

# Lightweight Real-time Semantic Segmentation Network with Efficient Transformer and CNN

Guoan Xu, Juncheng Li, Guangwei Gao, *Senior Member, IEEE*, Huimin Lu, *Senior Member, IEEE*  
 Jian Yang, *Member, IEEE* and Dong Yue, *Fellow, IEEE*,

**Abstract**—In the past decade, convolutional neural networks (CNNs) have shown prominence for semantic segmentation. Although CNN models have very impressive performance, the ability to capture global representation is still insufficient, which results in suboptimal results. Recently, Transformer achieved huge success in NLP tasks, demonstrating its advantages in modeling long-range dependency. Recently, Transformer has also attracted tremendous attention from computer vision researchers who reformulate the image processing tasks as a sequence-to-sequence prediction but resulted in deteriorating local feature details. In this work, we propose a lightweight real-time semantic segmentation network called LETNet. LETNet combines a U-shaped CNN with Transformer effectively in a capsule embedding style to compensate for respective deficiencies. Meanwhile, the elaborately designed Lightweight Dilated Bottleneck (LDB) module and Feature Enhancement (FE) module cultivate a positive impact on training from scratch simultaneously. Extensive experiments performed on challenging datasets demonstrate that LETNet achieves superior performances in accuracy and efficiency balance. Specifically, It only contains 0.95M parameters and 13.6G FLOPs but yields 72.8% mIoU at 120 FPS on the Cityscapes test set and 70.5% mIoU at 250 FPS on the CamVid test dataset using a single RTX 3090 GPU. Source code will be available at <https://github.com/IVIPLab/LETNet>.

**Index Terms**—Real-time semantic segmentation, Convolutional neural network, Lightweight network, Transformer.

## I. INTRODUCTION

The task of semantic segmentation aims to assign a semantic label to each pixel, which is widely used in augmented reality devices, autonomous driving, and video surveillance. Since the Fully Convolutional Network (FCN [1]) was proposed, existing semantic segmentation models have been using it as

This work was partly supported by the National Natural Science Foundation of China under Grants 61972212 and 61833011, and the Open Fund Project of Provincial Key Laboratory for Computer Information Processing Technology (Soochow University) under Grant KJS2274. (*Guoan Xu, Juncheng Li, and Guangwei Gao contributed equally to this work.*) (*Corresponding author: Guangwei Gao*)

Guoan Xu, Guangwei Gao, and Dong Yue are with the Institute of Advanced Technology, Nanjing University of Posts and Telecommunications, Nanjing 210023, China, and also with the Provincial Key Laboratory for Computer Information Processing Technology, Soochow University, Suzhou 215006, China (e-mail: xga\_njupt@163.com, csggao@gmail.com, medongy@vip.163.com).

Juncheng Li is with the School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China, also with Jiangsu Key Laboratory of Image and Video Understanding for Social Safety, Nanjing University of Science and Technology, Nanjing 210049, China (e-mail: cvjunchengli@gmail.com).

Huimin Lu is with the Kyushu Institute of Technology, Kitakyushu 804-8550, Japan (e-mail: dr.huimin.lu@ieee.org).

Jian Yang is with the School of Computer Science and Technology, Nanjing University of Science and Technology, Nanjing 210049, China (e-mail: csjyang@njust.edu.cn).

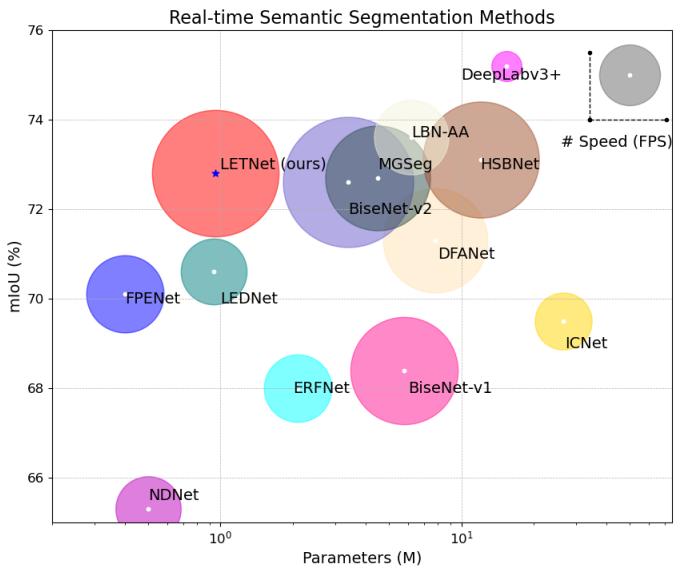


Fig. 1. Accuracy-Parameters-Speed evaluations on the Cityscapes test dataset. A larger radius of a circle indicates a faster inference speed.

a prototype for improvement. However, the receptive field of FCN-based models is limited. Thus, it is impossible to learn remote dependencies, which is not conducive to the extraction of global semantic information that is critical to intensive tasks, especially the semantic segmentation task. To address this limitation, some recent methods propose the use of large convolutional kernel [2], dilated convolution [3], and feature pyramids [4] to expand the sensory field. Another approach is to integrate Non-local [5] from the natural language processing (NLP) [6] domains into the FCN structure, which is designed to model the global interaction of all pixels in the feature map, but with high memory and high computational costs. On the other hand, researchers began experimenting with completely removing convolution and exploring a model that used only attention modules alone, Transformer [7], which was designed to model sequence-to-sequence long-range dependencies and capture relationships anywhere in the sequence.

Unlike previous CNN-based approaches, Transformer [7] is not only powerful in terms of global context modeling but also achieves good results on downstream tasks in the case of large-scale pre-training. In [8], a visual Transformer (ViT) is proposed to perform image recognition tasks using a two-dimensional image block with positional embedding as input. However, the disadvantage of ViT [8] compared to

CNN is that it must be pre-trained on large data sets. The image resolution is much larger than words in NLP [6]. Many computer vision tasks, such as semantic segmentation, require intensive prediction at the pixel level, which is difficult to process for Transformers on high-resolution images because its computational complexity of self-attention is related to the size of the image at the quadratic level. In addition, when Transformer is used in the image processing field, the two-dimensional image is sliced and fed into the model as a one-dimensional sequence, thus breaking the connection between local structures and focusing only on the global context at all stages. As a result, low-resolution features lack detailed localization information that cannot be effectively recovered by directly upsampling to full resolution, resulting in rough segmentation results.

Although Transformer can achieve global information modeling, it cannot extract fine spatial details. On the contrary, CNN can provide a way to extract low-level visual cues that compensate well for this fine spatial detail. Therefore, some methods try to combine CNN with Transformer to handle semantic segmentation tasks. For example, in the field of medical image segmentation, TransUNet [9], TransBTS [10], and TransFuse [11] have achieved satisfactory results. Inspired by this, we also propose a lightweight real-time semantic segmentation model, named LETNet, based on the CNN and Transformer. As depicted in Fig. 1, our LETNet achieves a good balance between the performance, model size, and inference speed of the model. The main contribution of this paper is three folds:

- We propose a Lightweight Dilated Bottleneck (LDB) to extract important semantic information. LDB consists of dilated convolution and depth-wise separable convolution, achieving extreme weight reduction in terms of parameters and computational quantities.
- We propose a hybrid network, LETNet, for semantic segmentation. LETNet adopts the most concise encoder-decoder structure and regards the efficient Transformer as a capsule network to learn global information. Meanwhile, a Feature Enhancement (FM) module is added to the jump connection to help supplement the boundary detail information when restoring the resolution.
- LETNet achieved 72.8% mIoU on the cityscapes test set on the single RTX3090 hardware platform with only 0.95M of parameter quantities and 70.5% of the good performance on the CamVid dataset. The performance is better than most existing models.

## II. RELATED WORK

### A. CNN-based Semantic Segmentation Methods

Owing to the powerful feature representation capabilities of convolutional neural networks, semantic segmentation methods have also made great progress [12]. The groundbreaking article based on CNN was FCN [1], after which many architectures have been refined on this basis. To alleviate the contradiction between image resolution and limited receptive field, DeepLab [3] and PSPNet [4] employed parallel atrous convolutions to build an atrous spatial pyramid pooling (ASPP)

module, which introduces good descriptors for various scale contextual information. Additionally, with the advantages of modeling feature dependencies, the self-attention mechanism has attracted the interest of many scholars. For instance, Based on SENet [13], a local cross-channel interaction strategy without dimensionality reduction and a method for adaptively selecting the size of one-dimensional convolution kernels are proposed in ECANet [14]. In addition, there is an attention mechanism commonly used in NLP [6] to model long-distance dependencies. Typical of these is Non-local neural networks [5], which uses the similarity of two points to weight the features of each position. DANet [15] used ResNet [16] as the backbone network, followed by an attention module composed of spatial dimension and channel dimension in parallel for capturing long-range deep features dependencies to improve the segmentation result. CCNet [17] was improved to calculate the association between the pixel and all the pixels in the row and column, which economizes the computational burden. LRNNet [18] proposed an effective simplified Non-local module that uses regional singular vectors to generate more simplified and representative features to model remote dependency and global feature selection.

Although these types of methods achieve good results, they do not change the fact that non-local is essentially a pixel-wise matrix algorithm, which still makes the computer face a huge computational challenge. So the lightweight network came into being. For example, ICNet [19] used multi-scale images as input where high-level semantic information and low-level spatial details are utilized. BiseNet [20] and BiseNet-v2 [21] proposed two-path architecture, one branch is responsible for extracting deep semantic information, and the other high-resolution shallow branch is responsible for providing detailed information supplement. DFANet [22] utilized a feature reuse policy, which enhanced the interaction and aggregation of features at different levels. Furthermore, point-wise attention is used at the end of each stage to enhance the feature expression ability while ensuring that the computation is small. ESPNet [23] and ESPNet-v2 [24] reduced the number of parameters and computation by integrating decomposed convolution into point-wise convolution and dilation convolution. In addition, NRD [25] used dynamic convolutional neural networks to extract feature information from images. However, CNN-based methods always have a problem that cannot be completely solved, and that is the limitation exhibited by modeling long-range relationships. While existing methods only resort to building a deep encoder and downsampling operations, the negative effects are redundant parameters and the loss of more local details.

### B. Transformer-based Semantic Segmentation Methods

Transformer was first proposed in [7] and has achieved great success in natural language processing. Unlike CNN, Transformer is not only powerful in terms of global context modeling but also achieves good results on downstream tasks in the case of large-scale pre-training. For example, ViT [8] proposed to perform image recognition tasks with 2D image patches with position embeddings as input. DETR [26] and

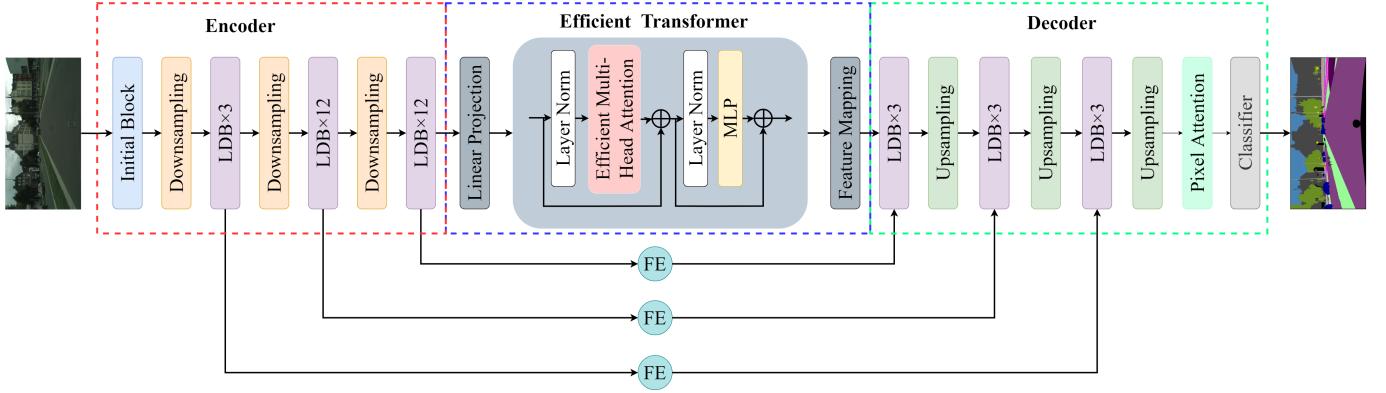


Fig. 2. The proposed Lightweight Real-time Semantic Segmentation Network with Efficient Transformer and CNN (LETNet).

the deformable version utilized Transformers Encoder-decoder to fuse context in the detection head. SETR [27] solved semantic segmentation from the perspective of sequence to sequence. It abandoned CNN and is a structure completely based on Transformer. Meanwhile, SegFormer [28] proposed a hierarchical encoder structure, output multi-scale features, and fuse them in the decoder. After that, some deformed structures were proposed for medical image segmentation [9]–[11], [29].

Although the above methods have achieved good results, since the computational complexity of the Transformer is proportional to the square of the image size, it will increase a lot of computational burdens. In addition, when using Transformer in the image domain, the input patches are regarded as a one-dimensional sequence input to the model, which destroys the connection between local structures and only focuses on the global context modeling at the stage. The lack of detailed localization information leads to coarse segmentation results. On the other hand, the CNN architecture provides a path to extract low-level vision cues that can compensate well for such fine spatial details. Therefore, we aim to explore a method combining CNN with Transformer to handle the segmentation task and use an efficient Transformer to reduce memory consumption.

### III. PROPOSED METHOD

#### A. Network Architecture

As shown in Fig. 2, LETNet comprises an Encoder, a Decoder, an Efficient Transformer, and three long skip connections. Specifically, the Encoder and Decoder are CNN structures used to extract local features for better image representation. The transformer can reflect complex spatial transformation and long-distance feature dependencies by self-attention and multi-layer perceptron (MLP) structure to obtain global feature representation. The three long-distance connections are inspired by UNet [30], which combines low-level spatial information with high-level semantic information for high-quality segmentation.

#### B. Lightweight Dilated Bottleneck (LDB)

As shown in Fig. 3, the structure of LDB adopted the idea of ResNet [16] on the whole, and the module is designed as a

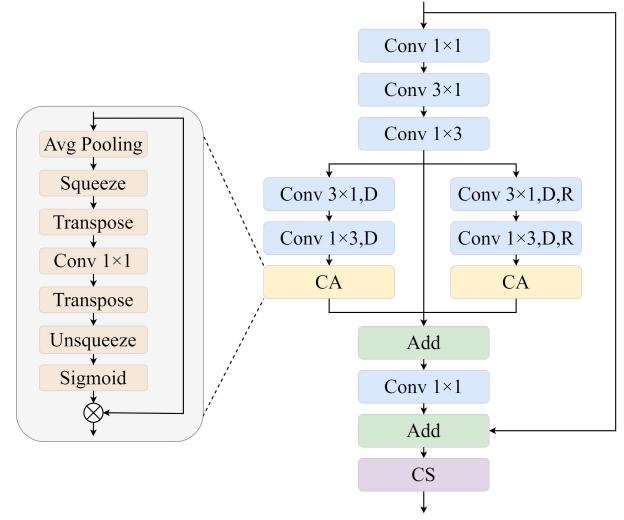


Fig. 3. The proposed Lightweight Dilated Bottleneck (LDB). Among them,  $D$  represents depth-wise convolution,  $R$  is the kernel of dilated convolution,  $CA$  means Channel Attention, and  $CS$  denotes the channel shuffle operation.

residual module to collect more feature information while the number of network layers is as small as possible. Specifically, at the bottleneck, the number of channels of the input feature is reduced to half by  $1 \times 1$  convolution. After reducing the number of channels, the amount of parameters and calculations is greatly reduced. Although this will lose a part of the accuracy, it will be more beneficial to stack two modules more than make up for the loss at this point. At the same time, due to the use of  $1 \times 1$  convolution, the network depth must be deepened to obtain a larger receptive field. Therefore, after the  $1 \times 1$  convolution, the decomposed convolutions of  $3 \times 1$  and  $1 \times 3$  are added to expand the feeling to capture a wider range of contextual information. Moreover, decomposed convolution is also based on considering the number of parameters and the amount of calculation. Similarly, in the next two-branch structure, both branches also use decompose convolution, one of which is responsible for local and short-distance feature information, and the other uses atrous convolution, which is responsible for extracting feature information from a larger receptive field under different atrous rates. Next to these two branches are channel attentions, inspired by ECANet [14], which aims to

build an attention matrix in the channel dimension to enhance feature expression and suppress noise interference because, for CNN, most of the feature information is contained in the channel. Then, the two low-dimensional branches and middle features are fused and input to a  $1 \times 1$  point-wise convolution below to restore the number of channels of the feature map to be the same as the number of channels of the input feature map. Finally, the strategy of channel shuffle is used to avoid the drawbacks of information independence and no correlation between channels caused by depth-wise convolution and to promote the exchange of semantic information between different channels. The complete operation is shown as follows:

$$F_1 = f_{1 \times 3}(f_{3 \times 1}(f_{1 \times 1}(x))), \quad (1)$$

$$F_{21} = f_{CA}(f_{1 \times 3, D}(f_{3 \times 1, D}(F_1))), \quad (2)$$

$$F_{22} = f_{CA}(f_{1 \times 3, D, R}(f_{3 \times 1, D, R}(F_1))), \quad (3)$$

$$y = f_{CS}(f_{1 \times 1}(F_1 + F_{21} + F_{22}) + x), \quad (4)$$

where  $x$  represents the input feature maps,  $y$  represents the output feature map, and  $f_{k \times k}(\cdot)$  are convolution operation.

### C. Efficient Transformer (ET)

As we mentioned before, despite its advantages in local feature extraction, the ability of CNN to capture global representations is still insufficient, which is important for many high-level computer vision tasks. To deal with this problem, we introduce Transformer to learn long-range dependencies. However, in image processing tasks, since the input image resolution is much larger than the words in the natural language processing field. We introduce the Efficient Transformer (ET), which is inspired by ETSR [31]. Different from the traditional Transformer, ET occupies fewer computing resources. Meanwhile, to avoid excessive memory usage and computational load, we abandon the series connection of multiple ETs in ETSR [31] and only use one ET as a capsule network, which is placed in the middle of the entire network. As we all know, Transformer consists of two layer normalizations, one Multi-Head Attention (MHA), and one Multi-Layer Perception (MLP). The biggest difference between ET and the original Transformer is that MHA. After the layer normalization, ET sets up a reduction layer to halve the number of channels, which reduces part of the computation. Then, a linear layer projects the feature map into three matrices,  $Q$  (query),  $K$  (key), and  $V$  (value). Specifically, in EMHA,  $Q$ ,  $K$ , and  $V$  are first split into  $s$  segments, and then a scaled dot product attention of  $Q_i$ ,  $K_i$ , and  $V_i$  is executed correspondingly. After that, we concatenate the obtained  $O_1, \dots, O_s$  to get the whole output  $O$ . In fact, it relies on the idea of group convolution, splitting large matrices into small matrices and then calculating and finally merging, so as to achieve the purpose of reducing the amount of calculation. Finally, the expansion layer is employed to restore the number of channels. The architecture of EMHA is shown in Fig. 4 (a) and the Scaled Dot-Product Attention (SDPA) operation can be defined as:

$$O_i = \text{soft max} \left( \frac{Q_i K_i^T}{\sqrt{d}} \right) V_i, \quad (5)$$

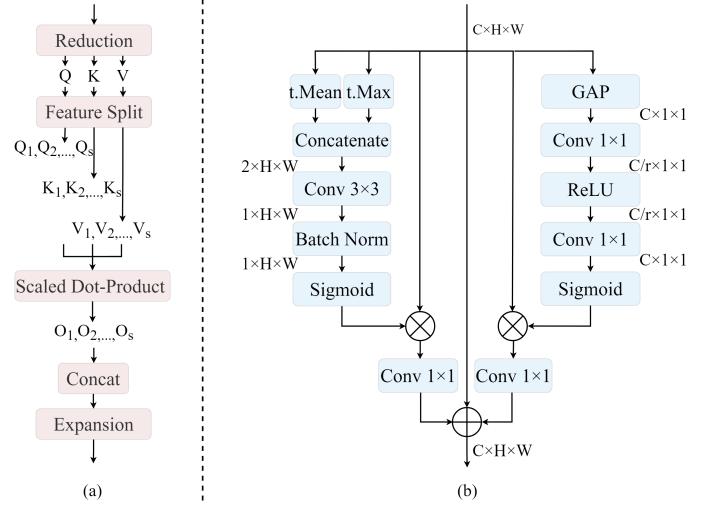


Fig. 4. Schematic diagram of the (a) Efficient Multi-Head Attention (EMHA) and (b) Feature Enhancement (FE) module. Please zoom in for details.

where  $Q$ ,  $K$ , and  $V$  represent *query*, *key*, and *value* metrices,  $d$  is the embedding dimension. Afterward, all the outputs ( $O_1, O_2, \dots, O_s$ ) of SDPA are concatenated together to generate the whole output feature  $O$ .

### D. Feature Enhancement (FE)

In the neural network, the lower layer has high resolution and accurate spatial information (the resolution corresponds to the spatial position) but has little semantic information. In contrast, the high layer has low resolution and lacks spatial position information but rich semantic information. Therefore, in the segmentation task, to make the high-level information also have enough spatial information, the low-level spatial information, and high-level semantic information are usually combined to perform high-quality segmentation. Therefore, we use the UNet-style structure to fuse the high-level and low-level feature maps of the same resolution. At the same time, in the process of three long connections, we propose a Feature Enhancement (FE) module to improve the ability of feature expression. As shown in the fig. 4 (b), feature dependency modeling is carried out from two dimensions, one is the channel dimension, the other is the spatial dimension, and the two dimensions are transformed at the same time and finally fused to transmit the low-level information to the high-level more effectively. The operation can be defined as:

$$M_C = X * \sigma \left( f_{1 \times 1}^C \left( \gamma \left( f_{1 \times 1}^r(f_{AP}(X)) \right) \right) \right), \quad (6)$$

$$M_S = X * \sigma \left( B(f_{3 \times 3}(\text{Concat}[f_{AP}(X), f_{MP}(X)])) \right), \quad (7)$$

$$Y = f_{1 \times 1}(M_C) + f_{1 \times 1}(M_S) + X, \quad (8)$$

where  $X$  is the input feature,  $M_C$  represents the output of channel dimension,  $M_S$  represents the output of spatial dimension,  $\sigma$  denotes the sigmoid function,  $\gamma$  means ReLU function,  $B$  represents Batch Normalization,  $C$  is the channel of the feature map,  $r$  means reduction,  $f_{AP}(\cdot)$  and  $f_{MP}(\cdot)$  denote the average pooling and max pooling operations, respectively.

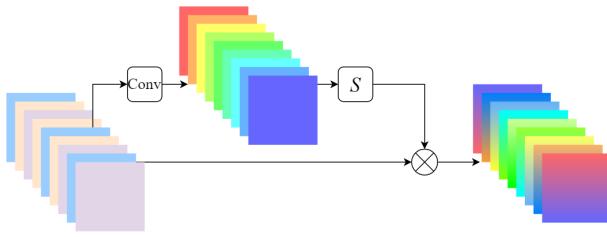


Fig. 5. Schematic diagram of the pixel attention mechanism. Among them,  $S$  means the *Sigmoid* function.

### E. Pixel Attention (PA)

The pixel attention (PA) mechanism learns weights based on the importance of features at different pixel positions. This means that each channel has the same weight but the weights are different at different pixel positions for the same channel. The pixel attention pays more attention to the edges and textures of objects in the image, so adding PA can facilitate the recovery of edge detail information, thereby improving the performance of segmentation. The operation is shown in Fig. 5 and the formula is as follows:

$$y = \sigma(f_{1 \times 1}(X)) * X, \quad (9)$$

where  $X$  is the input feature,  $\sigma$  denotes the sigmoid function, and  $f_{1 \times 1}(\cdot)$  is the convolutional layer with kernel size of 1.

## IV. EXPERIMENTS

### A. Datasets

**Cityscapes:** The resolution of images in this dataset is  $2048 \times 1024$ , collected from German and French urban road scenes in 50 different cities, including pedestrians, roads, vehicles, etc. It has 19 categories for the evaluation of semantic segmentation. Among them, 5000 finely annotated images are further divided into 2075, 500, and 1525 for training, validation, and testing, respectively.

**Camvid:** It contains 701 urban road images of  $960 \times 720$ , which have 11 categories, and the finely annotated images are divided into 367, 101, and 233 for training, validation, and testing, respectively.

### B. Model Settings

In this work, we use the PyTorch framework to build the model and train it on an RTX3090 GPU. In Table I, we show the detail of the model settings of LETNet on the Cityscapes and CamVid datasets. Meanwhile, the learning rate varies with iterations and can be calculated as follows:

$$lr = lr_{initial} \times \left(1 - \frac{iteration}{max\_iteration}\right)^{0.9}, \quad (10)$$

where  $lr_{initial}$  represents the initial learning rate. It is worth noting that we train Cityscapes and CamVid separately with different parameter settings since the resolution of the datasets is different.

TABLE I  
THE DETAIL OF THE MODEL SETTINGS.

Dataset	Cityscapes	CamVid
Batch size	6	8
Loss function	CrossEntropy Loss	
Optimization method	SGD(momentum 0.9)	Adam(momentum 0.9)
Weight decay	$1 \times 10^{-4}$	$2 \times 10^{-4}$
Initial learning rate	$4.5 \times 10^{-2}$	$1 \times 10^{-3}$
Learning rate policy	Poly	

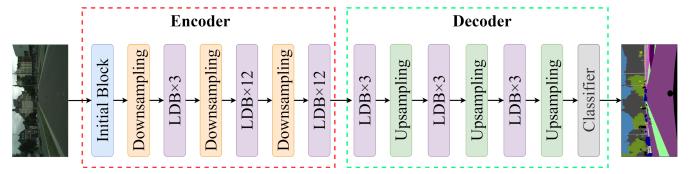


Fig. 6. The architecture of the baseline model.

### C. Ablation Study

As depicted in Table II, some ablation experiments on the proposed modules are designed to prove the validity of these modules. **A. Long Connection, B. Feature Enhancement, C. Efficient Transformer.** Meanwhile, the baseline model is shown in Fig. 6. It is worth noting that the baseline model does not introduce any proposed modules. The architecture of the baseline model is composed of an Encoder-Decoder composed of LDBs, which achieves 69.85% mIoU on the validation set.

In group A, it is the effect of gradually adding L1, L2, L3, and it can be seen that after adding L1, there is a significant performance improvement of 0.58%, proving that shallow information is greatly beneficial to the resolution of deep semantic information recovery. Meanwhile, when adding three long skip connections to the model, the performance increased by 1.16% mIoU. This further verifies the importance of long connections for image segmentation.

In Group B, it is the effect of gradually adding L1, L2, and L3 to the network after joining Feature Enhancement. Comparing B1 and A1, it can be seen that adding FE only adds 101K parameters and 0.0032G FLOPs, but it can improve the performance of the model by 0.2% mIoU, which is quite impressive.

In Group C, after introducing the Efficient Transformer, the model's performance has improved by 3.07% mIoU, and the introduction of the pixel attention mechanism has also brought 0.5% benefits to the model. Finally, C3\* is the final version of the proposed LETNet, which improves the performance of the baseline model by 5.15% mIoU.

All the above experiments fully demonstrate the necessity and effectiveness of the proposed modules and strategies.

### D. Comparisons with Advanced Models

In this part, we compared recent years of representative semantic segmentation methods on the Cityscapes and CamVid

TABLE II

ABLATION STUDY FOR THE PROPOSED MODULES ON CITYSCAPES. L1, L2, L3: LINE1, LINE2, LINE3, FE: FEATURE ENHANCEMENT, \* REPRESENTS THE FINAL VERSION, AND THE DIFFERENCE MEANS THE GAP IN PERFORMANCE BETWEEN THE MODEL AND BASELINE IN EACH GROUP.

	Method	Parameter (K) $\downarrow$	FLOPs (G) $\downarrow$	mIoU (%) $\uparrow$	Difference
<b>A:</b> Long Connection	Baseline	723,400	13.068300560	69.85	-
	<b>A1:</b> Baseline+L1	731,128	13.319958800	70.43	+0.58
	<b>A2:</b> Baseline+L1+L2	758,856	13.546451216	70.79	+0.94
	<b>A3:</b> Baseline+L1+L2+L3	761,976	13.552742672	71.01	+1.16
<b>B:</b> Feature Enhancement	Baseline	723,400	13.068300560	69.85	-
	<b>B1:</b> Baseline+L1(FE)	731,229	13.323170112	70.63	+0.78
	<b>B2:</b> Baseline+L1(FE)+L2(FE)	759,058	13.550465392	71.14	+1.29
	<b>B3:</b> Baseline+L1(FE)+L2(FE)+L3(FE)	762,279	13.556957648	71.46	+1.61
<b>C:</b> Efficient Transformer	Baseline	723,400	13.068300560	69.85	-
	<b>C1:</b> Baseline+ET	911,824	13.068487760	72.92	+3.07
	<b>C2:</b> Baseline+ET+L1(FE)+L2(FE)+L3(FE)	950,703	13.557144848	74.53	+4.68
	<b>C3*:</b> Baseline+ET+L1(FE)+L2(FE)+L3(FE)+PA	950,975	13.590699280	75.00	+5.15

TABLE III  
COMPARISONS WITH THE STATE-OF-ARTS METHODS ON THE CITYSCAPES DATASET.

	Methods	Year	Resolution	Backbone	Parameter (M) $\downarrow$	FLOPs (G) $\downarrow$	Speed (FPS) $\uparrow$	mIoU (%) $\uparrow$
Large Size	DeepLab [3]	2015	512×1024	ResNet-101	262.10	457.8	0.25	63.5
	DeepLab-v3+ [32]	2018	-	Xception	15.40	555.4	8.40	75.2
	DenseASPP [33]	2018	512×512	DenseNet	35.70	632.9	-	80.6
	PSPNet [4]	2017	713×713	ResNet-101	250.80	412.2	0.78	81.2
	DANet [15]	2019	1024×1024	ResNet-101	66.60	1298.8	4.00	81.5
	CCNet [17]	2019	1024×1024	ResNet-101	66.50	1153.9	4.70	81.9
	SETR-PUP [27]	2021	768×768	ViT-Large	318.30	-	0.50	82.2
	SegFormer [28]	2021	1024×2048	MiT-B5	84.70	1447.6	2.50	84.0
	Lawin Transformer [34]	2022	1024×1024	Swin-L	-	1797	-	84.4
Medium Size	SegNet [35]	2017	640×360	VGG-16	29.50	286.0	17	57.0
	SQNet [36]	2016	1024×2048	SqueezeNet	-	270.0	17	59.8
	BiseNet-v1 [20]	2018	768×1536	Xception	5.80	14.8	106	68.4
	ICNet [19]	2018	1024×2048	PSPNet-50	26.50	28.3	30	69.5
	DFANet [22]	2019	1024×1024	Xception	7.80	3.4	100	71.3
	STDC1-50 [37]	2021	512×1024	-	8.40	-	87	71.9
	FPANet [38]	2022	512×1024	-	14.10	-	-	72.0
	HSB-Net [39]	2021	512×1024	ResNet-34	12.10	-	124	73.1
	LBN-AA [40]	2021	448×896	No	6.20	49.5	51	73.6
Small Size	ENet [41]	2016	512×1024	No	0.36	3.8	135	58.3
	ESPNet [23]	2018	512×1024	ESPNet	0.36	-	113	60.3
	CGNet [42]	2020	360×640	No	0.50	6.0	-	64.8
	NDNet [43]	2021	1024×2048	No	0.50	14.0	40	65.3
	ESPNet-v2 [24]	2019	512×1024	ESPNet-v2	-	2.7	80	66.2
	ADSCNet [44]	2020	512×1024	No	-	-	77	67.5
	ERFNet [45]	2017	512×1024	No	2.10	-	42	68.0
	CFPNet [46]	2021	1045×2048	No	0.55	-	30	70.1
	FPENet [47]	2019	512×1024	No	0.40	12.8	55	70.1
	LEDNet [48]	2019	512×1024	No	0.94	-	40	70.6
	SGCPNet [49]	2022	1024×2048	No	0.61	4.5	103	70.9
	FBSNet [50]	2022	512×1024	No	0.62	9.7	90	70.9
	EdgeNet [51]	2021	512×1024	No	-	-	31	71.0
	MSCFNet [12]	2022	512×1024	No	1.15	17.1	50	71.9
	BiseNet-v2 [21]	2021	512×1024	Xception	3.40	21.2	156	72.6
	MGSeg [52]	2021	1024×1024	ShuffleNet-v2	4.50	16.2	101	72.7
LETNet (ours)		-	512×1024	No	0.95	13.6	150	72.8

TABLE IV  
PER-CLASS IoU (%) RESULTS ON THE CITYSCAPES TEST SET. “AVG” REPRESENTS THE AVERAGE RESULTS OF ALL THESE CATEGORIES. OBVIOUSLY, OUR FBSNET ACHIEVES THE BEST mIoU RESULTS.

Methods	Avg	Bic	Bus	Bui	Car	Fen	Mot	Pol	Per	Rid	Roa	Sid	Sky	Tru	Tra	TLi	Ter	TSi	Veg	Wal
SegNet [35]	57.0	51.9	43.1	84.0	89.3	29.0	35.8	35.1	62.8	42.8	96.4	73.2	91.8	38.1	44.1	39.8	63.8	45.1	87.0	28.4
ENet [41]	58.3	55.4	50.5	75.0	90.6	33.2	38.8	43.4	65.5	38.4	96.3	74.2	90.6	36.9	48.1	34.1	61.4	44.0	88.6	32.2
ESPNet [23]	60.3	57.2	52.5	76.2	92.3	36.1	41.8	45.0	67.0	40.9	97.0	77.5	92.6	38.1	50.1	35.6	63.2	46.3	90.8	35.0
ESPNet-v2 [24]	66.2	59.9	65.9	88.8	91.8	42.1	44.2	49.3	72.9	53.1	97.3	78.6	93.3	53.0	53.2	52.6	66.8	60.0	90.5	43.5
ICNet [19]	69.5	70.5	72.7	89.7	92.6	48.9	53.6	61.5	74.6	56.1	97.1	79.2	93.5	51.3	51.3	60.4	68.3	63.4	91.5	43.2
LEDNet [48]	70.6	71.6	64.0	91.6	90.9	49.9	44.4	62.8	76.2	53.7	98.1	79.5	94.9	64.4	52.7	61.3	61.2	72.8	92.6	47.7
FBSNet [50]	70.9	70.1	56.0	91.5	93.9	53.5	56.2	62.5	82.5	63.8	98.0	83.2	94.4	50.5	37.6	67.6	70.5	71.5	92.7	50.9
EdgeNet [51]	71.0	67.7	60.9	91.6	94.3	50.6	55.3	62.6	80.4	61.1	98.1	83.1	94.9	50.0	52.5	67.2	69.7	71.4	92.4	45.4
MSCFNet [12]	71.9	70.2	66.1	91.0	94.1	52.5	57.6	61.2	82.7	62.7	97.7	82.8	94.3	50.9	51.9	67.1	70.2	71.4	92.3	49.0
LETNe (ours)	72.8	69.3	72.4	91.6	94.4	53.7	56.1	61.0	82.3	61.7	98.2	83.6	94.9	55.0	57.0	66.7	70.5	70.5	92.5	50.9

TABLE V  
COMPARISONS WITH THE STATE-OF-ART METHODS ON THE CAMVID DATASET

Method	Year	Resolution	Backbone	Parameter (M)↓	Speed (FPS)↑	mIoU (%)↑
ENet [41]	2016	360×480	No	0.36	61	51.3
SegNet [35]	2017	360×480	VGG-16	29.50	29	55.6
NDNet [43]	2021	360×480	-	0.50	-	57.2
DFANet [22]	2019	720×960	Xception	7.80	120	64.7
BiseNet-v1 [20]	2018	720×960	Xception	5.80	116	65.6
DABNet [53]	2019	360×480	No	0.76	-	66.4
FDDWNet [54]	2020	360×480	No	0.80	79	66.9
ICNet [19]	2018	720×960	PSPNet-50	26.50	28	67.1
LBN-AA [40]	2021	720×960	No	6.20	39	68.0
BiseNet-v2 [21]	2020	720×960	ResNet	49.00	-	68.7
FBSNet [50]	2022	360×480	No	0.62	120	68.9
SGCPNet [49]	2022	720×960	No	0.61	278	69.0
MSCFNet [12]	2021	360×480	No	1.15	-	69.3
LETNet (ours)	2022	360×480	No	0.95	200	70.5

datasets to demonstrate that our method strikes a better balance between segmentation accuracy and segmentation efficiency.

**Evaluation on Cityscapes:** In Table III, we broadly divided the existing excellent methods into three categories: Large Size, Medium Size, and Small Size. Classification is based on parameters and calculations as the standard, the amount of parameters below 5M belongs to the Small Size category, and the calculation amount greater than 300G belongs to the Large Size category. It can be observed that the large-size models have obviously achieved outstanding segmentation effects, but their calculation complexity is high, the operation speed is slow, and they are not suitable for intelligent terminal hardware with high real-time requirements.

In the medium size, the performance of HSBNet [39] and LBN-AA [40] are slightly better than our LETNet. However, we should notice that the parameters of LBN-AA [40] and HSBNet [39] are 6 times and 12 times larger than that of LETNet, respectively.

In the small size category, our LETNet achieves the best results with fewer amount of parameters. This fully demonstrates that our LETNet can achieve a good balance between model size and performance. Indeed, BiseNet-v2 [21] is an

outstanding model, which achieves similar mIoU results with slightly faster speed. However, we should not ignore that the number of parameters of LETNet is only 1/4 of BiseNet-v2. Meanwhile, Bisenet-v2 uses Xception as the backbone, which results in extra computational costs. In addition, we also list some of the methods detailed for each intersection classification over the union in Table IV. Obviously, our LETNet achieves the best results in almost every class.

**Evaluation on CamVid:** In Table V, we provide a comparison of LETNet with other advanced methods on the CamVid dataset. According to the table, we can see that our LETNet still achieves the best result with only 0.95M parameters. This further verifies the effectiveness and excellence of the proposed LETNet.

**Visual Comparison:** In Figs. 7 and 8, we also show the visual comparison of these methods on the Cityscapes and CamVid datasets, respectively. Obviously, our LEFTNet can get more accurate segmentation results. This is due to the well-design structure, and the ability of the Transformer can capture global correlation information, which helps to improve the accuracy of segmentation.

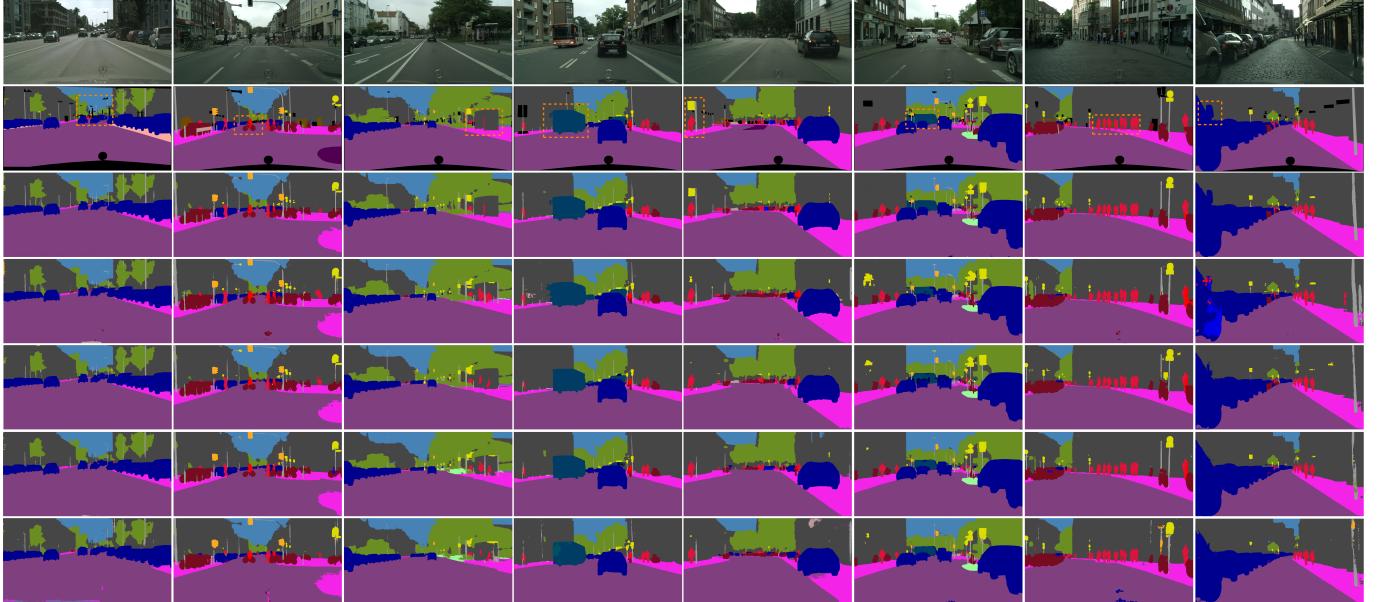


Fig. 7. Visual comparisons on the Cityscapes dataset. From top to bottom are original input images, ground truths, and segmentation results from our LETNet, LEDNet [48], ERFNet [45], ESPNet [23], and ENet [41].

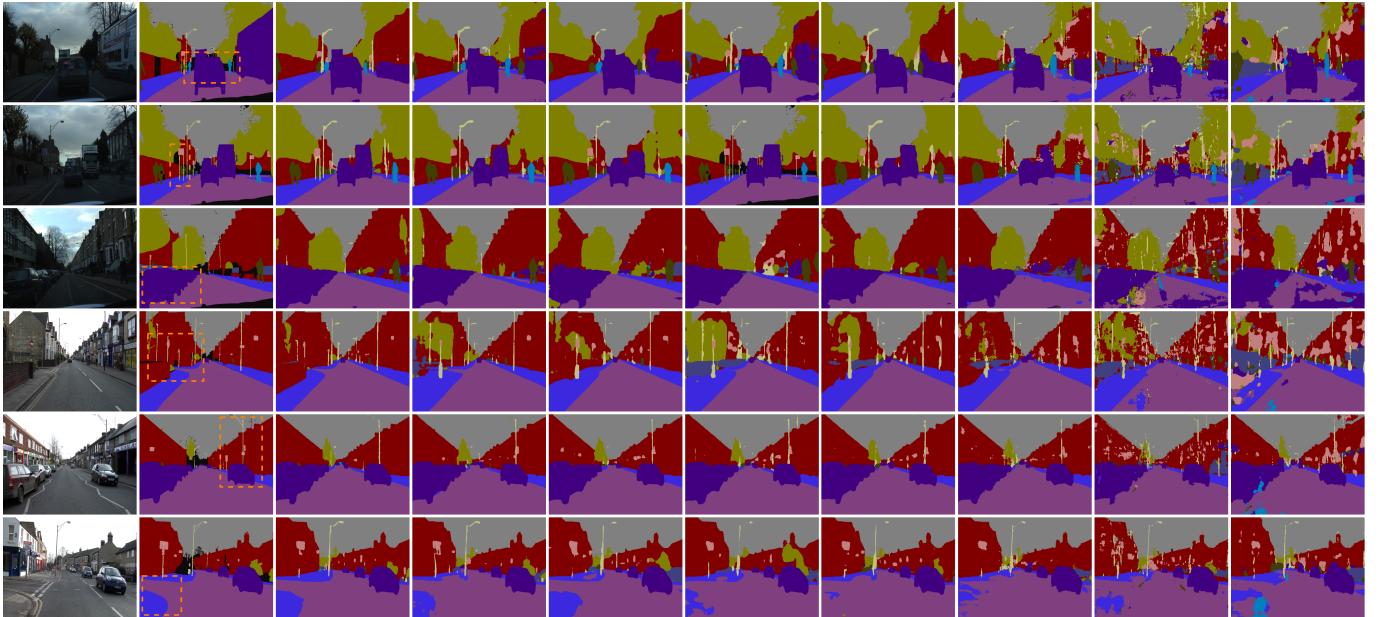


Fig. 8. Visual comparisons on the CamVid dataset. From left to right are original input images, ground truths, and segmentation results from our LETNet, FBSNet [50], ICNet [19], DABNet [53], BiseNet-v1 [20], DFApNet [22], SegNet [35], and ENet [41].

## V. CONCLUSION

In this paper, we proposed a Lightweight Real-time Semantic Segmentation Network with Transformer and CNN. We combine the local feature extraction capabilities of CNNs with the long-range dependency modeling capabilities of Transformers. Specifically, an efficient Transformer is introduced in the middle of the model as a capsule network. Unlike the traditional Transformer, a more lightweight MHA is used, which can significantly reduce GPU memory consumption. Meanwhile, the Lightweight Dilated Bottleneck (LDB) module designed in CNN can learn more features under the premise of

ensuring extreme simplicity and lightweight. Simultaneously, to make up for the shallow detail information lost by CNN in extracting deep semantic information, a U-shaped connection is used in the model. In connecting different levels, a Feature Enhancement (FM) module is also designed to improve the effective feature expression and suppress noise. Extensive experiment results show that our model makes an excellent balance between model size and performance.

## REFERENCES

- [1] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proceedings of the IEEE Conference*

- on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3431–3440.
- [2] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun, “Large kernel matters—improve semantic segmentation by global convolutional network,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4353–4361.
- [3] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2017.
- [4] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2881–2890.
- [5] X. Wang, R. Girshick, A. Gupta, and K. He, “Non-local neural networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7794–7803.
- [6] K. Chowdhary, “Natural language processing,” *Fundamentals of Artificial Intelligence*, pp. 603–649, 2020.
- [7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017, pp. 1–11.
- [8] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, “An image is worth 16x16 words: Transformers for image recognition at scale,” *arXiv preprint arXiv:2010.11929*, 2020.
- [9] J. Chen, Y. Lu, Q. Yu, X. Luo, E. Adeli, Y. Wang, L. Lu, A. L. Yuille, and Y. Zhou, “Transunet: Transformers make strong encoders for medical image segmentation,” *arXiv preprint arXiv:2102.04306*, 2021.
- [10] W. Wang, C. Chen, M. Ding, H. Yu, S. Zha, and J. Li, “Transbts: Multimodal brain tumor segmentation using transformer,” in *Proceedings of International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Springer, 2021, pp. 109–119.
- [11] Y. Zhang, H. Liu, and Q. Hu, “Transfuse: Fusing transformers and cnns for medical image segmentation,” in *Proceedings of International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Springer, 2021, pp. 14–24.
- [12] G. Gao, G. Xu, Y. Yu, J. Xie, J. Yang, and D. Yue, “Mscfnet: a lightweight network with multi-scale context fusion for real-time semantic segmentation,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 25489–25499, 2022.
- [13] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7132–7141.
- [14] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, “Eca-net: Efficient channel attention for deep convolutional neural networks,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11534–11542.
- [15] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, “Dual attention network for scene segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 3146–3154.
- [16] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.
- [17] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, “Ccnet: Criss-cross attention for semantic segmentation,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 603–612.
- [18] W. Jiang, Z. Xie, Y. Li, C. Liu, and H. Lu, “Lrnnet: A light-weighted network with efficient reduced non-local operation for real-time semantic segmentation,” in *Proceedings of the IEEE International Conference on Multimedia & Expo workshops (ICMEW)*. IEEE, 2020, pp. 1–6.
- [19] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia, “Icnet for real-time semantic segmentation on high-resolution images,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 405–420.
- [20] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, “Bisenet: Bilateral segmentation network for real-time semantic segmentation,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 325–341.
- [21] C. Yu, C. Gao, J. Wang, G. Yu, C. Shen, and N. Sang, “Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation,” *International Journal of Computer Vision*, vol. 129, no. 11, pp. 3051–3068, 2021.
- [22] H. Li, P. Xiong, H. Fan, and J. Sun, “Dfanet: Deep feature aggregation for real-time semantic segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 9522–9531.
- [23] S. Mehta, M. Rastegari, A. Caspi, L. Shapiro, and H. Hajishirzi, “Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 552–568.
- [24] S. Mehta, M. Rastegari, L. Shapiro, and H. Hajishirzi, “Espnetv2: A light-weight, power efficient, and general purpose convolutional neural network,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 9190–9200.
- [25] B. Zhang, Z. Tian, C. Shen *et al.*, “Dynamic neural representational decoders for high-resolution semantic segmentation,” in *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, vol. 34, 2021, pp. 17388–17399.
- [26] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, “Deformable detr: Deformable transformers for end-to-end object detection,” *arXiv preprint arXiv:2010.04159*, 2020.
- [27] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. Torr *et al.*, “Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 6881–6890.
- [28] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, “Segformer: Simple and efficient design for semantic segmentation with transformers,” in *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, vol. 34, 2021, pp. 12077–12090.
- [29] H. Cao, Y. Wang, J. Chen, D. Jiang, X. Zhang, Q. Tian, and M. Wang, “Swin-unet: Unet-like pure transformer for medical image segmentation,” *arXiv preprint arXiv:2105.05537*, 2021.
- [30] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Springer, 2015, pp. 234–241.
- [31] Z. Lu, J. Li, H. Liu, C. Huang, L. Zhang, and T. Zeng, “Transformer for single image super-resolution,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*, 2022, pp. 457–466.
- [32] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 801–818.
- [33] M. Yang, K. Yu, C. Zhang, Z. Li, and K. Yang, “Denseaspp for semantic segmentation in street scenes,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 3684–3692.
- [34] H. Yan, C. Zhang, and M. Wu, “Lawin transformer: Improving semantic segmentation transformer with multi-scale representations via large window attention,” *arXiv preprint arXiv:2201.01615*, 2022.
- [35] V. Badrinarayanan, A. Kendall, and R. Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for image segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.
- [36] M. Treml, J. Arjona-Medina, T. Unterthiner, R. Durgesh, F. Friedmann, P. Schubert, A. Mayr, M. Heusel, M. Hofmarcher, M. Widrich *et al.*, “Speeding up semantic segmentation for autonomous driving,” in *Proceedings of the Conference on Neural Information Processing Systems*, 2016, pp. 1–7.
- [37] M. Fan, S. Lai, J. Huang, X. Wei, Z. Chai, J. Luo, and X. Wei, “Rethinking bisenet for real-time semantic segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 9716–9725.
- [38] Y. Wu, J. Jiang, Z. Huang, and Y. Tian, “Fpanet: Feature pyramid aggregation network for real-time semantic segmentation,” *Applied Intelligence*, vol. 52, no. 3, pp. 3319–3336, 2022.
- [39] G. Li, L. Li, and J. Zhang, “Hierarchical semantic broadcasting network for real-time semantic segmentation,” *IEEE Signal Processing Letters*, vol. 29, pp. 309–313, 2021.
- [40] G. Dong, Y. Yan, C. Shen, and H. Wang, “Real-time high-performance semantic image segmentation of urban street scenes,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3258–3274, 2020.
- [41] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, “Enet: A deep neural network architecture for real-time semantic segmentation,” *arXiv preprint arXiv:1606.02147*, 2016.

- [42] T. Wu, S. Tang, R. Zhang, J. Cao, and Y. Zhang, "Cgnet: A light-weight context guided network for semantic segmentation," *IEEE Transactions on Image Processing*, vol. 30, pp. 1169–1179, 2020.
- [43] Z. Yang, H. Yu, Q. Fu, W. Sun, W. Jia, M. Sun, and Z.-H. Mao, "Ndnet: Narrow while deep network for real-time semantic segmentation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 9, pp. 5508–5519, 2021.
- [44] J. Wang, H. Xiong, H. Wang, and X. Nian, "Adscnet: asymmetric depthwise separable convolution for semantic segmentation in real-time," *Applied Intelligence*, vol. 50, no. 4, pp. 1045–1056, 2020.
- [45] E. Romera, J. M. Alvarez, L. M. Bergasa, and R. Arroyo, "Erfnet: Efficient residual factorized convnet for real-time semantic segmentation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 263–272, 2017.
- [46] A. Lou and M. Loew, "Cfpnet: channel-wise feature pyramid for real-time semantic segmentation," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 2021, pp. 1894–1898.
- [47] M. Liu and H. Yin, "Feature pyramid encoding network for real-time semantic segmentation," *arXiv preprint arXiv:1909.08599*, 2019.
- [48] Y. Wang, Q. Zhou, J. Liu, J. Xiong, G. Gao, X. Wu, and L. J. Latecki, "Lednet: A lightweight encoder-decoder network for real-time semantic segmentation," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 1860–1864.
- [49] S. Hao, Y. Zhou, Y. Guo, R. Hong, J. Cheng, and M. Wang, "Real-time semantic segmentation via spatial-detail guided context propagation," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.
- [50] G. Gao, G. Xu, J. Li, Y. Yu, H. Lu, and J. Yang, "Fbsnet: A fast bilateral symmetrical network for real-time semantic segmentation," *IEEE Transactions on Multimedia*, 2022.
- [51] H.-Y. Han, Y.-C. Chen, P.-Y. Hsiao, and L.-C. Fu, "Using channel-wise attention for deep cnn based real-time semantic segmentation with class-aware edge information," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 2, pp. 1041–1051, 2020.
- [52] J.-Y. He, S.-H. Liang, X. Wu, B. Zhao, and L. Zhang, "Mgseg: Multiple granularity-based real-time semantic segmentation network," *IEEE Transactions on Image Processing*, vol. 30, pp. 7200–7214, 2021.
- [53] G. Li, I. Yun, J. Kim, and J. Kim, "Dabnet: Depth-wise asymmetric bottleneck for real-time semantic segmentation," *arXiv preprint arXiv:1907.11357*, 2019.
- [54] J. Liu, Q. Zhou, Y. Qiang, B. Kang, X. Wu, and B. Zheng, "Fddwnet: A lightweight convolutional neural network for real-time semantic segmentation," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 2373–2377.



**Guoan xu** received the B.S degrees in Measurement Control Technology and Instrumentation from Changshu Institute of Technology, Jiangsu, China, in 2019. He is currently pursuing the M.S. degree with the College of Automation & College of Artificial Intelligence, Nanjing University of Posts and Telecommunications. His research interests include image semantic segmentation.



**Juncheng Li** received the Ph.D. degree in Computer Science and Technology from East China Normal University, in 2021, and was a Postdoctoral Fellow at the Center for Mathematical Artificial Intelligence (CMAI), The Chinese University of Hong Kong. He is currently an Assistant Professor at the School of Communication & Information Engineering, Shanghai University. His main research interests include image restoration, computer vision, and medical image processing. He has published more than 27 scientific papers in IEEE TIP, IEEE TNNLS, IEEE TMM, ICCV, ECCV, AAAI, and IJCAI.



**Guangwei Gao** (Senior Member, IEEE) received the Ph.D. degree in pattern recognition and intelligence systems from the Nanjing University of Science and Technology, Nanjing, in 2014. He was a Visiting Student of the Department of Computing, The Hong Kong Polytechnic University, in 2011 and 2013, respectively. He was also a Project Researcher with the National Institute of Informatics, Japan, in 2019. He is currently an Associate Professor in Nanjing University of Posts and Telecommunications. His research interests include pattern recognition and computer vision. He has published more than 60 scientific papers in IEEE TIP/TCSVT/TITS/TMM/TIFS, ACM TOIT/TOMM, AAAI, IJCAI, PR, etc. Personal website: <https://guangweigao.github.io>.



**Huimin Lu** (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from the Kyushu Institute of Technology in 2014. From 2013 to 2016, he was a JSPS Research Fellow (DC2, PD, and FPD) with the Kyushu Institute of Technology. He is currently an Assistant Professor with the Kyushu Institute of Technology and an Excellent Young Researcher of MEXT-Japan. His research interests include computer vision, robotics, artificial intelligence, and ocean observing.



**Jian Yang** (Member, IEEE) received the PhD degree from Nanjing University of Science and Technology (NUST), on the subject of pattern recognition and intelligence systems in 2002. In 2003, he was a postdoctoral researcher at the University of Zaragoza. From 2004 to 2006, he was a Postdoctoral Fellow at Biometrics Centre of Hong Kong Polytechnic University. From 2006 to 2007, he was a Postdoctoral Fellow at Department of Computer Science of New Jersey Institute of Technology. Now, he is a Chang-Jiang professor in the School of Computer Science and Engineering of NUST. His research interests include pattern recognition, computer vision and machine learning. Currently, he is/was an Associate Editor of Pattern Recognition Letters, IEEE Trans. Neural Networks and Learning Systems, and Neurocomputing. He is a Fellow of IAPR.



**Dong Yue** (Fellow, IEEE) received the Ph.D. degree in engineering from the South China University of Technology, Guangzhou, China, in 1995. He is currently a Professor and Dean of the Institute of Advanced Technology and College of Automation & AI, Nanjing University of Posts and Telecommunications. His current research interests include analysis and synthesis of networked control systems, multiagent systems, optimal control of power systems, and Internet of Things. Prof. Yue served as the Associate Editor for IEEE Industrial Electronics Magazine, IEEE Transactions on Industrial Informatics, IEEE Transactions on Systems, Man and Cybernetics: Systems, IEEE Transactions on Neural Networks and Learning Systems.