Article

Semantic Segmentation by Multi-Scale Feature Extraction Based on Grouped Dilated Convolution Module

Dong Seop Kim , Yu Hwan Kim and Kang Ryoung Park *

Division of Electronics and Electrical Engineering, Dongguk University, 30 Pildong-ro, 1-gil, Jung-gu, Seoul 04620, Korea; k_ds1028@naver.com (D.S.K.); taekkuon@dongguk.edu (Y.H.K.)

* Correspondence: parkgr@dgu.edu

Abstract: Existing studies have shown that effective extraction of multi-scale information is a crucial factor directly related to the increase in performance of semantic segmentation. Accordingly, various methods for extracting multi-scale information have been developed. However, these methods face problems in that they require additional calculations and vast computing resources. To address these problems, this study proposes a grouped dilated convolution module that combines existing grouped convolutions and atrous spatial pyramid pooling techniques. The proposed method can learn multi-scale features more simply and effectively than existing methods. Because each convolution group has different dilations in the proposed model, they have receptive fields of different sizes and can learn features corresponding to these receptive fields. As a result, multi-scale context can be easily extracted. Moreover, optimal hyper-parameters are obtained from an in-depth analysis, and excellent segmentation performance is derived. To evaluate the proposed method, open databases of the Cambridge Driving Labeled Video Database (CamVid) and the Stanford Background Dataset (SBD) are utilized. The experimental results indicate that the proposed method shows a mean intersection over union of 73.15% based on the CamVid dataset and 72.81% based on the SBD, thereby exhibiting excellent performance compared to other state-of-the-art methods.

Keywords: semantic segmentation; pixel-level classification; grouped dilated convolution module; multi-scale context

Citation: Kim, D.S.; Kim, Y.H.; Park, K.R. Semantic Segmentation by Multi-Scale Feature Extraction Based on Grouped Dilated Convolution Module. *Mathematics* **2021**, *9*, 947. <https://doi.org/10.3390/math9090947>

Academic Editors: Akemi Galvez Tomida and Andres Iglesias Prieto

Received: 12 March 2021

Accepted: 21 April 2021

Published: 23 April 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Various computer-vision tasks, such as object detection and image classification, have been examined by researchers. Among these, semantic segmentation is a significantly challenging task that requires pixel-level classification not only of landscape elements (e.g., sky, roads, and buildings) but also numerous objects (e.g., pedestrians and bicyclists), as shown in Figure 1. This can also be applied to other tasks, such as autonomous cars, closed-circuit televisions, security applications, and medical imaging. Given such potential, it has been actively examined across many fields. Advancements in deep neural networks have led to a significant increase in performance for these computer-vision tasks. In particular, fully convolutional networks (FCNs) [1] have been effectively utilized to redesign existing classification models for semantic segmentation, and recent studies have been carried out with a focus on methods that use FCNs. Initial research on semantic segmentation has mainly focused on the utilization of FCNs which have an encoder–decoder structure, represented by SegNet [2], U-Net [3], and DeepLab-LargeFov [4]. However, problems of semantic segmentation have not been solved with the application only of a simple encoder–decoder structure. This difficulty is based on the following issues. First, semantic segmentation requires an accurate detection of multi-scale objects.

Figure 1. Examples of semantic segmentation applying pixel-wise classification, with the left column showing input images and the right column showing ground truth in which challenging factors, such as various classes and multi-scale objects, exist.

For example, a car class in a road-scene database includes vehicles of different sizes according to distance. Moreover, road and sky classes occupy large areas, whereas the pedestrian class occupies a small one. In this regard, it is crucial to accurately detect such multi-scale objects in images. Second, because full understanding of images is required in semantic segmentation, the spatial context of images should be correctly identified. In other words, relations and patterns among objects should be precisely analyzed. For example, the car class tends to be located above the road class and far away from the sky class. Additionally, pedestrians are likely to be found on sidewalks. A crucial key point of semantic segmentation is detecting these relations and patterns. Third, semantic segmentation is interrupted by mislabeling, which occurs during the process of creating pixel-unit ground truth [5]. Utilization of FCNs with the aforementioned simple encoder–decoder structure accompanies problems related to such requirements for semantic segmentation. Thus, recent studies have been conducted to overcome these problems. Previous studies [6–10] identified multi-scale objects by adjusting input images of different sizes during the learning process. Specifically, larger or smaller images than the original ones were used. A new convolution technique was also utilized to solve these problems in a more sophisticated manner. Dilated (i.e., atrous) convolutions can increase receptive fields without the loss of resolution in feature maps. Thus, they can be applied to semantic segmentation. Other previous studies [4,11–14] effectively analyzed multi-scale objects and the spatial context of images by appropriately using dilated convolutions. More recently, various advanced methods, such as spatial pyramid pooling (SPP) and attention mechanisms, have been examined. In particular, recent studies have focused on the fusion of dilated convolutions and SPP techniques for the aggregation of major spatial features [15–21]. In natural language processing (NLP), attention has been given to applying weight to important words. This technique also adds weight to the major spatial and channel contexts in convolutional neural networks (CNNs), computer-vision tasks, and semantic segmentation applications [22–26]. Detailed analyses and comparisons of these studies are discussed in the next section.

To address the aforementioned problems of semantic segmentation, this study proposes a grouped dilated convolution module (GDCM). Inspired by the fusion of dilated convolutions and atrous spatial pyramid pooling (ASPP), this new convolution module can

be used to facilitate the effective learning of multi-scale features. Compared with previous works, this study is novel in the following four ways.

- The new GDCM developed in this study can robustly segment objects of different sizes and environments. Convolutions are classified into groups with different dilated parameters, and each group trains convolution filters that show high correlations with multi-scale features in different receptive fields.
- This module can learn multi-scale features more effectively by using fewer parameters than existing methods.
- A highly applicable method is provided to replace existing convolution blocks. Moreover, it can derive high semantic segmentation performance without additional modules, such as attention mechanisms.
- Our trained model and algorithm, with instructions for use, are being shared as shown in [27] in order to allow other researchers to conduct fair performance evaluations of the developed methods.

2. Related Work

In this section, existing semantic segmentation methods are divided into four types, as indicated below, and are discussed in terms of multi-scale objects and class imbalances.

2.1. Multi-Scale Input-Based Method

In semantic segmentation, classification becomes challenging because of multi-scale objects. Several methods have been developed to overcome this issue. Farabet et al. [6] divided input images into multi-scale objects based on a Laplacian pyramid for the learning process. Mostajabi et al. [7] obtained 14 sub-images based on super pixels from input images and used them as input data to a model. Chen et al. [8] used three images of different sizes as input data to models and combined the images for additional applications. The models were consistent with each other, and weights were shared. FeatureMap-Net [9] is similar to the aforementioned methods, but its only difference is that it uses convolution blocks with different weights. Dai et al. [10] developed a segmentation method based on a bounding box obtained by selective searching among region proposal strategies.

Although these methods were developed to control multi-scale objects, they have problems with the reduction of training speed, owing to several forwarding processes and the application of scale at a fixed ratio during the model training.

2.2. Atrous Convolution-Based Method

Atrous convolutions [4] can be used to effectively increase receptive fields without the loss of resolution of feature maps. They can also significantly increase effective receptive fields (ERFs) [12,28]. For this reason, they have been actively applied in semantic segmentation applications. These are also called dilated convolutions [11]. DeepLab-LargeFOV [4] applied atrous convolutions to the input of the last convolution layer to reduce loss of resolution. Yu et al. [11] presented a context module that applied dilated convolutions. In this module, a feature map obtained from inputs is used, and dilated convolutions are sequentially calculated. Through this process, large receptive fields are ensured without loss of resolution. Liu et al. [12] and Hamaguchi et al. [13] proposed segmentation models that exhibited shallow but excellent performance by intensively analyzing the relationship between ERFs and dilated convolutions in semantic segmentation tasks. Wang et al. [14] presented a method of using dilated convolutions in parallel to reduce the gridding effects known to be a problem of these convolutions. As described, dilated convolutions can increase receptive fields without loss of resolution while significantly expanding ERFs. However, they have limitations in that they generate gridding artifacts that cause lattice patterns on output images [29], and they show insufficient performance in semantic segmentation, which requires full understanding of the images.

2.3. Spatial Pyramid Pooling-Based Method

Semantic segmentation becomes challenging owing to multi-scale objects. To address this problem, SPP-based methods have been introduced, which are different from methods based on multi-scale input images [15–18]. These methods are distinguished from multi-scale input-based methods as explained in Section 2.1 in that they are operated at the feature-map level derived from a sufficiently trained model, instead of from the input-image level. Some SPP-based methods apply pooling at different ratios by using dilated convolutions. This technique is the ASPP [15,16,18]. A PSPNet [5] applies pooling at different ratios to feature maps obtained from a backbone model and combines them to output a prediction map that shows more precise and robust performance for managing multi-scale objects. Unlike the PSPNet, DeepLab [15,16,21] applies dilated convolutions to pooling at different ratios. Each pooling layer independently learns weights for pooling. Based on the aforementioned studies, a number of intensive SPP research projects have been carried out [18–20]. Although SPP-based methods are effective in handling multi-scale objects, they tend to focus on the last feature map, which has already lost a great amount of spatial information. The proposed method in this study is distinguished from these SPP-based methods in that it focuses on the entire feature map.

2.4. Attention-Based Method

Attention-based methods have been examined to identify relations of words located far from each other, while adding weights to them in NLP [30]. Recently, this attention mechanism has been actively applied to NLP and computer-vision tasks [31–35]. Wang et al. [33] developed a method that outputs a weight map for spatial context by replacing self-attention-based with convolution-based calculations to solve the long-range dependency problem. Unlike these researchers who adopted the attention mechanism to manage spatial information, a few researchers [34,35] have presented an attention module that places weights to channels by combining feature maps. As demonstrated, numerous studies have analyzed applications of attention mechanisms to convolutional feature maps for computer-vision tasks. Such applications have also been actively investigated for semantic segmentation. Similar to the method [33] developed by Wang et al., Zhang et al. [22] used a spatial attention module that was designed to derive weights for the spatial context to generate segmentation prediction maps. The CCNet [25] sequentially connected spatial attention modules to increase segmentation performance, and the DFANet [24] applied channel attention to feature maps. Research on integrating information on both spatial and channel attention modules was also conducted. A dual attention network (DANet) [23] was developed through the application of both spatial and channel attention modules. The output of each module was fused based on the sum rule. Zhu et al. [26] developed a method that combined both spatial and channel attention by implementing spatial pyramid pooling in an attention module, unlike the DANet, which integrated spatial and channel attention in parallel. As mentioned, various studies have analyzed the application of an attention mechanism. However, it has problems, such as its large number of additional calculations and reduced processing speed.

To address these problems, this study proposes the GDCM, which can effectively learn multi-scale features by applying both dilated convolutions and ASPP. Table 1 compares the advantages and disadvantages of the proposed method.

Table 1. Summarized comparisons of the proposed and previous works on semantic segmentation.

Category	Advantage	Disadvantage
Multi-scale input-based [6–10]	Multi-scale information can be easily learned through the application of multi-scale inputs	A great amount of training and inference time is required, owing to several forwarding processes. Training is carried out based on a scale at a fixed ratio
Atrous convolution-based [4,11–14]	Spatial information can be learned without loss of resolution	Limited performance and gridding artifacts are observed
Spatial pyramid pooling-based [15–21]	Spatial information can be learned more precisely through the application of dilated convolutions at different scales in the form of a pyramid to the feature map	Only the last feature map, which loses a great amount of spatial information, is focused. A great deal of calculation and large-capacity hardware memory are required
Attention-based [22–26]	Main features can be trained through the calculation of weights applied to spatial or channel contexts	A great deal of calculation and large-capacity hardware memory are required. Processing is performed slowly
Proposed method	Spatial information can be learned with fewer parameters. Performance of state-of-the-art methods can be delivered	Grouped convolution requires large memory for training, which reduces the batch size and consequently increases the training time

3. Proposed Method

It is essential to manage multi-scale information for semantic segmentation. That is, spatial information should be effectively extracted from feature maps. In this regard, the proposed method applies various filter groups that have different receptive fields in the convolution blocks. Because each group has different-sized views, they independently learn and aggregate multi-scale information to facilitate the learning of a global context based on feature maps. The next section presents the background for the design of the proposed model. Subsequently, it describes a method that can extract useful contextual and multi-scale information from the input image.

3.1. Grouped Convolution

Figure 2a presents an example of grouped convolutions. Unlike conventional convolutions, which consider the entire depth of feature-map input, grouped convolutions are classified according to group (G) parameters and input feature-map channels [36,37]. It is assumed that G is two in the example of Figure 2a. If so, input feature maps obtained from the calculation of height (H) $\times$ width (W) $\times$ channel of input feature map (C_{in}) are distributed into each group derived from the calculation of $H \times W \times (C_{in})/G$. Subsequently, a convolution calculation is performed for each group. When the kernel size is assumed to be 3×3 in this example, a filter group is calculated based on $3 \times 3 \times (C_{in})/G \times (\text{channel of output feature map } (C_{out}))$. Thus, the number of parameters required decreases by G in the grouped convolution calculation compared with that of a conventional convolution calculation. Moreover, the application of grouped convolutions is advantageous in that each filter group can learn weights having high correlation to the corresponding receptive fields. A previous study [37] verified the enhanced performance of a model based on the application of grouped convolutions through several experiments. Considering the advantages of this grouped convolution application, this study proposes a method of combining grouped convolution calculations with SPP.

3.2. Spatial Pyramid Pooling

The SPP technique applies filters with receptive fields of different sizes to objects of different sizes in the feature maps to achieve robust performance. This technique has long been utilized in diverse vision tasks. In the field of semantic segmentation, methods such

as DeepLab [15,16,21] and PSPNet [5] applied convolutions of different sizes in parallel with extracted feature maps to train spatial information. In particular, the ASPP technique uses dilated (atrous) convolutions, instead of convolution filters of different sizes, as shown in Figure 2b. Dilated convolutions have the same number of parameters as those of the same size. However, the former convolutions have larger receptive fields than the latter. For example, a 3×3 convolution filter has nine parameters and a 3×3 receptive field. On the other hand, a 3×3 convolution filter having a dilation of two has nine parameters and a 5×5 receptive field. SPP layers are applied in parallel to feature maps extracted by the CNNs. Each feature map obtained through the application of SPP layers is concatenated and transferred to the classification layer. However, the output of the last layer tends to include a large number of nodes (e.g., 2048 or 4096), and each filter is combined in parallel for the calculation. For this reason, the SPP technique requires high costs for calculation and memory. To reduce these costs, the GDCM provides the advantages of grouped convolutions and spatial pyramid pooling techniques, which are discussed in the following section.

Figure 2. Examples of each method: (a) grouped convolutions when G is two; (b) atrous pyramid pooling based on dilated convolutions among spatial pyramid pooling techniques; (c) structure of the proposed GDCM. * means convolution operation.

3.3. GDCM

This study develops the GDCM based on the assumption that the implementation of multi-scale context can serve as a key point for addressing the problems of semantic segmentation. Several existing studies have verified that the aforementioned ASPP technique increases the performance of different models. However, it requires a high calculation cost and a large-capacity memory. Thus, ASPP is applied to the grouped convolutions, which leads to excellent efficiency, owing to fewer parameters and smaller calculations. As shown in Figures 2c and 3, different dilations are applied to each grouped convolution using the proposed method. Specifically, G is a set of 32 in a grouped convolution, which is divided

into four subgroups. Each subgroup performs calculations based on convolution filters applying different dilations. Subsequently, the outputs of each group are delivered via concatenation. Each group learns corresponding receptive fields based on the effects of the calculation. Because aggregated feature maps are the sums of differently sized trained features, the proposed module can learn a multi-scale context and reflects the advantages of the grouped convolution technique instead of focusing on the last feature map obtained from a backbone model, similar to existing ASPP-based methods. For this reason, it can perform calculations in each convolution layer and does not require the cost generated by additional calculations and a large-capacity memory. Furthermore, the proposed module adopts the advantages of ASPP, which uses convolutions and applies different dilation parameters to the feature maps. These advantages enable the proposed module to train multi-scale information more conveniently.

Figure 3. Proposed GDCM in which all 32 grouped convolutions are applied in a convolution layer. All are divided into four subgroups that apply different dilation convolutions. The value of a dilation parameter is established to be 1, 2, 3 or 4 according to the group, and the final output is derived via concatenation.

4. Experimental Results

This section provides quantitative and qualitative experimental results from using the proposed method. Two open databases (i.e., Cambridge-driving Labeled Video Database (CamVid) [38] and the Stanford Background Dataset (SBD) [39]) were used to perform fair experiments. Each database is described in detail in the following sub-section and in Table 2.

Table 2. Descriptions of experimental datasets (Train, Val, and Test refer to training, validation, and testing images, respectively).

Dataset	Size (Pixels)	Number of Classes	Number of Images			
			Train	Val	Test	Total
CamVid	960 × 720	11	367	101	233	701
SBD	320 × 240	8	625	-	90	715

4.1. Experimental Datasets

As shown in Table 2, the CamVid dataset is a road-scene database that has 11 classes: cars, pedestrians, roads, side-walks, sky, trees, buildings, sign symbols, fences, bicyclists, and column poles. The targets of these classes can be easily found on roads. A void class also exists in this database, which cannot be identified and is not involved in learning or inferencing. Moreover, an experiment based on the SBD consisting of road scenes and various environmental elements was performed to verify the robust performance of the proposed method. The SBD comprised 715 images obtained from various open datasets (e.g., LabelMe, MSRC, Pascal VOC, and Geometric Context). The Pascal VOC dataset was initially considered for the experiment. However, it had problems in that the number of background classes was significantly greater and different objects were classified with background classes. Owing to these problems, this dataset was evaluated as inappropriate for semantic segmentation, requiring it to fully understand images. Therefore, the SBD was finally selected. Because the SBD consists of images in various environments, it has eight classes: roads, sky, water, trees, grass, buildings, mountains, and foreground. Moreover, the foreground class is regarded as a dataset that is unlikely to be segmented, because it includes various sub-classes of cars, humans, animals, and other objects. Both datasets are publicly available and provide for fair experiments and evaluation. The number and sizes of data used for learning and testing vary according to the datasets. Details of such data are described in the following section. Figure 4 shows image examples of the CamVid dataset and the SBD.

Figure 4. Examples of experimental databases: (a,b) CamVid; (c,d) SBD; In (a–d), the left and right images show the input and ground-truth images, respectively.

4.2. Training of the Proposed Model

The proposed model performed training using the train-from-scratch approach. All experiments were conducted fairly in the same training environment. The number of training epochs was 700, and the base learning rate was set to 0.01. Because a pretrained model was not used in accordance with the learning-rate policy, a learning-rate warm-up [40] was implemented to facilitate smooth learning. This method performs model warm-ups in advance of full-scale learning by considering the difficulty of learning in the initial stage and by applying a learning rate that gradually increases based on the difficulty level. A previous study [40] verified the effectiveness of this method. The number of epochs for warm-up was set to 50, and the base learning rate was designed to gradually increase from 0 to 0.01. The optimal epochs of 50 and 0 to 0.01 as base learning rate were experimentally determined with training data by trial and error. Experimental results showed that the different number of epochs or different values for the increase in base learning rate caused the failure of convergence of training loss to small value. Subsequently, according to a previous study [40], the learning rate was scheduled in the form of a “poly” Equation (1), where lr indicates the learning rate, power is 0.9, max_iter is equal to total number of iterations $\times$ the number of epochs, and $current_iter$ indicates the current iteration number.

$$lr = base_lr \cdot \left(1 - \frac{current_iter}{max_iter}\right)^{power} \quad (1)$$

For an optimizer, adaptive moment estimation [41] was used. A cross-entropy function was utilized as a loss function. The batch size was established as four for the CamVid dataset and eight for the SBD. The optimal batch sizes of four and eight were also experimentally determined with training data by trial and error. Experimental results showed that the different number of batch sizes caused the failure of convergence of training loss to a small value.

For augmentation of training data, this study applied random cropping and left–right random flipping with a probability of 50%. When random cropping was performed, each input image was randomly resized in a range between 0.8 and 1.5 times. Thereafter, the images were cropped to 512×512 . Flipping was applied randomly at a probability of 50%. The input size applied in the learning process of the model was established as 960×720 for the CamVid dataset and 512×512 for the SBD. In all experiments, standardization of the input data was conducted under the conditions that the average is zero and variance is one. This standardization assumes that the data is Gaussian distributed. All the state-of-the-art methods compared in our experiments used this standardization method for input data. Therefore, we used this same method for fair comparisons with the state-of-the-art methods. After the standardization process, the models began data learning. Furthermore, weights according to distribution of classes were added to the loss function during the learning process in consideration of a large data distribution for semantic segmentation [2]. Figure 5 shows the convergences of training loss graphs, which confirms that our model was successfully trained with the training data.

The proposed method was implemented on PyTorch (Facebook, Redwood City, CA, USA) [42]. Training and testing were performed on a desktop computer with an Intel® Core™ i7-6700 (Intel Corp., Santa Clara, CA, USA) central processing unit (CPU) at 3.47 GHz with 12-GB memory and two NVIDIA GeForce GTX 1070 (NVIDIA Corp., Santa Clara, CA, USA) (1920 compute unified device architecture (CUDA) cores and 8 GB memory) graphics processing units (GPUs) [43].

Figure 5. Training loss graphs.

4.3. Testing with CamVid

4.3.1. Ablation Studies

Regarding the matrices used for evaluation, pixel (global) accuracy, class (mean) accuracy, and mean intersection over union (mIoU) were used in accordance with conditions stated in previous studies [1,44]. Equations (2)–(4) present the detailed calculation. C refers to the number of classes, and TP, FP, and FN denote true positive, false positive, and false negative, respectively. TP, FP, and FN indicate that positive data was correctly predicted as positive data, negative data was incorrectly predicted as positive data, and positive data was incorrectly predicted as negative data, respectively. Pixel (global) accuracy in Equation (2) refers to the ratio of pixels of all classes predicted correctly. Class (mean) accuracy in Equation (3) refers to the average of the ratios of TP correctly predicted compared with pixels included in corresponding classes according to each class. Further, mIoU (i.e., the Jaccard index) refers to the average of the intersection of the entire union for each class, as expressed in Equation (4).

$$\text{pixel acc} = \frac{\sum_{i=1}^C TP_i}{\sum_{i=1}^C (TP_i + FP_i)}, \quad (2)$$

$$\text{class acc} = \frac{1}{C} \sum_{i=1}^C \frac{TP_i}{TP_i + FP_i}, \quad (3)$$

$$\text{mIoU} = \frac{1}{C} \sum_{i=1}^C \frac{TP_i}{TP_i + FP_i + FN_i}. \quad (4)$$

This study closely followed the scheme revealed in a previous study [45] in order to perform fair experiments. In terms of inference, images at the original size of 960×720 were used as input data. Moreover, various ablation studies were conducted to experimentally verify accurate parameter inference of the proposed model in this study. In this regard, two versions of GDCM-S (shallow) using two dilated groups and GDCM-W (wide) using four dilated groups were applied in experiments. These modules were classified into S (small) and L (large) according to the dilation parameters. Table 3 compares each group according to those tested, and according to each dilation parameter, S (small) and L (large) divided experiment. Table 3 shows the comparison of the number of dilated groups and dilation parameters of each group according to methods. GDCM-SS is a shallow module that uses two dilated groups that apply small dilations. GDCM-WL uses four dilated groups that apply large dilations (i.e., 1, 2, 4 and 8). The optimal numbers of groups (G) and subgroups

were experimentally determined with the training data, with which best accuracies of semantic segmentation were obtained.

Table 3. Comparison of the number of dilated groups and dilation parameters of each group per method. # means “the number”.

Method	# of Dilated Group	Dilation Parameter of Each Group
GDCM-SS	2	(1, 2)
GDCM-SL		(1, 4)
GDCM-WS	4	(1, 2, 3, 4)
GDCM-WL		(1, 2, 4, 8)

Ablation studies were conducted under these conditions. As shown in Table 4, GDCM-WS showed the highest segmentation accuracy. Segmentation accuracy is also compared according to the model depths of GDCM-WS and GDCM-SS, which showed the first and second rank performances, respectively. In addition, we compared the testing accuracies according to the various numbers of groups and subgroups. As shown in Table 4, GDCM-WS (including $G = 32$ and subgroups = 4) show the highest accuracies. As shown in Table 5, segmentation accuracy was higher with (4, 4, 6 and 6) repetitions of each block than when with (3, 3, 5 and 5). Nonetheless, the number of model parameters also increased.

Table 4. Comparison of accuracy per method (unit: %).

Method (#Groups, #Subgroups)	Pixel Acc	Class Acc	mIoU
GDCM-SS (32, 2)	93.24	80.94	72.93
GDCM-SS (16, 2)	92.45	80.13	71.76
GDCM-SL (32, 2)	92.87	80.11	71.68
GDCM-SL (16, 2)	91.98	79.94	70.97
GDCM-WS (32, 4)	94.48	81.62	73.15
GDCM-WS (16, 4)	93.14	79.89	71.33
GDCM-WL (32, 4)	93.64	80.25	71.92
GDCM-WL (16, 4)	91.17	79.12	70.63
Com 1	90.26	78.73	69.78
Com 2	91.17	79.14	70.14

Table 5. Comparison of accuracy by model depth per method.

Method	Repetitions of Each Block	mIoU (%)	# of Model Parameters
GDCM-SS	(3, 3, 5, 5)	72.93	13.3 M
	(4, 4, 6, 6)	73.09	15.6 M
GDCM-WS	(3, 3, 5, 5)	73.15	13.3 M
	(4, 4, 6, 6)	73.35	15.6 M
Com 1		69.78	17.5 M
Com 2		70.14	18.3 M

Moreover, we compared the accuracies and number of model parameters in our method with those by other combinations such as Com 1 (combination of dilated convolution and attention-based method) and Com 2 (combination of dilated convolution, ASPP, and attention-based method). As shown in Tables 4 and 5, the proposed method shows better accuracy with fewer number of model parameters than other combination methods.

4.3.2. Comparisons with State-of-the-Art Methods

Segmentation accuracies of the proposed method and the state-of-the-art methods were compared. We accessed some trained models in Table 6, and if the model was inacces-

sible, we implemented the models based on their paper. In any case, we performed training with our training data and executed testing with our testing data for fair comparisons. Therefore, identical testing data and resolution was used by each of these methods. As shown in Table 6, it was verified that the GDCM-based method proposed in this study showed higher accuracy than state-of-the-art methods.

Table 6. Comparisons of the proposed method with state-of-the-art methods (unit: %).

Method	Pixel Acc	Class Acc	mIoU
Souly et al. [46]	87	72.4	58.2
SegNet [2]	90.4	71.2	60.1
Yu et al. [11]	-	-	65.3
Jégou et al. [47]	91.5	-	66.9
ICNet [48]	-	-	67.1
BiSeNet [49]	-	-	68.7
Yang et al. [50]	89.79	-	69.94
ESSN [51]	92.74	79.66	71.67
Proposed method	94.48	81.62	73.15

4.4. Testing with SBD

4.4.1. Ablation Studies

Unlike images in the CamVid dataset, not all images in the SBD have the same size. However, their average size was 320×240 . Moreover, training and test sets were not separated in the SBD. In this regard, an eightfold cross-validation was conducted to ensure fair experiments, in which $\sim 7/8$ of the entire dataset was used for training and $\sim 1/8$ was used for testing. In this cross-validation, eight types of training and testing sets can be obtained. Given this condition, the mean of the entire experimental results obtained for eight iterations based on the SBD was applied. Data augmentation applied random left and right flips, random crops, and random scales, similar to the CamVid dataset tested earlier. Regarding the random flip, a probability of 50% was applied. Each image was resized to 640×640 and scaled randomly between 0.8 and 1.2 times. Subsequently, random cropping was performed based on the 512×512 size. During the testing process, images were resized to 512×512 for experimentation.

Ablation studies were carried out by applying the same method used in the CamVid database. The optimal numbers of groups (G) and subgroups were experimentally determined with training data, with which best accuracies of semantic segmentation were obtained. It was found that GDCM-WS showed the highest segmentation accuracy, as indicated in Table 7, which compares segmentation accuracy by model depth according to GDCM-WS and GDCM-SS, which exhibited the first and second rank performances, respectively. In addition, we compared the testing accuracies according to the various numbers of groups and subgroups. As shown in Table 7, GDCM-WS (including $G = 32$ and subgroups = 4) show the highest accuracies. As shown in Table 8, GDCM-SS exhibited higher segmentation accuracy when repetitions of each block were (4, 4, 6 and 6) than when repetitions of each block were (3, 3, 5 and 5). However, the number of model parameters also increased. On the other hand, GDCM-WS exhibited higher segmentation accuracy with fewer model parameters when repetitions of each block were 3, 3, 5 and 5, compared with 4, 4, 6 and 6.

In addition, we compared the accuracies and number of model parameters by our method with those by other combinations such as Com 1 (combination of dilated convolution and attention-based method) and Com 2 (combination of dilated convolution, ASPP, and attention-based method). As shown in Tables 7 and 8, the proposed method shows better accuracy with fewer number of model parameters than other combination methods.

Table 7. Comparison of accuracy per method (unit: %).

Method (#Groups, #Subgroups)	Pixel Acc	Class Acc	mIoU
GDCM-SS (32, 2)	88.92	81.24	71.95
GDCM-SS (16, 2)	87.23	80.08	70.43
GDCM-SL (32, 2)	87.81	80.72	71.04
GDCM-SL (16, 2)	86.25	79.97	70.84
GDCM-WS (32, 4)	89.27	81.98	72.81
GDCM-WS (16, 4)	87.13	79.83	70.17
GDCM-WL (32, 4)	87.81	80.72	71.04
GDCM-WL (16, 4)	85.33	78.53	70.12
Com 1	85.13	78.21	69.26
Com 2	86.29	79.05	70.02

Table 8. Comparison of accuracy by model depth per method.

Method	Repetitions of Each Block	mIoU (%)	# of Model Parameters
GDCM-SS	(3, 3, 5, 5)	71.95	13.3 M
	(4, 4, 6, 6)	72.24	15.6 M
GDCM-WS	(3, 3, 5, 5)	72.81	13.3 M
	(4, 4, 6, 6)	72.77	15.6 M
	Com 1	69.26	17.5 M
	Com 2	70.02	18.3 M

4.4.2. Comparisons with the State-of-the-Art Methods

In the following experiment, segmentation accuracies of the proposed method and those of the state-of-the-art methods were compared. We accessed some trained models in Table 9, and if the model was inaccessible, we implemented the models based on their paper. In any case, we performed training with our training data and executed testing with our testing data for fair comparisons. Therefore, identical testing data and resolution was used by each of these methods. As shown in Table 9, it was confirmed that the GDCM-based method proposed in this study showed higher accuracy than the state-of-the-art methods. Figure 6 shows the detected results by proposed method, which confirms that our method can correctly detect even small sized objects.

Table 9. Comparisons of the proposed method with state-of-the-art methods (unit: %).

Method	Pixel Acc	Class Acc	mIoU
Luc et al. [52]	68.7	75.2	54.3
Byeon et al. [53]	75.56	68.26	-
Souly et al. [46]	82.3	77.6	63.3
Liu et al. [54]	83.5	76.9	-
Sharma et al. [55]	82.3	79.1	64.5
Mostajabi et al. [7]	86.1	80.9	-
ESSN [51]	87.46	81.51	71.56
Proposed method	89.27	81.98	72.81

Figure 6. Detected results by proposed method. The images in the 1st and 2nd rows are from CamVid and those in the 3rd row are from SBD. In each row, input, ground-truth, and detected result are respectively shown from the left. Class information is shown in Figure 4.

4.5. Processing Time

In the following experiment, the processing speed of the proposed method was measured using the desktop computer described in Section 4.2 and the Jetson TX2 (NVIDIA Corp., Santa Clara, CA, USA) embedded system [56], which is widely used for onboard deep-learning processing for existing autonomous vehicles, as shown in Figure 7. Jetson TX2 has an NVIDIA Pascal™-family GPU (256 CUDA cores), with 8 GB memory for both CPU and GPU and 59.7-GB/s memory bandwidth, and uses less than 7.5 W power.

GPU including CPU and memory blocks

Figure 7. Jetson TX2 embedded system.

As indicated in Table 10, the proposed method showed a recognition speed per image of 25.23 ms on the desktop computer and 86.31 ms on the Jetson TX2 embedded system. These values correspond to processing speeds of 39.6 (1000/25.23) frames/s and 11.6 (1000/86.31) frames/s, respectively. In the case of the Jetson TX2 embedded system, the processing time of the embedded system was longer than that on the desktop computer, as computing resources of the Jetson TX2 embedded system were significantly limited compared with those of the desktop. Nevertheless, it verified that the proposed method in this study can be applied with an embedded system having limited computing resources and that it can also enable a front camera installed on an autonomous vehicle to detect target objects.

Table 10. Comparisons of processing speed by the proposed method on desktop computer and embedded system (unit: ms).

Platform	Time	Frames/Sec
Desktop computer	25.23	39.6
Jetson TX2	86.31	11.6

5. Conclusions

This study analyzed various types of existing semantic segmentation methods to develop a method for increasing performance by considering the characteristics of semantic segmentation tasks. Based on the analytic result, the GDCM, a new semantic segmentation method that can perform multi-scale information learning by using fewer parameters, was proposed. The module was designed to generate filter groups with different views through combinations of grouped and dilated convolutions. Each filter group learned in a multi-scale context. The proposed module exhibited far superior performance than existing methods in that it required fewer calculations and parameters, owing to its application range in the convolution layer, unlike existing methods, which require a large number of additional calculations. The result of an experiment using two open databases indicated that the proposed GDCM showed improved segmentation accuracy compared to the state-of-the-art methods.

The importance and applicability of our method is that it can produce high semantic segmentation performance without additional modules, such as attention mechanisms, as shown in Tables 4, 5, 7 and 8. However, our method has a limitation that grouped convolution, which is the basis of our GDCM, requires large memory for training, which reduces the batch size and consequently increases the training time. Through the intensive experiments with two open databases having different image characteristics such as image brightness, object size, and camera viewing angle, etc., we expect the generality of the proposed model even with other datasets. Although the proposed model has strength in segmenting small sized objects, it is expected to have a limitation in segmenting severely small sized objects, such as a small tumor or cancer cell in a large sized medical image.

In future work, we would apply our model to segment severely small sized objects in medical images. In addition, we would research a method of enhancing the training speed of our method by solving the problem of grouped convolution. Moreover, further research will be conducted to apply grouped dilated convolution not only to semantic segmentation but also to different vision tasks, including those on detection and recognition of human face, body, vehicle, etc., at a distance and in various kinds of images such as visible light, near-infrared, and thermal images, to verify the applicability of this technique.

Author Contributions: Methodology, D.S.K.; Conceptualization, Y.H.K.; Supervision, K.R.P.; Writing—original draft, D.S.K.; Writing—review and editing, K.R.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (MSIT) through the Basic Science Research Program

(NRF-2020R1A2C1006179), in part by the MSIT, Korea, under the ITRC (Information Technology Research Center) support program (IITP-2021-2020-0-01789) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation), and in part by the NRF funded by the MSIT through the Basic Science Research Program (NRF-2019R1A2C1083813).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Shelhamer, E.; Long, J.; Darrell, T. Fully convolutional networks for semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 640–651. [[CrossRef](#)] [[PubMed](#)]
- Badrinarayanan, V.; Kendall, A.; Cipolla, R. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [[CrossRef](#)] [[PubMed](#)]
- Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*; Springer: Munich, Germany, 2015; pp. 234–241.
- Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Semantic image segmentation with deep convolutional nets and fully connected CRFs. In *Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015*; pp. 1–14.
- Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017*; pp. 6230–6239.
- Farabet, C.; Couprie, C.; Najman, L.; LeCun, Y. Learning hierarchical features for scene labeling. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1915–1929. [[CrossRef](#)] [[PubMed](#)]
- Mostajabi, M.; Yadollahpour, P.; Shakhnarovich, G. Feedforward semantic segmentation with zoom-out features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015*; pp. 3376–3385.
- Chen, L.C.; Yang, Y.; Wang, J.; Xu, W.; Yuille, A.L. Attention to scale: Scale-aware semantic image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016*; pp. 3640–3649.
- Lin, G.; Shen, C.; Heng, A.V.; Reid, I. Efficient piecewise training of deep structured models for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016*; pp. 3194–3203.
- Dai, J.; He, K.; Sun, J. BoxSup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015*; pp. 1635–1643.
- Yu, F.; Koltun, V. Multi-scale context aggregation by dilated convolutions. In *Proceedings of the 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, 2–4 May 2016*; pp. 1–13.
- Liu, Y.; Yu, J.; Han, Y. Understanding the effective receptive field in semantic image segmentation. *Multimed. Tools Appl.* **2018**, *77*, 22159–22171. [[CrossRef](#)]
- Hamaguchi, R.; Fujita, A.; Nemoto, K.; Imaizumi, T.; Hikosaka, S. Effective use of dilated convolutions for segmenting small object instances in remote sensing imagery. In *Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV 2018), Lake Tahoe, NV, USA, 12–15 March 2018*; pp. 1442–1450.
- Wang, P.; Chen, P.; Yuan, Y.; Liu, D.; Huang, Z.; Hou, X.; Cottrell, G. Understanding convolution for semantic segmentation. In *Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV 2018), Lake Tahoe, NV, USA, 12–15 March 2018*; pp. 1451–1460.
- Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 834–848. [[CrossRef](#)]
- Chen, L.-C.; Papandreou, G.; Schroff, F.; Adam, H. Rethinking atrous convolution for semantic image segmentation. *arXiv* **2017**, arXiv:1706.05587v3.
- He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [[CrossRef](#)] [[PubMed](#)]
- Shen, Y.; Ji, R.; Wang, Y.; Wu, Y.; Cao, L. Cyclic guidance for weakly supervised joint detection and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2019, Long Beach, CA, USA, 16–20 June 2019*; pp. 697–707.
- Jiao, J.; Wei, Y.; Jie, Z.; Shi, H.; Lau, R.; Huang, T.S. Geometry-aware distillation for indoor semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2019, Long Beach, CA, USA, 16–20 June 2019*; pp. 2864–2873.

20. He, J.; Deng, Z.; Zhou, L.; Wang, Y.; Qiao, Y. Adaptive pyramid context network for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2019, Long Beach, CA, USA, 16–20 June 2019; pp. 7511–7520.
21. Chen, L.-C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-Decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the 15th European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 833–851.
22. Zhang, H.; Zhang, H.; Wang, C.; Xie, J. Co-occurrent features in semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2019, Long Beach, CA, USA, 16–20 June 2019; pp. 548–557.
23. Fu, J.; Liu, J.; Tian, H.; Li, Y.; Bao, Y.; Fang, Z.; Lu, H. Dual attention network for scene segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2019, Long Beach, CA, USA, 16–20 June 2019; pp. 3141–3149.
24. Li, H.; Xiong, P.; Fan, H.; Sun, J. DFANet: Deep feature aggregation for real-time semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2019, Long Beach, CA, USA, 16–20 June 2019; pp. 9514–9523.
25. Huang, Z.; Wang, X.; Huang, L.; Huang, C.; Wei, Y.; Liu, W. CCNet: Criss-Cross attention for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 603–612.
26. Zhu, Z.; Xu, M.; Bai, S.; Huang, T.; Bai, X. Asymmetric non-local neural networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 593–602.
27. Grouped Dilated Convolution Module (GDCM)-based Semantic Segmentation Network with Algorithm. Available online: <https://github.com/ddongk/GDCM> (accessed on 14 April 2021).
28. Luo, W.; Urtasun, Y.; Li, R.; Zemel, R. Understanding the effective receptive field in deep convolutional neural networks. In Proceedings of the NIPS 2016: The Thirtieth Annual Conference on Neural Information Processing Systems, Barcelona, Spain, 4–9 December 2016; pp. 4905–4913.
29. Koltun, F.; Yu, V.; Funkhouser, T. Dilated residual networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 636–644.
30. Lin, Z.; Feng, M.; Santos, C.N.D.; Yu, M.; Xiang, B.; Zhou, B.; Bengio, Y. A structured self-attentive sentence embedding. In Proceedings of the 5th International Conference on Learning Representations, Toulon, France, 24–26 April 2017; pp. 1–15.
31. Wang, F.; Jiang, M.; Qian, C.; Yang, S.; Li, C.; Zang, H.; Wang, X.; Tang, X. Residual attention network for image classification. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6450–6458.
32. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. CBAM: Convolutional block attention module. In Proceedings of the 15th European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 1–17.
33. Wang, X.; Girshick, R.; Gupta, A.; He, K. Non-local neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 7794–7803.
34. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 7132–7141.
35. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Vedaldi, A. Gather-Excite: Exploiting feature context in convolutional neural networks. In Proceedings of the NIPS 2018: The Thirty-second Annual Conference on Neural Information Processing Systems, Montreal, QC, Canada, 3–8 December 2018; pp. 1–11.
36. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. In Proceedings of the 26th Annual Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1–9.
37. Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; He, K. Aggregated residual transformations for deep neural networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5987–5995.
38. Brostow, G.J.; Fauqueur, J.; Cipolla, R. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognit. Lett.* **2009**, *30*, 88–97. [[CrossRef](#)]
39. Gould, S.; Fulton, R.; Koller, D. Decomposing a scene into geometric and semantically consistent regions. In Proceedings of the 2009 IEEE 12th International Conference on Computer Vision, Kyoto, Japan, 29 September–2 October 2009; pp. 1–8.
40. He, T.; Zhang, Z.; Zhang, H.; Zhang, Z.; Xie, J.; Li, M. Bag of tricks for image classification with convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2019, Long Beach, CA, USA, 16–20 June 2019; pp. 558–567.
41. Kingma, D.P.; Ba, J.L. Adam: A method for stochastic optimization. In Proceedings of the ICLR 2015: International Conference on Learning Representations 2015, San Diego, CA, USA, 7–9 May 2015; pp. 1–15.
42. Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; Lerer, A. Automatic differentiation in PyTorch. In Proceedings of the NIPS 2017: The Thirty-first Annual Conference on Neural Information Processing, Long Beach, CA, USA, 4–9 December 2017; pp. 1–4.
43. GeForce GTX 1070. Available online: https://en.wikipedia.org/wiki/GeForce_10_series (accessed on 3 April 2020).
44. Everingham, M.; Gool, L.V.; Williams, C.K.I.; Winn, J.; Zisserman, A. The PASCAL visual object classes (VOC) challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–388. [[CrossRef](#)]

45. Brostow, G.J.; Shotton, J.; Fauqueur, J.; Cipolla, R. Segmentation and recognition using structure from motion point clouds. In Proceedings of the 10th European Conference on Computer Vision, Marseille, France, 12–18 October 2008; pp. 44–57.
46. Souly, N.; Spampinato, C.; Shah, M. Semi supervised semantic segmentation using generative adversarial network. In Proceedings of the 2017 IEEE International Conference on Computer Vision: ICCV 2017, Venice, Italy, 22–29 October 2017; pp. 5689–5697.
47. Jégou, S.; Drozdal, M.; Vazquez, D.; Romero, A.; Bengio, Y. The one hundred layers tiramisu: Fully convolutional DenseNets for semantic segmentation. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1175–1183.
48. Zhao, H.; Qi, X.; Shen, X.; Shi, J.; Jia, J. ICNet for real-time semantic segmentation on high-resolution images. In Proceedings of the 15th European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 418–434.
49. Yu, C.; Wang, J.; Peng, C.; Gao, C.; Yu, G.; Sang, N. BiSeNet: Bilateral segmentation network for real-time semantic segmentation. In Proceedings of the 15th European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 334–349.
50. Yang, T.; Wu, Y.; Zhao, J.; Guan, L. Semantic segmentation via highly fused convolutional network with multiple soft cost functions. *Cognit. Syst. Res.* **2019**, *53*, 20–30. [[CrossRef](#)]
51. Kim, D.S.; Arsalan, M.; Owais, M.; Park, K.R. ESSN: Enhanced semantic segmentation network by residual concatenation of feature maps. *IEEE Access* **2020**, *8*, 21363–21379. [[CrossRef](#)]
52. Luc, P.; Couprie, C.; Chintala, S.; Verbeek, J. Semantic segmentation using adversarial networks. In Proceedings of the Thirtieth Conference on Neural Information Processing Systems, Workshops on Adversarial Training, Barcelona, Spain, 5–10 December 2016; pp. 1–9.
53. Byeon, W.; Breuel, T.M.; Raue, F.; Liwicki, M. Scene labeling with LSTM recurrent neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3547–3555.
54. Liu, F.; Lin, G.; Shen, C. CRF learning with CNN features for image segmentation. *Pattern Recognit.* **2015**, *48*, 2983–2992. [[CrossRef](#)]
55. Sharma, A.; Tuzel, O.; Jacobs, D.W. Deep hierarchical parsing for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 530–538.
56. Jetson TX2 Module. Available online: <https://www.nvidia.com/en-us/autonomous-machines/embedded-systems-dev-kits-modules/> (accessed on 30 October 2020).