BENet: Boundary-Enhanced Network for Real-time Semantic Segmentation

Xiaochun Lei
Guilin University of Electronic Technology
Zeyu Chen
Guilin University of Electronic Technology
Zhaoxin Yu
Guilin University of Electronic Technology
Zetao Jiang
zetaojiang@guet.edu.cn

Guilin University of Electronic Technology

Research Article

Keywords: semantic segmentation, deep neural networks, real-time inference, boundary-enhanced

Posted Date: December 8th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3707992/v1

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at The Visual Computer on March 27th, 2024.
See the published version at https://doi.org/10.1007/s00371-024-03320-7.

BENet: Boundary-Enhanced Network for Real-time Semantic Segmentation

Xiaochun Lei [1,2], Zeyu Chen [1], Zhaoxin Yu [1], Zetao Jiang [1,2,*]

[1]School of Computer Science and Information Security Guilin University of Electronic Technology, GuiLin Guangxi, 541010, China.
[2]Guangxi Key Laboratory of Image and Graphic Intelligent Processing, Guilin University of Electronic Technology, Guilin Guangxi, 541004, China.
[*]Correspondence: zetaojiang@guet.edu.cn.

Contributing authors: lxc8125@guet.edu.cn; 2100300312@mails.guet.edu.cn; 2100301436@mails.guet.edu.cn; zetaojiang@guet.edu.cn;

Abstract

In the realm of real-time semantic segmentation, deep neural networks have demonstrated promising potential. However, current methods face challenges when it comes to accurately segmenting object boundaries and small objects. This limitation is partly attributed to the prevalence of convolutional neural networks, which often involve multiple sequential down-sampling operations, resulting in the loss of fine-grained details. To overcome this drawback, we introduce BENet, a real-time semantic segmentation network with a focus on enhancing object boundaries. The proposed BENet integrates two key components: the Boundary Extraction Module (BEM) and the Boundary Adaption Layer (BAL). The proposed BEM efficiently extracts boundary information, while the BAL guides the network using this information to preserve intricate details during the feature extraction process. Furthermore, to address the challenges associated with poor segmentation of elongated objects, we introduce the Strip Mixed Aggregation Pyramid Pooling Module (SMAPPM). This module employs strip pooling kernels to effectively expand the contextual representation and receptive field of the network, thereby enhancing overall segmentation performance. Our experiments conducted on a single RTX 3090 GPU show that our method achieves an mIoU of 79.4% at a speed of 45.5 FPS on the Cityscapes test set without ImageNet pre-training.

Keywords: semantic segmentation, deep neural networks, real-time inference, boundary-enhanced

1 Introduction

Semantic segmentation stands as a foundational task in the realm of computer vision, where it ascribes a distinct category to every pixel within an image. As the associated applications continue to evolve, achieving an optimal equilibrium between the speed of inference and the precision of semantic segmentation is progressively gaining significance.

Some existing methods achieve real-time semantic segmentation through an encoder–decoder structure[1][2]. To mitigate information loss during down-sampling, various approaches have been used, for instance, enhancing the encoder or decoder, reusing high-resolution features, and utilizing attention mechanisms. However, for the general encoder–decoder architecture, achieving a balance between accuracy and speed is highly challenging. To address this challenge, a series of multi-branch architectures have been designed, including[3][4][5][6], which once achieved a state-of-the-art trade-off between speed and accuracy. PIDNet[7], which involves a novel three-branch network architecture and incorporates an Auxiliary Derivative Branch (ADB) to extract high-frequency features for boundary region prediction, demonstrated the significance of boundary information in image segmentation, providing valuable insights.

With remarkable feats achieved in the field of deep learning, implicit learning methods (e.g., convolutional neural networks, vision transformers, and attention mechanisms) are commonly employed for boundary information extraction, while explicit learning methods are often overlooked. However, explicit learning methods, such as edge detection operators[8][9][10], offer significant advantages over implicit learning methods in terms of simplicity, efficiency, and clarity of purpose. Recent work has introduced edge detection operators in deep learning methods as an assistive tool, confirming their effectiveness in enhancing detail and network localization capabilities[11][12][13][14][15].

For real-time semantic segmentation tasks, we posit that edge detection operators align well with the imperative for efficient inference. They facilitate network localization of boundary information with minimal computational and latency costs. Thus, we propose a novel module for boundary extraction, called the Boundary Extraction Module (BEM), which contains an edge operator. For the spatial detail features maintained by the high-resolution branch, BEM employs the Sobel operator[8] in conjunction with convolution layers to extract edge features. The Sobel operator serves as a strong prior, prompting the network to filter out information irrelevant to the edges. Recognizing that traditional edge detection operators do not account for semantic information, the BEM simultaneously introduces high-dimensional semantic features extracted from the low-resolution branch, which aims to filter out irrelevant texture information by employing rich semantic information. To separately adapt the boundary information extracted through the BEM to two branches and further extract meaningful boundary

features, we employ the Boundary Adaption Layer (BAL) to process it and adapt two branches separately. Furthermore, we present the Strip Mixed Aggregation Pyramid Pooling Module (SMAPPM) to extend the contextual representations and receptive fields. Compared with existing methods that alter receptive fields[16][17][18][19][20], our method is more straightforward to implement, with lower computational overhead and latency. Using these components, we introduce a real-time semantic segmentation model named BENet, successfully striking a balance between inference speed and accuracy. In summary, our primary contributions include:

1. We propose the BEM and BAL for extracting boundary information and focusing the network on boundaries, resulting in improved segmentation performance, particularly for small objects and boundary details.
2. We present the SMAPPM, which extends the network's contextual representations and receptive field, significantly improving the segmentation performance for elongated objects.
3. Our method exhibited an outstanding performance on a single RTX 3090 GPU, with an mIoU of 79.4% at 45.5 FPS on the Cityscapes test set, all achieved without pre-training. Importantly, our approach maintains strong competitiveness under the same training configuration that excludes pre-training.

2 Related Work

In this section, we will discuss representative methods of high-precision semantic segmentation and real-time semantic segmentation.

2.1 High-precision Semantic Segmentation

Semantic segmentation is a crucial task in computer vision. Ever since FCN[21] pioneered the integration of fully convolutional network into the realm of semantic segmentation, a wide range of FCN-based approaches have been introduced and have demonstrated massive potential. In general, these studies primarily focus on improving the extraction and aggregation of semantic information, contextual information, and detailed information about images. Networks such as PSPNet[22] and APCNet[23] employ pyramid pooling modules to aggregate multi-scale contextual information. In contrast, HyperSeg[24], ZigZagNet[25], and Large Kernel Matters[26] are explicitly designed for multi-level feature aggregation. Additionally, certain methods utilize attention mechanisms to capture contextual information, exemplified by non-local[27], DANet[28], and CCNet[29]. To cope with the challenge of detail loss during downsampling, DeepLabv1–3[30–32] introduce dilated convolutions into the network to expand receptive fields without reducing spatial resolution. SegNet[33] incorporates an innovative decoder, which employs max-pooling indices obtained from the corresponding encoder for nonlinear up-sampling of input feature maps, aiding in the recovery of spatial details. HRNet[34] utilizes three branches and repeated multi-scale fusion to maintain high-resolution representations, achieving high accuracy in semantic segmentation tasks. However, the aforementioned methods primarily prioritize model accuracy and overlook the crucial trade-off between inference latency and accuracy.

2.2 Real-time Semantic Segmentation

Semantic segmentation is gaining traction in diverse fields, including autonomous driving and robotics. Consequently, the need for swift response and inference in semantic segmentation poses a challenge. To tackle this challenge, researchers have proposed numerous efficient real-time semantic segmentation methods, striving for an optimal balance between inference speed and model accuracy. These CNN-based works can be broadly categorized into encoder–decoder architectures and multi-branch network architectures.

Encoder-Decoder Architectures: Earlier encoder–decoder approaches, such as ENet[35], adopted an early down-sampling strategy to mitigate computational costs. ERFNet[36] innovatively introduced one-dimensional separable residual blocks, replacing each 3×3 convolution with 3×1 and 1×3 convolutions, resulting in a substantial reduction in the number of parameters. ASFNet[37] proposes an adaptive multiscale segmentation fusion network to fuse multiscale contextual to obtain more precise segmentation results. Recent contributions, such as RegSeg[2], have introduced the D-Block, which utilizes two parallel 3×3 convolution layers with different dilation rates to enhance receptive fields. By maintaining a lower number of channels in the backbone and omitting contextual modules, RegSeg achieved a balanced trade-off between accuracy and inference latency. PP-LiteSeg[1] incorporates a flexible and lightweight decoder to reduce the computational cost of decoders.

Multi-Branch Architectures: Encoder–Decoder architectures often allocate a large number of parameters to recover detailed information, leading to increased inference latency. To address this challenge, multi-branch architectures have been introduced. ContextNet[3] was a pioneering model that incorporated a dual-branch structure, utilizing both full-resolution and low-resolution inputs. The former captures detailed information, while the latter captures global contextual information for efficient semantic analysis. Fast-SCNN[38] utilizes the "learning-to-down-sample" module, allowing two branches to share shallow features, thereby further reducing computational costs. DDRNet[5] utilizes bidirectional feature fusion between two branches, resulting in an impressive performance. LPS-Net[39] is a lightweight network that analyzes the design of convolution blocks, including convolution type and the number of channels as well as the interaction between multiple scales, providing a novel solution for semantic segmentation. PIDNet[7], inspired by PID controllers, has a three-branch architecture in addition to an ADB to the bilateral network to extract boundary information, achieving optimal results.

Overall, the objective of these research endeavors is to meet the demand for rapid and effective semantic segmentation in real-time applications, mindful of the balance between inference speed and model accuracy. Nevertheless, there is room for improvement in the segmentation of boundary details and small objects with these methods, presenting an opportunity for our approach.

3 Method

An overview of our method is presented in Figure 1. First, we illustrate the Boundary-Enhanced Network (BENet), which is composed of a dual-branch network, the BEM

and the BAL. Next, we provide details regarding the BEM and BEL. Finally, we discuss the SMAPPM. Due to the frequent utilization of basic convolutional blocks composed of convolution, batch normalization, and ReLU activation functions in our approach, we refer to it as the "CBR block" for the sake of brevity.

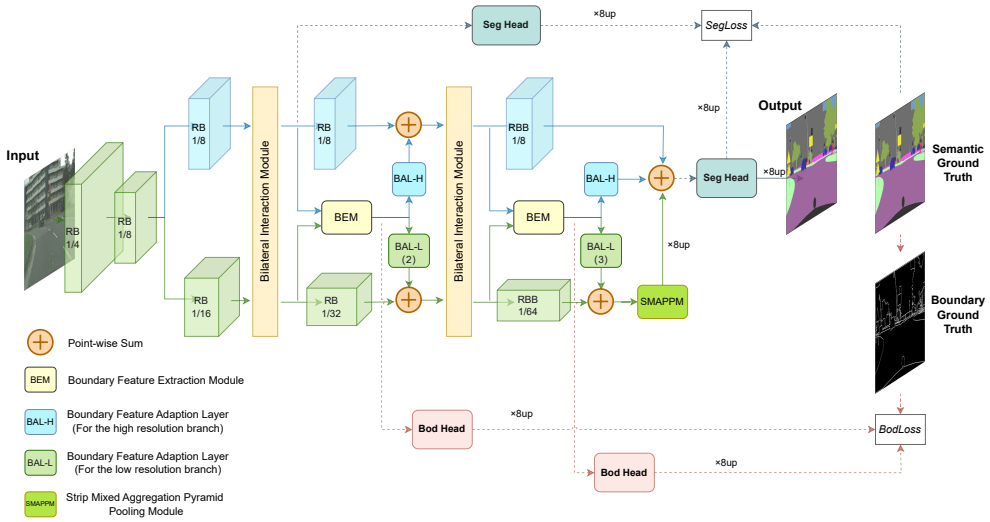

Fig. 1 The architecture overview. "RB" denotes sequential residual basic blocks, and "RBB" denotes a single residual bottleneck block. "SegHead" represents the segmentation head, while "BodHead" represents the boundary head. We utilize coarse boundaries as the ground truth for boundaries. Below "BAL-L", the numerical annotations represent the quantities for stacking CBR blocks, which are 2 and 3, respectively. During inference, all methods related to dashed lines will be discarded to prevent additional latency.

3.1 Boundary-Enhanced Network

To address the issue of detailed information loss during the semantic segmentation process, we devised the BENet. Based on the dual-branch network, we incorporated BEM and BEL to enhance boundary information, guiding the preservation of detailed information.

Following DDRNet, we used residual blocks[40] to form the backbone and employed the same bilateral interaction scheme by transforming low-resolution features and injecting them into high-resolution ones and vice versa. Considering network parallelism, our network extracts boundary features using a BEM and BAL after each bilateral interaction module and injects the boundary feature into each branch after the next residual block.

Utilizing a deep supervision strategy in line with prior studies [7][5][22][41][42], we employed the following approaches. The high-resolution feature after the first bilateral interaction module is fed into a segmentation head for computing semantic auxiliary loss. The boundary heads are placed after each BEM to generate boundary auxiliary loss. Both the boundary and semantic heads consist of a 3×3 convolutional layer

followed by a 1×1 convolutional layer. Each convolutional layer comprises a sequence of BN–ReLU–Convolution.

Three auxiliary losses are used to better optimize the entire network. The total loss can be represented as follows:

$$L_{total} = L_n + \lambda_1 L_{sa} + \lambda_2 L_{ba_1} + \lambda_3 L_{ba_2} \tag{1}$$

Where L_{total} is the total loss, L_n represents the loss obtained by the final output of the network, L_{sa} represents the semantic auxiliary loss, L_{ba_1} and L_{ba_2} are the boundary auxiliary losses obtained by the first and second boundary heads, respectively. Cross entropy loss is applied to L_n and L_{sa}, while L_{ba_1} and L_{ba_2} utilize weighted binary cross entropy loss. We set the parameters as $\lambda_1 = 0.4, \lambda_2 = 10, \lambda_3 = 10$.

3.2 Boundary Extraction Module

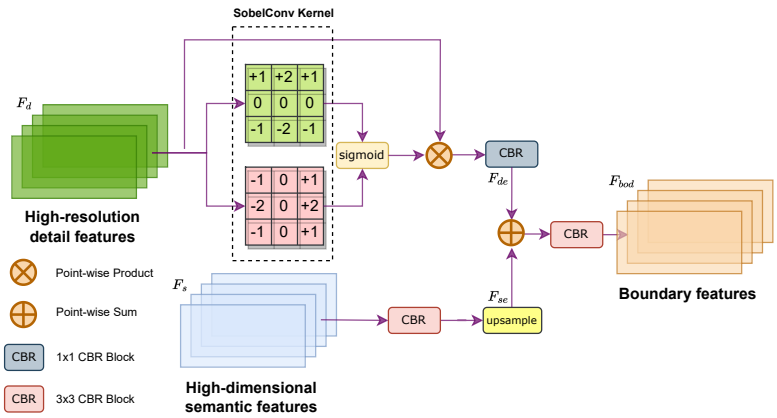

Fig. 2 Illustration of our BEM module. The "1×1 CBR Block" represents a block consisting of 1×1 convolution, batch normalization layer and ReLU activation function in series. The "3×3 CBR Block" follows the same principle. The specific method of the upsample is bilinear interpolation.

This module employs the Sobel operator[8], a traditional edge detection operator based on first-order differentiation. Given its efficiency, we employed it to furnish prior information to the module at minimal cost. Additionally, we intended to utilize it to filter out edge-irrelevant information from the features.

Our network utilizes features extracted from the two branches, F_d and F_s, as the input for the BEM to extract pure boundary features. For the high-resolution detail feature F_d, the Sobel operator filters out edge-irrelevant information. In the specific implementation, two parameter-fixed 3×3 convolutions with a stride of 1 are employed. The specific convolution kernel parameters are detailed in Figure 2. After passing through the upper and lower convolution kernels in the figure, we obtain horizontal and vertical derivative approximations, respectively, and then merge them to derive the gradient amplitude. While using the explicit learning method Sobel

operator for initial edge extraction is efficient, its unlearnable characteristics may lead to suboptimal performance in certain scenarios. Excessive prior information can also elevate the difficulty of network optimization. Therefore, we employ the sigmoid activation function to map the obtained gradient values to the range between 0 and 1. We then multiply them with the original input F_d to acquire features with edge enhancement. A 1×1 CBR block follows to reduce the number of channels, resulting in the edge-enhanced detail feature F_{de}. The specific process is as follows:

$$F_{de} = C_{1\times1}(F_d \odot \sigma(\sqrt{G_x(F_d)^2 + G_y(F_d)^2})) \tag{2}$$

Where $C_{1\times1}$ represents the 1×1 CBR block, G_x and G_y denote the Sobel convolution in the horizontal and vertical directions, $\odot$ denotes point-wise product, and σ denotes the sigmoid activation function.

For high-dimensional semantic features F_s, a straightforward approach involves using a 3×3 CBR block, where the convolution operation with a stride and padding both equal to 1 is applied. This is done to extract features highly correlated with the boundary while preserving its resolution. Subsequently, bilinear interpolation is employed for up-sampling to acquire boundary-related high-dimensional semantic features F_{se}, as illustrated in the following formula:

$$F_{se} = Upsample(C_{3\times3}(F_s)) \tag{3}$$

Finally, we perform a point-wise sum of F_{de} and F_{se}, followed by the application of a 3×3 CBR block to further refine the extraction of boundary information, resulting in the final boundary feature F_{bod}. The resolution of F_{bod} is set to be the same as that of F_d.

$$F_{bod} = C_{3\times3}(F_{de} + F_{se}) \tag{4}$$

Given the challenge of directly instructing the network to acquire boundary features, we implemented a deep supervision strategy by introducing a boundary auxiliary loss. This was done to ensure the effectiveness of the BEM, as detailed in Section 3.1.

3.3 Boundary Adaption Layer

To tailor the boundary features extracted by BEM to two branches with different functionalities, we devised BAL, which comes in two distinct forms: BAL-H and BAL-L, corresponding to high-resolution and low-resolution branches, respectively.

For BAL-H, given that the boundary features extracted by BEM maintain the same resolution as the high-resolution branch features, we opted for efficiency and directly applied a 3×3 CBR block with a stride of 1 and padding of 1 to further adapt to the high-resolution branch.

As for BAL-L, adapting the boundary features to the low-resolution branch involves downsampling. Considering that the low-resolution branch extracts high-level features while the boundary feature tends to be associated more with low-level detail information, using pooling or interpolation algorithms for downsampling may lead to significant feature offset or loss. Therefore, we employed stacked 3×3 CBR blocks with a stride of 2 and padding of 1. At the first BAL-L location, two CBR blocks are

utilized, and at the second location, three are employed. Not only does this achieve fourfold and eightfold down-sampling, respectively, but it also optimizes the boundary feature adaptation for the low-resolution branch.

3.4 Strip Mixed Aggregation Pyramid Pooling Module

In prior studies, Pyramid Pooling Module (PPM) and its variants were employed to build global scene priors, yielding significant results. Deep Aggregation PPM (DAPPM) drew inspiration from Res2Net[43] to fuse semantic information across different scales in a hierarchical-residual manner. Parallel Aggregation PPM (PAPPM) reduced channel numbers to mitigate information redundancy and devised parallelized PPM for enhanced efficiency.

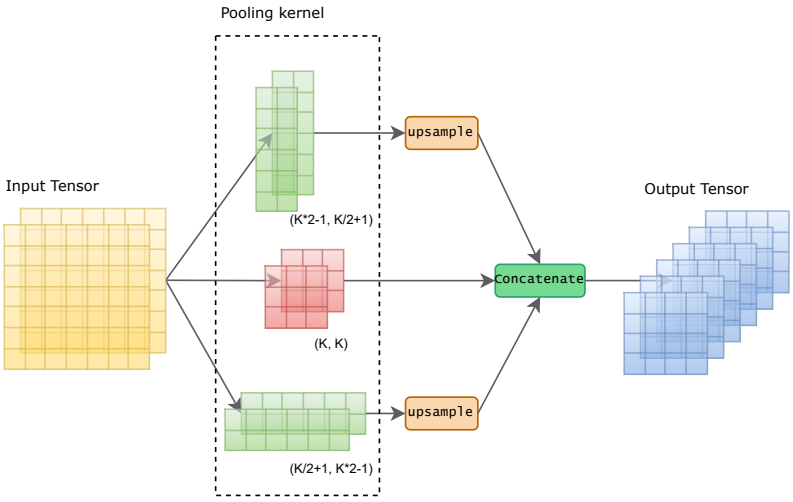

Fig. 3 Illustration of Strip Mixed Pooling Module. Use bilinear interpolation for upsampling and concatenate along the channel dimension.

These recent works all employed square pooling operations to extract multi-scale contextual representations. However, we contend that the absence of strip receptive fields in many scenarios renders the network insensitive to some elongated objects. Therefore, we introduced a Strip Mixed Pooling Module (SMPM), as illustrated in Figure 3.

Specifically, we introduce a parameter K for the SMPM. Initially, we employ conventional pooling with a shape of (K, K) as normal. Simultaneously, strip poolings with kernel shapes of $(K * 2 - 1, K/2 + 1)$ and $(K/2 + 1, K * 2 - 1)$ are utilized to extend the strip receptive field, obtaining a multi-scale pooling map of strip shape. The specific structure of our module, SMAPPM, is depicted in Figure 4.

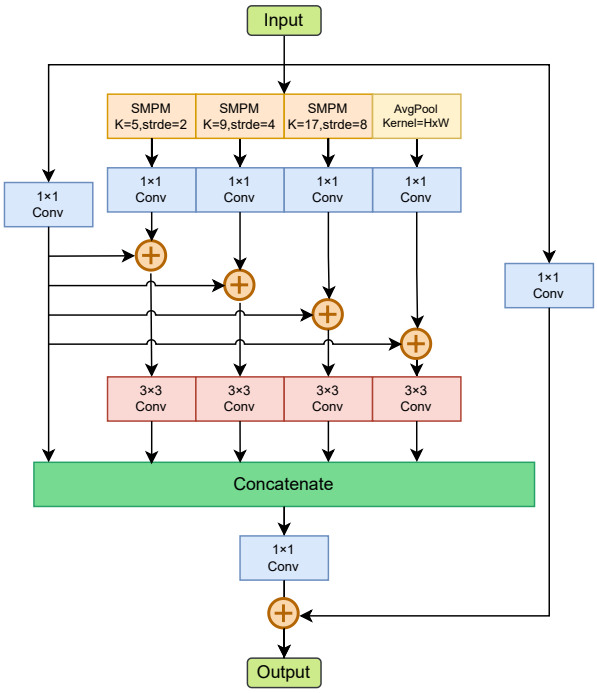

Fig. 4 The overview of Strip Mixed Aggregation PPM.

We adhere to the structure of PAPPM to maintain its parallelism and integrate SMPM into it. We have set the parameter K to 5, 9, and 17 in this study. Similar to DAPPM, multi-scale feature maps with resolutions of 1/128, 1/256, and 1/512 are obtained under an input image resolution of 1/64, encompassing the expansion of strip receptive fields. Experimental results demonstrate the outstanding performance of SMAPPM.

4 Experiments

4.1 Datasets

We utilized the Cityscapes[44] and CamVid[45] datasets to assess the performance of our proposed model.

Cityscapes: The Cityscapes dataset is widely employed in the field of semantic segmentation. It comprises a total of 5000 high-quality, pixel-level finely annotated urban street scene images. Among these, 2975 images constitute the training set, 500 images make up the validation set, and 1525 images form the test set. These images have a resolution of 1024×2048, and the objects within them are categorized into 19 classes.

CamVid: The CamVid dataset comprises 701 densely annotated street scene images, each with a resolution of 960×720. The dataset is split into 367 images for training, 101 images for validation, and 233 images for testing. During our training

process, the model is trained on the trainval set, focusing on utilizing 11 predefined classes, and its performance is evaluated on the test set.

4.2 Implementation Details

We implemented our method using PyTorch 1.13.1 and Nvidia CUDA 12.2. We conducted training on both Cityscapes and CamVid datasets without ImageNet pre-training. As the absence of ImageNet pre-training may lead to fluctuating training results, we executed our method thrice and averaged the results to ensure reliable outcomes.

Cityscapes: For the Cityscapes dataset, we utilized SGD with a momentum of 0.9, an initial learning rate of 0.05, a batch size of 8, 1000 epochs, and a weight decay of 0.0001. We applied data augmentation techniques, including random cropping, random horizontal flipping, and random scaling in the range of [0.5, 2.0]. All images were randomly cropped to a size of 1024×1024 for training, and the Online Hard Example Mining (OHEM)[46] loss was employed.

CamVid: For the CamVid dataset, we set the initial learning rate to 0.001, and images were randomly cropped to 960×720 during training. Consistent with the methodologies outlined in [2][7], we fine-tuned the Cityscapes pre-trained models for CamVid using a batch size of 12 and epochs of 200. Noteworthily, our Cityscapes pre-trained models did not undergo ImageNet pre-training. The other training settings are similar to those used for Cityscapes.

4.3 Ablation Studies

4.3.1 BEM Ablation Studies

This experiment aims to demonstrate the effectiveness of BEM with boundary auxiliary losses. We visualized the boundary features extracted by BEMs. Additionally, we trained our baseline network DDRNet-23 and the network with only BEMs added on the Cityscapes training set using the same configuration and analyzed the results.

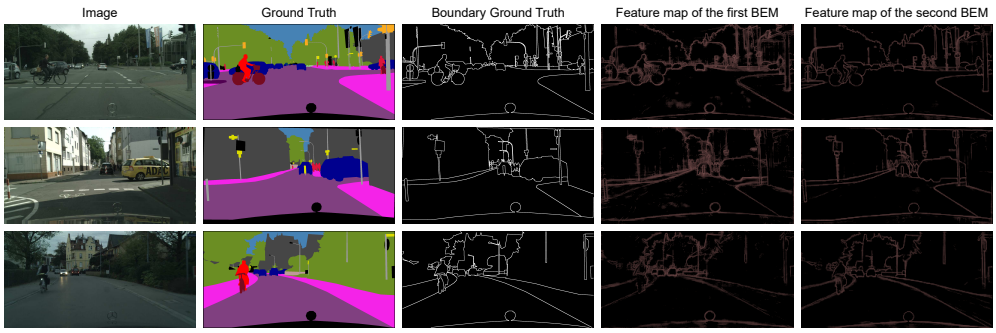

Fig. 5 Feature visualization of BEM. The first column of images is the original input image, the second column is the corresponding ground truth, the third column is the coarse boundary ground truth used in the network, and the fourth and fifth columns are visualizations of the first and second BEM output feature maps.

As depicted in Figure 5, we selected two meaningful feature maps for visualization, one from the output of the first BEM and another from the output of the second BEM in the network. By comparing them with the coarse boundary label, it can be observed that our BEM exhibits excellent performance in extracting boundary information.

Table 1 Comparison of the performance of the DDRNet-23 and the DDRNet-23 with only BEMs added on the Cityscapes validation set.

Model	mIoU (%)	FPS	Params (M)	GFLOPs
DDRNet-23	77.31 ± 0.3	55.2	20.1	148.5
DDRNet-23 with BEMs	$\mathbf{78.34 \pm 0.2}$	51.1	21.3	157.1

The network with the added BEMs achieved an improvement in mIoU of approximately 1.0% (Table 1), with the trade-off being a marginal reduction in speed and an increase in parameters.

4.3.2 BAL Ablation Studies

In section 3.3, we introduced BAL, which is divided into two forms: BAL-H and BAL-L, adapted to two branches, respectively. In this section, we explore the necessity of injecting boundary information into both branches and assess the effectiveness of BALs.

Table 2 Ablation study results of BAL.

Model	BAL-L	BAL-H	mIoU (%)
DDRNet-23 with BEMs			78.34
	✓		78.32
		✓	78.37
	✓	✓	**78.83**

Specifically, we tested the training effects of removing BAL-H or BAL-L separately and simultaneously adding both to the network. The experimental results are presented in Table 2. Adding BAL-H or BAL-L alone cannot achieve good results, although we have already verified the effectiveness of the BEM in the previous section. We believe that this is ascribable to the feature offset between the two branches. Simultaneously applying two adaption layers effectively alleviates the feature mismatch between branches while injecting boundary features into the network, thereby enhancing the overall network performance.

4.3.3 SMAPPM Ablation Studies

Recently, context aggregation modules proposed for real-time semantic segmentation models have demonstrated excellent results. DAPPM[5] has significantly improved

network accuracy, and PAPPM[7] has been introduced to enhance parallelism and speed up inference. Given the utilization of pooling layers in PPM, we designed the SMPM and proposed SMAPPM to further extend the context scale and receptive field with different shapes. To demonstrate its superiority, we conducted experiments on DDRNet-23 and BENet and compared the results with those obtained with other outstanding PPM modules.

Table 3 Comparison of DDRNet-23 and BENet using DAPPM, PAPPM, and SMAPPM.

Model	PPM			mIoU(%)
	DAPPM	PAPPM	SMAPPM	
DDRNet-23	✓			77.31
		✓		77.44
			✓	**78.39**
BENet(Our)	✓			78.83
		✓		78.99
			✓	**79.48**

As evident from the results presented in Table 3, our SMAPPM achieves better performance in both DDRNet-23 and our BENet under the same training configuration.

4.3.4 Auxiliary Losses Ablation Studies

To enhance overall network optimization and reinforce the functionality of the components, we incorporated two boundary auxiliary losses and one semantic auxiliary loss. To verify the effectiveness of these auxiliary losses, we conducted experiments in this section using our entire approach.

According to Table 4, these three auxiliary losses exhibit excellent performance. Specifically, the two boundary auxiliary losses yield a noteworthy enhancement of 1.07% in mIoU, while the semantic auxiliary loss contributes an improvement of 0.2%

Table 4 Ablation study results of Auxiliary Losses.

Auxiliary Loss			mIoU(%)
l_{sa}	l_{ba_1}	l_{ba_2}	
			78.21
✓			78.41
✓	✓		78.79
✓		✓	79.04
✓	✓	✓	**79.48**

mIoU. These results underscore the efficacy of the auxiliary losses in augmenting the performance of the overall network.

4.4 Exploring the Effectiveness of Sobel Operator in BEM

To substantiate the efficacy of the Sobel operator in the context of the BEM, three distinct methods were devised and denoted as BEM-A, BEM-B, and BEM-C for comparative experimentation. BEM-A adheres to the BEM as previously described. In BEM-B, a 3×3 standard convolution with equivalent size and stride is utilized instead of the Sobel operator. In BEM-C, the Sobel operator is omitted, and high-resolution detail features are directly fed into the 1×1 CBR block.

Table 5 The comparison of the three BEM methods on Cityscapes validation set.

Method	BEM-A	BEM-B	BEM-C
mIoU(%)	**79.48**	78.85	79.11

The results are encapsulated in Table 5. Evidently, the BEM-A method exhibits the best performance, which verifies the effectiveness of the Sobel operator in the BEM. For BEM-B, we believe that utilizing training models with standard convolution poses challenges in capturing edge information in this context, primarily due to limited parameters. By integrating the Sobel operator, we contend that it provides prior knowledge to the network for edge extraction, thereby facilitating the acquisition of edge representations and filtering out extraneous information. Consequently, this renders the training process more manageable.

4.5 Comparison on CamVid

As depicted in Table 6, BENet achieves an mIoU of 78.86%. Under identical training settings, it surpasses our baseline method DDRNet-23 by 1.37% in mIoU. Additionally, BENet demonstrates substantial improvement when compared with previously proposed outstanding methods such as BiseNetV2 and MSFNet.

Table 6 Comparison with previous work on CamVid. IM means ImageNet, C means Cityscapes and C(w/o IM) means the Cityscapes pretrained model not undergoing ImageNet pre-training.

Model	Extra Data	GPU	FPS	mIoU(%)
STDC2-Seg[47]	IM	Tesla T4	152.2	73.9
GAS[48]	-	Tesla T4	153.1	72.8
CAS[49]	-	Tesla T4	169	71.2
HyperSeg-S[24]	IM	GTX 1080Ti	38.0	78.4
BiSeNetV2[42]	C	GTX 1080Ti	124.0	76.7
MSFNet[50]	C	GTX 2080Ti	91.0	75.4
PP-LiteSeg-T[1]	C	GTX 2080Ti	**154.8**	75.0
DDRNet-23[5]	C(w/o IM)	RTX 3090	120.4	77.28
BENet	C(w/o IM)	RTX 3090	93.7	**78.65**

4.6 Comparison on Cityscapes

As illustrated in Figure 6, we also compared the prediction results of our model BENet with other real-time semantic segmentation methods on the Cityscapes validation set. Despite our comparative methods DDRNet and PIDNet being the latest state-of-the-art models, BENet consistently achieves superior results in small objects, elongated objects, and some detailed regions.

Table 7 Comparison with previous work on Cityscapes. IM means ImageNet, * means train on our device with the same configuration.

Model	Extra Data	mIoU(%)		FPS	GPU	Resolution	Params	GFLOPs
		Val	Test					
BiSeNet(Res18)[4]	None	74.8	74.7	65.5	GTX 1080Ti	1536 × 768	49M	55.3
BiSeNetV2-L[42]	None	75.8	-	47.3	GTX 1080Ti	1536 × 768	-	-
PP-LiteSeg-B2[1]	IM	78.2	77.5	102.6	GTX 1080Ti	1536 × 768	-	-
STDC2-Seg75[47]	IM	77.0	76.8	73.5	RTX 2080Ti	1536 × 768	22.2M	54.9
RTFormer-Base[51]	IM	79.3	-	39.1	RTX 2080Ti	2048 × 1024	16.8M	-
SFNet(Res18)[52]	IM	-	78.9	30.4	RTX 3090	2048 × 1024	12.87M	247.0
RegSeg[2]	None	78.50	78.3	51.2	RTX 3090	2048 × 1024	39.1M	3.34
DDRNet-23[5]	IM	79.1	**79.4**	55.2	RTX 3090	2048 × 1024	20.1M	142.1
DDRNet-23 *	None	77.3 ± 0.3	77.7	55.2	RTX 3090	2048 × 1024	20.1M	142.1
PIDNet-M *[7]	None	79.4 ± 0.2	**79.4**	42.2	RTX 3090	2048 × 1024	34.4M	197.4
BENet *	None	$\mathbf{79.5 \pm 0.25}$	**79.4**	45.5	RTX 3090	2048 × 1024	29.5M	183.74

In Table 7, we present a comparison of real-time semantic segmentation networks in terms of speed and accuracy on the Cityscapes dataset. Compared with RegSeg, the previous state-of-the-art method without extra data, BENet demonstrates higher accuracy, achieving a 1.1% mIoU improvement on the Cityscapes test set. Using the same training configuration, we trained DDRNet-23 and PIDNet-M, which achieved the best performance ever and had a network scale similar to BENet. Compared with DDRNet-23, BENet sacrifices some speed but achieves a 1.7% mIoU accuracy

improvement. With fewer parameters, a lower calculation amount, and faster speed, BENet achieves comparable accuracy to PIDNet-M.

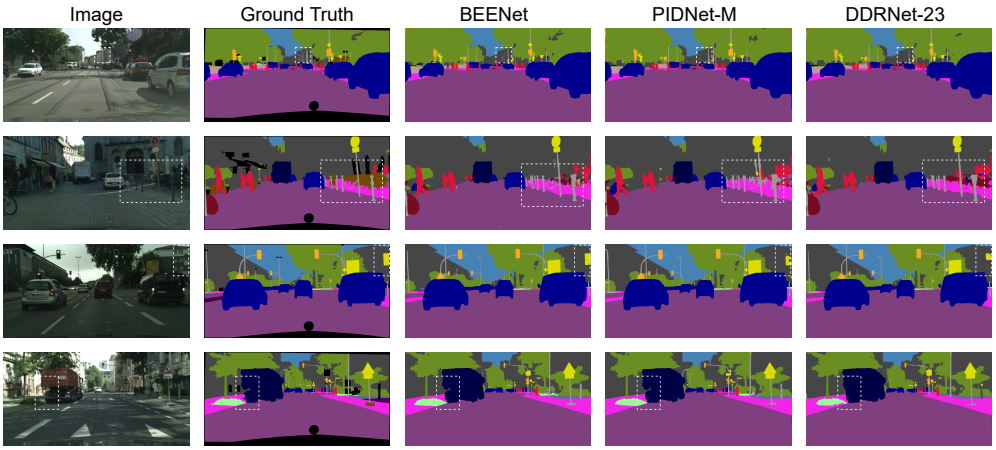

Fig. 6 Comparison of prediction results between BENet, DDRNet-23 and PIDNet-M on the cityscapes validation set. Due to the credit of BEMs, BALs and SMAPPM, BENet demonstrates better segmentation performance on challenging region such as elongated objects and small objects, exhibiting superior segmentation performance.

5 Conclusion

In this paper, we propose a boundary-enhanced network for real-time semantic segmentation, aiming to explicitly extract the boundary features of images to guide the network in maintaining detailed information. We introduce BEM and BAL to efficiently extract boundary information. The high-dimensional semantic features and high-resolution detail features maintained by the dual-branch network are used to further purify the boundary features. The effectiveness of BEM in boundary extraction has been demonstrated through experiments, and the resulting accuracy improvement also validates the promoting role of boundary information in semantic segmentation. Additionally, considering the lack of strip receptive fields in conventional CNN-based networks and the difficulty in segmenting elongated objects, we further propose SMAPPM to expand the network's receptive field and enhance contextual mapping, achieving significant results. In summary, BENet is an efficient real-time semantic segmentation approach that demonstrates strong competitiveness with existing state-of-the-art methods on the Cityscapes dataset.

6 Declarations

6.1 Conflict of interest

The authors declared that they have no conflicts of interest.

6.2 Funding

This work was supported by the National Natural Science Foundation of China (62172118) and Nature Science key Foundation of Guangxi (2021GXNSFDA196002); in part by the Guangxi Key Laboratory of Image and Graphic Intelligent Processing under Grants (GIIP2305) and Student's Platform for Innovation and Entrepreneurship Training Program under Grant (S202310595258, 202310595026).

6.3 Data availability

The data supporting the reported results for the Cityscape dataset can be accessed through the following link: https://www.cityscapes-dataset.com/. This dataset is publicly available for research purposes and can be downloaded upon registration on the website. Similarly, for the CamVid dataset, the data supporting the reported results is available at the following link: http://mi.eng.cam.ac.uk/research/projects/VideoRec/CamVid/. Like the Cityscape dataset, the CamVid dataset is also publicly available for research purposes, and access to it can be obtained by registering on the website.

6.4 Authors' contributions

Conceptualization, X.L. and Z.C.; methodology, X.L. and Z.C.; software, Z.C. and Z.Y.; validation Z.C.; formal analysis, Z.C.; investigation, X.L. and Z.C.; resources, X.L. and Z.J.; data curation, Z.C. and Z.Y.; writing—original draft preparation, Z.C.; writing—review and editing, Z.C., X.L., and Z.J.; visualization, Z.C. and Z.Y.; supervision, Z.J.; project administration, Z.J.; funding acquisition, Z.J. All authors have read and agreed to the published version of the manuscript.

References

[1] Peng, J., Liu, Y., Tang, S., Hao, Y., Chu, L., Chen, G., Wu, Z., Chen, Z., Yu, Z., Du, Y., et al.: Pp-liteseg: A superior real-time semantic segmentation model. arXiv preprint arXiv:2204.02681 (2022)

[2] Gao, R.: Rethinking dilated convolution for real-time semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4674–4683 (2023)

[3] Poudel, R.P., Bonde, U., Liwicki, S., Zach, C.: Contextnet: Exploring context and detail for semantic segmentation in real-time. arXiv preprint arXiv:1805.04554 (2018)

[4] Yu, C., Wang, J., Peng, C., Gao, C., Yu, G., Sang, N.: Bisenet: Bilateral segmentation network for real-time semantic segmentation. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 325–341 (2018)

[5] Hong, Y., Pan, H., Sun, W., Jia, Y.: Deep dual-resolution networks for real-time and accurate semantic segmentation of road scenes. arXiv preprint arXiv:2101.06085 (2021)

[6] Yan, M., Lou, X., Chan, C.A., Wang, Y., Jiang, W.: A semantic and emotion-based dual latent variable generation model for a dialogue system. CAAI Transactions on Intelligence Technology (2023)

[7] Xu, J., Xiong, Z., Bhattacharyya, S.P.: Pidnet: A real-time semantic segmentation network inspired by pid controllers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 19529–19539 (2023)

[8] Kanopoulos, N., Vasanthavada, N., Baker, R.L.: Design of an image edge detection filter using the sobel operator. IEEE Journal of solid-state circuits **23**(2), 358–367 (1988)

[9] Canny, J.: A computational approach to edge detection. IEEE Transactions on pattern analysis and machine intelligence (6), 679–698 (1986)

[10] Kanopoulos, N., Vasanthavada, N., Baker, R.L.: Design of an image edge detection filter using the sobel operator. IEEE Journal of solid-state circuits **23**(2), 358–367 (1988)

[11] Lin, Y., Zhang, D., Fang, X., Chen, Y., Cheng, K.-T., Chen, H.: Rethinking boundary detection in deep learning models for medical image segmentation. In: International Conference on Information Processing in Medical Imaging, pp. 730–742 (2023)

[12] Chen, X., Dong, C., Ji, J., Cao, J., Li, X.: Image manipulation detection by multi-view multi-scale supervision. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 14185–14193 (2021)

[13] Fan, D.-P., Ji, G.-P., Sun, G., Cheng, M.-M., Shen, J., Shao, L.: Camouflaged object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2777–2787 (2020)

[14] Lin, Y., Qu, Z., Chen, H., Gao, Z., Li, Y., Xia, L., Ma, K., Zheng, Y., Cheng, K.-T.: Label propagation for annotation-efficient nuclei segmentation from pathology images. arXiv preprint arXiv:2202.08195 (2022)

[15] Yan, M., Xiong, R., Shen, Y., Jin, C., Wang, Y.: Intelligent generation of peking opera facial masks with deep learning frameworks. Heritage Science **11**(1), 20 (2023)

[16] Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 764–773 (2017)

[17] Zhu, X., Hu, H., Lin, S., Dai, J.: Deformable convnets v2: More deformable, better results. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9308–9316 (2019)

[18] Qi, Y., He, Y., Qi, X., Zhang, Y., Yang, G.: Dynamic snake convolution based on topological geometric constraints for tubular structure segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6070–6079 (2023)

[19] Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122 (2015)

[20] Dou, W., Gao, S., Mao, D., Dai, H., Zhang, C., Zhou, Y.: Tooth instance segmentation based on capturing dependencies and receptive field adjustment in cone beam computed tomography. Comput. Animat. Virtual Worlds **33**(5) (2022) https://doi.org/10.1002/CAV.2100

[21] Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440 (2015)

[22] Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2881–2890 (2017)

[23] He, J., Deng, Z., Zhou, L., Wang, Y., Qiao, Y.: Adaptive pyramid context network for semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7519–7528 (2019)

[24] Nirkin, Y., Wolf, L., Hassner, T.: Hyperseg: Patch-wise hypernetwork for real-time semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4061–4070 (2021)

[25] Lin, D., Shen, D., Shen, S., Ji, Y., Lischinski, D., Cohen-Or, D., Huang, H.: Zigzagnet: Fusing top-down and bottom-up context for object segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7490–7499 (2019)

[26] Peng, C., Zhang, X., Yu, G., Luo, G., Sun, J.: Large kernel matters–improve semantic segmentation by global convolutional network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4353–4361 (2017)

[27] Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7794–7803 (2018)

[28] Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., Lu, H.: Dual attention network for scene segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3146–3154 (2019)

[29] Huang, Z., Wang, X., Huang, L., Huang, C., Wei, Y., Liu, W.: Ccnet: Criss-cross attention for semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 603–612 (2019)

[30] Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic image segmentation with deep convolutional nets and fully connected crfs. arXiv preprint arXiv:1412.7062 (2014)

[31] 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 **40**(4), 834–848 (2017)

[32] Chen, L.-C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587 (2017)

[33] Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE transactions on pattern analysis and machine intelligence **39**(12), 2481–2495 (2017)

[34] Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., *et al.*: Deep high-resolution representation learning for visual recognition. IEEE transactions on pattern analysis and machine intelligence **43**(10), 3349–3364 (2020)

[35] Paszke, A., Chaurasia, A., Kim, S., Culurciello, E.: Enet: A deep neural network architecture for real-time semantic segmentation. arXiv preprint arXiv:1606.02147 (2016)

[36] Romera, E., Alvarez, J.M., Bergasa, L.M., Arroyo, R.: Erfnet: Efficient residual factorized convnet for real-time semantic segmentation. IEEE Transactions on Intelligent Transportation Systems **19**(1), 263–272 (2017)

[37] Zha, H., Liu, R., Yang, X., Zhou, D., Zhang, Q., Wei, X.: Asfnet: Adaptive multi-scale segmentation fusion network for real-time semantic segmentation. Computer Animation and Virtual Worlds **32**(3-4), 2022 (2021)

[38] Poudel, R.P., Liwicki, S., Cipolla, R.: Fast-scnn: Fast semantic segmentation network. arXiv preprint arXiv:1902.04502 (2019)

[39] Zhang, Y., Yao, T., Qiu, Z., Mei, T.: Lightweight and progressively-scalable networks for semantic segmentation. International Journal of Computer Vision, 1–19

(2023)

[40] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)

[41] 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: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16, pp. 775–793 (2020)

[42] 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 **129**, 3051–3068 (2021)

[43] 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 **43**(2), 652–662 (2019)

[44] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)

[45] Brostow, G.J., Fauqueur, J., Cipolla, R.: Semantic object classes in video: A high-definition ground truth database. Pattern Recognition Letters **30**(2), 88–97 (2009)

[46] Shrivastava, A., Gupta, A., Girshick, R.: Training region-based object detectors with online hard example mining. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 761–769 (2016)

[47] Fan, M., Lai, S., Huang, J., Wei, X., Chai, Z., Luo, J., Wei, X.: Rethinking bisenet for real-time semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9716–9725 (2021)

[48] Lin, P., Sun, P., Cheng, G., Xie, S., Li, X., Shi, J.: Graph-guided architecture search for real-time semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4203–4212 (2020)

[49] Zhang, Y., Qiu, Z., Liu, J., Yao, T., Liu, D., Mei, T.: Customizable architecture search for semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11641–11650 (2019)

[50] Si, H., Zhang, Z., Lv, F., Yu, G., Lu, F.: Real-time semantic segmentation via multiply spatial fusion network. arXiv preprint arXiv:1911.07217 (2019)

[51] Wang, J., Gou, C., Wu, Q., Feng, H., Han, J., Ding, E., Wang, J.: Rtformer: Efficient design for real-time semantic segmentation with transformer. Advances in Neural Information Processing Systems **35**, 7423–7436 (2022)

[52] 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: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16, pp. 775–793 (2020)