Computing and Informatics, Vol. 43, 2024, 849–873, doi: 10.31577/cai_2024_4_849

MIDWRSEG: ACQUIRING ADAPTIVE MULTI-SCALE CONTEXTUAL INFORMATION FOR ROAD-SCENE SEMANTIC SEGMENTATION

Bing Su, Peng Jin, Yifeng Lin, Fuyang Wang

Changzhou University
School of Computer Science and Artifical Intelligence
No. 2468, YanZeng West Rd., Wujin District
Changzhou City, Jiangsu Province, China
e-mail: s22150812024@smail.cczu.edu.cn

Abstract. We present MIDWRSeg, a simple semantic segmentation model based on neural network architecture. For complex road scenes, a large receptive field gathered at multiple scales is crucial for semantic segmentation tasks. Currently, there is an urgent need for the CNN architecture to establish long-range dependencies (large receptive fields) akin to the unique attention mechanism employed by the Transformer architecture. However, the high complexity of the attention mechanism formed by the matrix operations of Query, Key and Value cannot be borne by real-time semantic segmentation models. Therefore, a Multi-Scale Convolutional Attention (MSCA) block is constructed using inexpensive convolution operations to form long distance dependencies. In this method, the model adopts a Simple Inverted Residual (SIR) block for feature extraction in the initial encoding stage. After downsampling, the feature maps with reduced resolution undergo a sequence of stacked MSCA blocks, resulting in the formation of multi-scale long-range dependencies. Finally, in order to further enrich the size of the adaptive receptive field, an Internal Depth Wise Residual (IDWR) block is introduced. In the decoding stage, a simple decoder similar to FCN is used to alleviate computational consumption. Our method has formed a competitive advantage with existing real-time semantic segmentation models for encoder-decoder on Cityscapes and CamVid datasets. Our MIDWRSeg achieves 74.2 % mIoU at a speed of 88.9 FPS at Cityscapes test and achieves 76.8 % mIoU at a speed of 95.2 FPS at CamVid test.

Keywords: Deep convolutional network, attention mechanism, semantic segmentation, autonomous driving

1 INTRODUCTION

Semantic segmentation essentially involves classifying pixels. Unlike image classification, it poses enormous challenges as a pixel-intensive task. This basic task is a prerequisite for achieving autonomous driving and virtual reality. When faced with complex road scenes populated by people, cars, trees, and other elements, the manual extraction of features often seems inadequate. Fortunately, with the development of hardware, computing power has rapidly improved, and neural networks that extract features in a black-box manner have regained their vitality once again. This has freed us from the inconvenience of manually extracting features.

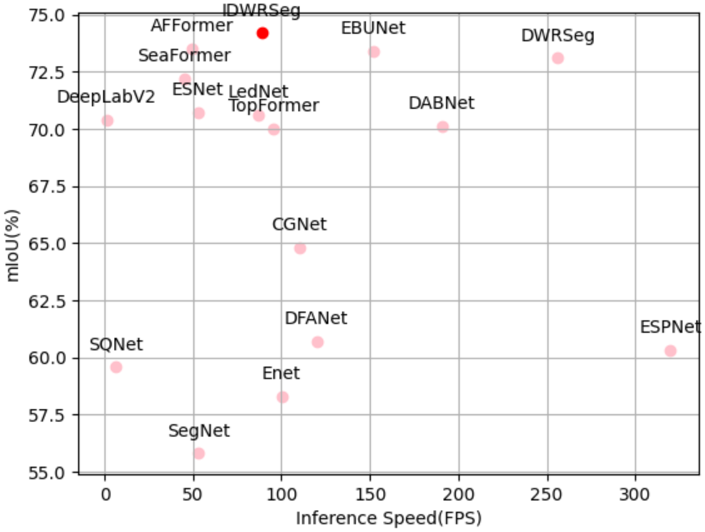

Figure 1. A comparison of speed-accuracy trade-off on Cityscapes test set

The creative model of fully convolutional neural networks (FCN) [1] has significantly advanced the fundamental task of semantic segmentation. However, FCN heavily relies on classification networks, merely converting the final fully connected layer into a 1 × 1 convolution. This adjustment causes the feature maps obtained in the final three stages to be excessively small. To address this, interpolation is employed to resize them back to the original image dimensions. Unfortunately, this process often leads to the loss of crucial spatial information. The DeepLab [2, 3, 4] series proposes dilated convolution to maintain the size of feature maps in subsequent stages, while also obtaining a considerable receptive field. However, improper setting of the dilation rate results in gridding artifacts in the predicted image. The model proposed by U-Net [5] for medical images performs upsampling step by step in decoding stages and then concatenates the features from the the preceding encoding stage along the channel dimension. By fully utilizing multi-scale features in each stage, the final segmentation result exhibits significantly improved accuracy. However, this multi-stage feature fusion comes with a significant computational cost. Based on the excellent works mentioned above, it is evident that global receptive fields and multi-scale feature learning are crucial for semantic segmentation. The question then arises: how can we effectively capture the global receptive field? Inspired by the self attention mechanism in the Transformer [6] architecture, its remarkable capability in globally modeling sequences in natural language processing is noteworthy. However, the high complexity of the attention mechanism, arising from matrix operations involving Query, Key, and Value, poses a significant computational burden. The Vision Transformer (ViT) [7] is the first to apply the Transformer architecture to computer vision tasks. It achieves this by dividing the input image into 16×16 patches, which are then transformed into tokens via patch embedding operations, enabling self-attention with manageable complexity. This work strongly demonstrates the practicality of the self-attention mechanism in computer vision, yet its time complexity remains prohibitive for real-time systems. This has prompted the exploration of inexpensive convolutional attention mechanisms, among which SegNeXt [8] proposed the Multi-Scale Convolutional Attention (MSCA) block to form multi-scale effective receptive fields. The design of our model integrates the MSCA block into the encoder part.

In the final stage of encoding, we integrate the Depth-Wise Residual (DWR) block from DWRSeg [9]. The feature maps are divided channel-wise, followed by the application of dilated convolutions with different ratios to further fuse features across scales.

Our motivation for adopting the MSCA and DWR blocks stems from the fact that most current segmentation models are simply derived from existing classification models, such as VGG [10] and ResNet [11], by discarding the final classification layer and using the features from each stage to perform segmentation through a decoding process similar to FCN. However, due to the simple stacking of similar modules, the overall feature extraction capability of these models is limited.

Taking ResNet18 as an example, it starts with a 7×7 convolution layer with a stride of 2 and padding of 3, followed by two stacked BasicBlocks in each of the four stages. These stages are identical and do not have distinct responsibilities, such as maintaining spatial information in the first stage, capturing multi-scale information in the second and third stages, and acquiring contextual information in the fourth stage. Even when a spatial pyramid pooling block is added in the final stage to aggregate contextual information, there is still a significant sawtooth effect along the segmentation boundaries, resulting in inaccurate segmentation. See Figure 2 for illustration.

The contributions made in this article are as follows:

1. In the initial encoding stage, a Simple Inverted Residual (SIR) block is used for noise reduction, which facilitates the effective formation of long-distance

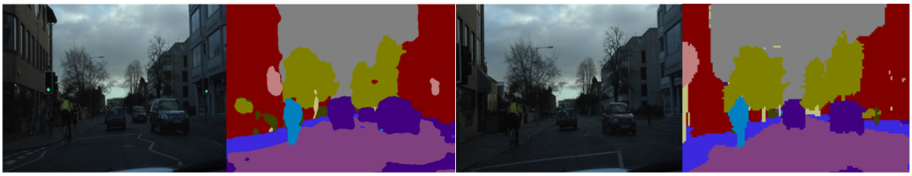

Figure 2. FCN based on ResNet18. On the left, there is no use of a spatial pyramid module to aggregate contextual information, resulting in poor boundary segmentation. On the right, a spatial pyramid module is used to aggregate contextual information. However, due to the weak feature extraction capability of the backbone, a severe sawtooth effect is produced.

dependencies in the subsequent MSCA blocks. Both of them jointly employ inexpensive convolution operations to achieve efficient receptive fields.

2. We introduce the Internal Depth-wise Residual (IDWR) block by incorporating internal residual connections into the DWR block. The IDWR block leverages a relatively rich multi-scale receptive field. Owing to the high efficiency of feature extraction in the early stages, the context aggregation module is excluded, thereby reducing the model's parameter size and indirectly enhancing inference speed.
3. Our model clarifies the responsibilities of each stage. The first stage maintains a larger feature map to retain more spatial information. The second and third stages rely on three-branch asymmetric convolution to adapt to different segmentation target sizes. The fourth stage relies on the IDWR block to achieve the aggregation of context information, making the target boundary segmentation accurate. A strong encoder we implemented is a prerequisite for successful semantic segmentation. Please refer to Figure 1. Our MIDWRSeg achieves an mIoU of 74.2 % at a speed of 88.9 FPS on the Cityscapes test and an mIoU of 76.8 % at a speed of 95.2 FPS on the CamVid test.

2 RELATED WORK

2.1 Attention Mechanism and Large Convolutional Kernel

Vision Transformer (Vit) [7] successfully utilizes self attention mechanism through patch embedding operation and achieves impressive results in image classification tasks. However, Transformers based on global attention mechanisms generally require a large amount of computation. Swin Transformer [12] proposes a self attention mechanism that includes sliding windows, which can introduce the local characteristics of CNN and save computation. In the semantic segmentation task, SeaFormer [13] follows a similar encoding approach as Vit [7]. However, it has introduced three distinct decoding methods: Naive Upsampling, Progressive Upsampling, and Multi Level Feature Aggregation. The segmentation accuracy has been greatly improved, but the number of model parameters has also increased significantly. The deployment of lightweight models remains challenging, resulting in the development of lightweight convolutional attention modules that employ simplified convolutional structures. Typical works in the field of attention modules include SE (Squeeze and Excitation attention) [14], CBAM (Convolutional Block Attention Module) [15], and CA (Coordinate Attention) [16]. These modules can be seamlessly integrated into various convolutional architectures.

Recently, some other works have reassessed the role of large convolutions, suggesting that traditional convolutional architectures can obtain long-range dependencies (large receptive fields) similar to attention mechanisms, provided that the convolution kernel is large enough. RepLKNet [17] efficiently utilizes 31×31 convolutions through the use of short cuts, re-parameterization, and optimized low-level depth-wise convolutions. On the other hand, SLaK [18] smoothly extends the convolution kernel to 51×51 by utilizing two parallel, rectangular convolutions instead of square large convolutions. The dynamic sparsity of the convolution kernel helps greatly improve the model capacity without increasing the model size.

In order to combine the advantages of attention mechanism and large convolutional kernels, the multi-scale convolutional attention module from SegNeXt [7] was adopted in the subsequent encoding stage of our model, with kernel sizes of 7, 11, and 21, respectively, to generate Q, K, and V to form attention and strengthen key information.

2.2 Real-Time Encoder-Decoder Architecture

The SegNet [19], proposed by Badrinarayanan et al., adopts an encoder-decoder structure and records the position index and value of the max pooling during each encoding process. It is used to restore feature maps during upsampling to save computational resources, thus achieving real-time requirements. ENet [20] still employs an encoder-decoder structure, characterized by its asymmetric design. In this structure, the encoder part dominates, being significantly larger than the decoder part, thus maintaining a compact model size and enhancing segmentation speed. To further optimize its parameters and introduce more nonlinearity, asymmetric convolution [21] is utilized, while dilated convolution is employed to expand the receptive field. ERFNet [22] is similar to ENet [20] in that it also employs asymmetric convolutions, decomposing all (3×3) convolutions in the entire network into (3×1) and (1×3) convolutions. LinkNet [23] shares a similar structure to Unet [5], but requires fewer layers for feature extraction. During feature fusion, it adopts element-wise addition instead of channel-wise addition as in Unet [5]. DABNet [24] proposes a new Deep Asymmetric Bottleneck (DAB) module that effectively utilizes asymmetric convolution and dilated convolution to construct bottleneck layers. The DABNet, composed of DAB modules, generates sufficient receptive domains and densely utilizes contextual information.

2.3 Real-Time Multiple-Branches Architecture

To address the issue of tight coupling between spatial and channel information in the encoder-decoder architecture, ICNet [25] employs a three-branch multi-resolution encoding structure. It utilizes 1/4 of the original image resolution as the primary means for extracting semantic information, while the full resolution (1/1) and half resolution (1/2) images compensate for the loss of spatial information. ContextNet [26] utilizes two branches to effectively extract spatial and contextual features, reducing the redundancy of the branches. BiSeNet [27] is a dual-branch network that operates with the input being the original image resolution. It features specially designed spatial and contextual paths, incorporating a Feature Fusion Module (FFM) and an Attention Refinement Module (ARM) for upsampling.

BiSeNetV2 [28] continues the dual branch structure of the v1 version, divided into a Detail Branch and a Semantic Branch. For the Detail Branch, a VGG [10] like network structure is still used for fast downsampling. The Semantic Branch runs parallel to the Detail Branch and is primarily designed to capture advanced semantic information. Due to the fact that detailed information can be supplemented by the Detail Branch, the number of channels in the Semantic Branch is strictly controlled by the parameter α to avoid redundancy.

STDCNet [29] employs Laplacian convolution on labels for the extraction of fine edge features. By utilizing densely connected STDC blocks as its backbone, the network eliminates redundant spatial paths, thereby compressing the model and enhancing both segmentation speed and accuracy.

DDRNet [10] introduces a novel bilateral network as an efficient backbone for real-time semantic segmentation. This network features deep dual resolution branches and multiple bilateral fusions. Additionally, a module has been devised to acquire abundant contextual information by integrating feature aggregation with pyramid pooling technology. Notably, when processing low-resolution feature maps, it requires minimal additional inference time.

3 METHOD

In this section, we will describe the architecture of our entire model, MIDWRSeg. The model maintains a mature and stable encoder-decoder architecture, which facilitates the reproduction of the entire model. We will first provide an overview of the overall architecture, followed by a detailed introduction of the individual blocks in the order of their appearance in the model.

3.1 Overall Architecture

Figure 3 shows the overall architecture of the MIDWRSeg. The entire model consists of four stages in the encoding, with the Stem block and SIR block as the first stage, completing denoising and primary feature extraction operations. In the second and

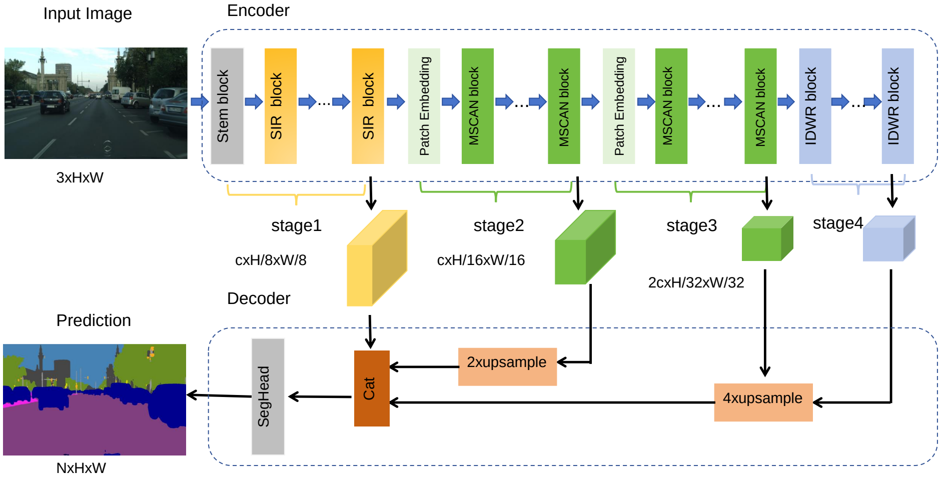

Figure 3. The overview of MIDWRSeg on segmentation. The overall architecture of the model employs an encoder-decoder structure. The first stage includes a Stem block and a Simple Inverted Residual (SIR) block, designed to maintain spatial information. The second, third stage involves Patch Embedding operations and the Muti-Scale Convolution Attention (MSCA) block, aimed at capturing multi-scale semantic information. The fourth stage incorporates the Internal Depth-Wise Residual (IDWR) block, which utilizes internal three-branch residual connections to acquire sufficient receptive fields and contextual information. The entire decoding stage resembles FCN. H and W denote the height and width of the feature map, while c represents the number of channels in the feature map. N denotes the number of classes.

third stages, the MSCA block is used to form dependencies on long distances and multi-scale features. In the fourth stage, the IDWR block is introduced to combine richer receptive fields and form adaptive regions based on internal residuals. The SegHead undergoes the conv1 and conv2 to the final number of classes N. The detailed description of the model is shown in Table 1.

3.2 Stem Block

The Stem Block is depicted in Figure 4. Firstly, the 3-channel input image f_{in} will undergo a 3×3 convolution with a stride of 2 for preliminary feature extraction. Subsequently, the resulting feature map f_{stage1} will be processed through two branches. The formula is described as follows:

$$f_{stage1} = \mathrm{Conv}_{k=3\times3}^{s=2}(f_{in}). \tag{1}$$

The f_{left} branch first relies on 1×1 convolution to further compress the number of channels, thereby alleviating computational pressure, and then expands back to

Stages	Rate	Channel	Repeat
Image	1	3	
Stem Block	1/4	64	1
SIR Block	1/8	64	8
MSCAN Block	1/16	64	3
MSCAN Block	1/32	128	5
IDWR Block	1/32	128	3
Decoder cat	1/8	384	
Decoder conv1	1/8	128	
Decoder conv2	1/8	N	
Flops (G)			14.13
Params (M)			3.86

Table 1. Detailed MIDWRSeg architecture

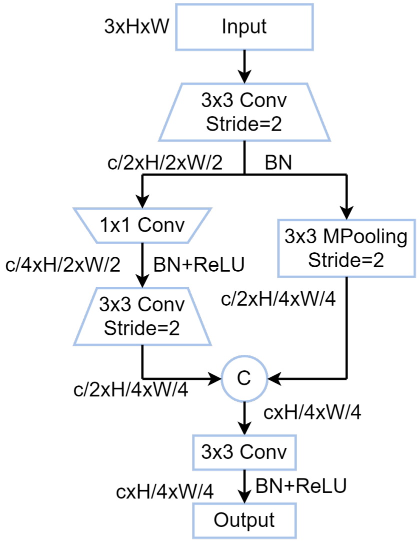

Figure 4. Stem block

original number of channels through a 3×3 convolution operation. The f_{left} can be described as follows:

$$f_{left} = \mathrm{Conv}_{k=3\times 3}^{s=2}(\mathrm{Conv}_{k=1\times 1}(f_{stage1})). \tag{2}$$

The f_{right} branch utilizes max pooling to rapidly identify the maximum value as the most representative feature, while concurrently carrying out preliminary noise reduction operations. The f_{right} is computed as follows:

$$f_{right} = \text{MaxPooling}_{k=3\times 3}^{s=2}(f_{stage1}). \tag{3}$$

Finally, the f_{left} and f_{right} branches stack the channels and then merge the features through a final 3×3 convolution. The final output feature map f_{out} can be described as follows:

$$\begin{aligned} f_{out} &= \text{Cat}(f_{left} + f_{right}), \\ f_{out} &= \text{Conv}_{k=3\times 3}(f_{out}). \end{aligned} \tag{4}$$

3.3 Simple Inverted Residual Block

The Simple Inverted Residual (SIR) block is depicted in Figure 5. In the early stages, the primary task is denoising and extracting primary features f_{in}, therefore the number of channels is increased by sixfold to combine multiple modes of features f_{out}. The feature map f_{out} is computed as follows:

$$\begin{aligned} f_{out} &= \text{Conv}_{k=1\times 1}(\text{Conv}_{k=3\times 3}(f_{in})), \\ f_{out} &= f_{out} + f_{in}. \end{aligned} \tag{5}$$

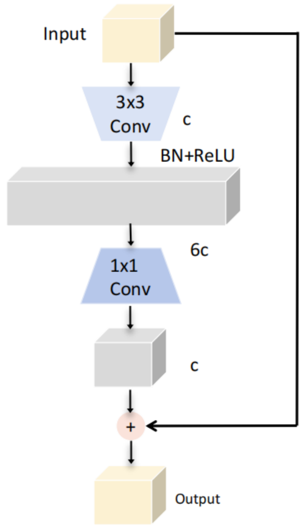

Figure 5. Simple Inverted Residual block (SIR)

Finally, to conserve computational resources, 1×1 convolution is utilized to decrease the number of channels. To maintain stable gradient updates and prevent model degradation, residual connection is added.

3.4 OverlapPatchEmbed

In order for subsequent multi-scale conv attention to proceed smoothly, it is necessary to divide the feature maps f_{in} into patches of size 7. As shown in Figure 6.

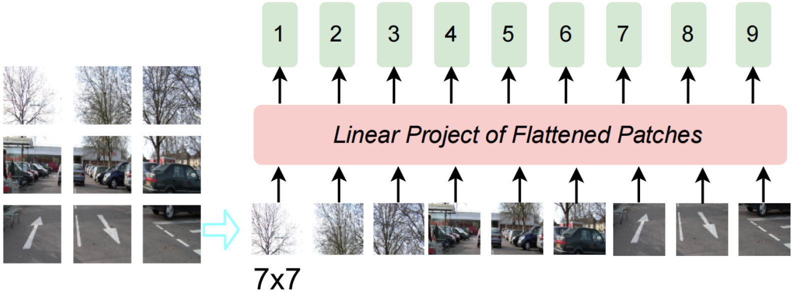

Figure 6. An overview of PatchEmbed. The feature map is divided into 7 × 7 patches, and the patches are formed into a sequence from left to right. Then, the Linear Project of Flattened Patches operation is performed to merge the dimensions of H and W of the feature map, and the last two dimensions are swapped to form a specific sequence-to-sequence vector.

Using a stride of 4 to move these patches creates overlapping blocks. The overlapping blocks of encoded features are normalized along the batch size dimension to obtain the final output f_{out}. The operation of dividing patches can be easily implemented using convolution, described as follows:

$$f_{out} = \mathrm{BatchNorm}(\mathrm{Conv}_{k=7\times 7}^{s=4}(f_{in})). \tag{6}$$

After the operation of dividing patches, in order to perform the convolutional attention operation, we need to reshape the tensor from the previous shape of (B, C, H, W) by flattening the last two dimensions H and W to obtain the total number of patches. After this step, the shape of the tensor changes to $(B, C, H \times W)$. Then, we transpose the last two dimensions to make it conform to the tensor shape commonly used in natural language processing, which is $(B, num\_patch, embed\_dim)$. Here, $num\_patch = H \times W$, and $embed\_dim = C$. The process of flattening and transposing dimensions is described as follows:

$$f_{out} = \mathrm{Transpose}_{dim=1,2}(\mathrm{Flatten}_{H,W}(f_{out})). \tag{7}$$

3.5 Multi-Scale Convolution Attention Block

The Multi Scale Convolution Attention (MSCA) Block is shown in Figure 7. Following the process of feature extraction and noise reduction in the preceding stage, the MSCA module proposed in SegNext [8] is employed to establish long-range dependencies.

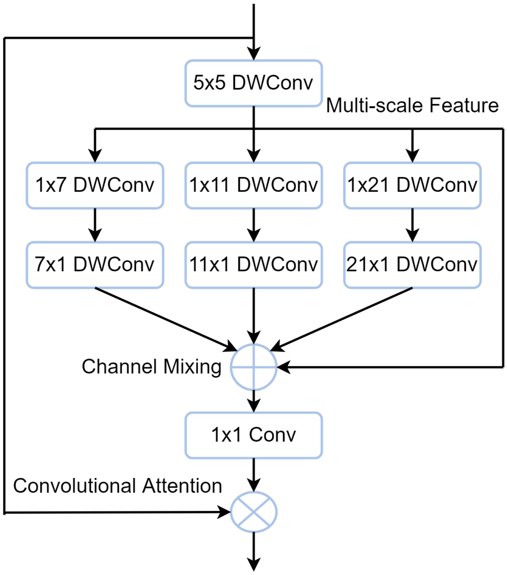

Figure 7. Multi-scale convolution attention

The attention mechanism formed by this module effectively strengthens semantic information. And for the situation where there are many slender targets such as street lights, railings, people, and tall buildings on the road, using the pair of asymmetric convolutions combined is appropriate. It is worth noting that the convolution kernel sizes of the three branches are 1×7, 1×11, and 1×21, respectively. The convolution attention can be described by the formula as follows:

$$\begin{aligned} &\mathrm{Att} = \mathrm{Conv}_{k=1\times 1}\left(\sum_{i=0}^{3} \mathrm{Scale}_k(\mathrm{DWConv}(F))\right), \quad k = 5, 7, 11, 21, \\ &\mathrm{Out} = \mathrm{Att} \otimes F. \end{aligned} \tag{8}$$

F denotes the input feature map. $DWConv$ denotes Depth-Wise Conv. $Scale_k$ denotes the size of the Conv kernel, and $\otimes$ denotes element-wise multiplication of the feature maps. Perhaps it will surprise you to learn why such a large convolution kernel is used. Studies such as RepLKNet [17] and SLaK [18] have found that large kernel convolutions can achieve more effective receptive fields, which are crucial for ensuring correct segmentation by the model.

A stage of MSCA, which corresponds one-to-one with the traditional attention mechanism, is shown in Figure 8.

The feature map f_{in} undergoes a 1×1 convolution, followed by the GELU

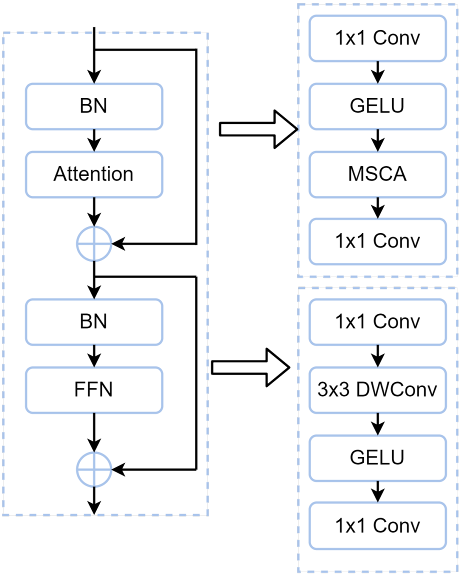

Figure 8. A stage of MSCA

activation function. The GELU activation function formula is as follows:

$$\mathrm{GELU}(x) = 0.5x\left(1 + \tanh\left(\sqrt{\frac{2}{\pi}}\left(x + 0.044715x^3\right)\right)\right). \tag{9}$$

Subsequently, it traverses the MSCA block and then undergoes another 1×1 convolution, which replaces the traditional combination of BatchNorm and Attention. The f_{stage1} is computed as follows:

$$\begin{aligned} f_{stage1} &= \mathrm{GELU}(\mathrm{Conv}_{k=1\times1}(f_{in})),\\ f_{stage1} &= \mathrm{Conv}_{k=1\times1}(MSCAN(f_{stage1})) + f_{in}. \end{aligned} \tag{10}$$

Subsequently, the feature map f_{stage2} undergoes a 1×1 convolution, followed by a Depth-Wise convolution with a kernel size of 3. Immediately after, it passes through a GELU activation function and another 1×1 convolution. This corresponds to the original BatchNorm and Feedforward Neural Network (FFN) operations. The f_{stage2} is computed as follows:

$$\begin{aligned} f_{stage2} &= \mathrm{DWConv}_{k=3\times3}(\mathrm{Conv}_{k=1\times1}(f_{stage1})),\\ f_{stage2} &= \mathrm{Conv}_{k=1\times1}(\mathrm{GELU}(f_{stage2})) + f_{stage2}. \end{aligned} \tag{11}$$

3.6 Internal Depth-Wise Residual Block

The Depth-Wise Residual module is shown in Figure 9. Firstly, the feature map f_{in} undergoes a 3×3 convolution to extract regional information. Then, to capture multi-scale receptive fields, a three-branch dilated convolution is employed with dilation rates of 1, 3, and 5, respectively. The f_{stage_1} is computed as follows:

$$\begin{aligned} f &= \mathrm{Conv}_{k=3\times 3}(f_{in}), \\ f_1 &= \mathrm{DWConv}_{k=3\times 3}(f), \\ f_2 &= \mathrm{DWConv}_{k=3\times 3}^{d=3}(f), \\ f_3 &= \mathrm{DWConv}_{k=3\times 3}^{d=5}(f), \\ f_{stage1} &= \mathrm{Cat}(f_1 + f_2 + f_3), \end{aligned} \tag{12}$$

$$f_{out} = f_{stage1} + f_{in}. \tag{13}$$

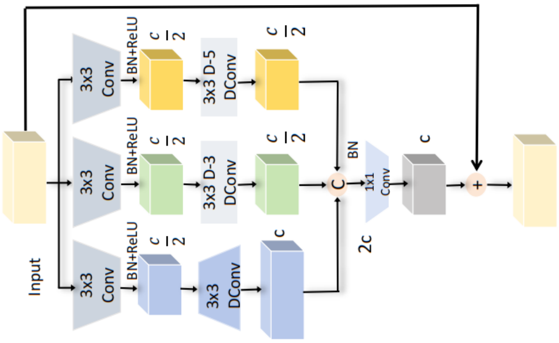

Figure 9. Depth-Wise Residual Module

Finally, the number of channels is adjusted through a 1×1 convolution, and then added to the original input through residual connections. The f_{out} is computed according Equation (13). However, the regional information obtained through the 3×3 convolution is not fully utilized. To make better use of the receptive field information from this stage, internal residual connections are employed to leverage the information from the previous stage during the application of dilated convolution convolutions across the three branches. This is illustrated in Figure 10. The f_{stage1}

can be redescribed as:

$$\begin{aligned} f_1 &= \mathrm{DWConv}_{k=3\times3}(f) + f, \\ f_2 &= \mathrm{DWConv}^{d=3}_{k=3\times3}(f) + f, \\ f_3 &= \mathrm{DWConv}^{d=5}_{k=3\times3}(f) + f, \\ f_{stage1} &= \mathrm{Cat}(f_1 + f_2 + f_3). \end{aligned} \tag{14}$$

The improvement can be attributed to the fact that short connections enable the model to explicitly combine multiple models with varying receptive field sizes (small and large receptive fields accumulate consecutively). This approach enhances the model's performance within a larger receptive field while maintaining its ability to capture fine-grained features. And the activation function used after the three-branch dilated convolution is ELU. The ELU activation function is described as follows:

$$\mathrm{ELU}(x) = \begin{cases} e^x - 1, & \text{if } x < 0, \\ x, & \text{if } x \geq 0. \end{cases}$$

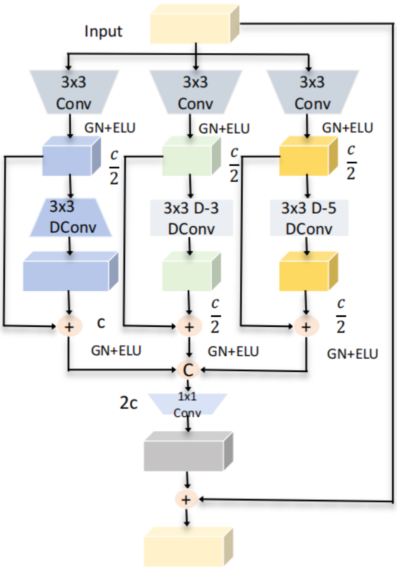

Figure 10. Internal Depth-Wise Residual Module

4 EXPERIMENT

4.1 Dataset

CamVid (Cambridge Driving Labeled Video Database) [30, 31] is the first driving dataset. It comprises over 700 finely annotated road scene images and is divided into training, validation, and testing sets. 32 and 11 categories is used separately for the experiment. They are roads, traffic signs, cars, sky, pedestrian walkways, utility poles, walls, pedestrians, buildings, and bicycles. The dataset consists of 367 training sets, 101 validation sets, and 233 testing sets.

The Cityscapes dataset [32] is an openly available dataset used in computer vision, and it has been widely utilized to provide data support for understanding and analyzing cities. The dataset is also divided into training, validation, and testing sets. Among them, there are 2 975 training sets, 500 validation sets, and 1 525 testing sets. Although the dataset comprises 30 semantic categories, most optimal models consider only 19 categories for experimentation. Our training and evaluation for this experiment also adhered to the use of these 19 categories.

4.2 Implementation Details

Our model implementation uses the PyTorch framework version 1.13.1 and CUDA version 11.7.

We employed the AdamW [33] optimization method in our training procedure for the Cityscapes dataset. We set the batch size to 8, the weight decay to $2e^{-4}$, the momentum to 0.9, and the initial learning rate to 0.009. It should be noted that the loss function we use is a cross-entropy loss function with Online Hard Example Mining (OHEM), which is utilized to learn from difficult samples and indirectly enhance the model's ability.

We use the SGD [34] optimization method. We set the batch size to 12, the weight decay to 0.0005, the momentum to 0.9, and the initial learning rate to 0.001 in the training procedure of the CamVid dataset. The loss function uses a cross entropy loss function with weight, where weight is used to handle classes imbalance issues in road scenes [35, 36, 37].

In terms of data augmentation, random mirror flipping and random scaling with ratios of 0.75, 1.0, 1.25, 1.5, 1.75, and 2.0 are employed. For the CamVid dataset, the input size is cropped to 720 × 960. Similarly, for the Cityscapes dataset, the input size is cropped to 512 × 1 024.

4.3 Ablation on MSCA Module

In this section, we conducted a series of ablation experiments to verify the effectiveness of our MIDWRSeg. We conduct ablation experiments on the MSCAN module and IDWR module respectively. In addition, we also investigate the stacking depth of MSCAN modules. All ablation experiments are conducted on the CamVid dataset.

Kernel Size	FPS	mIoU
$\{7, 11, 15\}$	83	69.62
$\{7, 15, 17\}$	87.6	66.4
$\{7, 15, 25\}$	76.7	69.49
$\{7, 17, 19\}$	83.2	70.48
$\{7, 17, 21\}$	75.6	69.72
$\{7, 11, 21\}$	88.9	**71.09**

Table 2. The experiment results about kernel size of MSCAN

Our model primarily relies on the large-scale convolution kernel employed by the MSCAN module for extracting deep semantic features. Subsequently, it implements attention mechanism operations to capture long-range dependencies (large receptive fields). In the first part, we explore the effect of the three different kernel sizes in the MSCAN module on feature extraction. In the second part, we investigate the impact of varying the depth of MSCAN stacking on the model's performance.

To investigate the impact of different convolution sizes on model performance within the three branches of the MSCAN module, we selected three numbers from the set $\{7, 9, 11, 13, 15, 17, 19, 21\}$. All ablation experiments were conducted by stacking eight layers of MSCAN modules. The specific convolution kernel sizes for the three branches of MSCAN are detailed in Table 2.

It should be noted that, in order to ensure experimental fairness, the influence of the IDWR module in the fourth stage of IDWRSeg has been excluded. Following an ablation experiment on the MSCAN convolution kernel size, it is observed that a larger difference between convolution kernel sizes is more beneficial for the model. But it can have an impact on large objects that suddenly appear in the four corners. Nonetheless, in this regard, a slight difference in the size of convolutional kernels can alleviate this situation. For further details, please refer to the last image in the bottom right corner of Figure 11, which depicts the car in the bottom left corner.

The next step involves conducting a depth ablation experiment on the MSCAN block, which will be performed on the CamVid dataset. In this experiment, the second and third stages of the model will be configured with different numbers of MSCAN blocks. The specific experimental results are presented in Table 3.

Stage2	Stage3	FPS	mIoU
3	5	88.9	71.09
4	5	62.11	72.32
5	6	65.69	72.94
3	9	61.49	72.98
4	9	56.37	73.42

Table 3. The experiment results about depths of MSCAN

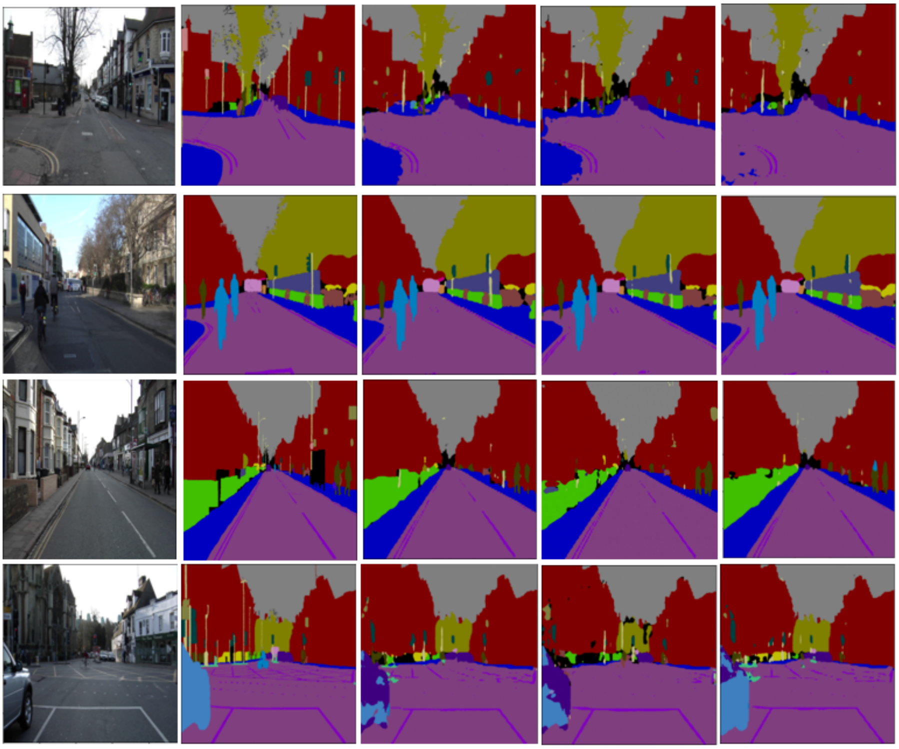

Figure 11. Visualize the differences in convolutional kernel size in MSCAN Modules. From left to right are the original image, ground truth, kernel size $\{7, 11, 21\}$, $\{7, 17, 19\}$, $\{7, 15, 17\}$.

4.4 Ablation on IDWR Module

In order to facilitate rapid validation of ablation experiments in the future, some categories of the original CamVid dataset have been merged, reducing the total number of categories from 32 to 11. The ablation experiments on the IDWR module are still conducted on the CamVid dataset, with a 2:1:1 ratio determined as the optimal configuration for the number of channels. This aspect has been previously explored in DWRSeg [9] and will not be repeated here. Instead, this section delves into the depth of ablation experiments and examines the influence of residual connections on accuracy. For detailed results, please refer to Table 4.

4.5 Comparisons with Other Works

We present a comparison between our MIDWRSeg and state-of-the-art real-time semantic segmentation methods. Our method is tested on a single RTX3060 GPU

Stage4	Connection	FPS	mIoU
3		77.95	73.2
3	✓	80.20	73.6
4		79.73	73.29
4	✓	83.22	73.91
5		79.83	73.8
5	✓	81.66	73.97
6		80.24	73.88
6	✓	76.9	73.90

Table 4. The experiment results about depths of IDWR

with an image resolution of 512 × 1 024 on the Cityscapes dataset. The resolution on the CamVid dataset is 720 × 960. We test the speed without employing any acceleration strategy and only use fine data to train the model.

Before comparing the performance of different models, it is necessary to briefly introduce the metrics that will be used. Mean Intersection over Union (mIoU): This is a commonly used standard metric in semantic segmentation, which calculates the average of the Intersection over Union (IoU) for each category. Frames Per Second (FPS): FPS denotes to the number of frames transmitted per second.

First, Table 5 shows a comparison of our model with other models on the CamVid dataset. In the experiment on the CamVid dataset, pre-trained weights from the Cityscapes dataset are used.

Method	mIoU ↑	FPS ↑
ICNet [25]	67.1	34.5
DFANet [38]	64.7	120
STDC1 [29]	73.0	197.6
STDC2 [29]	73.9	**152.2**
BiSeNetV2 [28]	76.7	124.5
ours	**76.8**	88.7

Table 5. Comparisons with other state-of-the-art methods on CamVid

Second, Table 6 demonstrates a comparison of our model with other models on the Cityscapes dataset, all of which adopt an encoder-decoder structure. And another metric, FLOPs, denotes the number of floating point operations, which can be used to measure the complexity of a model.

Third, Table 7 demonstrates a comparison of our model with other multi-branch structure models, using the same evaluation metrics as before.

Finally, we conducted visualizations on the Cityscapes test set. There are four columns in total, arranged in the order of original images and segmentation results. Since the official website does not provide the actual labels for segmentation, we are unable to display the labels. The segmentation results in the first two rows

Method	Input Size	Parameters (M)↓	FLOPs (G)	FPS↑	Platform	mIoU (%)↑
ENet [20]	512 × 1 024	**0.36**	4.4	100.2	RTX3090	58.3
SegNet [19]	360 × 640	29.5	286	53	RTX3090	55.8
ESPNet [39]	512 × 1 024	0.36	3.5	**320**	RTX3090	60.3
SQNet [40]	1 024 × 2 048	16.3	576.4	6.1	RTX3090	59.6
CGNet [41]	1 024 × 2 048	0.49	28	108	RTX3090	64.8
DABNet [24]	512 × 1 024	0.76	27.7	191	RTX3090	70.1
DeepLabV2 [3]	512 × 1 024	4	457	1	RTX2080Ti	70.4
ESNet [42]	512 × 1 024	1.66	24.4	53	RTX3090	70.7
DFANet [38]	512 × 1 024	4.8	2.1	120	RTX3080	67.1
LedNet [43]	512 × 1 024	0.94	–	87	RTX3090	70.6
DWRSeg [9]	512 × 1 024	3.53	16.42	256.2	RTX3080	73.1
EBUNet [44]	512 × 1 024	1.57	24.13	152	RTX3090	73.4
TopFormer [45]	512 × 1 024	5.1	–	95.7	RTX2080Ti	70.0
SeaFormer [13]	512 × 1 024	8.6	–	45.2	RTX2080Ti	72.2
AFFormer [46]	512 × 1 024	3.0	–	49.5	RTX2080Ti	73.5
ours	512 × 1 024	3.86	14.12	88.9	RTX3060	**74.2**

Table 6. Comparison with state-of-the-art encoder-decoder semantic segmentation methods on Cityscapes test set

Method	Input Size	Parameters (M)↓	FLOPs (G)	FPS↑	Platform	mIoU (%)↑
ICNet [25]	1 024 × 2 048	26.5	28.3	15.4	RTX3090	70.6
ContextNet [26]	1 024 × 2 048	0.85	7.2	121	RTX3090	66.1
BiseNetV1 [27]	768 × 1536	5.8	14.8	105	TitanXP	68.4
BiseNetV2 [28]	512 × 1 024	–	–	156	GTX1080Ti	72.4
STDC1 [29]	512 × 1 024	9.97	0.81	**250.4**	RTX3060	71.9
STDC2 [29]	512 × 1 024	14	1.45	188.6	RTX3060	73.4
DDRNet [10]	1 024 × 2 048	5.7	36.3	108.8	GTX1080Ti	77.8
PIDNet [47]	1 024 × 2 048	7.6	46.3	100.8	RTX3090	**78.8**
SCTNet [48]	512 × 1 024	4.6	–	160.3	RTX2080Ti	72.8
ours	512 × 1 024	3.86	14.12	88.9	RTX3060	74.2

Table 7. Comparison with state-of-the-art multi-branch semantic segmentation methods on Cityscapes test set

are relatively good, while there are some larger flaws in the last two rows. The visualization results are shown in Figure 12.

5 CONCLUSION

Through ablative experiments on the MSCAN and IDWR blocks, it can be observed that multi-scale feature extraction and effective receptive field extraction are crucial. The adoption of MSCAN, which forms long-range dependencies through inexpensive

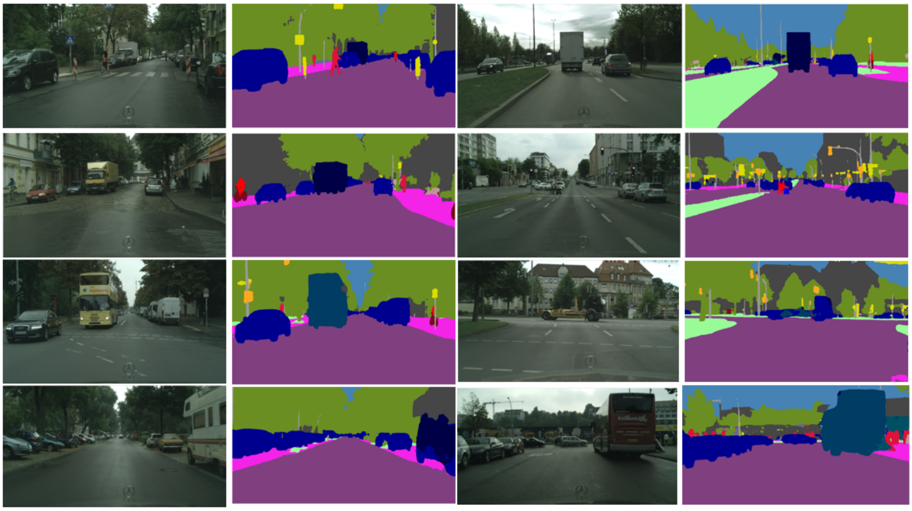

Figure 12. Visualization of the Cityscapes tests. The first and second row predictions are good results. There is a significant deviation in the predicted results of the rows three or four.

convolutions, proves to be effective. Our method has formed a competitive advantage with existing real-time semantic segmentation models for encoder-decoder on Cityscapes and CamVid datasets. Our MIDWRSeg achieves mIoU of 74.2 % at a speed of 88.9 FPS at Cityscapes test and achieves mIoU of 76.8 % at a speed of 95.2 FPS at CamVid test. With a model parameter size of 3.86 M, this lightweight structure provides convenience for deployment on memory-constrained embedded devices.

REFERENCES

[1] Long, J.—Shelhamer, E.—Darrell, T.: Fully Convolutional Networks for Semantic Segmentation. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3431–3440, doi: 10.1109/CVPR.2015.7298965.

[2] Chen, L. C.—Papandreou, G.—Kokkinos, I.—Murphy, K.—Yuille, A. L.: Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. In: Bengio, Y., LeCun, Y. (Eds.): 3rd International Conference on Learning Representations (ICLR 2015). 2015, doi: 10.48550/arXiv.1412.7062.

[3] 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 Transactions on Pattern Analysis and Machine Intelligence, Vol. 40, 2018, No. 4, pp. 834–848, doi: 10.1109/TPAMI.2017.2699184.

[4] Chen, L. C.—Papandreou, G.—Schroff, F.—Adam, H.: Rethinking Atrous Convolution for Semantic Image Segmentation. CoRR, 2017, doi: 10.48550/arXiv.1706.05587.

[5] Ronneberger, O.—Fischer, P.—Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab, N., Hornegger, J., Wells, W. M., Frangi, A. F. (Eds.): Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. Springer, Cham, Lecture Notes in Computer Science, Vol. 9351, 2015, pp. 234–241, doi: 10.1007/978-3-319-24574-4_28.

[6] Vaswani, A.—Shazeer, N.—Parmar, N.—Uszkoreit, J.—Jones, L.—Gomez, A. N.—Kaiser, Ł.—Polosukhin, I.: Attention Is All You Need. 2017, pp. 5998–6008, doi: 10.48550/arXiv.1706.03762.

[7] Dosovitskiy, A.—Beyer, L.—Kolesnikov, A.—Weissenborn, D.—Zhai, X.—Unterthiner, T.—Dehghani, M.—Minderer, M.—Heigold, G.—Gelly, S.—Uszkoreit, J.—Houlsby, N.: An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale. CoRR, 2020, doi: 10.48550/arXiv.2010.11929.

[8] Guo, M. H.—Lu, C. Z.—Hou, Q.—Liu, Z.—Cheng, M. M.—Hu, S. M.: SegNeXt: Rethinking Convolutional Attention Design for Semantic Segmentation. In: Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., Oh, A. (Eds.): Advances in Neural Information Processing Systems 35 (NeurIPS 2022). Curran Associates, Inc., 2022, pp. 1140–1156, doi: 10.48550/arXiv.2209.08575.

[9] Wei, H.—Liu, X.—Xu, S.—Dai, Z.—Dai, Y.—Xu, X.: DWRSeg: Dilation-Wise Residual Network for Real-Time Semantic Segmentation. CoRR, 2022, doi: 10.48550/arXiv.2212.01173.

[10] Hong, Y.—Pan, H.—Sun, W.—Jia, Y.: Deep Dual-Resolution Networks for Real-Time and Accurate Semantic Segmentation of Road Scenes. CoRR, 2021, doi: 10.48550/arXiv.2101.06085.

[11] He, K.—Zhang, X.—Ren, S.—Sun, J.: Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[12] Liu, Z.—Lin, Y.—Cao, Y.—Hu, H.—Wei, Y.—Zhang, Z.—Lin, S.—Guo, B.: Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 9992–10002, doi: 10.1109/ICCV48922.2021.00986.

[13] Wan, Q.—Huang, Z.—Lu, J.—Yu, G.—Zhang, L.: SeaFormer: Squeeze-Enhanced Axial Transformer for Mobile Semantic Segmentation. CoRR, 2023, doi: 10.48550/arXiv.2301.13156.

[14] Hu, J.—Shen, L.—Sun, G.: Squeeze-and-Excitation Networks. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 7132–7141, doi: 10.1109/CVPR.2018.00745.

[15] Woo, S.—Park, J.—Lee, J. Y.—Kweon, I. S.: CBAM: Convolutional Block Attention Module. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (Eds.): Computer Vision – ECCV 2018. Springer, Cham, Lecture Notes in Computer Science, Vol. 11211, 2018, pp. 3–19, doi: 10.1007/978-3-030-01234-2_1.

[16] Hou, Q.—Zhou, D.—Feng, J.: Coordinate Attention for Efficient Mobile Network Design. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 13708–13717, doi: 10.1109/CVPR46437.2021.01350.

[17] Ding, X.—Zhang, X.—Han, J.—Ding, G.: Scaling Up Your Kernels to 31×31: Revisiting Large Kernel Design in CNNs. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 11963–11975, doi: 10.1109/CVPR52688.2022.01166.

[18] Liu, S.—Chen, T.—Chen, X.—Chen, X.—Xiao, Q.—Wu, B.—Kärkkäinen, T.—Pechenizkiy, M.—Mocanu, D.—Wang, Z.: More ConvNets in the 2020s: Scaling Up Kernels Beyond 51x51 Using Sparsity. CoRR, 2022, doi: 10.48550/arXiv.2207.03620.

[19] Badrinarayanan, V.—Kendall, A.—Cipolla, R.: SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 39, 2017, No. 12, pp. 2481–2495, doi: 10.1109/TPAMI.2016.2644615.

[20] Paszke, A.—Chaurasia, A.—Kim, S.—Culurciello, E.: ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation. CoRR, 2016, doi: 10.48550/arXiv.1606.02147.

[21] Ding, X.—Guo, Y.—Ding, G.—Han, J.: ACNet: Strengthening the Kernel Skeletons for Powerful CNN via Asymmetric Convolution Blocks. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 1911–1920, doi: 10.1109/ICCV.2019.00200.

[22] Romera, E.—Álvarez, J. M.—Bergasa, L. M.—Arroyo, R.: ERFNet: Efficient Residual Factorized ConvNet for Real-Time Semantic Segmentation. IEEE Transactions on Intelligent Transportation Systems, Vol. 19, 2017, No. 1, pp. 263–272, doi: 10.1109/TITS.2017.2750080.

[23] Chaurasia, A.—Culurciello, E.: LinkNet: Exploiting Encoder Representations for Efficient Semantic Segmentation. 2017 IEEE Visual Communications and Image Processing (VCIP), 2017, pp. 1–4, doi: 10.1109/VCIP.2017.8305148.

[24] Li, G.—Yun, I.—Kim, J.—Kim, J.: DABNet: Depth-Wise Asymmetric Bottleneck for Real-Time Semantic Segmentation. CoRR, 2019, doi: 10.48550/arXiv.1907.11357.

[25] Zhao, H.—Qi, X.—Shen, X.—Shi, J.—Jia, J.: ICNet for Real-Time Semantic Segmentation on High-Resolution Images. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (Eds.): Computer Vision – ECCV 2018. Springer, Cham, Lecture Notes in Computer Science, Vol. 11207, 2018, pp. 418–434, doi: 10.1007/978-3-030-01219-9_25.

[26] Poudel, R. P. K.—Bonde, U.—Liwicki, S.—Zach, C.: ContextNet: Exploring Context and Detail for Semantic Segmentation in Real-Time. CoRR, 2018, doi: 10.48550/arXiv.1805.04554.

[27] Yu, C.—Wang, J.—Peng, C.—Gao, C.—Yu, G.—Sang, N.: BiSeNet: Bilateral Segmentation Network for Real-Time Semantic Segmentation. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (Eds.): Computer Vision – ECCV 2018. Springer, Cham, Lecture Notes in Computer Science, Vol. 11217, 2018, pp. 334–349, doi: 10.1007/978-3-030-01261-8_20.

[28] Yu, C.—Gao, C.—Wang, J.—Yu, G.—Shen, C.—Sang, N.: BiSeNetV2: Bilateral Network with Guided Aggregation for Real-Time Semantic Segmentation. International Journal of Computer Vision, Vol. 129, 2021, pp. 3051–3068, doi: 10.1007/s11263-021-01515-2.

[29] Fan, M.—Lai, S.—Huang, J.—Wei, X.—Chai, Z.—Luo, J.—Wei, X.: Rethinking BiSeNet for Real-Time Semantic Segmentation. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 9711–9720, doi: 10.1109/CVPR46437.2021.00959.

[30] Brostow, G. J.—Shotton, J.—Fauqueur, J.—Cipolla, R.: Segmentation and Recognition Using Structure from Motion Point Clouds. In: Forsyth, D., Torr, P., Zisserman, A. (Eds.): Computer Vision – ECCV 2008. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 5302, 2008, pp. 44–57, doi: 10.1007/978-3-540-88682-2_5.

[31] Brostow, G. J.—Fauqueur, J.—Cipolla, R.: Semantic Object Classes in Video: A High-Definition Ground Truth Database. Pattern Recognition Letters, Vol. 30, 2009, No. 2, pp. 88–97, doi: 10.1016/j.patrec.2008.04.005.

[32] Cordts, M.—Omran, M.—Ramos, S.—Rehfeld, T.—Enzweiler, M.—Benenson, R.—Franke, U.—Roth, S.—Schiele, B.: The Cityscapes Dataset for Semantic Urban Scene Understanding. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 3213–3223, doi: 10.1109/CVPR.2016.350.

[33] Loshchilov, I.—Hutter, F.: Decoupled Weight Decay Regularization. CoRR, 2017, doi: 10.48550/arXiv.1711.05101.

[34] Bottou, L.: Large-Scale Machine Learning with Stochastic Gradient Descent. In: Lechevallier, Saporta, Y. G. (Eds.): Proceedings of COMPSTAT'2010. Physica-Verlag HD, Heidelberg, 2010, pp. 177–186, doi: 10.1007/978-3-7908-2604-3_16.

[35] Tan, J.—Wang, C.—Li, B.—Li, Q.—Ouyang, W.—Yin, C.—Yan, J.: Equalization Loss for Long-Tailed Object Recognition. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 11659–11668, doi: 10.1109/CVPR42600.2020.01168.

[36] Wang, T.—Zhu, Y.—Zhao, C.—Zeng, W.—Wang, J.—Tang, M.: Adaptive Class Suppression Loss for Long-Tail Object Detection. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 3102–3111, doi: 10.1109/CVPR46437.2021.00312.

[37] Zhu, B.—Jiang, Z.—Zhou, X.—Li, Z.—Yu, G.: Class-Balanced Grouping and Sampling for Point Cloud 3D Object Detection. CoRR, 2019, doi: 10.48550/arXiv.1908.09492.

[38] Li, H.—Xiong, P.—Fan, H.—Sun, J.: DFANet: Deep Feature Aggregation for Real-Time Semantic Segmentation. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 9514–9523, doi: 10.1109/CVPR.2019.00975.

[39] Mehta, S.—Rastegari, M.—Caspi, A.—Shapiro, L.—Hajishirzi, H.: ESPNet: Efficient Spatial Pyramid of Dilated Convolutions for Semantic Segmentation. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (Eds.): Computer Vision –

ECCV 2018. Springer, Cham, Lecture Notes in Computer Science, Vol. 11214, 2018, pp. 561–580, doi: 10.1007/978-3-030-01249-6_34.

[40] Treml, M.—Arjona-Medina, J.—Unterthiner, T.—Durgesh, R.—Friedmann, F.—Schuberth, P.—Mayr, A.—Heusel, M.—Hofmarcher, M.—Widrich, M.—Nessler, B.—Hochreiter, S.: Speeding Up Semantic Segmentation for Autonomous Driving. 2016, https://openreview.net/forum?id=S1uHiFyyg.

[41] Wu, T.—Tang, S.—Zhang, R.—Cao, J.—Zhang, Y.: CGNet: A Light-Weight Context Guided Network for Semantic Segmentation. IEEE Transactions on Image Processing, Vol. 30, 2020, pp. 1169–1179, doi: 10.1109/TIP.2020.3042065.

[42] Wang, Y.—Zhou, Q.—Xiong, J.—Wu, X.—Jin, X.: ESNet: An Efficient Symmetric Network for Real-Time Semantic Segmentation. In: Lin, Z., Wang, L., Yang, J., Shi, G., Tan, T., Zheng, N., Chen, X., Zhang, Y. (Eds.): Pattern Recognition and Computer Vision: (PRCV 2019). Springer, Cham, Lecture Notes in Computer Science, Vol. 11858, 2019, pp. 41–52, doi: 10.1007/978-3-030-31723-2_4.

[43] Wang, Y.—Zhou, Q.—Liu, J.—Xiong, J.—Gao, G.—Wu, X.—Latecki, L. J.: LEDNet: A Lightweight Encoder-Decoder Network for Real-Time Semantic Segmentation. 2019 IEEE International Conference on Image Processing (ICIP), 2019, pp. 1860–1864, doi: 10.1109/ICIP.2019.8803154.

[44] Shen, S.—Zhai, Z.—Yu, G.—Yan, Y.—Dai, W.: EBUNet: A Fast and Accurate Semantic Segmentation Network with Lightweight Efficient Bottleneck Unit. Complex & Intelligent Systems, Vol. 9, 2023, No. 5, pp. 5975–5990, doi: 10.1007/s40747-023-01054-y.

[45] Zhang, W.—Huang, Z.—Luo, G.—Chen, T.—Wang, X.—Liu, W.—Yu, G.—Shen, C.: TopFormer: Token Pyramid Transformer for Mobile Semantic Segmentation. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 12073–12083, doi: 10.1109/CVPR52688.2022.01177.

[46] Dong, B.—Wang, P.—Wang, F.: Head-Free Lightweight Semantic Segmentation with Linear Transformer. Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 37, 2023, pp. 516–524, doi: 10.1609/aaai.v37i1.25126.

[47] Xu, J.—Xiong, Z.—Bhattacharyya, S. P.: PIDNet: A Real-Time Semantic Segmentation Network Inspired by PID Controllers. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023, pp. 19529–19539, doi: 10.1109/CVPR52729.2023.01871.

[48] Xu, Z.—Wu, D.—Yu, C.—Chu, X.—Sang, N.—Gao, C.: SCTNet: Single-Branch CNN with Transformer Semantic Information for Real-Time Segmentation. Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 38, 2024, pp. 6378–6386, doi: 10.1609/aaai.v38i6.28457.

Bing Su received his B.Sc. and Ph.D. degrees from the Nanjing University of Aeronautics and Astronautics (NUAA), China. He is currently Associate Professor with the Department of Computer Science, School of Information and Mathematics, Changzhou University. His current research interests include network security, wireless sensor networks, the Internet of Things, routing protocols, and cloud computing.

Peng Jin received his B.Sc. degree in computer science and technology from the Wuzhou University, Wuzhou, China, in 2021, and now he is pursuing his M.Sc. degree in computer science and technology from the Changzhou University, Changzhou, China. His current research interests include semantic segmentation and time series model.

Yifeng Lin received his B.Sc. degree in software engineering from the Jilin University, Jilin, China, in 2005, his M.Sc. degree in software engineering from the Jilin University, Jilin, China, in 2007, and his Ph.D. degree in computer application technology from the Jilin University, Jilin, China, in 2012. His current research interest is deep learning.

Fuyang Wang received his B.Sc. degree in computer science and technology from the Changzhou University, Changzhou, in 2024, and he is pursuing his M.Sc. degree in software engineering from the Xi'an Jiaotong University, Xi'an, China. His current research interests include semantic segmentation and basic neural network models.