of each method, we compare the performance of DeepLabV3+ [10] (based on ResNet50 [33]) as a baseline. The results are shown in **Table 7**.

In the case of depth-aware convolution [15] and 2.5D convolution [16], it can be seen that the performance decreases despite the use of depth information. Since these methods have a large difference in the range of depth values in the indoor environment and outdoors, the parameters for the depth values used in the indoor environment do not work well, resulting in reduced performance. In the case of Malleable 2.5D convolution [17], which has solved this problem, it can be confirmed that performance is improved even in outdoor environments by using parameters that can learn the difference in the range of depth values. As the number of kernels used increases from 1 to 3, it can be seen that the performance is further improved when the dimension of the convolution operation is expanded.

It is confirmed that the performance is improved to 80.56% when the **DPA** module is applied. This is slightly less performance than 80.81% when Malleable 2.5D convolution [17] kernel=3 is applied, but it has the advantage of improving performance although the amount of computation and increase in parameters is relatively small. Also, since the depth-based convolution methods [15], [16], [17] change the convolution method, it is necessary to decide which part of the convolution operation to replace in the encoder, and there is a tricky part when applying the pre-trained weights of the encoder. However, in the case of **DPA**, it is easier to apply in the form of a plug-in module to the semantic segmentation network of the encoder-decoder structure. Through this, it can be easily applied to HRNet [29], which is a more effective encoder for semantic segmentation, and it can be observed that the performance is further improved compared to the reduced amount of computation.

F. QUALITATIVE RESULT

Fig. 5 shows the segmentation result of the network to which the baseline and **DPA** are applied. First, **Fig. 5(a)** shows the results of applying BiSeNetV2 [27] and **DPA**. In the first row, the truck on the left occupies a large part of the image and some of it is cropped. The RGB-based network resulted in some parts being incorrectly segmented owing to the limited receptive field and ambiguity. However, through the depth information, it is inferred that the incorrectly segmented part is also a part of the truck, and as a result, the error is corrected. In the case of the second row of **Fig. 5(a)**, only a small part is shown in the image, and there is significant difficulty in segmentation. **DPA** suppresses a lot of erroneous segmentation by using depth information. **Fig. 5(b)** and **Fig. 5(c)** visualize the segmentation results in STDC [28]. In the yellow box in **Fig. 5(b)**, it can be observed that the incorrectly segmented part owing to the darkness of the light or the complexity of the background is effectively improved. However, in the second row of **Fig. 5(b)**, the vanilla STDC [28] shows better performance on the sidewalk and grass at the bottom left. There is no improvement through depth because there is no noticeable difference in depth value between the grass part and the sidewalk even when looking through the depth information. **Fig. 5(c)** is the result of STDC2-Seg75 [28], a model with a larger scale than STDC1-Seg75 [28]. It also corrects errors caused by cropping or darkness of light. **Fig. 5(d)** shows the result of HRNet [29]. It can be observed that the part that cannot distinguish the sky from the building is well distinguished through the depth information. As the performance of the baseline network improves, the qualitative difference in segmentation decreases, but cropped objects or darkness still cause difficulties, and **DPA** can effectively improve this.



(d)

FIGURE 5. Qualitative segmentation examples of the Cityscapes [31] validation dataset. (a) Results of BiSeNetV2 [27] and proposed method. (b) Results of STDC1-Seg75 [28] and proposed method. (c) Results of STDC2-Seg75 [28] and proposed method.(d) Results of HRNet [29] and the proposed method. In the parts marked with yellow boxes, it can be observed that the segmentation is not performed well with RGB information owing to the cutoff of an object, darkness of light, or ambiguity with the background, which is improved by using depth information through DPA.

V. CONCLUSION

In this paper, we present the Depth and Pixel-distance based Attention(**DPA**) module. The similarity between pixels is computed using the depth information and pixel position. Based on the computed similarity, the area requiring attention is expressed in the form of an attention weight, and through this, the RGB-based feature is augmented as a depth contextual feature. The scale parameter efficiently aligns the axes of different vector spaces with depth and pixel positions, and thus, the correlation between pixels is computed. Experiments show that our method can improve the performance of the existing RGB-based segmentation network using depth information by adding a module without changing the main structure. In the future, we would like to explore how to form a graph that expresses the correlation between pixels based on depth information, and improve coarse segmentation to fine segmentation.

REFERENCES

[1] L. C. Chen, G. Papandreou, and I. Kokkinos, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Jun. 2017.

[2] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2881–2890.

[3] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3431–3440.

[4] C. Hazirbas et al., "FuseNet: Incorporating depth into semantic segmentation via fusion-based CNN architecture," in *Proc. 13th Asian Conf. Comput. Vis. (ACCV)*. Taipei, Taiwan: Springer, Nov. 2017.

[5] J. Jiang, L. Zheng, F. Luo, and Z. Zhang, "RedNet: Residual encoder–decoder network for indoor RGB-D semantic segmentation," 2018, *arXiv:1806.01054*.

[6] X. Hu, K. Yang, L. Fei, and K. Wang, "ACNET: Attention based network to exploit complementary features for RGBD semantic segmentation," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 1440–1444.

[7] D. Seichter, M. Kohler, B. Lewandowski, T. Wengefeld, and H.-M. Gross, "Efficient RGB-D semantic segmentation for indoor scene analysis," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, p. 13.

[8] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1520–1528.

[9] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder–decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Oct. 2017.

[10] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder–decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 801–818.

[11] T. Takikawa, D. Acuna, V. Jampani, and S. Fidler, "Gated-SCNN: Gated shape CNNs for semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5229–5238.

[12] Z. Zhang, X. Zhang, C. Peng, X. Xue, and J. Sun, "ExFuse: Enhancing feature fusion for semantic segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 269–284.

[13] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia, "ICNet for real-time semantic segmentation on high-resolution images," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 405–420.

[14] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Medical Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2015.

[15] W. Wang and U. Neumann, "Depth-aware CNN for RGB-D segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 135–150.

[16] Y. Xing, J. Wang, X. Chen, and G. Zeng, "2.5 D convolution for RGB-D semantic segmentation," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 1410–1414.

[17] Y. Xing, J. Wang, and G. Zeng, "Malleable 2.5 D convolution: Learning receptive fields along the depth-axis for RGB-D scene parsing," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020.

[18] Y. Chen, T. Mensink, and E. Gavves, "3D neighborhood convolution: Learning depth-aware features for RGB-D and RGB semantic segmentation," in *Proc. Int. Conf. 3D Vis. (3DV)*, Sep. 2019, pp. 173–182.

[19] M. Yang, K. Yu, C. Zhang, Z. Li, and K. Yang, "DenseASPP for semantic segmentation in street scenes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3684–3692.

[20] 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.

[21] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7794–7803.

[22] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 3146–3154.

[23] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, "CCNet: Criss-cross attention for semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 603–612.

[24] Y. Yuan, L. Huang, J. Guo, C. Zhang, X. Chen, and J. Wang, "OCNet: Object context network for scene parsing," 2018, *arXiv:1809.00916*.

[25] H. Zhang, H. Zhang, C. Wang, and J. Xie, "Co-occurrent features in semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 548–557.

[26] Y. Yuan, X. Chen, and J. Wang, "Object-contextual representations for semantic segmentation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020.

[27] C. Yu, C. Gao, J. Wang, G. Yu, C. Shen, and N. Sang, "BiseNet V2: Bilateral network with guided aggregation for real-time semantic segmentation," *Int. J. Comput. Vis.*, vol. 129, pp. 3051–3068, Sep. 2021.

[28] M. Fan, S. Lai, J. Huang, X. Wei, Z. Chai, J. Luo, and X. Wei, "Rethinking BiSeNet for real-time semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 9716–9725.

[29] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, and W. Liu, "Deep high-resolution representation learning for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3349–3364, Apr. 2020.

[30] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 761–769.

[31] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3213–3223.

[32] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, and A. Desmaison, "Pytorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–12.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

MYUNG-WOO WOO (Graduate Student Member, IEEE) was born in Busan, South Korea, in 1989. He received the B.S. degree in aeronautical science and flight operation from Korea Aerospace University, South Korea, in 2012. He is currently pursuing the master's degree with the Department of Electrical and Computer Engineering, Seoul National University, South Korea. Since 2012, he has been serving as a Major with the Republic of Korea Air Force and is an F-16 Fighter Pilot. His research interests include artificial intelligence and computer vision.

SEUNG-WOO SEO (Member, IEEE) received the B.S. and M.S. degrees from Seoul National University, Seoul, South Korea, and the Ph.D. degree from Pennsylvania State University, University Park, PA, USA, all in electrical engineering. He was a Faculty Member at the Department of Computer Science and Engineering, Pennsylvania State University. He has served as a member of the Research Staff at the Department of Electrical Engineering, Princeton University, Princeton, NJ, USA. In 1996, he joined the School of Electrical Engineering, Institute of New Media and Communications, and the Automation and Systems Research Institute, Seoul National University, as a Faculty Member. He is currently a Professor of electrical engineering with Seoul National University and the Director of the Intelligent Vehicle IT (IVIT) Research Center funded by the Korean Government and Automotive Industries.