Article

Real-Time Semantic Segmentation of Road Scenes via Hybrid Dilated Grouping Network

Yan Zhang [1], Xuguang Zhang [1,*], Deting Miao [1], and Hui Yu [2]

[1] School of Communication Engineering, Hangzhou Dianzi University, Hangzhou 310018, Zhejiang, China
[2] cSCAN, University of Glasgow, G12QB, United Kingdom
[*] Correspondence: zhangxuguang78@163.com

Received: 10 January 2024
Accepted: 23 April 2024
Published: 25 March 2025

Abstract: Real-time semantic segmentation is a critical step for various real-world application scenarios such as autonomous driving systems. How to achieve a high accuracy while keeping a high inference speed has become a challenging issue for real-time semantic segmentation. To tackle this challenge, we propose a Hybrid Dilated Grouping Network (HDGNet) for real-time semantic segmentation of outdoor scenes in this study, which not only improves the accuracy of image segmentation, but also considers the inference speed. To reduce model parameters to speed up inference, we propose to use factorization convolution to replace ordinary two-dimensional convolution. However, simply reducing the amount of model parameters may sacrifice segmentation accuracy. We thus further introduce dilated convolution to extract multi-scale spatial information. The HDG module is constructed by combining factorization convolution and dilated convolution, which not only reduces the model parameters and improves the model inference speed, but also extracts local and more contextual information. And furthermore, to enhance the feature expression ability of the network, we introduce a channel attention mechanism to capture the information interaction between channels. After obtaining the shallow features and deep high-level semantic information, we design the skip layer connections to fuse the feature branches from different stages to improve the segmentation accuracy. The experiments conducted on the widely used datasets show that the proposed model achieves superior real-time performance over existing methods but using significantly fewer model parameters.

Keywords: real-time semantic segmentation; factorization convolution; depth-wise separable convolution; dilated convolution; channel attention mechanism

1. Introduction

Semantic segmentation is an increasingly popular research topic in recent years, that aims to predict the semantic class labels of all pixels in a given input image and present the segmentation results of different color region masks. It requires accurate segmentation of objects, edges and other details in the image [1], which can help us better analyze scene information. Different from image classification [2], semantic segmentation requires dense pixel-level prediction capabilities and greater computational costs. With the development and demand of autonomous driving, semantic segmentation of outdoor scenes has been attracted increasing attention, since it is a fundamental step for understanding the scene for the safe driving of autonomous vehicles. Figure 1 shows an example of semantic segmentation of outdoor scenes. For practical applications, it is critical to perform semantic segmentation of the scene as fast as possible to obtain effective information for making real-time judgments on the environment.

In traditional methods of semantic segmentation, the image is usually segmented and labeled according to the underlying features. Based on different segmentation standards, traditional image segmentation algorithms can be classified as the following three methods: threshold-based segmentation, cluster-based segmentation, and graph theory-based segmentation. The threshold-based image segmentation method is to divide the pixels according to the gray level, with a certain threshold as the segmentation line, and finally the entire image is divided into target and background. The threshold is generally a gray value used in an early method of Otsu [3]. Clustering segmentation is the process of distinguishing and classifying each pixel in an image according to certain requirements and rules, and finally determining that each pixel belongs to a certain region. Mean-shift [4] and SLIC [5] are widely used in cluster segmentation. Image segmentation based on graph theory is to transform the segmentation problem into graph segmentation. Traditional semantic segmentation generally uses Markov Random Field (MRF) and Conditional Random Field (CRF) to build a probabilistic graphical model, and then uses the graph theory to solve the problem [6]. The main idea is to assign a random vector to each feature and pixel. The probability that each pixel belongs to a class determines the classification of that pixel. Traditional image segmentation methods often need to find suitable features and further optimization [7, 8]. The feature extraction of traditional methods mainly relies on artificially designed extractors. At the same time, each method aims at specific applications, and the generalization ability and robustness are poor with a weak segmentation effect.

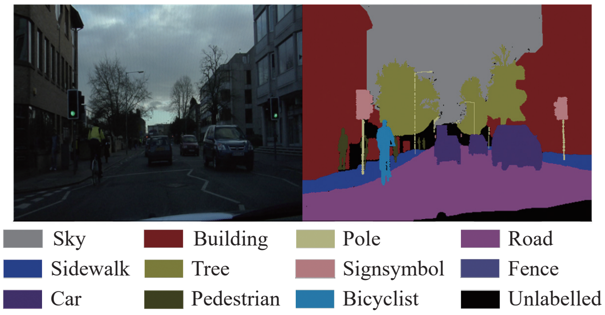


Figure 1. Semantic segmentation of outdoor scenes.

The advancement of artificial intelligence [9, 10] and deep learning technologies [11] facilitates the improvement of semantic segmentation in recent years. The multi-task learning framework combines depth estimation and semantic segmentation to improve segmentation performance, and image segmentation in specific scenarios. It has significantly enhanced the quality of semantic segmentation in real life applications, like in autonomous driving, drones [12], medical diagnosis, remote sensing image segmentation, and video surveillance [13]. Deep convolutional neural networks have also demonstrated powerful capabilities in high-resolution image classification [14]. In particular, Fully Convolutional Networks (FCNs) [15], which are the forerunners of CNNs for semantic segmentation tasks. Encoder-decoder networks are emerging as effective frameworks for segmentation problems. Although previous networks have achieved remarkable results in terms of segmentation accuracy [16, 17], most of them ignore the segmentation efficiency. That is, the high computation cost and the storage space needs make it difficult to meet the requirements for rapid interaction with the environment in real time [18]. Thus, designing lightweight and efficient networks is the main trend to solve these existing problems. In general, the smaller the number of model parameters, the shorter the inference time and the less redundancy. In the current segmentation network, researchers prefer to compress the network and shrink the amount of parameters to speed up the process [19–21]. Although the inference speed of the model has reached a certain level, model's accuracy has suffered as a result. Therefore, our purpose is to explore how to extract a high-precision and high-efficiency real-time outdoor scene semantic segmentation network.

Most of the previous works have demonstrated the effectiveness of dilated convolution [22]. It may both broaden the receptive field and keep the visual resolution intact. Those real-time-targeted semantic segmentation models employ dilated convolutions in their networks. Depth-wise separable convolution is another type of methods that effectively reduces the amount of parameters. Traditional convolution combines spatial and channel correlation to extract features, while depth-wise separable convolution separates spatial information from channel information. It independently computes cross-channel and spatial information to analyze the correlations between them and is often used in lightweight networks [23, 24]. However, directly replacing standard convolutions with depth-wise separable convolutions results in significant performance degradation, as depth-wise separable convolutions greatly reduce parameters.

To sum up, the contribution of this paper can be summarized in the following four points:

(1) Aiming at the problem of slow inference speed of semantic segmentation network model, this paper proposes to use factorization convolution to replace ordinary convolution, and combines the advantage of depthwise separable convolution to reduce the number of parameters to design a lightweight network module to speed up the network inference speed.

(2) In order to avoid the problem of insufficient network feature expression ability caused by the reduction of parameter number, a method is proposed to apply dilated convolution to factorization convolution in the depth direction. The advantage of dilated convolution is that more abundant long-distance information can be obtained, so that

the network can extract enough semantic information while reducing the number of parameters.

(3) an efficient hybrid dilated grouping module is designed by combining dilated convolution, factorization convolution and depthwise separable convolution. In order to avoid gradient vanishing, the module adopts ResNet residual structure and connects the input with the output through a series of convolution transformations to compensate for information loss. The deep features are extracted by reusing the module at different stages in the network.

(4) In order to further improve the feature expression ability of the network, considering that the network proposed above has used dilated convolution to capture enough context-space information, a lightweight attention module is integrated in the encoder stage of the network to capture the information correlation between channels.

The rest of this paper is structured as follows: In Section II, we introduce related works which have employed different approaches in real-time semantic segmentation. In Section III, IV and V, we introduce the model structure proposed in this paper in detail, including attention module and the hybrid dilated grouping module, which is composed of dilated convolution and factorization convolution. In Section VI, we conduct ablation experiments and global experiments to verify that the network we designed is effective. A conclusion and future works are given in Section VII.

2. Related Work

2.1. Lightweight segmentation networks

In recent years, with the increasing attention paid to practical applications in challenging scenarios, many works have explored the potential of more lightweight architectures. In intelligent vehicle systems, speed is an important indicator to consider in outdoor driving scenarios. Real-time semantic segmentation requires not only high precision prediction results, but also fast reasoning speed. There are generally two types of real-time semantic segmentation models: (1) The models' initial features are extracted using a lightweight backbone network, and subsequently deep semantic features and shallow features are fused using various feature fusion modules. (2) These approaches generally design a lightweight module that can be used for pixel-level prediction, in which both factorization convolution and dilated convolution are present. ENet [19] is an efficient lightweight network for semantic segmentation in real-time. To decrease the quantity of computation, it cuts off a lot of convolution kernels. However, the lightweight network destroys the spatial information through the compression channel, and the segmentation accuracy is not good. A better real-time semantic segmentation network for high-resolution images is a priority, Zhao et al. [25] proposed the Image Cascade Network (ICNet), which utilized an image cascade strategy to improve segmentation efficiency. When using lower resolution input images, however, performance suffers. ESPNet [22] and ERFNet [13] are two other efficient networks for real-time semantic segmentation. They have similar model parameters as ENet. However, the segmentation accuracy and inference speed are improved compared with ENet. Mehta et al. [26] proposed an efficient spatial pyramid module in ESPNet, which could obtain multi-scale context information. Romera et al. [20] proposed the ERFNet, which used residual connections and factorization convolution to ensure efficiency and accuracy. Yu et al. [27] proposed two branches in BiseNet. The first fork is concerned with retaining the initial network layer's spatial information, while the second is concerned with extracting the network's deeper characteristics. These networks balance speed and accuracy to a certain extent, but speed and accuracy are mutually restricted. How to minimize computation while ensuring the network's performance or how to feed more image information to the real-time network still needs to be further explored.

2.2. Factorization Convolution, Depth-wise Separable Convolution

Factorization convolution and depth-wise separable convolution are frequently employed in place of conventional two-dimensional convolution to preserve segmentation performance while decreasing the amount of parameters. Factorization convolution, also known as asymmetric convolution, is the factorization of standard convolution. In other words, replacing $n \times n$ convolution with $n \times 1$ convolution and $1 \times n$ convolution can reduce computational cost and memory, and can be widely used in lightweight segmentation networks. Szegedy et al. [28–31] replaced the large convolution kernel with some small convolution kernels while guaranteeing the receptive field in Inception networks. To reduce computing costs, Chollet [23] and Howard [24] used depth-wise separable convolutions in Xception and MobileNet, respectively, but the accuracy dropped slightly. More and more researchers tend to use factorization convolution to decompose the 3×3 convolution into 3×1 and 1×3 convolutions in real-time semantic segmentation, such as EDANet [33], DABNet [41] and LEDNet [41], which greatly reduces the amount of computation.

2.3. Dilated Convolution

Simply reducing the quantity of model parameters may reduce the segmentation accuracy. Thus, dilated convolution combined with factorization convolution is usually used to design effective modules for real-time semantic segmentation. Dilated convolution increases the receptive field of the model by padding zeros in the ordinary 2D convolution without sacrificing the resolution of input images, and scene parsing studies have shown that long-distance information is essential, so it is often used in semantic segmentation models. Compared with the original normal convolution operation, for the dilated convolution, there is an additional parameter known as the dilation rate, which represents the number of intervals for inserting hole zeros into the convolutional kernel. In the Deeplab series [16], an Atrous Spatial Pyramid Pooling (ASPP) module was proposed that is mainly based on several different convolution kernels to obtain more contextual information, and each convolution kernel has a different dilation rate; Wang et al. [34] proposed the LEDNet, which used factorization and dilated convolutions to form the basic module; Lo et al. [32] used dilated convolutions and dense connections to design efficient network structures in the EDANet. At the moment, most semantic segmentation networks use dilated convolutions, which have been shown to work well for pixel-level prediction tasks.

2.4. Attention Mechanism

In recent years, we often see that attention mechanism is applied to various deep learning tasks. It is analogous to the visual attention system in humans. It examines the entire image to find the place that needs special attention, and then concentrates all of its efforts there. Huang et al. [35] proposed the CCNet, which designed an efficient cross-attention module for detecting visual correlations; Fu et al. [22] proposed the DANet, which adaptively integrates local information and global dependencies, fusing channel and spatial attention to further enhance feature representation. That attention mechanism enabled each pixel to fully capture the global information, but it needed to generate a huge attention map since the attention map of each pixel needed to be calculated for the whole image. The above attention models require complex matrix multiplications at the pixel level, which means they are not suitable for the network structures working in real time. Researchers have developed an increasing number of lightweight attention networks in recent years to improve feature representation. SENet [36] is lightweight, mainly to increase the model's sensitivity to channel attributes by focusing on the correlation between channels. It could learn the relevance of various channel elements on its own; Woo et al. [37] proposed the CBAM combining spatial and channel attention, which achieved better results while ensuring the amount of computation; Wang et al. [38] proposed an efficient cross-channel interaction strategy without dimensionality reduction in ECANet, and it not only maintains segmentation accuracy but also greatly reduces network complexity.

3. Network Structure

In this study, our goal is to create a lightweight network with fewer parameters and removes unnecessary structures to strike a trade-off between speed and accuracy for segmentation. Based on the hybrid dilated grouping module and attention module, we design a lightweight real-time network structure for semantic segmentation of outdoor scenes, as shown in Figure 2.

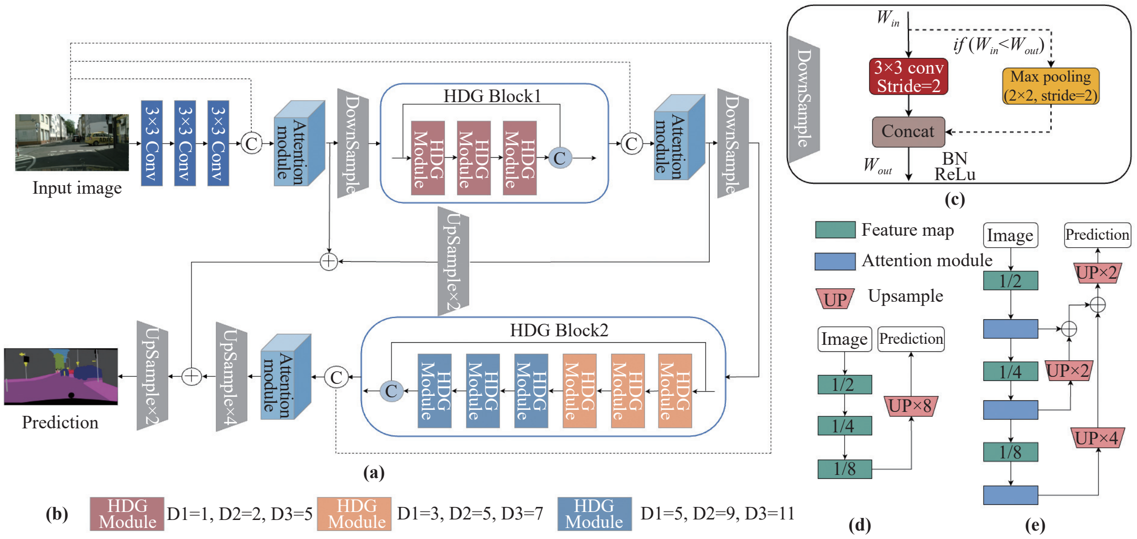


Figure 2. Architecture of proposed Hybrid Dilated Grouping Network. (A) HDGNet detailed network architecture; (B) Combinations of different dilation rates used by HDG modules; (C) DownSample module; (D) DABNet backbone architectures; (E) HDGNet backbone architectures.

Different from DABNet backbone architecture shown in Figure 2(D), It can be seen that we design an efficient hybrid dilated grouping module in the encoder for the full extraction of image features, and the attention module corrects the characteristics of the channels to improve the expressive power of the neural network, so that we do not use a complex module at the decoding end to recover the image information. Our network that shown in Figure 2(E) has higher segmentation accuracy than other methods with fewer parameters and can achieve a certain inference speed.

Early processing of high-resolution input is computationally expensive, to reduce the amount of calculation, three 3×3 convolutions are used to shrink the size of the input image to extract the initial features of the image. Referring to the design of the initial block in ENet, we use the same block, all downsampling layers share a common building block that is then extended to two different modes in the network. If the amount of input channels exceeds the amount of output channels, this module is a simple 3×3 convolution with a stride of 2. Otherwise, a 2×2 max pooling layer with a stride of 2 is added, and the convoluted and pooled feature maps are then connected to form the final downsampled output.

Although the downsampling operation reduces the input image's resolution through convolution operation to increase the receptive field, some visual details will be lost in the process. In contrast to many semantic segmentation network models, we only perform three downsampling operations to get 1/8 of the image size, while they do five downsampling operations to get 1/32 of the image resolution, which will cause a huge loss of information. To this end, we use skip layer connections between the features after each downsampling block and characteristics of the middle layer to make up for the information that was lost.

Each downsampling block is followed by a hybrid dilated grouping module, which includes different numbers of consecutive hybrid dilated grouping blocks. The first and second hybrid dilated grouping modules consist of 3 and 6 HDG blocks, respectively, which are used for dense feature extraction. To reduce information loss and better perform feature propagation, a skip layer connection is adopted in the hybrid dilated grouping module to combine the input with the deepest features, which means that in each HDG module, the input of the first HDG block is added to the output of the last HDG block. The skip-layer connection can strengthen information dissemination and can be seen as a residual connection that avoids gradient disappearance. This operation alleviates the conflict between the loss of semantic information in shallow features and the lack of boundary and detail information in deep features. In addition, we use hybrid dilated convolution in the HDG module. The dilated rate of each hybrid dilated grouping module shown in Figure 3 gradually expands the receptive field with different settings of the dilation rate in each block. In addition to obtaining spatial context information, we also need to capture more channel features. An efficient channel attention module is used after each skip layer connection, aiming to focus on the extraction of channel features.

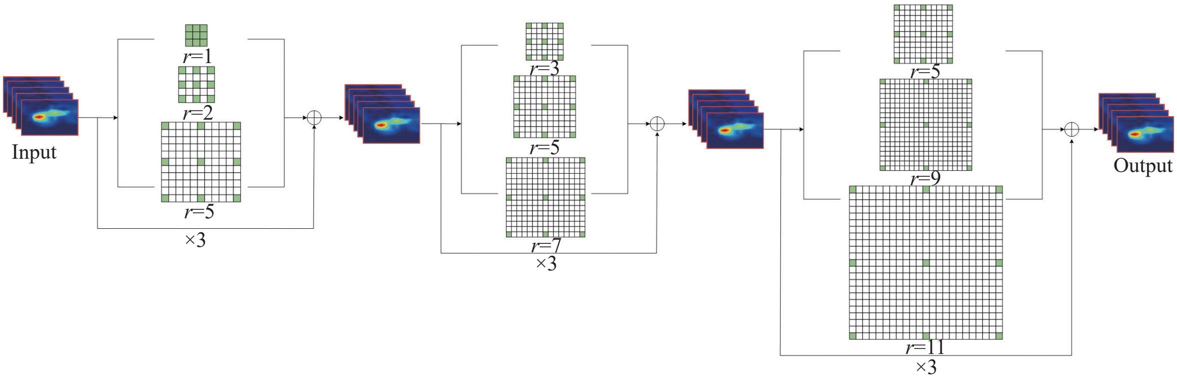


Figure 3. Hybrid dilated convolution module.

Our network is different from many other segmentation networks that use an encoder-decoder structure. At the decoder end, we do not design complex modules and complex network structures to recover image information. Considering that reducing the number of parameters may affect the extraction and restoration of features and thus reduce the accuracy, we propose the reuse of feature maps at different stages in the decoder to make up for the lost information. The low complexity and computing cost of the network guarantee the real-time performance of the network. The experimental results show that our model has few parameters and low complexity, which can achieve certain precision and meet the design requirements of real-time semantic segmentation network.

4. Hybrid Dilated Grouping Module

This paper is to build a lightweight semantic segmentation network, which can not only attain a certain segmentation accuracy but also have a certain reasoning speed. In the design method of real-time semantic segmentation network model, we choose to design a lightweight module to extract deep semantic information, and then reuse it in the whole network. We design a hybrid dilated grouping module to be an effective module for dense prediction. For real-time semantic segmentation, obtaining a certain precision in segmentation and also meeting the real-time requirements is essential. Therefore, we take these two requirements into account, first of all, from the perspective of reducing the amount of model parameters, we use factorization convolution and depth separable convolution to replace the traditional two-dimensional convolution, which can greatly reduce the amount of model parameters, thus speeding up the network's inference; Secondly, considering that reducing the number of parameters may affect the extraction of features and thus reduce the accuracy, we propose to use dilated convolution to capture long-distance context information and extract deeper semantic information. Combining factorization convolution, depth-wise separable convolution, and dilated convolution, our HDG Module can capture long-distance context information with fewer parameters. We will discuss each component in the HDG Module in detail.

4.1. Factorization Convolution, Depth-wise Separable Convolution

Current lightweight real-time semantic segmentation usually uses depth-wise separable convolution and factorization convolution to design efficient modules for dense prediction. For a standard $N \times N$ convolution kernel, it can be replaced by a $1 \times N$ convolution and an $N \times 1$ convolution. Factorization convolution can reduce the $N \times N$ convolution kernels per-pixel complexity from $O(N^2)$ to $O(N)$. Depth-wise separable convolution integrates ordinary standard convolution into depth-wise convolution followed by point-wise convolution. Each convolution kernel of depth convolution corresponds to one channel, while each convolution kernel of conventional convolution operates on each channel of the input image. For a $K \times K$ convolution kernel, the input's size is $F \times F \times M$ and the output's size is $F \times F \times N$. M and N are the number of input and output channels respectively. The calculation amount of conventional convolution is: K^2F^2MN, and the calculation amount of depth-wise convolution is: K^2F^2M. It can be seen that we apply factorization convolution to depth-wise convolution can significantly decrease the amount of computation, as shown in Figure 4. Referring to EDANet [32], it can be expressed as follows:

$$\begin{aligned} Map_w &= \sum_{i=-M}^{M}\sum_{j=-N}^{N} K(i,j)I_w(x-i,y-j) \\ &= \sum_{i=-M}^{M} K_x(i)\left[\sum_{j=-N}^{N} K_y(j)I_w(x-i,y-j)\right]. \end{aligned} \quad (1)$$

where I_w represents a 2D images with w number of channels, K denotes a standard convolution kernel, K_x and K_y are 1D convolution kernels in different dimensional directions, respectively. M, N represent the size of the convolution kernel. And (x, y) represents the position of the pixel. Map_w is a feature map obtained by using factorization convolution in the depth direction.

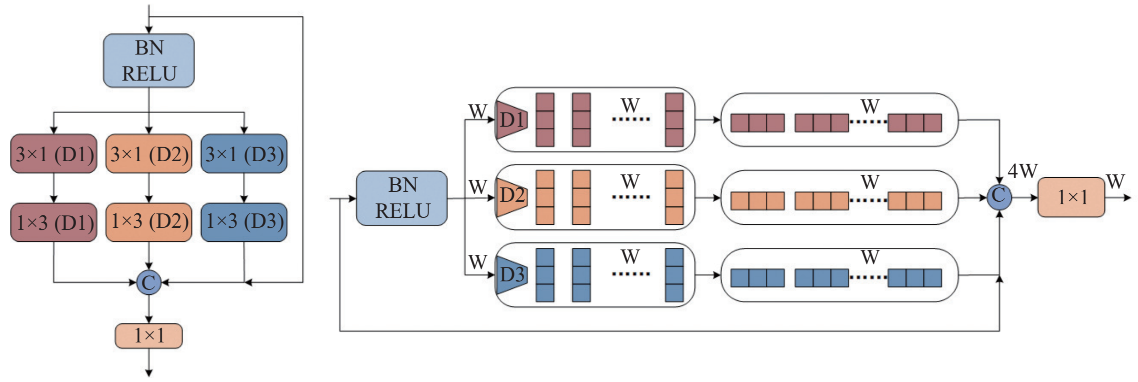


Figure 4. HDGModule (D: dilation rate, w: input channels).

4.2. Hybrid Dilated Convolution

In 1-D, dilated convolution is described as:

$$output[i] = \sum_{l=1}^{L} input[i + r \cdot l]h[l]. \quad (2)$$

where $input[i]$ and $output[i]$ represent input and output signals, respectively, $h[l]$ is the filter of length L, and r corresponds to the dilation rate we use to sample the input signal. In standard convolution, $r = 1$.

To enrich the multi-scale feature extraction capability of the CNN network, dilated convolution is used to gather long-distance context information for semantic segmentation by inserting holes (zeroes) between pixels in the convolution kernel to increase the receptive field without reducing the resolution of feature maps. This enables denser feature extraction in relatively deep layers. The $k \times k$ convolution kernel is added with a dilation rate, then the size of the dilated convolution kernel is $k_d \times k_d$, where $k_d = k + (k-1) \times (r-1)$, r is a dilation rate.

However, when the dilation rate of the dilated convolution becomes large in the high layers, the sampling of the input becomes sparse. This can cause some new problems: local information may be lost, some information over long distances may not be relevant, and the grid effect [39] may break the link between regional details, which is not conducive to learning features. To eliminate the grid effect, we propose a novel hybrid dilated convolution. While capturing multi-scale context information, it can eliminate the grid effect in traditional dilated convolution, so that the network can gather more meaningful information and improve the model's capacity to convey features.

A hybrid dilated grouping module based on three branches in parallel is proposed to capture more context information of multi-scale objects by mixing three extended convolution cores, as shown in Figure 3. This mixing of information is helpful to model learning and model robustness. According to the hybrid dilated convolution theory [40], we set different dilation rates for each hybrid dilated module. The common divisor of the dilation rate in each module cannot be greater than 1, and all dilation rates should show a gradually increasing trend. As shown in Figure 3, it has two advantages: First, it eliminates the grid effect caused by inserting holes in the convolution kernel; second, it captures more contextual information about multi-scale objects by combining three dilated convolution kernels. In order to make up for the information loss produced by the dilated convolution operation, the hybrid dilated convolution module connects the input feature with the output.

4.3. HDGModule

Previous work has designed many lightweight network structures [20, 33], Inspired by these network structures, this paper designs an efficient hybrid dilated grouping module. Unlike other lightweight modules, our module does not use a bottleneck structure to reduce dimensions but uses a non-bottleneck structure to obtain better accuracy. The original input is sent to three branches to extract multi-scale contexts at different scales. The three branches use different dilation rates to capture information at different scales of a single image. With the common advantages of dilated convolution, factorization convolution and residual connection, better results can be obtained in the case of limited computing power. It can jointly capture the surrounding and context information, which is very suitable for high-resolution urban scenes.

Our HDG module is shown in Figure 4. Inspired by most lightweight structures, the HDG module combines dilated convolution with factorization convolution, the input channel remains unchanged. It is then fed into three branches with different dilation rates. Finally, the number of channels is recovered by 1×1 point-wise convolution and all channel information is fused through residual connection. Each branch is composed of factorization convolutions, which greatly reduces the amount of computation. To broaden the network's receptive field, in this work, we further perform dilated convolution on the factorization convolution to capture more long-distance feature information. However, when the dilation rate is set relatively high, the model needs to fill a large number of holes in the convolution kernel to maintain the resolution of the feature map, which brings a huge amount of computation and does not meet the design principles of lightweight networks. To solve the above contradiction, we apply dilated convolution to the factorization convolution in the depth direction on each branch to decrease the amount of computation, which is called the hybrid dilated grouping convolution in the depth direction.

In the hybrid dilated grouping module, residual connection is to make up for the lack of information. In the deep network, when the number of layers increases, issues like gradient dissipation and explosion become more prevalent. The residual connection connects the input and output, which can solve these problems well. To this end, we adopt residual connections to solve the degradation problem of deep neural networks, making the forward and backward propagation of information more fluent.

We apply activation functions in the model to boost the nonlinearity of the network model in order to improve its nonlinear expression capabilities. A pre-activation function is adopted in the hybrid dilated grouping module in this work, and we use batch normalization before each nonlinear function. As with ENet, we use PReLU as a nonlinear operation because it performs better than ReLU in the shallow network model.

The HDGModule can be summarized as follows: First, the module uses depth-wise separable convolution in each branch and applies factorization convolution on depth-wise convolution. Three branches with different dilation rates are responsible for extracting the corresponding context information. Then, the three branches are stacked together and a residual connection is taken with the input. Finally, in order to make the number of channels the same as the original, pointwise convolution is used so that all channel information can be fused. This series of operations can be described as follows:

$$x_b = p(x_{HDGin}). \tag{3}$$

$$y_1 = C_{1\times3,d1}(C_{3\times1,d1}(x_b)). \tag{4}$$

$$y_2 = C_{1\times3,d2}(C_{3\times1,d2}(x_b)). \tag{5}$$

$$y_3 = C_{1\times3,d3}(C_{3\times1,d3}(x_b)). \tag{6}$$

$$y_{HDGout} = C_{1\times1}(p(concat(y_1, y_2, y_3, x_{HDGin}))) \tag{7}$$

where x_{HDGin} and y_{HDGout} denote the input and output of the HDG module respectively, p is the PReLU nonlinear activation function, y_1, y_2 and y_3 are the outputs of three branches in HDG module, $C_{m\times n}$ is the convolution kernel of $m\times n$, d represents the dilation rate, and $concat$ denotes feature concatenation.

5. Attention Module

In general, due to the small number of network layers, it is difficult for lightweight networks to extract enough deep features as the same as large networks. Therefore, generating more useful features and adding them to our network is an efficient strategy to improve segmentation accuracy. We introduce the attention mechanism into the proposed network model, which is helpful for the model to extract valuable information. Considering that we have already used HDG modules with dilated rate combinations in the network, which can capture long-distance spatial information, we only introduce the channel attention module and assign different weights to each channel to strengthen the characteristics of important channels, avoid the interference of redundant information and noise between channels, so that the model can adapt to the instability brought by data changes, and enhance the robustness of the model. In order to improve the feature expression ability of the network model, we introduce channel attention mechanisms at different stages of the network encoder to capture channel correlation. At the same time, in order to improve the semantic segmentation performance without increasing the amount of network parameters, we use a lightweight attention module to improve the model's expression ability.

We use ECANet [38] as a module to extract channel information in the network, which is a lightweight attention model improved based on SENet module, as shown in Figure 5. Although it only involves a small number of parameters, it can achieve significant performance gains. The local cross-channel interaction technique proposed by this channel attention network does not include dimensionality reduction, which can effectively avoid the impact of dimension reduction on the channel attention learning effect. By adding the channel attention mechanism at different locations of the network, the feature expression ability of the model can be adaptively improved, which greatly promotes the connection between local and contextual information. The channel attention module we introduced in the network is placed after each feature fusion, which can well extract the channel with the strongest semantic information relevance in the network, enable the network model to pay more attention to the feature extraction of the required target, make the whole process of feature extraction and fusion more effective, and improve the fusion efficiency between features.

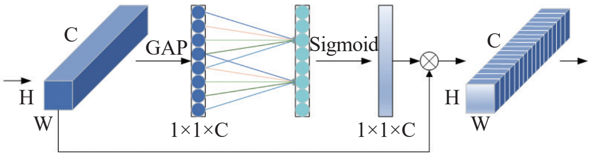


Figure 5. Efficient Channel Attention Module.

This attention module first performs channel-by-channel global average pooling without reducing dimensions and then obtains the weight of each channel through one-dimensional convolution with the size of k to complete cross channel information interaction. The ECA module captures local cross channel interaction by evaluating each channel and its k nearest neighbors, where k reflects the coverage of local cross channel interaction, i.e. how many neighbors participate in a channel's attention prediction, and k is adaptively calculated. The weight of each channel is then determined using the sigmoid activation function, and it is eventually weighted to the original feature.we should expand to understand that this attention module is actually a two-step operation. The first step is to use global average pooling to compress the global spatial information into the global information. The second step is to obtain the weight of each channel through one-dimensional convolution to complete the information interaction across channels. The operating principle of ECANet is described as follows:

$$ECA(F) = \sigma(f^{k\times k}(T(AvgP(F)))) \times F. \tag{8}$$

where T represents the transposition, compression operations of the tensor dimensions, specifically, the input of $1\times C\times W\times H$ will become $1\times C\times 1\times 1$ after the global average pooling operation. Our goal is to change it into the tensor form of $1\times 1\times C$. Therefore, the T operation is to compress the tensor into $1\times C\times 1$ and then transpose it into $1\times 1\times C$. $f^{k\times k}$ represents the adaptive convolution kernel, and k denotes the coverage of local cross channel interaction, $AvgP$ denotes global average pooling, σ is the sigmoid nonlinear activation function, F is the input feature map.

6. Experiments

In this section, we conduct experiments on the proposed network on Cityscapes and CamVid datasets to verify the effectiveness of the model. At first, the dataset and parameter settings are introduced. Then ablation experiments are performed on the Cityscapes validation set to demonstrate our network's efficacy. Finally, we compare our network with other existing semantic segmentation networks.

6.1. Experimental Settings

The experiment has been conducted on a computer with specifications of Intel i7-8700K CPU and a single GeForce GTX 1080Ti GPU. The programming language is Python 3.8, and the deep learning framework is PyTorch 1.8.1. We use the computing platform CUDA 11.0.

We primarily conduct ablation experiments on the Cityscapes validation set to validate the proposed algorithm model's performance. The Cityscapes dataset, which includes 5000 high-quality pixel-level annotated photographs of urban driving situations from 50 locations, is given collaboratively by three German companies, including Daimler AG. There are 19 semantic categories in the dataset, such as people, ground, vehicles, buildings, etc. The image resolution is 1024×2048, including a total of 2975 samples were used for training, 500 for validating, and 1525 for testing. Because the whole experiment does not use any pre-trained models, we use another low-resolution dataset, CamVid for outdoor scenarios to facilitate training and quickly judge the quality of the model, which is divided into 11 semantic categories and contains 367 training samples, 101 validation samples, and 233 test samples. The image resolution in the CamVid dataset is 720×960. To reduce computation and memory usage, the raw inputs for network training are resized to 512×1024 and 360×480 for Cityscapes and CamVid, respectively.

In the training of neural networks, in addition to finding optimal parameters such as weights and biases, setting the right hyperparameters is equally important in the training of the network. For example, too small a learning rate can lead to slow convergence of the model, while too large a learning rate can cause the model to oscillate or diverge around the point of minima. Smaller batch sizes can increase the convergence speed of the model but can lead to increased noise during training. Larger batch sizes reduce noise but take up more memory. Too few iterations can lead to model underfitting, while too many iterations can lead to model overfitting.To train the network end-to-end for the Cityscapes dataset, we employ the stochastic gradient descent (SGD) approach with a batch size of 4 to fully utilize GPU memory. We set momentum to 0.9 to accelerate convergence, and weight decay is set to 1×10^{-4} to prevent overfitting. The learning rate is adaptively modified after each iteration using the ``poly" learning approach, according to the following formula:

$$lr = lr_{base} \times \left(1 - \frac{iteration}{max\_iteration}\right)^{power} \tag{9}$$

where lr_{base} denotes the initial learning rate, lr is the learning rate after each iteration. In order to ensure model's learning effectiveness while quickening the convergence and enhancing the efficiency of model operation, we set the initial learning rate to 0.045 during training; $iteration$ and $max\_iteration$ respectively represent the order of the current iteration and the total number of iterations for each epoch; the default value of power is 0.9. We apply online hard example mining (OHEM) loss on the Cityscapes dataset to help the model focus more on difficult-to-classify examples during training, which is proposed for the problem of class imbalance. It is based on the cross-entropy loss:

$$Loss = -\sum_{i=0}^{C-1} y_i \log(p_i) \tag{10}$$

where C is the number of classes, y_i is the true value, and p_i is the probability of the predicted value.

Through multiple experiments, considering the advantages of Adam's ability to adaptively and dynamically adjust the learning rate, we employ the Adam optimizer to better train the CamVid dataset. The batch size is set to 12, the momentum is set to 0.9, the weight decay is 2×10^{-4}, and the "poly" learning strategy with an initial learning rate of 0.001 is used. Referring to ENet [19], to eliminate the weight imbalance issue on the CamVid dataset, we utilize the following equation.

$$w_{class} = \frac{1}{\ln(c + p_{class})} \tag{11}$$

where w_{class} is the weight, p_{class} is the propensity score and c is an additional hyper-parameter, we set it to 1.10.

Since our proposed overall network does not use any pretrained model, we set the training epoches to 1000. The limited amount of data in the dataset can not meet the application of actual scenarios. We use reasonable data enhancement strategies to make the model better adapt to different data distributions and improve the robustness of the model. Before model training, we used random flipping, mean reduction and random scale scaling to reduce the sensitivity of the model to images and avoid the problem of sample imbalance. Random scale scaling includes $\{0.75, 1.0, 1.25, 1.5, 1.75, 2.0\}$ to improve the generalization performance of the model. Finally, the image is randomly cropped into a fixed size for training. To demonstrate that our network can converge well on Cityscapes and CamVid datasets, referring to MSCFNet [41], we give the training process curves on the above two datasets. Figure 6 plots HDGNet training curves on different datasets. We can see that the two curves decline smoothly, representing that our network can fit the segmentation process well. Specifically, at the beginning of the training, the loss value decreases greatly, which indicates that the learning rate set by us on the Cityscapes or CamVid dataset is suitable for the gradient decline process. The loss value on Cityscapes datasets tends to be flat at epoch = 200, and then starts to decline, which indicates that our learning process is continuing. Because this dataset is very large, it can continue to learn more useful information. On the CamVid dataset, when the epoch is 200, the loss curve has become flat. This is due to the tiny size of the CamVid dataset, and most of the features of the dataset have been learned at this time. From the Figure 6, we can see that there are many burrs on the loss curve of the Cityscapes dataset, and the fluctuation frequency is relatively high. This is because under the same experimental hardware conditions, the batchsize we set on the Cityscapes dataset is relatively small, causing significant fluctuations. The CamVid dataset is relatively small, and the batchsize we set for it is larger, so the fluctuation frequency of the loss curve is smaller.

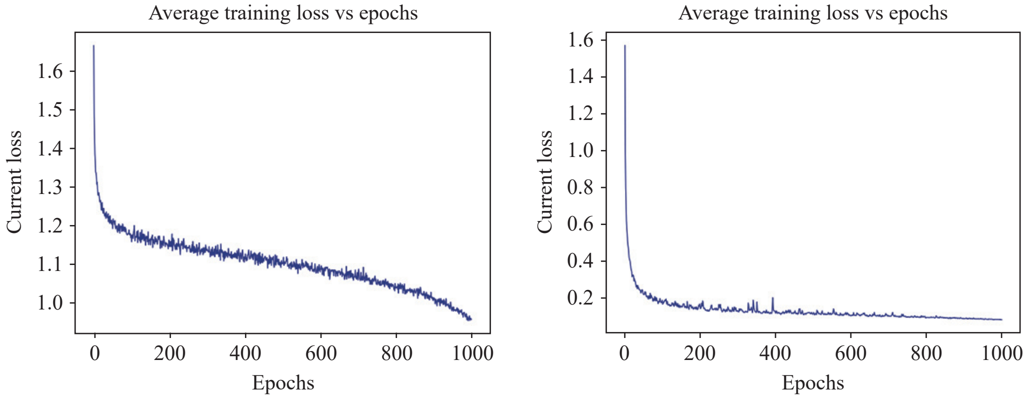


Figure 6. Loss curves of HDGNet on Cityscapes and CamVid datasets.

The proposed method's accuracy is assessed using the mean Intersection over Union (mIoU), a typical measure of semantic segmentation, to give the average of the ratio of the intersection and union of the two sets of all category prediction results and the ground truth, calculated as follows:

$$mIoU = \frac{1}{k}\sum_{i=0}^{k} \frac{p_{ii}}{\sum_{j=0}^{k} p_{ij} + \sum_{j=0}^{k} p_{ji} - p_{ii}} \tag{12}$$

where p_{ij} represents the number of pixels whose real value is i and is predicted to be j, k is the number of class.

Algorithm efficiency is measured by running time and frames per second (FPS). Operation time refers to the time required for the network to complete the segmentation of a single image. FPS represents the number of images the network completes semantic segmentation per second.

6.2. Ablation Study

In this section, we prove the effectiveness of the proposed network through some ablation experiments, which are based on Cityscapes datasets. Figure 8 shows the segmentation visualization results of HDGNet using different modules to prove the effectiveness of each module. For both the context fusion module and the attention module, we adopt dilation rates of [1, Figure 7. Segmentation visualization based on different dilation group settings 2, 5], [1, 2, 5], [1, 2, 5], [3, 5, 7], [3, 5, 7], [3, 5, 7], [5, 9, 11], [5, 9, 11], [5, 9, 11] for ablation experiments. Figure 8(a)(b) are the segmentation visualization result using different context fusion modules and different attention modules, respectively. In the ground truth, the highlighted borders are used to mark the parts with large gaps in the Cityscapes dataset after using different modules. The black parts represent the categories that are ignored in semantic segmentation.

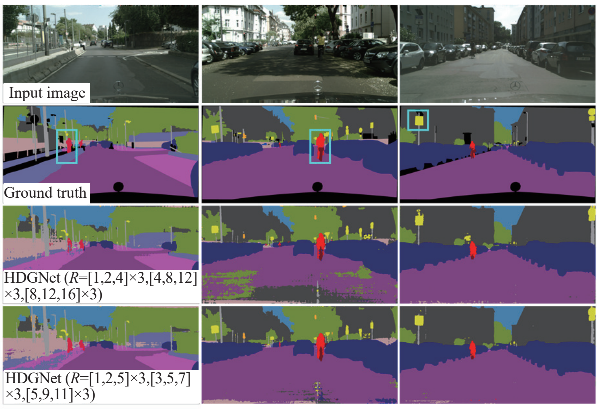
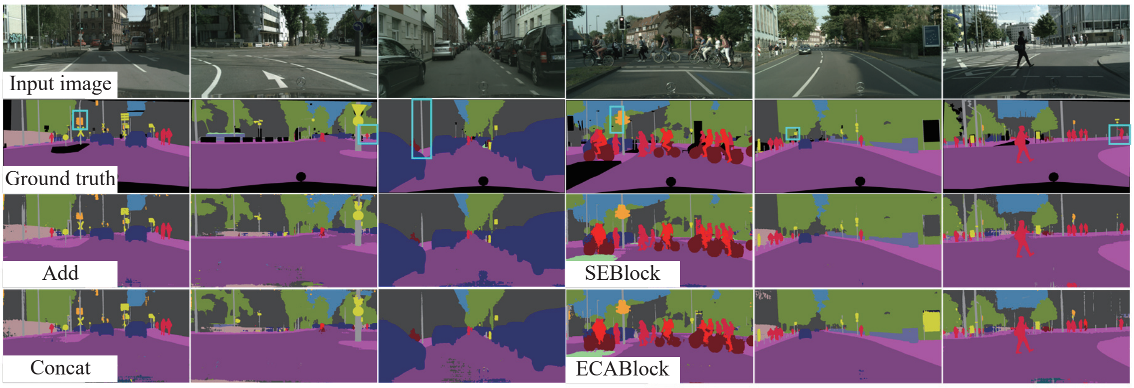

Input image
Ground truth
HDGNet (*R*=[1,2,4]×3,[4,8,12]×3,[8,12,16]×3)
HDGNet (*R*=[1,2,5]×3,[3,5,7]×3,[5,9,11]×3)

Figure 7. Segmentation visualization based on different dilation group settings.



(a) Context fusion module **(b)** Attention module

Figure 8. Segmentation visualization results of HDGNet using different modules.

Context Fusion Module. To investigate the impact of contextual features on segmentation accuracy, we first discard the influence of attention module to conduct experiments. Multi-scale context information is very important for the pixel-level classification task. In terms of context fusion, as can be seen from Table 1(a), feature concatenation has more advantages than feature adding. Although feature concatenation will bring some redundant parameters, there is no obvious loss in training speed. Long-distance skip connections are also often used in semantic segmentation networks to compensate for information loss and reduce the issue of disappearing gradients. To cut down on the amount of parameters, we use feature adding in the skip layer connection. The features of the two middle layers of the network are respectively upsampled and added together to form the final output features. Therefore, in order to strike a trade-off between the segmentation precision and inference speed, the residual connection of input and output is performed by feature concatenation in HDGModule, and we use feature adding in the skip layer connection as the final setting.

Attention Module. The channel attention module is used to capture channel characteristics. Table 1(b) shows the effectiveness of the channel attention module, adding SEBlock can bring about 1.71% improvement in accuracy, while using ECABlock can get a more significant improvement of 2.06% in accuracy with fewer parameters. Experimental results show that by utilizing the attention module we can obtain better segmentation performance without improving the parameter quantity.

Table 1 Results of ablation study based on cityscapes validation set

Model	add	concat	SEBlock	ECABlock	mIoU(%)	Parameters(M)
(a)context fusion module						
HDGNet	√	×	×	×	65.59	0.34
HDGNet	×	√	×	×	69.24	0.68
(b)attention module						
HDGNet	×	√	√	×	70.95	0.69
HDGNet	×	√	×	√	71.30	0.68
(c)dilation rate						
HDGNet(R = [1,2,3]×3,[5,7,9]×3,[11,13,17]×3)	×	√	×	√	70.49	0.68
HDGNet(R = ([1,2,5],[3,5,7],[5,9,11])×3)	×	√	×	√	70.79	0.68
HDGNet(R = [1,2,4]×3,[4,8,12]×3,[8,12,16]×3)	×	√	×	√	69.29	0.69
HDGNet(R = [1,2,5]×3,[3,5,7]×3,[5,9,11]×3)	×	√	×	√	71.30	0.68

Dilation Rate. We can draw a conclusion from the hybrid dilated convolution theory that the dilation rate in each block must not have a common divisor that is higher than 1, and it should be set to a gradually increasing zigzag structure. At the same time, the dilation rate of mutually prime numbers has a better effect. Therefore, we use an increasing dilation rate sequence in the hybrid dilated grouping convolution block, and the dilation rates are [1, 2, 5], [1, 2, 5], [1, 2, 5], [3, 5, 7], [3, 5, 7], [3, 5, 7], [5, 9, 11], [5, 9, 11], [5, 9, 11]. In order to study the effectiveness of the dilation rate parameter sequence scheme, we carry out different dilation rate parameter sequences for comparison, as shown in Table 1(c). When the dilation rate is set to [1, 2, 4], [1, 2, 4], [1, 2, 4], [4, 8, 12], [4, 8, 12], [4, 8, 12], [8, 12, 16], [8, 12, 16], [8, 12, 16], the common divisor of the dilation rate is greater than 1, and the segmentation accuracy is reduced by 2.01%. Moreover, in the segmentation visualization based on different dilation group settings, the experimental results of the second group of sequences are obviously better than the first group. From the second column of Figure 7, we can see that in the segmentation of people and bicycles, the dilation rate sequence of the first group has a common divisor greater than 1, so in the process of capturing multi-scale information, the dilated convolution operation with a dilation rate of 4 and 2 capture repeated spatial information, resulting in discontinuous spatial information and grid effect. The second set of sequences we set can increase the diversity of receptive fields to a greater extent and build a diversity feature map. Therefore, in order to avoid grid effect, a set of dilation rates must not contain a common divisor that is higher than 1. In our network, the assignment of dilation rate follows a sawtooth wave-like heuristic: the dilation rate of the first group is set to 1, 2, 5, and the next group is set to 3, 5, and 7 in the same repeated mode. The overall dilation rate is gradually increasing. By doing so, the final feature map can access effective information from a wider range. In addition, we also carry out two other sets of dilation rate sequence experiments. In our network, neither a larger dilation rate nor a cross dilation rate improved the experimental results. By analyzing the results, it was verified that our settings can lead to better performance.

6.3. Comparison with State-Of-The-Arts

To further validate the effectiveness and generalization of the lightweight network suggested in this article, we compare HDGNet with some existing semantic segmentation methods from different perspectives. The experiments are conducted on the Cityscapes and Camvid datasets. The evaluation results of the proposed algorithm on the above two datasets are first reported, and then using the same conditions, our model is contrasted with alternative semantic segmentation models. The comparison results are shown in Table 3 and 4, and “Pretrain” indicates whether the model has pre-training.

Accuracy and Parameter Comparisons. Our HDGNet exhibits good segmentation performance on two challenging datasets with fewer model parameters. On the Cityscapes dataset, the comparison results with other semantic segmentation algorithms are shown in Table 2 and 3. HDGNet achieves 70.5% in mIoU with only 680,000 parameters on the Cityscapes test set, outperforming existing real-time semantic segmentation algorithms. Table 2 lists the IoU of a single class in the Cityscapes test set, showing good segmentation performance in some classes. Compared to ICNet, we have significant accuracy gains in some classes with large areas. For instance, the IoU of Wall increases from 43.2% to 49.0%. In some categories with few training samples or small areas, our HDGNet’s performance is also better than ICNet, such as Train, Bus, Fence, Sky, Pedestrian and Rider.

Generally speaking, the more parameters, the more complex the model is. Most large models have a huge amount of redundant parameters. Table 3 shows that HDGNet occupies only about 2.6% of ICNet in terms of model parameters without pre-trained model, but the segmentation accuracy of our network has increased by 1% compared with ICNet. Although FSFNet's inference is faster than ours, it is less accurate and has 140,000 more parameters than our model. While EdgeNet outperformed us by 0.5% in accuracy, our model can process 16 more frames per second than it does in terms of inference speed. DFANet has a slightly higher speed, but it takes 11 times as many parameters as our model and its accuracy is slightly lower. In short, the parameters and GFLOPs in Table 3 show that our model can achieve a certain segmentation accuracy with less parameters and low model complexity.

Table 2 Per-class IoU(%) based on Cityscapes test set

Methods	Roa	Sid	Bui	Wal	Fen	Pol	Tli	TSi	Veg	Ter	Sky	Ped	Rid	Car	Tru	Bus	Tra	Mot	Bic	Cla	Cat
SegNet [21]	96.4	73.2	84	28.4	29	35.7	39.8	45.1	87	63.8	91.8	62.8	42.8	89.3	38.1	43.1	44.1	35.8	51.9	57	79.1
ENet [19]	96.3	74.2	75	32.2	33.2	43.4	34.1	44	88.6	61.4	90.6	65.5	38.4	90.6	36.9	50.5	48.1	38.8	55.4	58.3	80.4
ESPNet [26]	97	77.5	76.2	35	36.1	45	35.6	46.3	90.8	63.2	92.6	67	40.9	92.3	38.1	52.5	50.1	41.8	57.2	60.3	82.2
CGNet [42]	95.5	78.7	88.1	40	43	54.1	59.8	63.9	89.6	67.6	92.9	74.9	54.9	90.2	44.1	59.5	25.2	47.3	60.2	64.8	−
ERFNet [20]	**97.7**	**81**	**89.8**	42.5	48	56.3	59.8	65.3	91.4	68.2	94.2	76.8	57.1	92.8	50.8	60.1	51.8	47.3	61.7	68	86.5
ICNet [25]	97.1	79.2	89.7	43.2	48.9	**61.5**	60.4	63.4	**91.5**	68.3	93.5	74.6	56.1	92.6	51.3	**72.7**	51.3	**53.6**	**70.5**	69.5	86.4
HDGNet(ours)	96.3	80.8	88.8	**49**	**50.5**	58.6	**61.7**	**66.1**	91.4	**69**	**94.3**	**80**	**60.5**	**93.7**	**51.4**	70.5	**56.9**	52.8	67.4	**70.5**	**86.9**

Table 3 Evaluation results based on the cityscapes test set

Methods	Pretrain	InputSize	mIoU(%)	FPS	GPU	Parameters(M)	GFLOPs
SegNet [21]	ImageNet	360×640	56.1	14.6	TitanX	29.5	286
ENet [19]	No	512×1024	58.3	76.9	TitanX	0.36	4.4
FSSNet [43]	No	512×1024	58.8	51	TitanXp	**0.2**	−
ESPNet [26]	No	512×1024	60.3	112	TitanX	0.36	4.7
NDNet [44]	No	512×1024	61.1	101.1	Titan X	0.5	3.5
ContextNet [45]	No	1024×2048	66.1	18.3	TitanX	0.85	−
EDANet [32]	No	512×1024	67.3	81.3	TitanX	0.68	8.95
ADSCNet [46]	No	−	67.5	76.9	−	−	−
ERFNet [20]	No	512×1024	68	42	TitanX	2.1	26.86
BSDNet-Xception39 [47]	No	512×1024	68.3	84.6	RTX 2070	1.2	3.45
BiseNet [27]	ImageNet	768×1536	68.4	106	TitanXp	5.8	14.8
FSFNet [48]	No	1024×2048	69.1	**203**	1080Ti	0.82	13.5
DSNet [49]	coarse	360×640	69.3	100	1080Ti	0.9	−
ICNet [25]	ImageNet	1024×2048	69.5	30.3	TitanX	26.5	28.3
AGLNet [50]	No	512×1024	70.1	52	1080Ti	1.12	13.88
Farsee [51]	ImageNet	512×1024	70.2	68.5	TitanX	−	−
DFANet [52]	ImageNet	512×1024	70.3	160	TitanX	7.8	1.7
EdgeNet [53]	No	512×1024	71	31.4	TitanX	−	−
HDGNet(ours)	No	512×1024	70.5	48	1080Ti	0.68	10.69

To demonstrate that our network is effective not only in high-resolution images, such as Cityscapes dateset, but also in low-resolution images, we conduct experiments in the CamVid datasets. It can process 360×480 CamVid images at a speed of 110FPS. As shown in 4, on the CamVid test set, our HDGNet achieves 65.7% mIoU, which is better than most current semantic segmentation algorithms. ICNet sacrifices a huge number of model parameters in exchange for accuracy. Although its accuracy is improved slightly, it takes about 39 times as many parameters as our model. As seen in Tables 3 and 4, our network successfully strikes a trade-off between inference speed and precision.

Table 4 Evaluation results based on the CamVid test set

Methods	Pretrain	mIoU(%)	Parameters(M)
ENet [19]	No	51.3	0.36
ESPNet [26]	No	55.6	0.36
SegNet [21]	ImageNet	55.6	29.5
FCN-8s [15]	ImageNet	57	134.5
NDNet [44]	No	57.2	0.5
EFSNet [55]	No	60.7	**0.1**
BSDNet-Xception39 [47]	No	63.9	1.2
DFANet [52]	ImageNet	64.7	7.8
Dilation8 [56]	ImageNet	65.3	140.8
CGNet [42]	No	65.6	0.5
BiseNet [27]	ImageNet	65.6	5.8
ICNet [25]	ImageNet	**67.1**	26.5
HDGNet(ours)	No	65.7	0.68

Speed Comparison. The speed experiments compared with other state-of-the-art methods refer to DABNet [54]. For the fairness of the experiment, a single 1080Ti GPU is utilized for all speed experiments, and conduct experiments in a unified Pytorch framework. Table 5 shows the comparison of inference speed between HDGNet and other existing networks using full, half, and quarter resolution of images in Cityscapes respectively. The HDGNet can process a 512×1024 image at a speed of 48FPS. Although ESPNet is faster than HDGNet, its accuracy on Cityscapes datasets is 10.2% lower than ours, sacrificing too much accuracy. The speed comparison experiments show that our HDGNet can not only process low-resolution images, but also hign-resolution images at a relatively fast speed.

Table 5 Speed comparison (all operations on a single GTX 1080Ti)

	GTX 1080Ti					
Model	256 × 512		512 × 1024		1024 × 2048	
	ms	**fps**	**ms**	**fps**	**ms**	**fps**
SegNet [21]	16	64.2	56	17.9	–	–
ERFNet [20]	7	147.8	21	48.2	74	13.5
ENet [19]	10	99.8	13	74.9	44	22.9
ICNet [25]	9	107.9	15	67.2	40	25.1
ESPNet [26]	5	182.5	9	115.2	30	33.3
HDGNet(ours)	9	112	21	48	80	12.6

The network proposed in this paper combines the advantages of factorization convolution, depthwise separable convolution, dilated convolution and residual connection to build the core module of Hybrid Dilated Grouping Network, that is, HDG module. By taking advantage of the features of factorization convolution and depthwise separable convolution, the computing load of the module is reduced. At the same time, the HDG module is used to solve the grid effect caused by traditional dilated convolution by setting different dilated rates, so as to better capture spatial context information at different scales. In order to improve the performance of the network and enhance the ability of feature expression, we introduce a lightweight attention module in the encoder stage of the network to capture the information correlation between channels. Figure 9 demonstrates that our network has higher segmentation precision than other methods with fewer parameters, and can achieve a certain inference speed, which well proves that our network can balance the segmentation precision and inference speed.

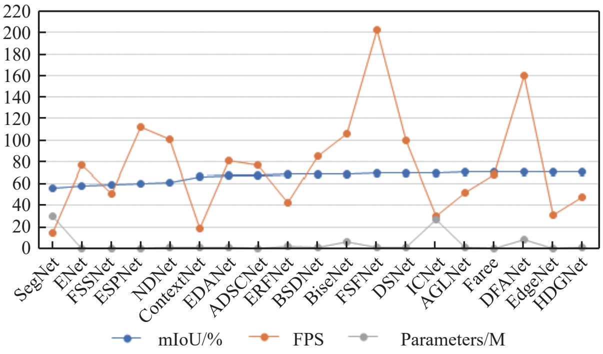


Figure 9. Speed, accuracy and parameters performance comparison on the Cityscapes test set.

7. Conclusions

We propose a real-time outdoor scene segmentation algorithm based on HDGNet, which maintains the image segmentation accuracy while taking into account the lightness and real-time performance of the algorithm model. The designed hybrid dilated grouping module uses factorization convolution, depth-wise separable convolution, and dilated convolution to obtain more valid context information. To capture more channel features of the model, a channel attention module is introduced to capture the inter-channel information. In addition, the HDGNet has feature branches from different stages, and skip-layer connections are used to connect features from distinct branches, so that the shallow features and deep high-level semantic information are fused, which enhances the feature representation, extremely promotes the interaction of local information and global information, and further optimizes the segmentation results. Experiments show that, compared with several other methods, our model can achieve a better balance between efficiency and accuracy. We will investigate the scene semantic segmentation approach on the mobile platform in future work to make the algorithm more suitable to completing the driving scene segmentation task.

Our method performs well on the Cityscapes and CamVid datasets, but there is room for further improvement and optimization. First of all, since the proposed method is only tested on two types of samples, it is not suitable for some specific samples. Therefore, we will try to combine the network with specific image targets in the future research to improve its practical application effect. Secondly, the network structure designed in this paper is an asymmetric structure of large encoder and small decoder, and the decoding level does not use complex modules to recover image information, so there is still a lack of structural design in restoring image resolution. In the follow-up research, the decoder can be more refined design to further improve the segmentation accuracy of the model. Finally, we will study the semantic scene segmentation method on the mobile platform in the future, so as to make the algorithm more suitable for the task of driving scene segmentation.

Author Contributions: Yan zhang: Data curation, formal analysis, investigation, methodology, software, valida-

tion, visualization, writing – original draft, writing – review & editing. **Xuguang Zhang:** Conceptualization, formal analysis, investigation, methodology, project administration, supervision, validation, writing – original draft, writing – review & editing. **Deting Miao:** investigation, methodology, software, writing – original draft, writing – review & editing. **Hui Yu:** Conceptualization, supervision, validation, writing – original draft, writing – review & editing.

Funding: National Natural Science Foundation of China (no.61771418).

Data Availability Statement: The datasets analyzed during the current study are available in the publicly archived datasets. Cityscapes: https://www.cityscapes-dataset.com/downloads. Camvid: http://mi.eng.cam.ac.uk/research/projects/VideoRec/CamVid.

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

References

1. Zhao, H.S.; Shi, J.P.; Qi, X.J.; *et al*. Pyramid scene parsing network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017*; IEEE: New York, **2017**; pp. 6230–6239. doi:10.1109/CVPR.2017.660
2. Yu, N.X.; Yang, R.; Huang, M.J. Deep common spatial pattern based motor imagery classification with improved objective function. Int. J. Netw. Dyn. Intell., **2022**, *1*: 73−84. doi: 10.53941/ijndi0101007
3. Otsu, N. A threshold selection method from gray-level histograms. IEEE Trans. Syst., Man, Cybern., **1979**, *9*: 62−66. doi: 10.1109/TSMC.1979.4310076
4. Comaniciu, D.; Meer, P. Mean shift: A robust approach toward feature space analysis. IEEE Trans. Pattern Anal. Mach. Intell., **2002**, *24*: 603−619. doi: 10.1109/34.1000236
5. Achanta, R.; Shaji, A.; Smith, K.; *et al*. SLIC superpixels compared to state-of-the-art superpixel methods. IEEE Trans. Pattern Anal. Mach. Intell., **2012**, *34*: 2274−2282. doi: 10.1109/TPAMI.2012.120
6. Rother, C.; Kolmogorov, V.; Blake, A. "GrabCut": Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graphics* (*TOG*) **2004**, *23*, 309–314. doi:10.1145/1015706.1015720
7. Li, M.C.; Wang, Z.D.; Li, K.L.; *et al*. Task allocation on layered multiagent systems: When evolutionary many-objective optimization meets deep Q-learning. IEEE Trans. Evol. Comput., **2021**, *25*: 842−855. doi: 10.1109/TEVC.2021.3049131
8. Liu, W.B.; Wang, Z.D.; Zeng, N.Y.; *et al*. A novel randomised particle swarm optimizer. Int. J. Mach. Learn. Cybern., **2021**, *12*: 529−540. doi: 10.1007/s13042-020-01186-4
9. Alicja, K.; Maciej, S. Can AI see bias in X-ray images. Int. J. Netw. Dyn. Intell., **2022**, *1*: 48−64. doi: 10.53941/ijndi0101005
10. Zhao, G.Y.; Li, Y.T.; Xu, Q.R. From emotion AI to cognitive AI. Int. J. Netw. Dyn. Intell., **2022**, *1*: 65−72. doi: 10.53941/ijndi0101006
11. Xu, X.; Zhang, J.R.; Li, Y.J.; *et al*. Adversarial attack against urban scene segmentation for autonomous vehicles. IEEE Trans. Ind. Inf., **2021**, *17*: 4117−4126. doi: 10.1109/TII.2020.3024643
12. Li, X.; Duan, H.B.; Mo, H.; *et al*. A novel visual perception framework for unmanned aerial vehicles: Challenges and approaches. In *Proceedings of 2021 China Automation Congress* (*CAC*), *Beijing, China, 22–24 October 2021*; IEEE: New York, 2021; pp. 8359–8363. doi:10.1109/CAC53003.2021.9727934
13. Ahmed, I.; Din, S.; Jeon, G.; *et al*. Towards collaborative robotics in top view surveillance: A framework for multiple object tracking by detection using deep learning. IEEE/CAA J. Autom. Sinica, **2021**, *8*: 1253−1270. doi: 10.1109/JAS.2020.1003453
14. Dong, G.S.; Yan, Y.; Shen, C.H.; *et al*. Real-time high-performance semantic image segmentation of urban street scenes. IEEE Trans. Intell. Transp. Syst., **2021**, *22*: 3258−3274. doi: 10.1109/TITS.2020.2980426
15. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 07–12 June 2015*; IEEE: New York, 2015; pp. 3431–3440. doi:10.1109/CVPR.2015.7298965
16. Chen, L.C.; Papandreou, G.; Kokkinos, I.; *et al*. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Trans. Pattern Anal. Mach. Intell., **2018**, *40*: 834−848. doi: 10.1109/TPAMI.2017.2699184
17. Fu, J.; Liu, J.; Tian, H.J.; *et al*. Dual attention network for scene segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019*; IEEE: New York, 2019; pp. 3141–3149. doi:10.1109/CVPR.2019.00326
18. Shakiba, F.M.; Shojaee, M.; Azizi, S.M.; *et al*. Real-time sensing and fault diagnosis for transmission lines. Int. J. Netw. Dyn. Intell., **2022**, *1*: 36−47. doi: 10.53941/ijndi0101004
19. Paszke, A.; Chaurasia, A.; Kim, S.; *et al*. ENet: A deep neural network architecture for real-time semantic segmentation. arXiv: 1606.02147, 2016
20. Romera, E.; Alvarez, J.M.; Bergasa, L.M.; *et al*. ERFNet: Efficient residual factorized ConvNet for real-time semantic segmentation. IEEE Trans. Intell. Transp. Syst., **2018**, *19*: 263−272. doi: 10.1109/tits.2017.2750080
21. 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. doi: 10.1109/TPAMI.2016.2644615
22. Lian, X.H.; Pang, Y.W.; Han, J.G.; *et al*. Cascaded hierarchical atrous spatial pyramid pooling module for semantic segmentation. Pattern Recognit., **2021**, *110*: 107622. doi: 10.1016/j.patcog.2020.107622
23. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017*; IEEE: New York, **2017**; pp. 1800–1807. doi:10.1109/CVPR.2017.195
24. Howard, A.G.; Zhu, M.L.; Chen, B.; *et al*. MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv: 1704.04861, 2017

25. Zhao, H.S.; Qi, X.J.; Shen, X.Y.; *et al*. ICNet for real-time semantic segmentation on high-resolution images. In *Proceedings of the 15th European Conference on Computer Vision* (*ECCV*), *Munich, Germany, 8–14 September 2018*; Springer: Berlin/Heidelberg, 2018; pp. 418–434. doi:10.1007/978-3-030-01219-9_25
26. Mehta, S.; Rastegari, M.; Caspi, A.; *et al*. ESPNet: Efficient spatial pyramid of dilated convolutions for semantic segmentation. In *Proceedings of the 15th European Conference on Computer Vision* (*ECCV*), *Munich, Germany, 8–14 September 2018*; Springer: Berlin/Heidelberg, 2018; pp. 561–580. doi:10.1007/978-3-030-01249-6_34
27. Yu, C.Q.; Wang, J.B.; Peng, C.; *et al*. BiSeNet: Bilateral segmentation network for real-time semantic segmentation. In *Proceedings of the 15th European Conference on Computer Vision* (*ECCV*), *Munich, Germany, 8–14 September 2018*; Springer: Berlin/Heidelberg, 2018; pp. 334–349. doi:10.1007/978-3-030-01261-8_20
28. Szegedy, C.; Liu, W.; Jia, Y.Q.; *et al*. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015*; IEEE: New York, 2015; pp. 1–9. doi:10.1109/CVPR.2015.7298594
29. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv:1502.03167, 2015
30. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; *et al*. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016*; IEEE: New York, 2016; pp. 2818–2826. doi:10.1109/CVPR.2016.308
31. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; *et al*. Inception-v4, inception-ResNet and the impact of residual connections on learning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, California, USA, 4–9 February 2017*; AAAI Press: Washington, DC, USA, 2017; pp. 4278–4284
32. Lo, S.Y.; Hang, H.M.; Chan, S.W.; *et al*. Efficient dense modules of asymmetric convolution for real-time semantic segmentation. In *Proceedings of the 1st ACM International Conference on Multimedia in Asia, Beijing, China, 15–18 December 2019*; ACM: New York, 2019; p. 1. doi:10.1145/3338533.3366558
33. Li, G.; Yun, I.; Kim, J.; *et al*. DABNet: Depth-wise asymmetric bottleneck for real-time semantic segmentation. arXiv: 1907.11357, 2019
34. Wang, Y.; Zhou, Q.; Liu, J.; *et al*. Lednet: A lightweight encoder-decoder network for real-time semantic segmentation. In *Proceedings of 2019 IEEE International Conference on Image Processing* (*ICIP*), *Taipei, China, 22–25 September 2019*; IEEE: New York, 2019; pp. 1860–1864. doi:10.1109/ICIP.2019.8803154
35. Huang, Z.L.; Wang, X.G.; Huang, L.C.; *et al*. CCNet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea* (*South*), *27 October 2019–2 November 2019*; IEEE: New York, 2019; pp. 603–612. doi:10.1109/ICCV.2019.00069
36. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018*; IEEE: New York, 2018; pp. 7132–7141. doi:10.1109/CVPR.2018.00745
37. Woo, S.; Park, J.; Lee, J.Y.; *et al*. CBAM: Convolutional block attention module. In *Proceedings of the 15th European Conference on Computer Vision* (*ECCV*), *Munich, Germany, 8–14 September 2018*; Springer: Berlin/Heidelberg, 2018; pp. 3–19. doi:10.1007/978-3-030-01234-2_1
38. Wang, Q.L.; Wu, B.G.; Zhu, P.F.; *et al*. ECA-Net: Efficient channel attention for deep convolutional neural networks. In *Proceedings of 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), *Seattle, WA, USA, 13–19 June 2020*; IEEE: New York, 2020; pp. 11531–11539. doi:10.1109/CVPR42600.2020.01155
39. Wang, P.Q.; Chen, P.F.; Yuan, Y.; *et al*. Understanding convolution for semantic segmentation. In *Proceedings of 2018 IEEE Winter Conference on Applications of Computer Vision* (*WACV*), *Lake Tahoe, NV, USA, 12–15 March 2018*; IEEE: New York, 2018; pp. 1451–1460. doi:10.1109/WACV.2018.00163
40. Wu, H.S.; Liang, C.X.; Liu, M.S.; *et al*. Optimized HRNet for image semantic segmentation. Expert Syst. Appl., **2021**, *174*: 114532. doi: 10.1016/j.eswa.2020.114532
41. Gao, G.W.; Xu, G.A.; Yu, Y.; *et al*. MSCFNet: A lightweight network with multi-scale context fusion for real-time semantic segmentation. IEEE Trans. Intell. Transp. Syst., **2021**, *23*: 25489−25499. doi: 10.1109/TITS.2021.3098355
42. Wu, T.Y.; Tang, S.; Zhang, R.; *et al*. CGNet: A light-weight context guided network for semantic segmentation. IEEE Trans. Image Process., **2021**, *30*: 1169−1179. doi: 10.1109/TIP.2020.3042065
43. Zhang, X.T.; Chen, Z.X.; Wu, Q.M.J.; *et al*. Fast semantic segmentation for scene perception. IEEE Trans. Ind. Inf., **2019**, *15*: 1183−1192. doi: 10.1109/TII.2018.2849348
44. Yang, Z.G.; Yu, H.S.; Fu, Q.; *et al*. NDNet: Narrow while deep network for real-time semantic segmentation. IEEE Trans. Intell. Transp. Syst., **2021**, *22*: 5508−5519. doi: 10.1109/TITS.2020.2987816
45. Poudel, R.P.K.; Bonde, U.; Liwicki, S.; *et al*. ContextNet: Exploring context and detail for semantic segmentation in real-time. In *Proceedings of the British Machine Vision Conference 2018, Newcastle, UK, 3–6 September 2018*; BMVA: Durham, UK, 2018
46. Wang, J.W.; Xiong, H.Y.; Wang, H.B.; *et al*. ADSCNet: Asymmetric depthwise separable convolution for semantic segmentation in real-time. Appl. Intell., **2020**, *50*: 1045−1056. doi: 10.1007/s10489-019-01587-1
47. Ye, L.; Zeng, J.X.; Yang, Y.; *et al*. BSDNet: Balanced sample distribution network for real-time semantic segmentation of road scenes. IEEE Access, **2021**, *9*: 84034−84044. doi: 10.1109/ACCESS.2021.3087510
48. Kim, M.; Park, B.; Chi, S. Accelerator-aware fast spatial feature network for real-time semantic segmentation. IEEE Access, **2020**, *8*: 226524−226537. doi: 10.1109/ACCESS.2020.3045147
49. Wang, W.F.; Fu, Y.J.; Pan, Z.J.; *et al*. Real-time driving scene semantic segmentation. IEEE Access, **2020**, *8*: 36776−36788. doi: 10.1109/ACCESS.2020.2975640
50. Zhou, Q.; Wang, Y.; Fan, Y.W.; *et al*. AGLNet: Towards real-time semantic segmentation of self-driving images via attention-guided lightweight network. Appl. Soft Comput., **2020**, *96*: 106682. doi: 10.1016/j.asoc.2020.106682
51. Zhang, Z.P.; Zhang, K.P. FarSee-Net: Real-time semantic segmentation by efficient multi-scale context aggregation and feature space super-resolution. In *Proceedings of 2020 IEEE International Conference on Robotics and Automation* (*ICRA*), *Paris, France, 31 May 2020–31 August 2020*; IEEE: New York, 2020; pp. 8411–8417. doi:10.1109/ICRA40945.2020.9196599
52. Li, H.C.; Xiong, P.F.; Fan, H.Q.; *et al*. DFANet: Deep feature aggregation for real-time semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019*; IEEE: New York, 2019; pp. 9514–9523. doi:10.1109/CVPR.2019.00975

53. Han, H.Y.; Chen, Y.C.; Hsiao, P.Y.; *et al*. Using channel-wise attention for deep CNN based real-time semantic segmentation with class-aware edge information. IEEE Trans. Intell. Transp. Syst., **2021**, *22*: 1041−1051. doi: 10.1109/TITS.2019.2962094
54. Li, G.; Jiang, S.L.; Yun, I.; *et al*. Depth-wise asymmetric bottleneck with point-wise aggregation decoder for real-time semantic segmentation in urban scenes. IEEE Access, **2020**, *8*: 27495−27506. doi: 10.1109/ACCESS.2020.2971760
55. Hu, X.G.; Wang, H.B. Efficient fast semantic segmentation using continuous shuffle dilated convolutions. IEEE Access, **2020**, *8*: 70913−70924. doi: 10.1109/ACCESS.2020.2987080
56. Yu, F.; Koltun, V. Multi-scale context aggregation by dilated convolutions. In *Proceedings of the 4th International Conference on Learning Representations, San Juan, Puerto Rico, 2–4 May 2016*; ICLR: San Juan, Puerto Rico, 2016

Citation: Zhang, Y.; Zhang, X.; Miao, D.; *et al*. Real-Time Semantic Segmentation of Road Scenes via Hybrid Dilated Grouping Network. *International Journal of Network Dynamics and Intelligence*. 2025, 4(1), 100006. doi: 10.53941/ijndi.2025.100006