Computing and Informatics, Vol. 44, 2025, 71–91, doi: 10.31577/cai_2025_1_71

DELITESEG: A REAL-TIME SEMANTIC SEGMENTATION MODEL FOR PREDICTING SMALL OBJECTS AND OBJECT CONTOURS

Bing Su, Yifei Luan*, Yifeng Lin

School of Alibaba Cloud and Big Data, Changzhou University
Changzhou, 213159, China
e-mail: s22150812033@smail.cczu.edu.cn

Abstract. Semantic segmentation is one of the key technologies in the development of autonomous vehicles. Practical applications are increasingly pursuing a balance between effectiveness and efficiency. Many lightweight segmentation models nowadays have some problems, often making it difficult to predict small objects and edges between different objects. In this work, we propose a model of encoder-decoder structure, DeliteSeg. Firstly, we added deformable convolutional layers to the encoder, leveraging the advantages of deformable convolution to enable the model to better predict object edges. Then we proposed a new deep context aggregation module DLPPM, which improves the context information aggregation ability by fusing low-resolution feature maps of different scales multiple times, enabling the model to better predict small objects. Finally, we designed a new lightweight attention decoder (LMD) that utilizes a spatial channel attention mechanism to refine feature maps at different levels, effectively recovering information. After extensive experiments, our network achieved 73.6 % mIou and 123.7 FPS on the Cityscapes dataset and 73.9 % mIou and 116.4 FPS on the CamVid dataset. The experimental results confirm that our proposed model can make appropriate trade-offs between accuracy and real-time performance.

Keywords: Real-time performance, deformable convolution, deep pyramid aggregation module, lightweight attention decoder

[*] Corresponding author

1 INTRODUCTION

Semantic segmentation is a dense classification task in computer vision, which assigns corresponding labels to each pixel in the input image. It is used in many aspects of life, including medical image segmentation, autonomous driving, virtual reality, scene understanding, and so on. With the rise of deep learning technology, some advanced semantic segmentation methods have made significant progress in accuracy using convolutional neural networks (CNN). Since the proposal of fully convolutional network (FCN) [1], many novel networks have emerged. Such as DeepLab [2], PSPNet [3], DenseASPP [4], RefineNet [5], etc. However, in order to extract more information, their structures are often complex with too many convolutional layers and feature channels. Due to their lack of lightweight, they are not easy to use in some real-time scenarios. Therefore, designing a lightweight network that can achieve real-time performance and meet accuracy requirements has always been our goal. Nowadays, many lightweight semantic segmentation methods have been proposed. In order to achieve high precision and speed, the methods they use can be roughly divided into two categories: 1. Model compression: achieving network simplification by simplifying the model and removing its redundant parts. The main implementation methods include pruning, knowledge distillation, parameter quantification, architecture design, and dynamic computing. 2. Convolutional decomposition: By using unique convolution methods to reduce the number of model parameters, such as depthwise separable convolution and group convolution. Like the classic network MobileNet [6], which uses depthwise separable convolution to construct the backbone, reducing the number of parameters and running faster than traditional convolution.

In recent years, the most advanced real-time semantic segmentation models are mainly divided into dual branch structure, encoder decoder, and multi branch structure. For encoding and decoding structures, the information extraction ability of the encoder has a significant impact on the accuracy of the model. In order to better predict the edge contours between small objects and different objects, many high-performance information extraction modules have been proposed. STDCNet [7] designs an efficient and simple feature extraction module called STDC by reducing the dimensionality of feature maps and utilizing their aggregation for image representation. DWRSeg [8] proposes a new efficient feature extraction module, DWR, to collect detailed semantic information.

In this paper, we propose a new lightweight semantic segmentation method, DeliteSeg, using an encoding and decoding structure. Our model encoder is composed of deSTDC blocks and utilizes deformable convolution to enhance the module's ability to extract features. We propose the DLPPM module, which enhances feature aggregation between different pooling layers, allowing the model to obtain more useful contextual information and better predict small objects. We have designed a new lightweight decoder LMD, which integrates different features multiple times and uses channel and spatial attention mechanisms to effectively restore feature maps. Figure 1 shows our comparison results with other networks.

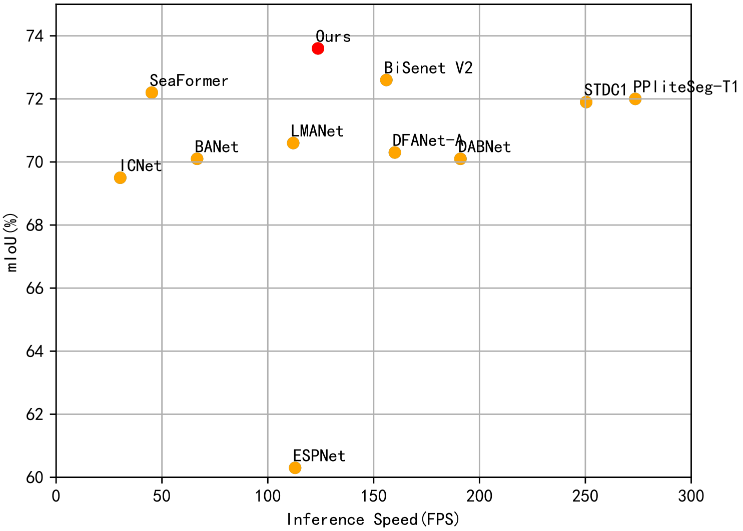


Figure 1. The comparison of segmentation accuracy (mIoU) and inference speed (FPS) on the Cityscapes test set. The red mark represents our net.

Our main contributions are summarized as follows:

- We propose the deSTDC module, which improves the STDC module by incorporating deformable convolutions, enabling the model to better predict the edges of objects.
- We propose a new deep context aggregation module DLPPM, which enhances feature aggregation between different pooling layers, enabling the model to better predict objects of different scales. In addition, its position is in the low resolution stage, so the inference time during network execution will not increase too much.
- We have designed a new lightweight attention decoder that can effectively recover information. It adopts a spatial channel attention mechanism to refine feature maps at different levels and employs multiple fusion methods to restore features.
- Our network achieved competitive results, achieving 73.6 % and 73.9 % mIoU on the Cityscapes and CamVid test sets, with FPS of 123.7 and 116.4, respectively.

2 RELATED WORK

In this section, we will review some related works on semantic segmentation, including lightweight semantic models, different convolutional methods, attention modules, and context information extraction module.

2.1 Lightweight Semantic Network

In real life, many scenarios require efficient and fast semantic segmentation techniques, so lightweight semantic segmentation has begun to develop. The earlier lightweight network ENet [9] was modified from SegNet [10], which used an asymmetric network structure and asymmetric convolution to reduce parameters. Later, ICNet [11] proposed a new image cascade network that utilizes both low resolution and high resolution information for effective segmentation. ESPNet [12] proposed an ESP module, which includes point wise convolution and dilated convolution pyramids, which can reduce computational complexity and perform multi domain feature extraction. BiseNet [13] proposed a dual branch structure network, where the contextual branch is used to extract contextual information, the spatial branch is used to extract spatial information, and finally the information from the two branches is fused. Both BiseNetv2 [14] and STDC-Seg [7] are improvements based on it, resulting in more efficient and accurate results.

2.2 Convolution Method

In addition to standard convolution, people have proposed various convolution methods to better meet their needs. As AlexNet [15] proposed group convolution, it divides the input feature map of the convolution into multiple groups, and also divides each convolution kernel into multiple groups. Convolution is performed within the corresponding groups, which can achieve the goal of reducing parameters. DeepLab [2] proposed dilated convolution, which reduces downsampling steps by expanding the receptive field, as excessive downsampling can result in information loss. MobileNet [6] proposed depthwise separable convolution, which divides the standard convolution into two steps: depth convolution and point convolution. This can reduce a large number of parameters, making it more suitable for use in lightweight networks. ACNet [16] proposed asymmetric convolution, whose core idea is to decompose the standard convolution and formally utilize spatially separable convolution to reduce the number of parameters. Deformable Convolutional Networks [17] proposed deformable convolution, which works by adding an additional parameter direction parameter to each element in the convolution kernel. This allows the kernel to expand to a large range during training, allowing it to adjust its shape according to actual conditions and better extract input features.

2.3 Attention Module

In order to improve the accuracy of the network, people add attention modules to the model. There are many categories of attention modules, including channels, space, self attention, etc. SENet [18] proposed the Channel Attention SE module, which extracts important channel features through average pooling. GeNet [19] also proposed using spatial attention to mine contextual information between features. CBAM [20] proposed a model that combines channel attention and spatial attention, aiming to enhance the attention ability of convolutional neural networks to images. DANet [21] proposed the idea of simultaneously introducing self attention into the channel space attention module. Polarized Self-Attention [22] proposed the PSA module, which can maintain high latitude in both channel and space to reduce information loss caused by dimensionality reduction, thereby improving the accuracy of the model.

2.4 Context Information Extraction Module

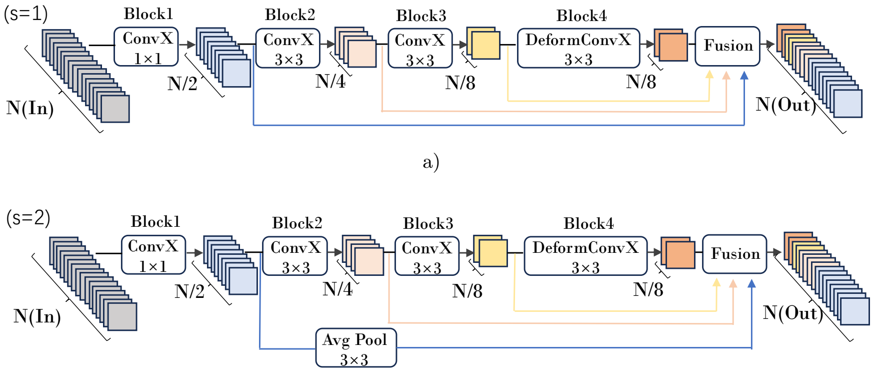
Predicting small objects has always been a challenge in semantic segmentation, and one effective method is to improve the model's ability to capture contextual information. Many models have proposed relevant solutions for this. DeepLab [2] proposed the ASPP module, which allows the model to capture information at different scales by using dilated convolutions with different dilation rates, and dilated convolutions can expand the receptive field. PSPNet [3] proposed the Spatial Pyramid Pooling Module (PPM), which utilizes different pooling kernels for image pooling, and its performance is better than ASPP. Later, the model utilized self attention to enhance feature extraction capabilities. Unlike the local characteristics of convolutional kernels, self attention mechanisms are good at capturing global dependencies. DANet [21] further improves feature representation by utilizing both positional and channel attention in parallel. Later, Object Context Network (OCNet) [23] utilized self attention mechanism for scene parsing, which defined object context as a set of pixels belonging to the same object category. However, the self attention mechanism also has its drawbacks, as it has high computational complexity. Later, CCNet [24] proposed the CCA module to replace the self attention mechanism, which is more effective in obtaining contextual information. PPlitesSeg [25] also proposed the fusion context information module SPPM, which is a simplified version of PPM and is more suitable for lightweight networks. It improves model speed by removing unnecessary branches.

3 PROPOSED METHOD

In this section, we will introduce the components of our DeliteSeg, including the deSTDC module, DLPPM module, and decoder LMD. Our DeliteSeg architecture design will be discussed at the end of this section.

3.1 DeSTDC Module

Our encoder uses the deSTDC module, which is an improvement from the STDC module.The STDC module was proposed by STDC-Seg [7]. Its structure is very simple, consisting of a 1×1 size convolutional block and three 3×3 size convolutional blocks. During the convolution process, its number of channels will decrease, and the previous results will be fused at the end of the module to enhance the



a)

b)

Figure 2. DeSTDC module structure

ability of feature extraction. Its advantage is that it can extract multi-scale features with fewer parameter quantities. The commonly used convolution kernels are square, while the shapes of objects in images are irregular, so block shaped convolution kernels cannot effectively extract the contour features of objects. The schematic of deformable convolution is as follows, its convolution kernel adds an extra direction parameter to each element so that the convolution kernel can be extended to a large range during training, allowing the convolution kernel to adjust its own shape according to the actual situation, and better extract the input features. Considering deformable convolution improves the accuracy of the model, but the complexity of the operations reduces the speed of the model. Therefore, we change the fourth nugget of the STDC module to deformable convolution, where the number of channels in this layer of the module is less, so the number of parameters involved is also less, and it does not have too much impact on the model speed. When the step size is different, the structure of the deSTDC module also varies. Figure 2 shows the structure of the deSTDC module with step sizes of 1 and 2, respectively.

3.2 DLPPM Module

We propose a new Deep Context Aggregation Module (DLPPM), which is an improved version of PPM. It processes feature maps through multi-scale pooling, enabling the model to better predict small objects. Figure 4 is its internal structure diagram. Inspired by Res2Net [26], we added three small convolutional blocks to blend multi-scale contextual information. Additionally, we enhanced the information fusion between different branches, further enhancing the module's ability to extract context. Firstly, it takes the feature maps of the original image at 1/32 resolution as input, and then generates feature maps of different scales through different

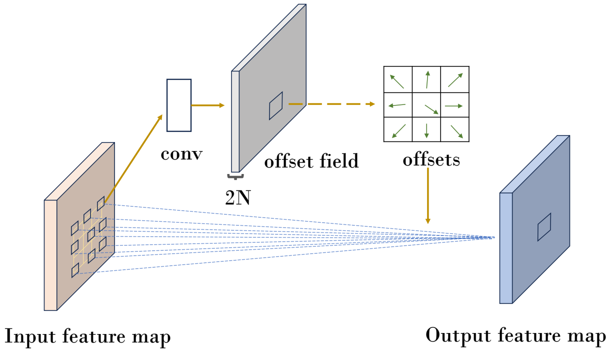


Figure 3. Deformable convolution

pooling kernels. Then further upsampling the output features, interacting with adjacent branches for information exchange, and finally integrating the features through small convolutional blocks. Finally, the feature map is restored through concatenation and convolution operations. Experimental results have shown that our module can effectively improve its ability to fuse contextual information.

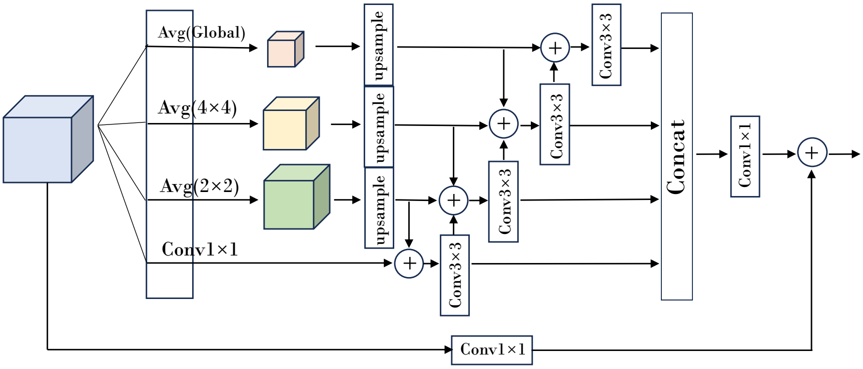


Figure 4. DLPPM module

3.3 Lightweight Attention Module Decoder (LMD)

We have designed a new decoder (LMD) to restore feature maps, and the performance of the decoder is crucial for the entire model. The structure of our decoder is shown in Figure 6. In order to better restore features, we used two different attention modules, with the spatial attention module (SAM) used to refine low-level feature maps and the channel attention module (CAM) used to refine high-level feature maps. We perform channel dimensionality reduction on the output of the fourth layer, then use SAM to process it, and additionally use CAM to process the output of the aggregation context module (DLPPM). Then add the results together. After upsampling, concatenate with the output of the third layer. Finally, after passing through several convolutional layers, the image is restored. We have added several 3 × 3 convolutional layers to the decoder, which are used to further mix information. After each 3 × 3 convolutions, BatchNorm and ReLU are added.

3.3.1 Spatial Attention Module

We use the Spatial Attention Module (SAM) to process low-level feature maps. As shown in Figure 5, for low-level input $F_{low} \in R^{C \times H \times W}$, let us first perform maximum and mean operations along the channel. Then, concatenate the obtained feature maps and perform convolution and sigmoid operations to obtain the spatial attention map P_{low}. Finally, P_{low} is multiplied element by element with the input F_{low} to obtain a refined feature M_{low}.

The calculation formula for spatial attention graph P_{low} is as follows:

$$P_{low} = \sigma \left(f_{conv} f_{cat} \left(f_{mean}, f_{max} \right) \right). \tag{1}$$

In formula (1), σ represents the sigmoid function. The shape of the transformed low-level feature map is represented by $C \times H \times W$ becomes $1 \times H \times W$. f_{conv} is a regular convolution operation, and f_{mean} and f_{max} are the mean and maximum values of channel dimensions, respectively.

Refined features M_{low}:

$$M_{low} = P \otimes F_{low}. \tag{2}$$

Among them, $\otimes$ represents element by element multiplication.

3.3.2 Channel Attention Module

We use Channel Attention Module (CAM) to process high-level feature maps. As shown in Figure 5, for high-level inputs $F_{high} \in R^{C \times H \times W}$. The module utilizes max pooling and average pooling operations to compress the spatial dimension of input features. Then, concatenate the obtained feature maps and perform convolution and sigmoid operations to obtain the spatial attention map P_{high}. Finally, P_{high} performs a multiplication operation with the input F_{high} to obtain a refined feature M_{high}.

The calculation formula for channel attention graph P_{high} is as follows:

$$P_{high} = \sigma(f_{conv} f_{cat}(f_{avgpool}, f_{maxpool})). \tag{3}$$

In formula (3), σ represents the sigmoid function. f_{conv} is a regular convolution operation, and $f_{avgpool}$ and $f_{maxpool}$ are average pooling and maximum pooling operations, respectively.

Refined features M_{high}:

$$M_{high} = P \otimes F_{high}. \tag{4}$$

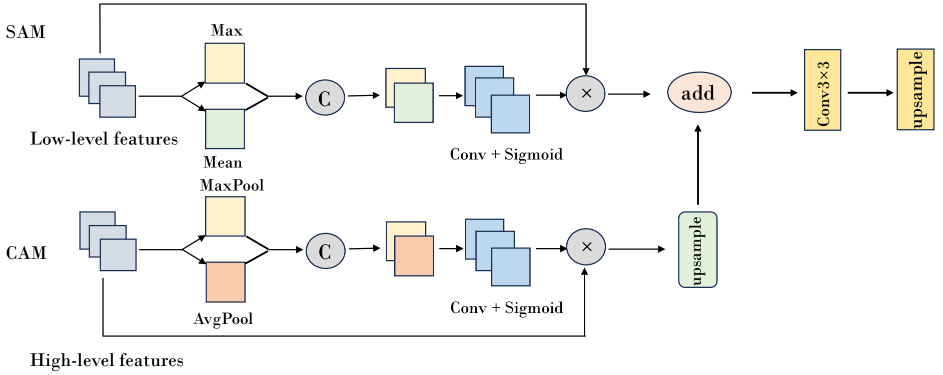


Figure 5. Attention module structure diagram

3.4 Network Structure

The structure of our proposed DeliteSeg is shown in Figure 5 and is listed in Table 1, which adopts an encoder decoder architecture. It mainly consists of three modules: an encoder composed of an improved deSTDC module, a deep aggregation module DLPPM for extracting contextual information, and a lightweight attention decoder (LMD).

Firstly, our encoder has a total of 5 layers in its structure. Considering the high computational cost of deformable convolution, we chose to use the deSTDC module in the last two layers with lower resolution. The first three layers use the same structure as STDC-Seg [7], consisting of two simple convolutions and one STDC module layer. Their advantages are simple structure and strong feature extraction ability. In the last two layers, we chose to add a deformable convolutional deSTDC module, which can further improve the encoder's ability to extract features and better segment the edges of objects.

Secondly, for information aggregation, we used a new deep aggregation context module DLPPM, which inputs low resolution feature maps, extracts multi-scale context information, and concatenates them in a cascading manner to generate features containing global context information.

Finally, we use an LMD decoder to restore the feature map. LMD adopts different attention mechanisms for different levels of feature mapping, which can produce more accurate output.

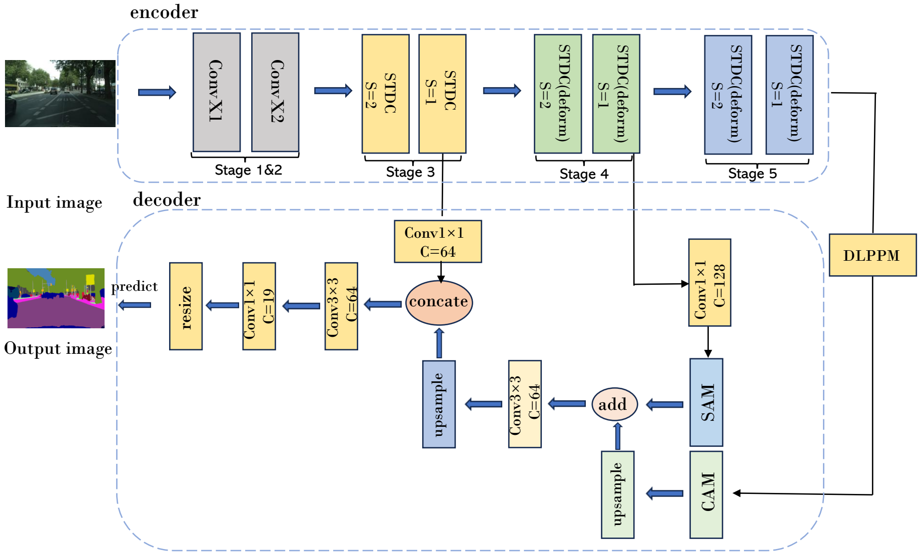


Figure 6. Network structure

Module	Composition	Channel Number	Output Size
Stage1	3 × 3 Conv	32	256 × 512
Stage2	3 × 3 Conv	64	128 × 256
Stage3	2 × STDC module	256	64 × 128
Stage4	2 × deSTDC module	512	32 × 64
Stage5	2 × deSTDC module	1 024	16 × 32
DLPPM	DLPPM	128	16 × 32
LMD	LMD	Number classes	512 × 1 024

Table 1. The structural composition of DeliteSeg

4 EXPERIMENTS

In this section, we will first introduce the two datasets used in the experiment, Cityscapes and CamVid datasets. Then, introduce the parameter settings and experimental process of our experiment. Subsequently, we will conduct ablation experiments on the improved module and analyze the results. Finally, we will compare the performance with other models.

4.1 Datasets

Cityscapes: The Cityscapes dataset is a semantic segmentation dataset that focuses on urban scene analysis. There are a total of 5 000 images in the Cityscapes dataset, of which 2 975 are used for network training, 500 are used for network validation, and 1 525 are used for network testing. It is used for semantic segmentation tasks in 19 categories. Its each image size is 1 024 × 2 048.

CamVid: The CamVid dataset is another famous urban scene dataset. It has a total of 701 images, of which 367 are used for training, 101 are used for validation, and 233 are used for testing. The image size of the CamVid dataset is 720 × 960. It is used for semantic segmentation tasks in 11 categories.

4.2 Training and Inference Settings

Training settings: We select the Random Gradient Descent (SGD) algorithm with a momentum of 0.9 as the optimizer. We also adopted a preheating strategy and a "multiple" learning rate scheduling. The training rounds for both datasets are 1 000 rounds. For Cityscapes, the batch size is 10, the initial learning rate is 0.005, and the weight decay in the optimizer is $5\mathrm{e}^{-4}$. For CamVid, the batch size is 16, the initial learning rate is 0.01, and the weight decay is $1\mathrm{e}^{-4}$. For data augmentation, we use random scaling, random cropping, random horizontal flipping, random color jitter, and normalization. The random scale ranges of Cityscapes and CamVid are $[0.125, 1.5]$, $[0.5, 2.5]$, respectively. The cropping resolution of Cityscapes is 512 × 1 024, CamVid has a cropping resolution of 720 × 960.

Inference settings: For the Cityscapes dataset, adjust the image size to 512 × 1 024, then the inference model generates a predicted image from the scaled image, and finally adjusts the predicted image to the original size of the input image. The time required for these three steps is calculated as inference time. For CamVid, the input image resolution is 720 × 960.

All experiments are conducted on NVIDIA RTX 3090 GPU, CUDA 11.6, CUDNN v8, PyTorch platform, Ubuntu 20.04 operating system, and 32 GB memory environment.

4.3 Quantitative Analysis

In this section, we conducted ablation studies on the deSTDC module, DSPPM module, and decoder LMD to verify the effectiveness of the proposed modules. We conducted all ablation experiments on the Cityscapes dataset.

4.3.1 The Ablation of DeSTDC Modules

Our encoder is mainly composed of deSTDC modules. Due to the time-consuming operation of deformable convolution, in order to achieve a balance between accuracy and speed, the number of deSTDC modules should be appropriate. We designed ablation experiments to determine whether selecting different numbers of deSTDC modules would affect network performance. From Table 2, it can be seen that adding a deformable convolutional layer to the STDC module can improve the performance of the benchmark network, although the number of parameters may increase slightly. When the deSTDC module is used in the last three layers of the encoder, the parameter count increases by 0.22 M, resulting in the best model performance. However, due to the slightly more computational complexity of deformable convolution compared to regular convolution, the speed of the model is not fast. When using a layer of deSTDC module, the number of model parameters increased by 0.06 M and mIoU increased by 0.3 %, but the improvement was not significant. Therefore, we choose to use the deSTDC module in the last two layers of the encoder to achieve a balance of speed accuracy. Compared to the baseline, our parameter count has increased by 0.12 M, but our mIoU has increased by 0.7 %.

Method	FPS	mIoU (%)	Parameters (M)
DeliteSeg-Baseline	202.6	72.9	6.49
Baseline (deSTDC = 2)	146.5	73.2	6.54
Baseline (deSTDC = 4)	123.7	73.6	6.61
Baseline (deSTDC = 6)	87.2	73.8	6.71

Table 2. Experiment results on deSTDC modules

4.3.2 The Ablation of DLPPM Module

We compare DLPPM with Pyramid Pooling Module (PPM), Simple Pyramid Pooling Module (SPPM), and Self-Attention Module (Base OC), all of which enhance the model's ability to obtain contextual information. From the results of Table 3, it can be inferred that compared to other modules, the DLPPM module can achieve better performance in the model. Compared to the PPM module, the mIoU of the model has been improved by 0.5 points. Compared to the SPPM module, the model mIoU has been improved by one point. Compared to the Base OC module, the mIoU of the model has been improved by 1.7 points. Compared to other modules, our DLPPM adds several internal convolutions to better fuse features from different branches, improving the model's ability to extract contextual information and better predict small objects.

Model	Module				Flops	Parameters	mIoU
	PPM	SPPM	Base-OC	DLPPM	(G)	(M)	(%)
DeliteSeg	√				9.06	6.46	73.1
DeliteSeg		√			8.94	6.39	72.6
DeliteSeg			√		8.62	6.15	71.9
DeliteSeg				√	9.12	6.61	73.6

Table 3. Experiment results on DLPPM modules

4.3.3 Decoder LMD Ablation

Decoder LMD is used to restore feature maps. We compare LMD with decoders of other models, using metrics such as intersection to union ratio (mIoU) and FPS. From Table 4, it can be seen that when using LMD, the mIoU of DeliteSeg reaches 73.6 %. Compared to the PPliteSeg's decoder, it has improved by 0.7 points, and the decrease in speed is also very small. Compared to DABNet's decoder, it has improved by 1.8 points. Compared to LRASPP, it has improved by 1.4 points. Compared to other decoders, our LMD utilizes both spatial attention module and channel attention module. In addition, we added convolutional layers to fuse the feature maps multiple times, further restoring the features. Therefore, we can conclude that a decoder (LMD) using two attention modules and multiple fusion methods can improve segmentation accuracy.

Method	Decoders				Flops	Parameters	FPS	mIoU
	DABNet	PPliteseg	LRASPP	LMD	(G)	(M)	(%)	
DeliteSeg	√				8.05	5.56	153.7	71.8
DeliteSeg		√			8.43	6.02	127.2	72.9
DeliteSeg			√		8.43	5.78	142.3	72.2
DeliteSeg				√	9.12	6.61	123.7	73.6

Table 4. Experiment results on LMD modules

5 COMPARISONS WITH OTHER WORKS

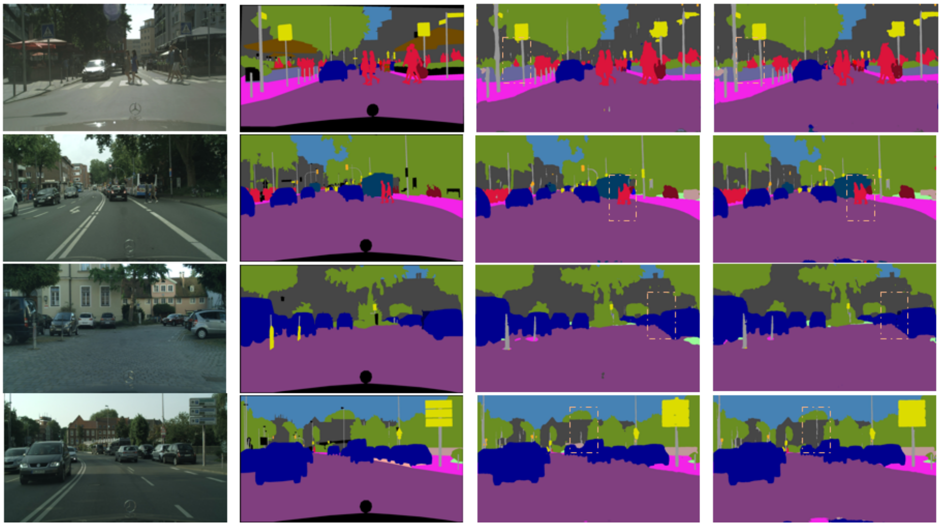
In this section, we compare the performance of DeliteSeg with some existing semantic segmentation methods on the test sets of two datasets and analyze the advantages of our network.

5.1 Comparisons on Cityscapes

Based on previous training and inference settings, we have chosen some networks with the same settings as much as possible for comparison. Including STDC1-Seg50 [7], ENet [9], ESPNet [12], BiSeNet V1 [13], PPliteSeg-T1 [25], etc. In addition, we also compared some traditional precision-oriented semantic segmentation networks, such as DeepLabV2 and RefineNet, to show the progressiveness of our DeliteSeg.

To provide a comprehensive comparison, Table 5 provides model information for various methods, input resolution of images, forward inference speed (FPS), and accuracy (mIoU), etc. As shown in Table 5, our DeliteSeg achieves 73.6 % mIoU at a speed of 123.7 FPS. Compared with some lightweight real-time semantic segmentation methods, our network has made some progress in mIoU. Compared with DFANet-A' [27], our mIoU has increased by 3.3 %, the parameter count has decreased by 1.19 M, and the model's Flops has increased by 5.72 G, making our network lighter and more efficient. Compared with STDC1-Seg [7], our model has decreased FPS but increased mIoU by 1.7 %. Compared with PPliteSeg-T1 [25], our model has increased mIoU by 1.6 %. We have also compared with some large models. Compared to DeepLabV2 [28], our mIoU has increased by 3.2 % and FPS is much faster. Compared to RefineNet [5], our network has achieved the same level of accuracy. But our speed and parameter count are better than it.

We also visually compared the results on the urban landscape validation set, as shown in Figure 7. From the figure, it can be seen that our DeliteSeg performs better in recognizing small objects and object edges than PPliteSeg-T1 [25]. Based on the above discussion, it can be concluded that our DeliteSeg achieves a good balance between segmentation accuracy and operational efficiency on urban landscape datasets.



a) Input image b) Ground truth c) PPliteSeg-T1 d) DeliteSeg

Figure 7. Visualized segmentation results on Cityscapes val set

5.2 Comparisons on CamVid

We also compared it with other models on CamVid. The image resolution used for inference is 720 × 960. Table 6 provides information on various methods, including pre training, speed (FPS), and accuracy (mIoU). From the table, it can be seen that our DeliteSeg performs well in terms of speed and accuracy. It reached 73.9 mIoU at a speed of 116.4 FPS. Compared to STDC1-Seg50 [7], the speed is slightly lower, but the accuracy is improved by 0.9 %. Compared to BiSeNet V2 [14], the

Model	Encoder	Resolution	GPU	mIoU (%)		FPS	Flops	Para-
				val	test	(G)	(M)	meters
ENet [9]	–	512 × 1 024	TitanX	–	58.3	76.9	4.4	0.4
ICNet [11]	PSPNet50	1 024 × 2 048	TitanX M	–	69.5	15.4	28.3	26.5
ESPNet [12]	ESPNet	512 × 1 024	TitanX	–	60.3	113	3.5	0.4
DFANet-A' [27]	Xception A	512 × 1 024	TitanX	–	70.3	100	3.4	7.8
CAS [29]	–	768 × 1 536	TitanXp	–	70.5	108.0	–	–
BiSeNet V1 [13]	Xception39	768 × 1 536	GTX 1080Ti	69.0	68.4	105.8	14.8	5.8
BiSeNet V2 [14]	–	512 × 1 024	GTX 1080Ti	73.4	72.6	156	55.3	49
STDC1-Seg50 [7]	STDC1	512 × 1 024	GTX 1080Ti	72.2	71.9	250.4	–	–
PPliteSeg-T1 [25]	STDC1	512 × 1 024	GTX 1080Ti	73.1	72.0	273.6	–	–
Fast-SCNN [30]	–	1 024 × 2 048	TitanXp	68.6	68.0	123.5	–	1.1
DeepLabV2 [20]	–	512 × 1 024	RTX 2080Ti	–	70.4	1	457	4
LMANet [31]	–	512 × 1 024	RTX 3090	–	70.6	112	–	0.95
RefineNet [5]	ResNet101	512 × 1 024	RTX 3090	–	73.6	9	428.3	118.1
DABNet [32]	–	1 024 × 2 048	RTX 3090	–	70.1	191	27.7	0.76
FBSNet [33]	–	512 × 1 024	RTX 3090	–	70.9	24	9.7	0.62
SeaFormer [34]	–	512 × 1 024	RTX 3090	–	72.2	45.2	–	8.6
DeliteSeg	–	512 × 1 024	RTX 3090	74.8	73.6	123.7	9.12	6.61

Table 5. Comparison with other semantic segmentation methods on Cityscapes test set

accuracy has increased by 1.5 %. Compared to AGLNet [35], DeliteSeg achieves faster speed and higher accuracy. In addition, we conducted a visual comparison with PPliteSeg-T1 [25] on the CamVid test set, as shown in Figure 8.

Model	Pretrained	GPU	mIoU (%)	FPS
ENet [9]	–	TitanX	51.3	61.2
DFANet-A [27]	ImageNet	TitanX	64.7	120
SFNet [35]	ImageNet	RTX 3090	72.58	134
SwiftNet [36]	–	GTX 1080Ti	72.58	–
BiSeNet V1 [13]	–	GTX 1080Ti	65.6	175
AGLNet [37]	–	RTX 3090	69.4	90.1
TD4-PSP18 [38]	ImageNet	RTX 3090	72.6	25
STDC1-Seg [7]	–	RTX 3090	73.0	197.6
PPliteSeg-T1 [25]	–	RTX 3090	73.3	222.3
DeliteSeg	–	RTX 3090	73.9	116.4

Table 6. Comparisons results on CamVid dataset with other works

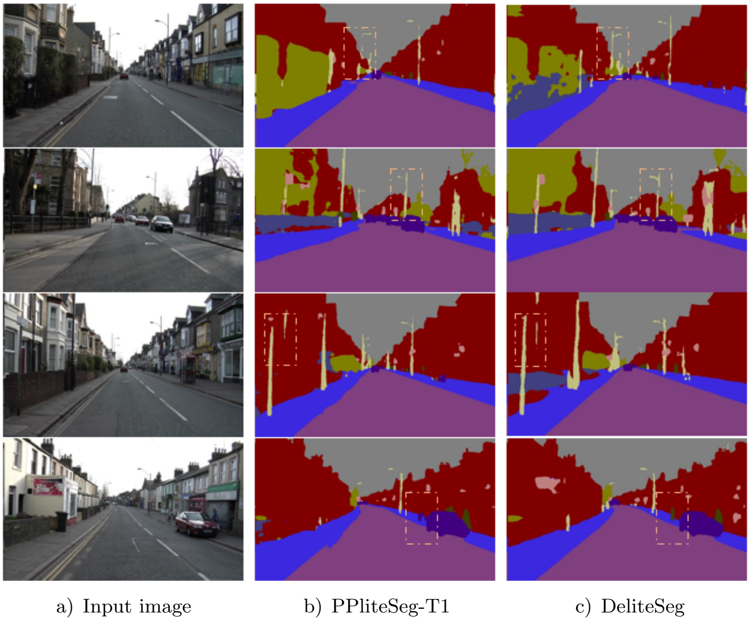


Figure 8. Visualized segmentation results on CamVid test set

6 CONCLUSION

In order to better predict small objects and object edges, this paper proposes a new lightweight semantic segmentation method called DeliteSeg. Our model consists of three main components: deSTDC block, DLPPM, and LMD. The deSTDC block adds deformable convolution to the STDC module to enhance its ability to extract features. The DLPPM module is mainly used to obtain more useful contextual information to better predict small objects. The decoder LMD restores feature maps by using channel space attention modules and using multiple feature fusion methods. We also designed a series of ablation experiments to verify the effectiveness of each module. We also conducted some comprehensive comparisons with other methods on the Cityscape and CamVid datasets. Specifically, our DeliteSeg achieved 73.6 % and 73.9 % mIoU on the aforementioned dataset, with FPS of 123.7 and 116.4, respectively. In summary, our network has achieved a good balance between segmentation accuracy and efficiency.

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.: 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.

[3] Zhao, H.—Shi, J.—Qi, X.—Wang, X.—Jia, J.: Pyramid Scene Parsing Network. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 6230–6239, doi: 10.1109/CVPR.2017.660.

[4] Yang, M.—Yu, K.—Zhang, C.—Li, Z.—Yang, K.: DenseASPP for Semantic Segmentation in Street Scenes. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 3684–3692, doi: 10.1109/CVPR.2018.00388.

[5] Lin, G.—Milan, A.—Shen, C.—Reid, I.: RefineNet: Multi-Path Refinement Networks for High-Resolution Semantic Segmentation. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 5168–5177, doi: 10.1109/CVPR.2017.549.

[6] Howard, A. G.—Zhu, M.—Chen, B.—Kalenichenko, D.—Wang, W.—Weyand, T.—Andreetto, M.—Adam, H.: MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. CoRR, 2017, doi: 10.48550/arXiv.1704.04861.

[7] 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.

[8] Wei, H.—Liu, X.—Xu, S.—Dai, Z.—Dai, Y.—Xu, X.: DWRSeg: Rethinking Efficient Acquisition of Multi-Scale Contextual Information for Real-Time Semantic Segmentation. CoRR, 2022, doi: 10.48550/arXiv.2212.01173.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] Yu, C.—Gao, C.—Wang, J.—Yu, G.—Shen, C.—Sang, N.: BiSeNet V2: Bilateral Network with Guided Aggregation for Real-Time Semantic Segmentation. International Journal of Computer Vision, Vol. 129, 2021, No. 11, pp. 3051–3068, doi: 10.1007/s11263-021-01515-2.

[15] Alom, M. Z.—Taha, T. M.—Yakopcic, C.—Westberg, S.—Sidike, P.—Nasrin, M. S.—Van Esesn, B. C.—Awwal, A. A. S.—Asari, V. K.: The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches. CoRR, 2018, doi: 10.48550/arXiv.1803.01164.

[16] 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.

[17] Dai, J.—Qi, H.—Xiong, Y.—Li, Y.—Zhang, G.—Hu, H.—Wei, Y.: Deformable Convolutional Networks. 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 764–773, doi: 10.1109/ICCV.2017.89.

[18] 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.

[19] Hu, J.—Shen, L.—Albanie, S.—Sun, G.—Vedaldi, A.: Gather-Excite: Ex-

ploiting Feature Context in Convolutional Neural Networks. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (Eds.): Advances in Neural Information Processing Systems 31 (NeurIPS 2018). Curran Associates, Inc., 2018, pp. 9401–9411, doi: 10.48550/arXiv.1810.12348.

[20] 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.

[21] Fu, J.—Liu, J.—Tian, H.—Li, Y.—Bao, Y.—Fang, Z.—Lu, H.: Dual Attention Network for Scene Segmentation. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 3141–3149, doi: 10.1109/CVPR.2019.00326.

[22] Liu, H.—Liu, F.—Fan, X.—Huang, D.: Polarized Self-Attention: Towards High-Quality Pixel-Wise Regression. CoRR, 2021, doi: 10.48550/arXiv.2107.00782.

[23] Yuan, Y.—Huang, L.—Guo, J.—Zhang, C.—Chen, X.—Wang, J.: OCNet: Object Context for Semantic Segmentation. International Journal of Computer Vision, Vol. 129, 2021, No. 8, pp. 2375–2398, doi: 10.1007/s11263-021-01465-9.

[24] Huang, Z.—Wang, X.—Huang, L.—Huang, C.—Wei, Y.—Liu, W.: CCNet: Criss-Cross Attention for Semantic Segmentation. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 603–612, doi: 10.1109/ICCV.2019.00069.

[25] Peng, J.—Liu, Y.—Tang, S.—Hao, Y.—Chu, L.—Chen, G. et al.: PP-LiteSeg: A Superior Real-Time Semantic Segmentation Model. CoRR, 2022, doi: 10.48550/arXiv.2204.02681.

[26] Gao, S. H.—Cheng, M. M.—Zhao, K.—Zhang, X. Y.—Yang, M. H.—Torr, P.: Res2Net: A New Multi-Scale Backbone Architecture. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 43, 2021, No. 2, pp. 652–662, doi: 10.1109/TPAMI.2019.2938758.

[27] 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.

[28] 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.

[29] Zhang, Y.—Qiu, Z.—Liu, J.—Yao, T.—Liu, D.—Mei, T.: Customizable Architecture Search for Semantic Segmentation. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 11633–11642, doi: 10.1109/CVPR.2019.01191.

[30] Poudel, R. P. K.—Liwicki, S.—Cipolla, R.: Fast-SCNN: Fast Semantic Segmentation Network. CoRR, 2019, doi: 10.48550/arXiv.1902.04502.

[31] Hu, X.—Liu, Y.: Lightweight Multi-Scale Attention-Guided Network for Real-

Time Semantic Segmentation. Image and Vision Computing, Vol. 139, 2023, Art. No. 104823, doi: 10.1016/j.imavis.2023.104823.

[32] 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.

[33] Gao, G.—Xu, G.—Li, J.—Yu, Y.—Lu, H.—Yang, J.: FBSNet: A Fast Bilateral Symmetrical Network for Real-Time Semantic Segmentation. IEEE Transactions on Multimedia, Vol. 25, 2022, pp. 3273–3283, doi: 10.1109/TMM.2022.3157995.

[34] Li, X.—You, A.—Zhu, Z.—Zhao, H.—Yang, M.—Yang, K.—Tan, S.—Tong, Y.: Semantic Flow for Fast and Accurate Scene Parsing. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J. M. (Eds.): Computer Vision – ECCV 2020. Springer, Cham, Lecture Notes in Computer Science, Vol. 12346, 2020, pp. 775–793, doi: 10.1007/978-3-030-58452-8_45.

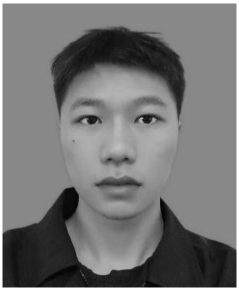
[35] Wan, Q.—Huang, Z.—Lu, J.—Yu, G.—Zhang, L.: SeaFormer: Squeeze-Enhanced Axial Transformer for Mobile Semantic Segmentation. The Eleventh International Conference on Learning Representations (ICLR 2023), 2023, doi: 10.48550/arXiv.2301.13156.

[36] Oršic, M.—Krešo, I.—Bevandic, P.—Šegvic, S.: In Defense of Pre-Trained ImageNet Architectures for Real-Time Semantic Segmentation of Road-Driving Images. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 12599–12608, doi: 10.1109/CVPR.2019.01289.

[37] Zhou, Q.—Wang, Y.—Fan, Y.—Wu, X.—Zhang, S.—Kang, B.—Latecki, L. J.: AGLNet: Towards Real-Time Semantic Segmentation of Self-Driving Images via Attention-Guided Lightweight Network. Applied Soft Computing, Vol. 96, 2020, Art. No. 106682, doi: 10.1016/j.asoc.2020.106682.

[38] Hu, P.—Caba, F.—Wang, O.—Lin, Z.—Sclaroff, S.—Perazzi, F.: Temporally Distributed Networks for Fast Video Semantic Segmentation. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 8815–8824, doi: 10.48550/arXiv.2004.01800.

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.



Yifei Luan received his B.Sc. degree in computer science and technology from the Anhui Institute of Science and Technology (AIST), 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 interest is deep learning.

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.