IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 11, 2019, accepted October 14, 2019, date of publication October 23, 2019, date of current version November 4, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2949076

Efficient Semantic Segmentation Using Spatio-Channel Dilated Convolutions

JAESEON KIM AND YONG SEOK HEO
Department of Electrical and Computer Engineering, Ajou University, Suwon 16449, South Korea

Corresponding author: Yong Seok Heo (ysheo@ajou.ac.kr)

This work was supported by the Ministry of Science and ICT (MSIT), South Korea, under the Information Technology Research Center (ITRC) Support Program supervised by the Institute for Information and Communications Technology Promotion (IITP) under Grant IITP-2019-2018-0-01424.

ABSTRACT There has been an increasing interest in reducing the computational cost to develop efficient deep convolutional neural networks (DCNN) for real-time semantic segmentation. In this paper, we introduce an efficient convolution method, *Spatio-Channel dilated convolution* (SCDC) which is composed of structured sparse kernels based on the principle of *split-transform-merge*. Specifically, it employs the kernels whose shapes are dilated, not only in spatial domain, but also in channel domain, using a channel sampling approach. Based on SCDC, we propose an efficient convolutional module named *Efficient Spatio-Channel dilated convolution* (ESC). With ESC modules, we further propose ESCNet based on ESPNet architecture which is one of the *state-of-the-art* real-time semantic segmentation network that can be easily deployed on edge devices. We evaluated our ESCNet on the Cityscapes dataset and obtained competitive results, with a good trade-off between accuracy and computational cost. The proposed ESCNet achieves 61.5 % mean intersection over union (IoU) with only 196 K network parameters, and processes high resolution images at a rate of 164 frames per second (FPS) on a standard Titan Xp GPU. Various experimental results show that our method is reasonably accurate, light, and fast.

INDEX TERMS Semantic segmentation, spatio-channel dilated convolution, efficient neural network, real-time processing.

I. INTRODUCTION

Semantic segmentation is to predict a label for each pixel in an image with corresponding predefined classes. This pixel-wise semantic information plays a crucial role in enabling various applications to fully understand the scenes. Thus, it has been one of fundamental and challenging problems in computer vision for decades.

Recently, deep convolutional neural network (DCNN) based methods [2]–[7] have shown remarkably accurate results for the semantic segmentation problem. Most of these deep learning based methods focused on improving the accuracy by considerably enlarging their depth and width. Thus, their networks have an enormous number of parameters because a greater number of parameters generally lead to a higher accuracy in deep learning.

Because of their numerous parameters, however, these accurate but ponderous networks suffer from heavy computational costs and time-consuming inferences. For example, as shown in Fig. 1, PSPNet [3] and DRN [9] show remarkably accurate performance, but they have huge amount

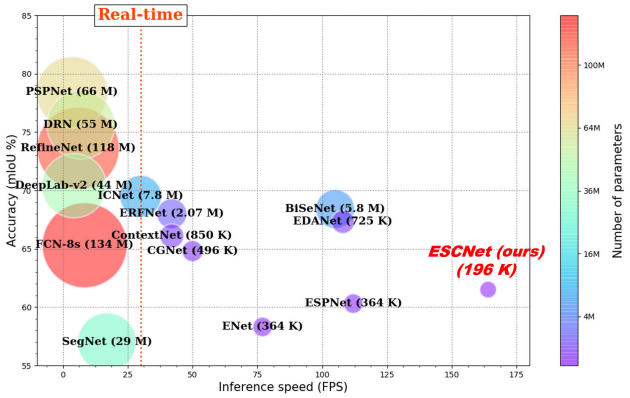


FIGURE 1. Visualization of inference speed, accuracy and the number of network parameters of the *state-of-the-art* methods. Here, the accuracy is mean intersection over union (mIoU) score for Cityscapes [8] test set. The circle size represents the number of network parameters.

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

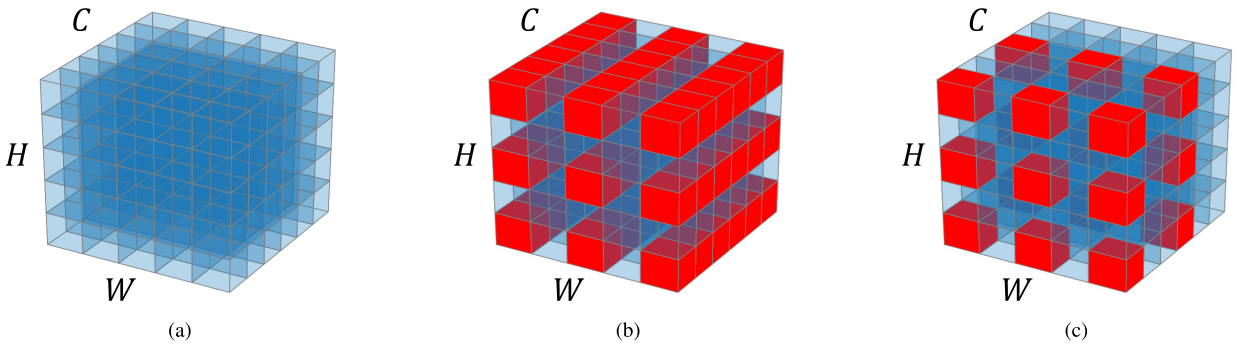


FIGURE 2. Illustration of conventional dilated convolution and the proposed SCDC operations. (a) Input feature map. H, W, and C represent the height, width and number of channels of the input feature map, respectively. Here, $H = 5$, $W = 5$, and $C = 6$. (b) Operation of a conventional dilated convolution at the center location in the spatial domain. Red cells represent the features used for the dilated convolution operation to produce an output feature at the center location of the output feature map. Here, 3 × 3 convolutional kernel with dilation rate 2 is used. (c) SCDC operation at the center location in the spatial domain. Here, 3 × 3 convolutional kernel with a dilation rate of 2 and a channel sampling rate of 3 is used.

of parameters and are noticeably slow to use for real-time tasks. Although the computational resources have been developed, many recent real-world applications that use embedded devices, *e.g.*, autonomous driving, augmented reality, and intelligent surveillance, still require light-weight networks to perform real-time data processing. Therefore, there has been an increasing interest in reducing the computational cost to develop efficient networks for real-time semantic segmentation.

Currently, *state-of-the-art* networks for real-time semantic segmentation highly employ the principle of convolution factorization, which aims to decompose a standard convolution into smaller, more efficient convolutions to reduce computational cost. ENet [10], ERFNet [11], and EDANet [12] exploit asymmetric convolutions [10], [13], whereas ContextNet [14] and BiSeNet [15] employ depthwise separable convolutions [16], [17] in their networks.

In addition, ESPNet [1] which can be easily deployed on edge devices, effectively decomposed a standard convolutional kernel into small kernels using a 1×1 convolution (also called pointwise convolution in [17]) and multiple dilated convolutions with different dilation rates. It has shown a good trade-off between accuracy and inference speed, with much fewer network parameters than the above-mentioned *state-of-the-art* methods. Moreover, ESPNetv2 [18] further improved its efficiency using depthwise separable convolutions [16], [17]. Fundamentally, their methods are based on the *feature re-sampling* approach, which uses an input feature map multiple times with different pooling rates [3], [19] or kernel sizes [5] for learning the representations. More specifically, in each layer, they reduce the dimension of the input feature map using pointwise convolution, and then repeat dilated convolution operations for this low-dimensional feature map with different dilation rates. However, even though they re-sample the low-dimensional feature map, their *feature re-sampling* approach is still computationally expensive in itself because it uses all channels of the feature map for multiple times.

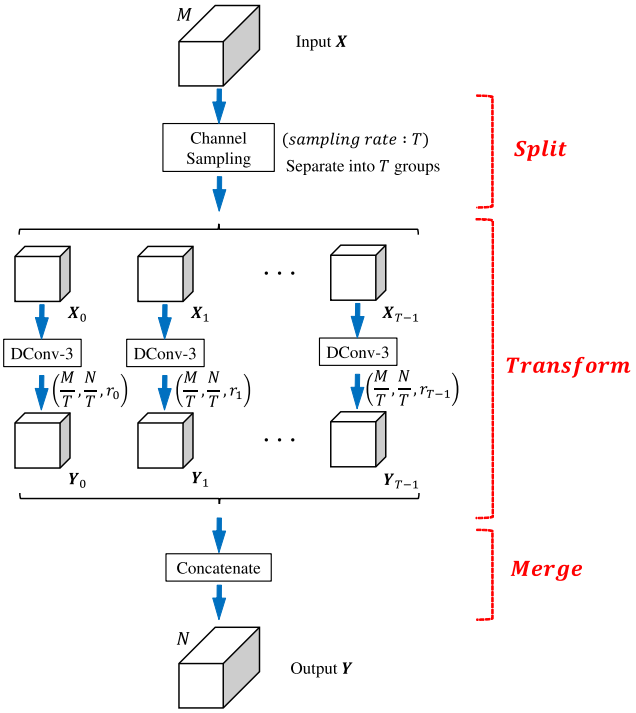
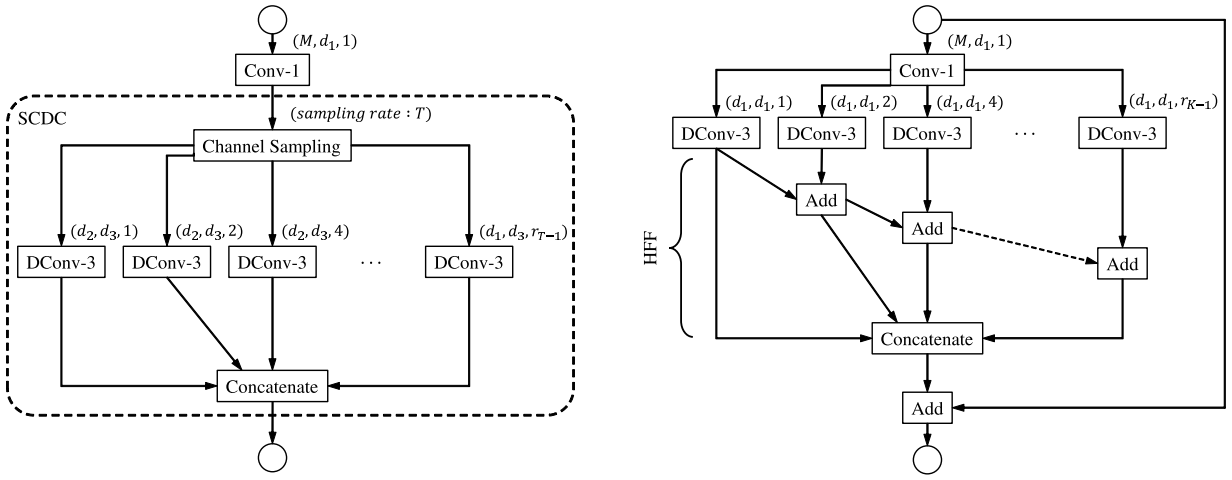


FIGURE 3. Proposed SCDC. Here, M and N represent the numbers of the input and output channels, respectively. DConv-n represents $n \times n$ dilated convolutional layer that is denoted as (# input channels, # output channels, and a dilation rate). r_t represents the dilation rate for the t-th group.

To deal with this problem, we propose an efficient convolution method, *Spatio-Channel dilated convolution* (SCDC) (see Fig. 2(c) and Fig. 3), which exploits multiple sparse dilated convolutions without repeatedly using all the channels of input feature map; rather, it uses only a fraction of the channels for each dilated convolution which has different size of receptive fields. Thus, it can simultaneously reduce the computational cost and exploit the convolutional kernels that have varying effective receptive fields. It is so named because



(a) ESC Module (b) ESP Module [1]

FIGURE 4. Proposed ESC module and the ESP module [1]. (a) Our proposed ESC module. (b) ESP module [1]. Here, Conv-n and DConv-n represent $n \times n$ standard and $n \times n$ dilated convolutional layers, respectively. Each convolutional layer is denoted by (# input channels, # output channels, and a dilation rate). The dilatation rates of each module are $r_k = 2^k$, where $k \in [0, K-1]$ for the ESP module, and $r_t = 2^t$, where $t \in [0, T-1]$ for the ESC module. K represents the dimension reduction factor and T represents the channel sampling rate. HFF represents the hierarchical feature fusion adopted in [1] for degridding.

it uses kernels whose shapes are dilated not only in spatial domain, but also in channel domain. In other words, it is composed of sparse dilated convolutions, each of which uses specific channel-indexed feature map. To select the channel indices of the feature map, we use a channel sampling approach. With channel sampling rate T, we design T different kernels, each of which has a different spatial dilation rate, and then uses the channel indices that do not overlap the other kernels.

With the proposed SCDC, we propose an efficient convolutional module, *Efficient Spatio-Channel dilated convolution* (ESC) as shown in Fig. 4(a). Inspired by ESPNet [1] architecture, we further propose an efficient network, ESCNet. Unlike the typical vision algorithms that are comprised of several steps, such as preprocessing [20] and feature clustering [21], we propose the end-to-end semantic segmentation framework. As shown in Fig. 1, our proposed ESCNet generates reasonably accurate results at a significantly fast speed with extremely fewer network parameters than other *state-of-the-art* methods.

The main contributions of our work are summarized as follows:

1) We propose an efficient convolution method, SCDC, which uses structured sparse kernels with different spatial dilation rates to efficiently enlarge the effective receptive field, and reduces computational complexity using a regular channel sampling approach.
2) We propose a novel convolutional module, ESC, which takes advantage of SCDC. Using ESC modules, we design ESCNet, based on the ESPNet [1] architecture, for real-time semantic segmentation.
3) We achieve competitive results on the Cityscapes [8] test set. With the proposed ESCNet, which has only 196 K parameters, we obtain a 61.5% mean intersection over union (IoU) with a speed of 164 FPS. It is reasonably accurate, light, and fast.

II. RELATED WORK

A. ENCODER-DECODER ARCHITECTURE

Most DCNN models perform convolutions followed by a pooling operation to expand the receptive field and learn strong feature representations which have rich semantic information. However, because of the pooling operations, there are inevitable loss of spatial information, especially for boundaries of small and/or sharp objects in the original input image. To recover the spatial information, early works [22]–[24] designed the encoder-decoder architecture. Inspired by the achievements of these early works, many recent methods [3], [4], [7] adopted the encoder-decoder architecture to combine spatial and semantic information.

B. EFFICIENT SEMANTIC SEGMENTATION METHODS

Recent *state-of-the-art* DCNN based methods for efficient semantic segmentation adopt various efficient convolution methods, *e.g.*, dilated convolutions [25], [26], group convolutions [27], [28], depthwise separable convolutions [16], [17] and asymmetric convolutions [10], [13].

1) DILATED CONVOLUTION

A large receptive field is crucial for obtaining high-level abstract semantic information, which is directly related to accurate results in semantic segmentation. Deepening the depth of network, and employing a large size kernel, or downsampling the feature map can help enlarge the effective receptive field of the network. However, most of these methods require extra parameters which increase computational

cost, or suffer from loss of the spatial information of the feature. To remedy these problems, [25], [26] proposed dilated convolution (also called atrous convolution in [26]).

Dilated convolution [25], [26] is a special form of standard convolution that effectively enlarges the receptive field without introducing extra parameters and computational cost. In addition, compared with the standard convolutional layer that has the same size of receptive field, the dilated convolutional layer effectively reduces computational cost. Therefore, dilated convolution has been adopted in many recent *state-of-the-art* methods [1], [10]–[12], [18], [29]–[31] for efficient semantic segmentation.

Although the conventional dilated convolution can reduce computational complexity, it is restricted only in spatial domain. Meanwhile, the proposed SCDC can reduce computational complexity, not only in spatial domain, but also in channel domain.

2) GROUP CONVOLUTION

To reduce the computational cost and eliminate redundancy, group convolution [27], [28] is also widely used in many computer vision tasks including efficient semantic segmentation [18], [31]. Group convolution is composed of structured sparse kernels and greatly reduces the computational cost. However, if multiple group convolutional layers are stacked together, a certain channel of output feature map is only derived from a small fraction of the input channels [32]. Because this property obstructs information flow between channel groups, it weakens representation of the feature map.

To address this issue, recent works [32]–[36] permute the divided groups of feature map (also called channel shuffle in [32], [33]) after group convolution operations. When they operate group convolutions or permutations, they separate groups with successive channels. In contrast, the sparse-complementary convolution [37] exploits two structured sparse kernels that use even and odd indexed channels respectively.

These works, however, exploit the same size of receptive field for each kernel. Because scene images usually contain objects of various sizes, networks with fixed-size of receptive fields for scene parsing often produce inconsistent predictions for large objects, and omit small objects [38]. Meanwhile, the proposed SCDC uses structured sparse kernels with various sizes of receptive fields for specific indexed channels.

3) DEPTHWISE SEPARABLE CONVOLUTION

Recent *state-of-the-art* efficient networks for image classification, *e.g.*, MobileNets [17], [39], and ShuffleNets [32], [33] exploit depthwise separable convolutions [16], [17] which factorize a standard convolution into depthwise and pointwise convolutions. In other words, they replace $n \times n$ convolution with a sequence of $n \times n$ depthwise convolution and pointwise convolution. References [16], [17] have shown the efficiency of the depthwise separable convolutions for effectively reducing the computational cost. Likewise, the depthwise separable convolution is also adopted in various efficient networks [14], [15], [18] for semantic segmentation task.

4) ASYMMETRIC CONVOLUTION

ENet [10], ERFNet [11], and EDANet [12] employ asymmetric convolutions [10], [13] which factorize a standard two-dimensional convolution into two consecutive one-dimensional convolutions to reduce the computational cost. In other words, they replace an $n \times n$ convolution with a sequence of $n \times 1$ and $1 \times n$ convolutions.

Aside from the above efficient convolution methods, ContextNet [14], BiSeNet [15], CGNet [29], and ICNet [30] use efficient multi-paths or branches in a light-weight network to extract various features that have different spatial, contextual, and scale information.

III. OUR APPROACH

In this section, we describe the concept of our proposed SCDC, as illustrated in Fig. 2(c) and Fig. 3. Then we introduce our proposed ESC module in detail. Furthermore, we illustrate the proposed ESCNet with the ESC modules. Finally, we elaborate on its effectiveness for reducing computational complexity and enlarging the receptive field compared to other convolution methods.

A. SPATIO-CHANNEL DILATED CONVOLUTION

Generally, a dilated convolutional layer [25], [26] with a convolutional kernel $\boldsymbol{W} \in \mathbb{R}^{N \times M \times n \times n}$ transforms an input feature map $\boldsymbol{X} \in \mathbb{R}^{M \times A_{in} \times B_{in}}$ into an output feature map $\boldsymbol{Y} \in \mathbb{R}^{N \times A_{out} \times B_{out}}$, where M and N represent the number of input and output channels, respectively. n represents the height and width of the kernel. Moreover, A_{in} and B_{in} and A_{out} and B_{out} represent the height and width of the input and output feature maps, respectively. With a stride of one and padding, the conventional dilated convolution operation with dilation rate r produces an output feature value $\boldsymbol{Y}(p, k, l)$ which is at the spatial location (k, l) in the p-th channel of $\boldsymbol{Y}$ as follows:

$$\boldsymbol{Y}(p, k, l) = \sum_{c} \sum_{i,j} \boldsymbol{W}(p, c, i, j) \cdot \boldsymbol{X}(c, k + ir, l + jr), \quad (1)$$

where $c \in [0, M-1]$ and $p \in [0, N-1]$ are the indices for the channels of the input and the output feature map, respectively. $i \in [-\lfloor \frac{n}{2} \rfloor, -\lfloor \frac{n}{2} \rfloor + n - 1]$ and $j \in [-\lfloor \frac{n}{2} \rfloor, -\lfloor \frac{n}{2} \rfloor + n - 1]$ are the indices along the height and width directions of kernel $\boldsymbol{W}$. For ease of presentation, the biases are omitted.

As shown in Fig. 2(b), the conventional dilated convolution operation saves computational cost in spatial domain; however, it uses all channels of input feature map. To further reduce the computational cost in channel domain, we sample the input feature map in channel domains, as well as spatial domains. As shown in Fig. 2(c), we use only specific channels of the input feature map for a dilated convolution. Since other channels of the feature map do not affect the selected channels, this is equivalent to collecting the selected channels

from the feature map and then operating conventional dilated convolution with corresponding kernel.

To select specific channels for a dilated convolution, we use a regular sampling operation in the channel axis with sampling rate T, which is an integer divisor of M. Using the regular sampling approach, we can generate a set of selected channel indices of the input feature map C, and collect the corresponding feature values of the selected channels. Then, we generate T different sets that do not overlap each other to separate the input feature map into T different feature map groups (*split*). A t-th set of selected channel indices C_t can be defined as follows:

$$C_t = \{m | m \bmod T = t\}, \tag{2}$$

where $m \in [0, M-1]$ is the channel index of the input feature map, $t \in [0, T-1]$ is the index of the selected feature map groups, and mod is a modulo operation.

Using the selected channel indices in C_t, we extract $\boldsymbol{X}_t$ from $\boldsymbol{X}$ by collecting the selected channels from $\boldsymbol{X}$, as follows:

$$\boldsymbol{X}_t = \{\boldsymbol{X}(m, \cdot, \cdot) | m \in C_t\}, \tag{3}$$

where $\boldsymbol{X}(m, \cdot, \cdot)$ is the m-th channel feature values of $\boldsymbol{X}$, and $\boldsymbol{X}_t \in \mathbb{R}^{\frac{M}{T} \times A_{in} \times B_{in}}$.

Then, we generate the t-th corresponding convolutional kernel $\boldsymbol{W}_t \in \mathbb{R}^{\frac{N}{T} \times \frac{M}{T} \times n \times n}$ with spatial dilation rate r_t. We assume r_t is an integer and is a function of t. Next, we obtain the t-th output feature map group $\boldsymbol{Y}_t \in \mathbb{R}^{\frac{N}{T} \times A_{out} \times B_{out}}$ through the conventional dilated convolution operation using $\boldsymbol{X}_t$ and $\boldsymbol{W}_t$ (*transform*). The output feature value $\boldsymbol{Y}_t(p', k, l)$, which is at the spatial location (k, l) in the p'-th channel of $\boldsymbol{Y}_t$ is computed as

$$\boldsymbol{Y}_t(p', k, l) = \sum_{c'} \sum_{i,j} \boldsymbol{W}_t(p', c', i, j) \cdot \boldsymbol{X}_t(c', k + ir_t, l + jr_t), \tag{4}$$

where $c' \in [0, \frac{M}{T} - 1]$ and $p' \in [0, \frac{N}{T} - 1]$ are the indices for the channels of $\boldsymbol{X}_t$ and $\boldsymbol{Y}_t$, respectively. i and j are the indices along the height and width directions of the kernel $\boldsymbol{W}_t$. Finally, we obtain an output feature map $\boldsymbol{Y} \in \mathbb{R}^{N \times A_{out} \times B_{out}}$ by concatenating all $\boldsymbol{Y}_t$ along the channel axis (*merge*), as follows:

$$\boldsymbol{Y} = \boldsymbol{Y}_0 \oplus \boldsymbol{Y}_1 \oplus \boldsymbol{Y}_2 \oplus \cdots \oplus \boldsymbol{Y}_{T-1}, \tag{5}$$

where $\oplus$ represents an operation to concatenate feature maps in the channel axis. This SCDC mechanism is illustrated in Fig. 3.

B. ESC MODULE

Now, we propose a new efficient module, ESC that takes advantage of SCDC. The proposed ESC module is based on the principle of *reduce-split-transform-merge* similar to [1] to reduce the computational complexity. As shown in Fig. 4(a), the ESC module replaces a standard convolution with a pointwise convolution and SCDC.

The ESC module first reduces the dimension of the input feature map using a pointwise convolution [1] with dimension reduction factor K. Here, we assume that K is an integer to make $\frac{N}{K}$ as an integer in our experiments. The input feature map with M channels is reduced to d_1 channels after a pointwise convolution, where d_1 is defined by $d_1 = \frac{N}{K}$ *(reduce)*. Then, the ESC module performs SCDC with the dimension-reduced feature map *(split-transform-merge)*. Here, channel sampling rate is T and the dilation rate is $r_t = 2^t$ for each t-th group, where $t \in \{1, 2, ..., T\}$.

In SCDC, the dimension-reduced feature map with d_1 channels are separated into T groups with d_2 channels, where $d_2 = \frac{d_1}{T} = \frac{N}{KT}$. As a result of T parallel dilated convolutions in SCDC, T feature map groups are individually transformed into feature maps with d_3 channels, where $d_3 = \frac{N}{T}$. The final output feature map with N channels is generated by concatenating these T feature map groups along the channel axis.

C. ESCNet

We propose ESCNet for real-time semantic segmentation based on ESPNet [1] architecture with ESC modules. Our architecture is fully depicted in Fig. 5 and Table 1. It is a fully convolutional network [2] of encoder-decoder architecture, utilizing both spatial information from the low level features and semantic information from the high level features.

The encoder part is structured with the ESC modules to extract semantic information. The hyper-parameter α_l is the number of the ESC modules stacked together to control the depth of the network, where l is the spatial level of the network; ESC $\times \alpha_l$ means that α_l number of ESC modules are stacked together. The spatial size of the input feature maps, which are fed into these stacked modules, is 2^l times smaller than the original input image.

We concatenate downsampled original images and intermediate feature maps to improve the information flow. Because the downsampling operations are performed in the red colored layers of the encoder, as shown in Fig. 5, the output of the encoder has $\frac{1}{8}$ times smaller spatial size than the original input image. To perform downsampling, we replaced a pointwise convolution in the proposed ESC module by a standard convolution with 3×3 kernel size and a stride of 2. Following [1], we set $\alpha_2 = 2$ and $\alpha_3 = 8$.

The decoder part, has much fewer parameters than the encoder network. When it performs upsampling and convolution operations, it combines high-level features with low-level features, which have enough semantic information and rich spatial information, respectively. We use the transposed convolution with 2×2 kernel size and a stride of 2 as the deconvolution. Because the upsampling operations are performed in the green colored layers of decoder, as shown in Fig. 5, the segmentation mask, which is the output of the decoder, has the same spatial size as the original input image.

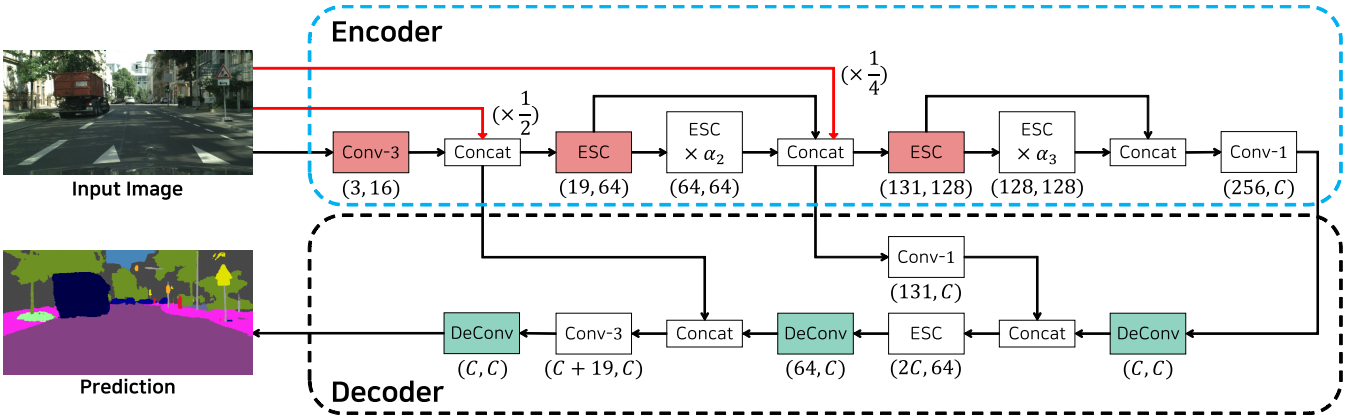


FIGURE 5. Illustration of our proposed ESCNet. Here, Conv-*n* represents $n \times n$ convolution and DeConv represents transposed convolution. The red and green boxes represent the layers responsible for downsampling and upsampling feature maps, respectively. We denote each module as (# input channels, # output channels), and C as the number of classes to predict.

TABLE 1. Layer disposal of our proposed network (ESCNet). Here, "Symbol" represents the symbol of the input and output feature maps of each layer, $\oplus$ represents the concatenate operation, and "Channels" represents the number of channels of the feature map. In addition, C represents the number of classes of the dataset, and "Output resolution" represents the output resolution for an example input image size of 1024×512. I^s : Resized input image s times the size of an original input image. F_l : An output feature map of each layer with index $l \in [1, 12]$.

Part	Layer	Type	Kernel Size	Stride	Input		Output		Output resolution
					Symbol	Channels	Symbol	Channels	
Encoder	Conv-3	Downsampler block	3×3	2	I^1	3	F_1	16	512×256
	ESC	Downsampler block		2	$F_1 \oplus I^{\frac{1}{2}}$	19	F_2	64	256×128
	ESC ×2			1	F_2	64	F_3	64	256×128
	ESC	Downsampler block		2	$F_3 \oplus F_2 \oplus I^{\frac{1}{4}}$	131	F_4	128	128×64
	ESC ×8			1	F_4	128	F_5	128	128×64
	Conv-1		1×1	1	$F_4 \oplus F_5$	256	F_6	C	128×64
Decoder	DeConv	Transposed convolution	2×2	2	F_6	C	F_7	C	256×128
	Conv-1		1×1	1	$F_3 \oplus F_2 \oplus I^{\frac{1}{4}}$	131	F_8	C	256×128
	ESC			1	$F_7 \oplus F_8$	2C	F_9	64	256×128
	DeConv	Transposed convolution	2×2	2	F_9	64	F_{10}	C	512×256
	Conv-3		3×3	1	$F_{10} \oplus F_1 \oplus I^{\frac{1}{2}}$	C+19	F_{11}	C	512×256
	DeConv	Transposed convolution	2×2	2	F_{11}	C	F_{12}	C	1024×512

TABLE 2. Comparison of different convolution types. Here, M and N represent the number of input and output channels, respectively, and n represents the height and width of the convolutional kernel. g and T represent the number of groups for the group convolution [27], [28] and the channel sampling rate in SCDC, respectively.

Convolution Type	Parameters	Effective Receptive Field
Standard	n^2MN	$n \times n$
Dilated [25], [26]	n^2MN	$n_r \times n_r$
Group [27], [28]	$\frac{n^2MN}{g}$	$n \times n$
SCDC (ours)	$\frac{n^2MN}{T}$	$n_r \times n_r$

IV. COMPARISON OF COMPLEXITY

Table 2 shows a comparison of the different types of convolutions and the proposed SCDC. The number of parameters of both $n \times n$ standard convolutional layer and $n \times n$ dilated convolutional layer is $M \cdot N \cdot n^2$. Here, M and N represent the number of input and output channels, respectively. When we operate dilated convolution with dilation rate r, the effective receptive field is $n_r \times n_r$, where $n_r = (n-1) \cdot r + 1$. While conventional dilated convolutions only reduce computational complexity in spatial domain, SCDC can reduce computational complexity in both spatial domain and channel domain.

Because SCDC separates the input feature map into T groups and operates different dilated convolutions, it can be thought as a type of group convolution [27], [28]. Although the parameters of SCDC and the group convolution with T groups are equal to $\frac{M \cdot N \cdot n^2}{T}$, the effective receptive field of SCDC is larger than that of the group convolution, because SCDC efficiently exploits multiple dilated convolutions.

Another difference between SCDC and the conventional group convolution is the channel shuffle operation. If multiple group convolutional layers are stacked together, outputs from a particular group are only related to the inputs within

TABLE 3. Comparison of the proposed ESC and the ESP [1] modules. Here, *M* and *N* represent the number of input and output channels, respectively, and *n* represents the height and width of the convolutional kernel. *K* and *T* represent the dimension reduction factor and the channel sampling rate, respectively.

Module	Parameters	Effective Receptive Field
ESP [1]	$\frac{MN}{K} + \frac{(Nn)^2}{K}$	$[(n-1)2^{K-1}+1]^2$
ESC (ours)	$\frac{MN}{K} + \frac{(Nn)^2}{KT}$	$[(n-1)2^{T-1}+1]^2$

the group. This property weakens representation because it blocks information flow between different channel groups. However, SCDC automatically avoids this property by using a channel sampling operation that regularly samples feature maps from other channel groups. In other words, the channel sampling operation in SCDC has the effect of a channel shuffle.

Because the proposed ESC module is inspired by the ESP module [1], which is based on *reduce-split-transform-merge* strategy, they have some similarities. Both reduce the dimension of the input feature maps with factor K, and split for different dilated convolutions in parallel. Because both modules exploit dilated convolutions, they have larger effective receptive fields than the standard convolutional layer. The ESP module has convolutional kernels with dilation rate $r = 2^k$, $k \in [0, K-1]$. The proposed ESC module has convolutional kernels with dilation rate $r = 2^t, t \in [0, T-1]$. When the ESP module and ESC module use $n \times n$ dilated convolutional kernels, their effective receptive field sizes are $[(n-1)2^{K-1}+1]^2$ and $[(n-1)2^{T-1}+1]^2$, respectively.

However, unlike the ESP module, our ESC module does not repeatedly use all channels of the feature map for multiple dilated convolutions. The ESC module samples the regular channels with sampling rate T, so that it separates T different feature map groups from the original feature map. Then, it feeds each feature map group into the corresponding dilated convolutional layer which is a function of the group index t.

Because of the above efficient method, ESC module can reduce the number of parameters further than that of the ESP module without severe degradation of accuracy. The ESP module has $\frac{MN}{K} + \frac{(Nn)^2}{K}$ parameters; however, ESC module has $\frac{MN}{K} + \frac{(Nn)^2}{KT}$ parameters. Compared to the standard convolutional layer, ESC module reduces the number of parameters by a factor of $\frac{n^2MKT}{MT+n^2N}$. Moreover, compared to the ESP module, the ESC module reduces the number of parameters by a factor of $\frac{(M+n^2N)T}{MT+n^2N}$. A comparison of both modules is provided in Table 3.

V. EXPERIMENTS

In this section, we evaluate our ESCNet on the Cityscapes [8] and CamVid [40], the challenging road scene understanding datasets. Firstly, we describe the details of the datasets, the implementation of our method, and the evaluation metrics. Then, we perform a series of ablation studies on our ESC module to validate its effectiveness and identify an efficient design. Finally, we report our competitive evaluation results and compare them with other *state-of-the-art* methods on the Cityscapes and CamVid test set.

A. DATASET

Cityscapes: The Cityscapes dataset [8] is a large scale dataset for semantic segmentation on urban traffic scene understanding. It consists of 5000 finely annotated high-resolution (2048×1024) images; this includes 2975 images for training, 500 images for validation and 1525 images for testing. Each pixel of these images is annotated into pre-defined 19 classes. Although another 20,000 coarse annotated images are available, we only used the fine annotated images for training in our experiment. We evaluated our results for the test set using the Cityscapes online servers.

CamVid: We also evaluated our ESCNet on the CamVid dataset [40], which contains images extracted from high resolution video sequences with size of 960×720. As many other prior works, we adopt the split of Sturgess *et al.* [41], which divided the dataset that includes 11 semantic classes into 367 images for training, 100 images for validation, and 233 images for testing. Following [22], we downsampled the images to 480×360 before training and testing.

B. IMPLEMENTATION DETAILS

We trained our networks using Pytorch with CUDA 9.0 and cuDNN v7 back-ends with a single NVIDIA Titan Xp (Pascal) GPU with 3840 CUDA cores. We also used an NVIDIA Titan X GPU (3584 CUDA cores) for testing to compare with other *state-of-the-art* networks under the same environment.

Almost all of our training strategies follow [1]. We did not use any extra dataset for pretrain, *e.g.*, ImageNet [42]. We trained the encoder first and then trained the full network, after attaching a light-weight decoder to the trained encoder. ADAM optimization [43] was used with a weight decay of 0.0005. We initialized the network parameters using He initialization [44]. We adopted batch normalization [45] and PReLU [44] activation function, which followed all layers except the Conv-1, and the last DeConv layer in the decoder as shown in Fig. 5. We adopted the poly learning rate strategy inspired by [5], [46] for training, where the learning rate lr is defined as

$$lr = lr_{init} \times \left(1 - \frac{epoch}{epoch_{max}}\right)^{power}, \tag{6}$$

where lr_{init}, $epoch_{max}$ and $power$ represent the initial learning rate, the number of total epochs, and the power of the polynomial, respectively. We set $lr_{init} = 0.0005$, $epoch_{max} = 500$, and $power = 0.9$. To address the class imbalance, we used the inverse class weighting scheme [1], [10], [11] in the cross-entropy loss function. To enlarge the dataset, we adopted standard data augmentation strategies, *e.g.*, scaling, cropping and flipping. Our network was trained and tested

TABLE 4. Analysis on the effectiveness of the channel sampling operation in the proposed ESC modules. # Params represents the number of network parameters. mIoU represents the class mIoU of the cityscapes validation set. We measured the inference speed (FPS) by averaging 100 iterations for 1024 × 512 size of inputs on an NVIDIA Titan Xp GPU.

Channel sampling	# Params	mIoU (%)	Speed (FPS)
✓	176 K	**54.9**	247.2
		54.4	**255.5**

on 1024 × 512 images that were downsampled by two for the Cityscapes dataset. Hence, we upsampled the output feature map using bilinear interpolation to 2048 × 1024 for evaluation on the Cityscapes online servers.

C. EVALUATION METRICS

We evaluated our model using standard strategies to measure network performance, *e.g.*, segmentation accuracy, computational complexity, and network size. For the segmentation accuracy, we evaluated with the mean IoU (Intersection over Union), class-wise IoU, category-wise IoU as metrics. Furthermore, following [22], we also report our results using the global accuracy, class average accuracy, and boundary F1 score (BF) [47].

For the computational complexity, we evaluated both floating-point operations per second (FLOPs) and frames per second (FPS) for the inference speed. The inference speed is a direct metric; however, it could differ depending on the software and hardware environments. Hence, although FLOPs is an indirect metric [33], it is widely used for approximation. For the network size, which is the amount of required storage space, we measured the number of network parameters, which is directly connected to the network size.

D. ABLATION STUDY

1) CHANNEL SAMPLING OPERATION

Firstly, we evaluated the effectiveness of the channel sampling operation in our ESC module. For comparison, we used ESC module in Fig. 4(a), which set $K = 4$ and $T = 4$. Then, we replaced the channel sampling layer with the layer that separates input feature maps into 4 groups with successive channels. With these two modules, we structured two ESCNet encoders. Because the encoder network of ESCNet is a fully convolutional network [2] for semantic segmentation in itself, we evaluated the encoders first to save computational resources. Table 4 shows the effectiveness of the channel sampling operation in our ESC module. The encoder with the channel sampling operation was slightly slower, but more accurate.

2) DIFFERENT SETTINGS IN ESC MODULE

Because we found that the channel sampling operation in our ESC module is effective, we performed a series of ablation studies to further validate the effectiveness of our ESC module and to select the design that improves the performance while maintaining efficiency, as shown in Table 5.

TABLE 5. Ablation studies for the K, T, residual connection (RC), and hierarchical feature fusion (HFF) in our ESC module. Here, K is the dimension reduction factor and T is the channel sampling rate. We evaluated the encoder networks of the ESCNet, which consist of the ESC modules listed below.

Module	K	T	RC	HFF	# Params	mIoU (%)	Speed (FPS)
ESP [1]	5	-	✓	✓	349 K	53.3	162.6
ESC	4	2			266 K	53.1	**247.2**
			✓			50.6	226.9
				✓		52.8	234.2
			✓	✓		51.5	217.1
	4	4			**176 K**	54.9	242.7
			✓			52.2	224.2
				✓		55.9	220.6
			✓	✓		53.7	204.1
	2	4			341 K	**56.4**	206.7
			✓			54.8	192.9
				✓		56.0	191.7
			✓	✓		54.1	179.5

To evaluate the performance of each setting in the ESC module, we performed various experiments with different dimension reduction factors (K) and channel sampling rates (T). In addition, we evaluated how much the the residual connection [48] and hierarchical feature fusion (HFF), which is adopted in [1] for degridding, affect the accuracy and inference speed in our proposed ESC module. For comparison, the results of the ESPNet [1] encoder are included in the first row of Table 5.

a: DIMENSION REDUCTION FACTOR K

To investigate the effect of the dimension reduction factor K, we changed the value of K from 2 to 4 while fixing the value of T to 4. Because the encoder group with different K values has the same T value, the effective receptive fields are all the same. However, the encoders with $K = 2$ have approximately twice as many parameters as the encoders with $K = 4$. Although the encoders with $K = 2$ were slightly more accurate than the corresponding encoders with $K = 4$, the inference speed was considerably slower.

b: SAMPLING RATE T

To investigate the effect of the channel sampling rate T, we changed the value of T from 2 to 4 while fixing the value of K to 4. The encoders with $T = 2$ have approximately 90 K more parameters than the encoders with $T = 4$, and the overall speed of the former was slightly faster. However, the encoders with $T = 4$ were more accurate than corresponding encoders with $T = 2$, because increasing the number of T enlarges the effective receptive field, as shown in Table 3.

TABLE 6. Ablation studies on the Cityscapes validation set. Here, we evaluated the full networks of the ESCNet that consist of the ESC modules listed below.

Module	K	T	HFF	# Params	mIoU (%)	Speed (FPS)
ESC	4	4		**196 K**	62.2	**164.6**
			✓		61.3	153.2
	2	4		364 K	**65.5**	145.5

TABLE 7. Results on cityscapes validation set. We quantify the performance of our proposed ESCNet using global accuracy (G), class average accuracy (C), mean intersection over union (mIoU) and boundary F1 score (BF). ESCNet[1] consists of ESC modules with $K = 4$, $T = 4$ and ESCNet[2] consists of ESC modules with $K = 2$, $T = 4$.

Network	# Params	G (%)	C (%)	mIoU (%)	BF (%)
ESPNet [1]	364 K	91.8	74.8	61.4	45.93
ESCNet[1] (ours)	**196 K**	93.3	73.2	62.2	46.74
ESCNet[2] (ours)	364 K	**93.9**	**75.4**	**65.5**	**48.40**

c: RESIDUAL CONNECTION

Yu *et al.* [9] removed residual connections from some layers in the rear part of the network because it can propagate gridding artifacts caused by previous dilated convolutional layers. To verify how much the residual connection would affect the improvement of accuracy in our proposed ESC module, we evaluated ESC modules with and without residual connection. For the residual connection, we compared pairs of encoders with and without the residual connection in each encoder group with the same K and T.

As shown in Table 5, the encoders without both residual connection and HFF performed more accurately and faster than the encoders with only residual connection. Similarly, the encoders with only HFF performed more accurately and faster than the encoders with both residual connection and HFF.

d: HIERARCHICAL FEATURE FUSION (HFF)

For the HFF, we compared pairs of encoders with and without HFF in each encoder group with the same K and T. In the first encoder group, where $K = 4$ and $T = 2$, the encoder without both residual connection and HFF performed slightly more accurately and noticeably faster than the encoder with only HFF. In the second encoder group, where $K = 4$ and $T = 4$, the encoder with only HFF performed 1% more accurately than the encoder without anything, but was considerably slower. In the last encoder group, where $K = 2$ and $T = 4$, the encoder without both residual connection and HFF performed slightly more accurately and noticeably faster than the encoder with only HFF.

From the results of the above series of ablation studies with the encoder network of ESCNet, we found that the ESC module with $K = 4$ and $T = 4$ has a good trade-off between the number of parameters, accuracy, and inference speed.

TABLE 8. Network parameters, inference speed and accuracy comparison with other *state-of-the-art* semantic segmentation networks on the Cityscapes test set. Here, mIoU represents the class mIoU and each networks speed is from their original papers or the reported runtime on the Cityscapes online server. †: Titan X, ‡: Titan Xp, ††: 1080Ti, ‡‡: Tesla K80. ESCNet[1] consists of ESC modules with $K = 4$, $T = 4$ and ESCNet[2] consists of ESC modules with $K = 2$, $T = 4$.

Network	# Params	Speed (FPS)	mIoU (%)
FCN-8s [2]	134 M	-	65.3
RefineNet [4]	118 M	-	73.6
PSPNet [3]	66 M	-	**78.4**
DRN-D-105 [9]	55 M	-	75.6
DeepLab-v2 [5]	44 M	-	70.4
SegNet [22]	29.45 M	17†	57.0
ICNet [30]	7.8 M	30†	69.5
BiSeNet [15]	5.8 M	105‡	68.4
ERFNet [11]	2.07 M	42†	68.0
ContextNet [14]	0.85 M	42†	66.1
ESPNetv2 [18]	0.789 M	-	66.2
EDANet [12]	0.689 M	108‡†	67.3
CGNet [29]	0.496 M	50‡‡	64.8
ESPNet [1]	0.364 M	112†	60.3
ENet [10]	0.364 M	77†	58.3
ESCNet[1] (ours)	0.196 M	**164**‡	61.5
ESCNet[2] (ours)	0.364 M	145‡	63.4

TABLE 9. FLOPs and inference speed (FPS) comparison with other *state-of-the-art* semantic segmentation networks for the Cityscapes datset in the same environment. We measured the FLOPs for a 224 × 224 size of input. In addition, we measured the inference speed (FPS) of the networks by averaging 100 iterations for 1024 × 512 size of inputs on an NVIDIA Titan X GPU and an NVIDIA Titan Xp GPU.

Network	FLOPs	Speed (FPS)	
		Titan X	Titan Xp
PSPNet [3]	82.8 B	3.5	6.8
DeepLab-v2 [5]	37.4 B	4.3	7.5
DRN-D-105 [9]	43.3 B	3.2	6.9
RefineNet [4]	50.3 B	5.9	11.4
FCN-8s [2]	62.7 B	8.4	11.9
SegNet [22]	31.2 B	10.0	19.3
ESPNetv2 [18]	**258 M**	20.6	32.4
CGNet [29]	647 M	23.3	38.2
ERFNet [11]	2.5 B	32.0	55.3
ENet [10]	358 M	56.5	91.9
EDANet [12]	848 M	61.3	116.1
ESPNet [1]	431 M	72.3	124.1
ESCNet[1] (ours)	265 M	**97.9**	**164.6**
ESCNet[2] (ours)	431 M	85.6	145.5

Moreover, we found that the residual connection hindered the performance of the ESC module. HFF sometimes led to higher accuracy, but it always led to a slower inference speed in our experiments. Therefore, we structured full networks by attaching the light-weight decoder to several encoders that have good performances as listed in Table 5. Then, we conducted further ablation studies with these networks (Table 6).

TABLE 10. Class-wise IoU comparison on the cityscapes test set.

Network	mIoU (%)	Road	Sidewalk	Building	Wall	Fence	Pole	Traffic light	Traffic sign	Vegetation	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	Motorcycle	Bicycle
ENet [10]	58.29	96.33	74.24	85.05	32.16	33.23	43.45	34.10	44.02	88.61	61.39	90.64	65.51	38.43	90.60	36.90	50.51	48.08	38.80	55.41
ESPNet [1]	60.34	95.68	73.29	86.60	32.79	36.43	47.06	46.92	55.41	89.83	65.96	92.47	68.48	45.84	89.90	40.00	47.73	40.70	36.40	54.89
ERFNet [11]	**68.02**	97.74	80.99	89.83	42.46	47.99	56.25	59.84	65.28	91.38	68.20	94.19	76.75	57.08	92.76	50.77	60.09	51.80	47.27	61.65
ESCNet[1] **(ours)**	61.49	95.73	73.27	88.47	33.65	38.99	49.56	49.97	59.22	90.80	65.57	93.70	69.81	46.87	88.96	45.34	51.69	35.55	38.07	53.07

As shown in Table 6, in the network group that sets $K = 4$ and $T = 4$, the network without HFF performed slightly more accurately and faster than the other. The network that set $K = 2$ and $T = 4$ performed more accurately than the networks which set $K = 4$ and $T = 4$; however, it was slower and had more parameters.

From the results of these series of ablation studies, we found that the residual connection and HFF were not helpful for the proposed ESC modules. Therefore, we did not use them in our experiments.

E. COMPARATIVE RESULTS ON CITYSCAPES

In this section, we present our evaluation results on Cityscapes dataset [8]. We measured the inference speed of ours by averaging 100 iterations for 1024 × 512 size of inputs on NVIDIA Titan Xp GPU. With our ESCNet that has only 196 K parameters, we obtain 62.2 % mIoU on validation set and 61.5 % on test set with 164 FPS. In addition, with our ESCNet that has 364 K parameters, we obtain 65.5 % mIoU on validation set and 63.4 % on test set with 145 FPS.

Because our proposed ESCNet is based on the ESPNet [1] architecture, we first compared the performances of both. As shown in Table. 7, ESCNet outperformed ESPNet in terms of network parameters, global accuracy, class average accuracy, mIoU, and BF score with much fewer or similar network parameters.

Next, we compared the performance of our proposed ESCNet with *state-of-the-art* networks for semantic segmentation in terms of network parameters, mIoU and inference speed, as shown in Table 8. Here, each inference speed of other networks is from their original papers or reported runtime on the Cityscapes online server.

We can see that the networks that have enormous number of parameters achieve high accuracy (*e.g.*, PSPNet [3], RefineNet [4], DeepLab-v2 [5]). ESCNet[1] and ESCNet[2] indicate the networks that consist of ESC modules with $K = 4,\ T = 4$ and $K = 2,\ T = 4$, respectively.

Comparisons between methods that are designed for real-time semantic segmentation show that ESCNet[1] is 3.2 %, and 1.2 % more accurate than ENet [10] and ESPNet [1], respectively. Meanwhile, ESCNet[1] has 1.86 times fewer parameters than either of them. Moreover, ESCNet[1] is 2.13 and 1.46 times faster than those of ENet and ESPNet, respectively. ESCNet[2] has parameters similar to ENet [10] and ESPNet [1]; however, it outperforms them in terms of accuracy and inference speed.

TABLE 11. Category-wise IoU comparison on the Cityscapes test set.

Network	mIoU (%)	Flat	Nature	Object	Sky	Construction	Human	Vihicle
ENet [10]	80.40	97.34	88.28	46.75	90.64	85.40	65.50	88.87
ESPNet [1]	82.18	95.49	89.46	52.94	92.47	86.67	69.76	88.45
ERFNet [11]	**86.46**	98.18	91.12	62.42	94.19	90.06	77.43	91.87
ESCNet[1] **(ours)**	83.49	97.27	90.34	55.37	93.70	88.88	70.66	88.18

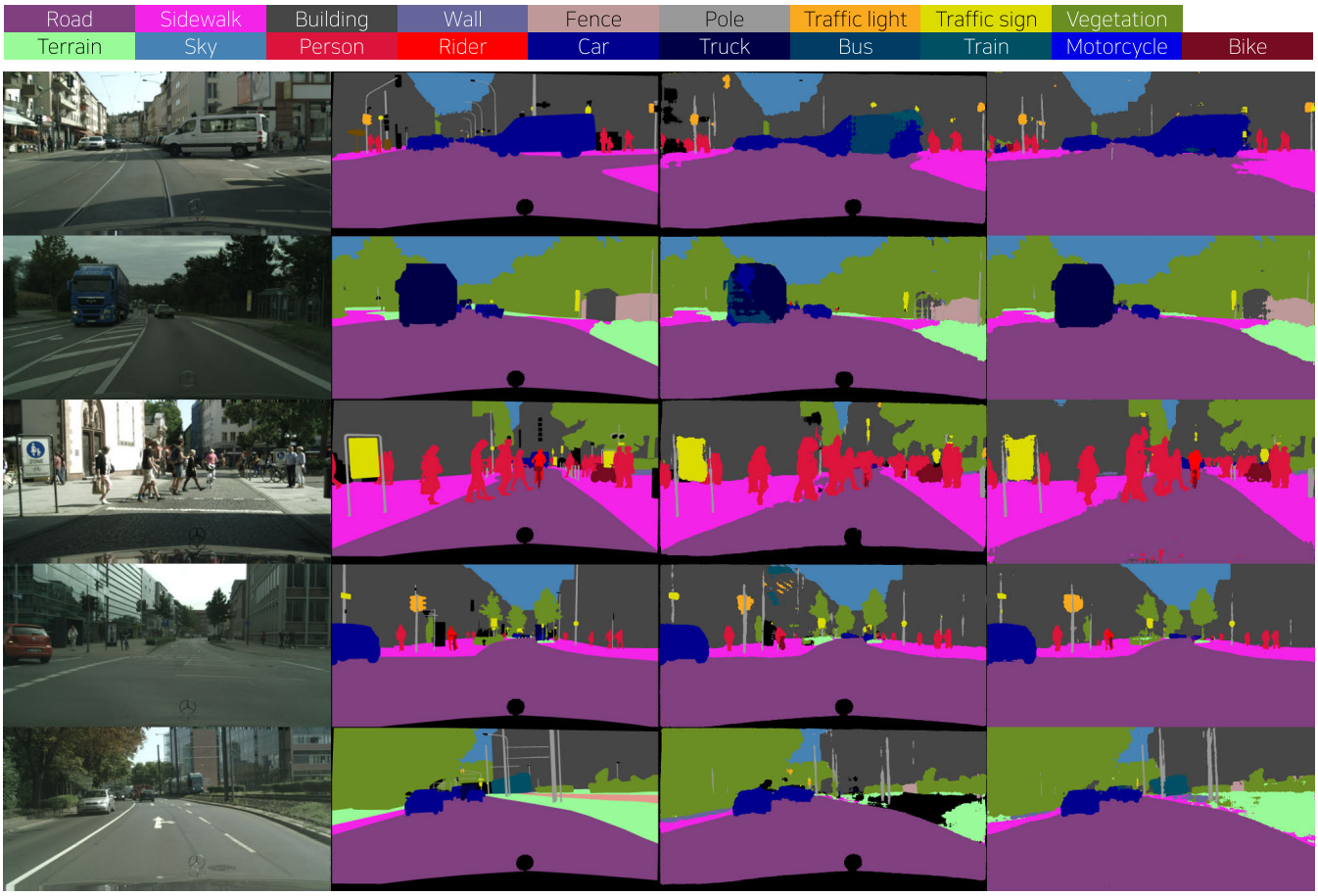
As shown in Table 9, to investigate the computational complexity, we measured FLOPs and FPS of the *state-of-the-art* networks under the same environment for fair comparisons. ESCNet[1] is faster than any other networks listed in Table 9 on both Titan X and Titan Xp GPUs. ESCNet[1] requires a computational budget of approximately 265 million FLOPs, which is 1.63 times less than that of ESPNet [1], and is remarkably faster.

Several qualitative comparison results on the Cityscapes validation set are shown in Fig. 6. The first to fourth columns in Fig. 6 represent the input image, ground truth, prediction of ESPNet [1], and prediction of ESCNet[1] (ours) respectively. As shown in Fig. 6, we can see that our results are more accurate than those of ESPNet [1]. For example, in the third row in Fig. 6, ESPNet [1] mispredicts a rider in the center of the picture as a person while ours produces accurate results.

Table 10 and Table 11 respectively show the class-wise IoU and category-wise IoU of ESCNet[1] (Ours) compared with ENet [10], ESPNet [1], and ERFNet [11] on the Cityscapes [8] test set. We can see that the overall class-wise IoU and category-wise IoU of ESCNet[1] (ours) are higher than those of ENet and ESPNet.

F. COMPARATIVE RESULTS ON CAMVID

In this section, we present our evaluation results on CamVid dataset [40]. Furthermore, we measured the inference speed of proposed ESCNet by averaging 100 iterations for 480 × 360 size of inputs on NVIDIA Titan Xp GPU. Our proposed ESCNet with 185 K parameters achieved 56.1 mIoU on CamVid test set and processes the inputs at a rate of 293.7 FPS. As shown in Table 12, we compared the performance of our ESCNet with *state-of-the-art* networks for semantic segmentation on the CamVid dataset in terms of network parameters, global accuracy, class average accuracy,



(a) Input (b) Ground Truth (c) ESPNet [1] (d) Ours

FIGURE 6. Qualitative visual results on the Cityscapes validation set with colormap for each class.

TABLE 12. Results on CamVid test set. We quantify the performance of our proposed ESCNet using global accuracy (G), class average accuracy (C), mean intersection over union (mIoU) and boundary F1 score (BF).

Network	# Params	Input size	G (%)	C (%)	mIoU (%)	BF
SegNet [22]	29.45 M	480 × 360	82.8	62.3	46.3	36.67
ENet [10]	0.364 M	480 × 360	-	68.3	51.3	-
ESPNet [1]	0.354 M	480 × 360	-	68.3	55.6	-
EDANet [12]	0.686 M	480 × 360	90.8	76.7	66.4	-
BiSeNet [15]	5.8 M	960 × 720	-	-	65.6	-
ICNet [30]	7.8 M	960 × 720	-	-	**67.1**	-
ESCNet[1] (ours)	**0.185 M**	480 × 360	87.1	70.9	56.1	37.74

mIoU and boundary F1 score (BF). We can see that our proposed ESCNet generates reasonably accurate results with much fewer network parameters than other methods.

VI. DISCUSSION

A. RESIDUAL CONNECTION

In general, identity mapping using residual connection is a good solution to improve the performance and avoid gradient vanishing in DCNN [48]. However, we noted that Yu *et al.* [9] removed the residual connections from some layers in the rear part of their network because it can propagate gridding artifacts caused by dilated convolutions. Therefore, we tried to verify how much the residual connection would affect the improvement of accuracy in our proposed ESC module.

As reported in V-D, the proposed ESC module has no residual connection because we found it was not helpful for accuracy and inference speed in our ESCNet.

In our experiment, we had to avoid the gradient vanishing problem regardless of residual connection, because our purpose was to verify how much the residual connection would affect the improvement of accuracy in our ESC module. Accordingly, we took measures to prevent the gradient vanishing problem. To prevent gradient vanishing regardless of residual connection, we adopted PReLU for activation function and initialized weights using He initialization method [44] in our experiment. Furthermore, we concatenate low-level feature maps with high-level feature maps for multiple times in our network. In the concatenation layer, the gradient values during back propagation split to their

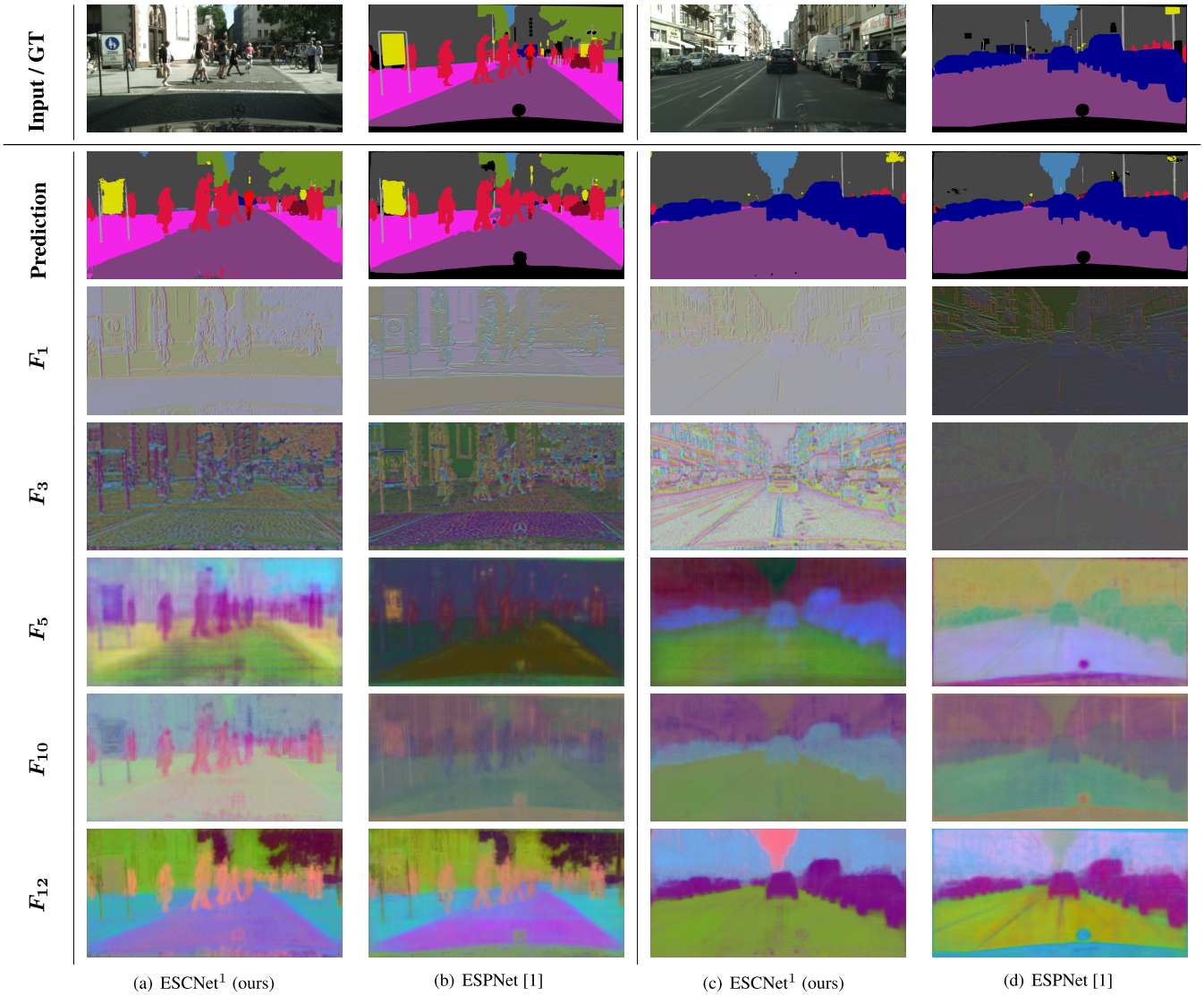


FIGURE 7. Visualization of intermediate feature maps in our propsed ESCNet and the ESPNet [1] for input images in Cityscapes validation set. To implement visualization, we compressed feature maps into three dimensions using PCA, following [49].

respective source layers. Therefore, it can prevent gradient vanishing up to a point.

Another benefit of concatenation is that it can combines spatial and context information using low-level and high-level features. The low-level features have rich spatial information while the high-level features have rich context information. We can combine both features simply by concatenating them.

B. FEATURE MAP VISUALIZATION

We compared proposed ESCNet and ESPNet [1] in feature-level by visualizing feature maps as shown in Fig. 7. To implement visualization, we compressed feature maps into three dimensions using PCA, following [49]. The first row in Fig. 7 represents the input and ground truth images. The second to seventh rows represent the prediction, F_1, F_3, F_5, F_{10} and F_{12}, respectively. The symbols of each feature map are marked in Table. 1.

As shown in Fig. 7, our proposed ESCNet generates as informative and sharp features as ESPNet with much fewer network parameters. However, we found F_5 and F_{10} generated by our ESCNet have gridding artifacts that were caused by dilated convolutions in previous layers.

Unlike the ESPNet, which exploits hierarchical feature fusion (HFF) method for degridding, we did not use this method because it was not helpful for accuracy and inference speed in our network as reported in Table. 5. Therefore, we had to deal with the gridding artifact problem in a different way.

To reduce gridding artifacts for final result, we utilized F_1 and input image by fusing them with F_{10} in decoder. As shown in Fig. 7, F_1 does not have any gridding artifact because it is extracted from an input image using only a 3×3 standard convolutional layer in our ESCNet. Moreover, we further utilized an input image to reconstruct spatial detail.

Therefore, to fuse those feature maps and input image, we first concatenated F_1 and input image with F_{10} directly. Then we convolved them twice using consecutive standard convolutional layer and transposed convolutional layer as depicted in Fig. 5 and Table. 1. Finally, we could obtain F_{12}, which has as less gridding artifacts as ESPNet, as shown in Fig. 7.

VII. CONCLUSION

In this paper, we proposed a novel module named ESC, which takes advantage of the proposed SCDC to efficiently use sparse kernels with different effective receptive fields, while simultaneously reducing the computational cost. Based on the ESC module, we further proposed ESCNet, an extremely efficient network for semantic segmentation. The experimental results showed its efficiency, which had a good trade-off between accuracy and computational cost for scene understanding. Our proposed ESCNet is reasonably accurate, light, and fast.

REFERENCES

[1] S. Mehta, M. Rastegari, A. Caspi, L. Shapiro, and H. Hajishirzi, "ESPNet: Efficient spatial pyramid of dilated convolutions for semantic segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 552–568.

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

[3] 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. 6230–6239.

[4] G. Lin, A. Milan, C. Shen, and I. Reid, "RefineNet: Multi-path refinement networks for high-resolution semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5168–5177.

[5] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "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, Apr. 2017.

[6] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017, *arXiv:1706.05587*. [Online]. Available: https://arxiv.org/abs/1706.05587

[7] 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)*, Sep. 2018, pp. 833–851.

[8] 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.

[9] F. Yu, V. Koltun, and T. Funkhouser, "Dilated residual networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 636–644.

[10] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "ENet: A deep neural network architecture for real-time semantic segmentation," 2016, *arXiv:1606.02147*. [Online]. Available: https://arxiv.org/abs/1606.02147

[11] E. Romera, J. M. Álvarez, L. M. Bergasa, and R. Arroyo, "ERFNet: Efficient residual factorized ConvNet for real-time semantic segmentation," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 1, pp. 263–272, Jan. 2018.

[12] S.-Y. Lo, H.-M. Hang, S.-W. Chan, and J.-J. Lin, "Efficient dense modules of asymmetric convolution for real-time semantic segmentation," 2018, *arXiv:1809.06323*. [Online]. Available: http://arxiv.org/abs/1809.06323

[13] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.

[14] R. P. K. Poudel, U. Bonde, S. Liwicki, and C. Zach, "ContextNet: Exploring context and detail for semantic segmentation in real-time," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, Sep. 2018, p. 146. [Online]. Available: http://bmvc2018.org/contents/papers/0286.pdf

[15] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "BiSeNet: Bilateral segmentation network for real-time semantic segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 325–341.

[16] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1800–1807.

[17] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*. [Online]. Available: https://arxiv.org/abs/1704.04861

[18] S. Mehta, M. Rastegari, L. Shapiro, and H. Hajishirzi, "ESPNetv2: A light-weight, power efficient, and general purpose convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9190–9200.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.

[20] W. Wei, B. Zhou, D. Połap, and M. Woźniak, "A regional adaptive variational PDE model for computed tomography image reconstruction," *Pattern Recognit.*, vol. 92, pp. 64–81, Aug. 2019.

[21] M. Woźniak and D. Połap, "Object detection and recognition via clustered features," *Neurocomputing*, vol. 320, pp. 76–84, Dec. 2018.

[22] 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, Dec. 2017.

[23] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent. (MICCAI)*, 2015, pp. 234–241.

[24] 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.

[25] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, May 2016, pp. 1–13. [Online]. Available: http://arxiv.org/abs/1511.07122

[26] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, May 2015, pp. 1–14. [Online]. Available: http://arxiv.org/abs/1412.7062

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[28] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5987–5995.

[29] T. Wu, S. Tang, R. Zhang, and Y. Zhang, "CGNet: A light-weight context guided network for semantic segmentation," 2018, *arXiv:1811.08201*. [Online]. Available: https://arxiv.org/abs/1811.08201

[30] 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.

[31] Y. Gu, Z. Zhong, S. Wu, and Y. Xu, "Enlarging effective receptive field of convolutional neural networks for better semantic segmentation," in *Proc. 4th IAPR Asian Conf. Pattern Recognit. (ACPR)*, Nov. 2017, pp. 388–393.

[32] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 6848–6856.

[33] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "ShuffleNet V2: Practical guidelines for efficient CNN architecture design," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2018, pp. 122–138.

[34] T. Zhang, G.-J. Qi, B. Xiao, and J. Wang, "Interleaved Group Convolutions," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4383–4392.

[35] G. Xie, J. Wang, T. Zhang, J. Lai, R. Hong, and G.-J. Qi, "Interleaved structured sparse convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 8847–8856.

[36] K. Sun, M. Li, D. Liu, and J. Wang, "IGCV3: Interleaved low-rank group convolutions for efficient deep neural networks," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, Sep. 2018, p. 101. [Online]. Available: http://bmvc2018.org/contents/papers/0330.pdf

[37] C.-F. Chen, J. Oh, Q. Fan, and M. Pistoia, "SC-Conv: Sparse-complementary convolution for efficient model utilization on CNNs," in *Proc. IEEE Int. Symp. Multimedia (ISM)*, Dec. 2018, pp. 97–100.

[38] R. Zhang, S. Tang, Y. Zhang, J. Li, and S. Yan, "Scale-adaptive convolutions for scene parsing," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2050–2058.
[39] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 4510–4520.
[40] G. J. Brostow, J. Fauqueur, and R. Cipolla, "Semantic object classes in video: A high-definition ground truth database," *Pattern Recognit. Lett.*, vol. 30, no. 2, pp. 88–97, 2009.
[41] P. Sturgess, K. Alahari, L. Ladicky, and P. H. S. Torr, "Combining appearance and structure from motion features for road scene understanding," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, Sep. 2009, pp. 1–11. doi: 10.5244/C.23.62.
[42] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.
[43] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, May 2015, pp. 1–15. [Online]. Available: http://arxiv.org/abs/1412.6980
[44] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.
[45] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*. [Online]. Available: http://arxiv.org/abs/1502.03167
[46] W. Liu, A. Rabinovich, and A. C. Berg, "ParseNet: Looking wider to see better," 2015, *arXiv:1506.04579*. [Online]. Available: http://arxiv.org/abs/1506.04579
[47] G. Csurka, D. Larlus, F. Perronnin, and F. Meylan, "What is a good evaluation measure for semantic segmentation?" in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, Sep. 2013, pp. 32.1–32.11. doi: 10.5244/C.27.32.
[48] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[49] L. Fan, W.-C. Wang, F. Zha, and J. Yan, "Exploring new backbone and attention module for semantic segmentation in street scenes," *IEEE Access*, vol. 6, pp. 71566–71580, 2018.

JAESEON KIM was born in Ansan, Gyeonggi, South Korea, in 1993. He received the B.S. degree in electrical and computer engineering from Ajou University, Suwon, South Korea, in 2018, where he is currently pursuing the M.S. degree in electrical and computer engineering.

His research interests include computer vision, deep learning, and semantic segmentation.

YONG SEOK HEO received the B.S. degree in electrical engineering and the M.S. and Ph.D. degrees in electrical engineering and computer science from Seoul National University, South Korea, in 2005, 2007, and 2012, respectively.

From 2012 to 2014, he was with the Digital Media and Communications Research and Development Center, Samsung Electronics. He is currently with the Department of Electrical and Computer Engineering, Ajou University, as an Associate Professor. His research interests include segmentation, stereo matching, 3-D reconstruction, and computational photography.