Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.060244

ARTICLE

Bilateral Dual-Residual Real-Time Semantic Segmentation Network

Shijie Xiang, Dong Zhou, Dan Tian* and Zihao Wang

Institute of Electronic Science and Technology, University of Electronic Science and Technology of China, Chengdu, 611731, China

*Corresponding Author: Dan Tian. Email: tiandan@uestc.edu.cn

Received: 28 October 2024; Accepted: 30 December 2024; Published: 26 March 2025

ABSTRACT: Real-time semantic segmentation tasks place stringent demands on network inference speed, often requiring a reduction in network depth to decrease computational load. However, shallow networks tend to exhibit degradation in feature extraction completeness and inference accuracy. Therefore, balancing high performance with real-time requirements has become a critical issue in the study of real-time semantic segmentation. To address these challenges, this paper proposes a lightweight bilateral dual-residual network. By introducing a novel residual structure combined with feature extraction and fusion modules, the proposed network significantly enhances representational capacity while reducing computational costs. Specifically, an improved compound residual structure is designed to optimize the efficiency of information propagation and feature extraction. Furthermore, the proposed feature extraction and fusion module enables the network to better capture multi-scale information in images, improving the ability to detect both detailed and global semantic features. Experimental results on the publicly available Cityscapes dataset demonstrate that the proposed lightweight dual-branch network achieves outstanding performance while maintaining low computational complexity. In particular, the network achieved a mean Intersection over Union (mIoU) of 78.4% on the Cityscapes validation set, surpassing many existing semantic segmentation models. Additionally, in terms of inference speed, the network reached 74.5 frames per second when tested on an NVIDIA GeForce RTX 3090 GPU, significantly improving real-time performance.

KEYWORDS: Real-time; residual structure; semantic segmentation; feature fusion

1 Introduction

The semantic segmentation task aims to assign a class label to each pixel in an image. High-performance semantic segmentation requires not only a large receptive field but also high-resolution spatial information. A large receptive field helps to obtain highly extracted semantic information, while high-resolution information helps to distinguish the edge details of objects. Many methods with high computational complexity have been proposed [1–3]. Although such networks can achieve higher segmentation accuracy, their large number of parameters and high computational complexity greatly limit their application in real-time scenarios such as video surveillance and autonomous driving. Real-time tasks require rapid execution of extensive image computations, and complex network architectures often struggle to meet this demand. To improve inference speed, many researchers have designed lightweight networks and decoders [4–6], leading to increasing attention on real-time segmentation algorithms.

In general, the key to improving the speed of semantic segmentation lies in reducing the computational complexity of the model. This can typically be achieved by lowering the resolution of input images, reducing the number of network channels, or employing techniques such as depthwise separable convolutions. Most

real-time semantic segmentation methods utilize these strategies to optimize computational efficiency. For instance, the design of efficient backbone networks like ResNet18 [7] and lightweight encoders such as Lednet [8] has significantly enhanced the inference speed of semantic segmentation. However, these optimizations often result in a decrease in model accuracy. Therefore, a crucial challenge remains in achieving accurate segmentation while reducing channel numbers, image resolution, and computational time.

In this paper, we enhanced the original residual blocks and connections of ResNet by incorporating auxiliary residual links, creating a dual-residual module, which was then applied to a dual-branch semantic segmentation network. To better utilize the semantic and spatial information extracted by the dual branches, we designed a feature interaction module between the branches and a feature fusion module at the end of the two branches. Using these modules, we constructed a complete real-time semantic segmentation network. We validated the performance of our network on three widely recognized benchmark datasets: Cityscapes, COCO-Stuff, and CamVid. Experimental results demonstrate that our model achieves a great balance between performance and inference speed, as shown in Fig. 1. The red points in the figure represent the experimental results of our network, while the red dashed line denotes the boundary between real-time and non-real-time models.

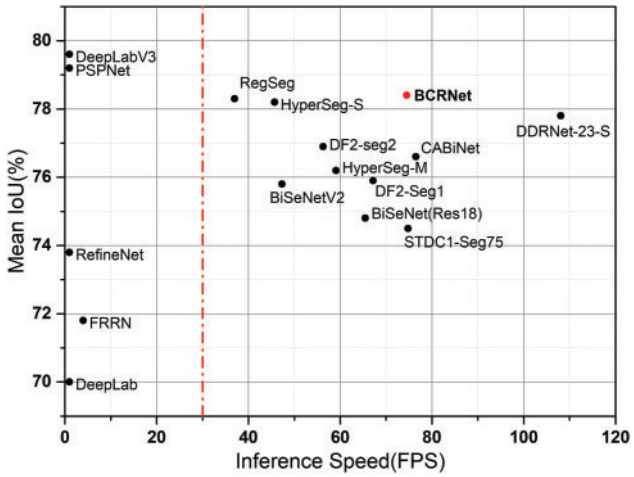

Figure 1: Performance comparison of some real-time semantic segmentation models on the Cityscapes dataset

The main contributions of this paper are as follows:

1 We proposed a novel residual structure, formulated a dual-residual link sampling module, and incorporated it into the network architecture. Empirical validation substantiated the efficacy of the novel residual module in enhancing the performance of the segmentation network, facilitating improved preservation of feature information.

2 We developed feature interaction modules within the network propagation process and feature fusion modules at the ends of each branch specifically tailored for the dual-branch network. These modules facilitate the fusion of features extracted from detail and semantic branches, thereby enhancing the network's capacity to preserve detailed features while acquiring semantic information.

3 Employing the newly introduced residual modules in conjunction with feature fusion modules, we constructed a comprehensive real-time semantic segmentation network, yielding promising experimental outcomes on the dataset, as illustrated in Fig. 1. Specifically, attaining a mean Intersection over Union (mIoU) of 78.41% and a framerate of 74.5 FPS on the Cityscape dataset, we achieved a commendable equilibrium between performance and inference speed.

2 Related Work

In recent years, image semantic segmentation has made considerable progress and has found extensive applications across a range of industries. This section provides an overview of the related literature on semantic segmentation, with a particular emphasis on high-performance segmentation models and methods designed for real-time semantic segmentation.

2.1 High-Performance Semantic Segmentation

From the early stages, semantic segmentation has been tackled using graphical algorithms such as filtering, super-resolution [9], and edge detection [10]. The introduction of networks like VGG [11], and ResNet [7] laid the foundation for the development of neural network-based semantic segmentation methods. UNet [12] avoids the tedious process of manually designing features. Its encoder-decoder structure and skip connections enable the network to simultaneously utilize low-level and high-level features to achieve accurate image segmentation. Fully convolutional networks(FCN) [13] improved segmentation accuracy by merging features through shortcut connections, leading to the widespread application of a new generation of segmentation algorithm networks in semantic segmentation tasks.

The network structure of SegNet [14] is similar to that of UNet. It records the pooling position during downsampling and directly performs deconvolution during depooling, better preserving the boundary feature information. With the introduction of dilated convolution, the DeepLab [15–17] series introduced dilated convolutions with different dilation rates in the network, which effectively solved the problem of gradual loss of position information due to pooling and downsampling of the network, thereby improving the segmentation accuracy of the network. In subsequent research, Segmenter [18] constructed an encoder-decoder architecture based on transformer and set a class mask, which performed better than the traditional linear structure. SegFormer [19] avoids complex decoders and proposes an lightweight multilayer perceptron decoder. By aggregating information from different levels and combining local attention with global attention, it generates powerful feature representations and reaches a new state-of-the-art level in large-scale model training.

2.2 Real-Time Semantic Segmentation

As the application scenarios for semantic segmentation become increasingly diverse, the real-time requirements for various segmentation networks are growing, especially in fields like autonomous driving [20,21] and robotics [22], which demand fast interaction and response times. Real-time inference networks can be categorized into two main types: single encoder-decoder backbone networks and dual-branch segmentation networks.

The encoder-decoder architecture network mainly has a backbone branch for inference. The network performs fast downsampling during encoding to compress features and extract semantic information, while upsampling is employed during decoding to restore features. Lightweight networks can be utilized as encoders to improve segmentation speed and reduce computational complexity. With the introduction of depth-wise convolution and separable convolution, the design of lightweight network architectures has experienced rapid development. MobileNet [23] replaces standard convolution with depth-wise separable convolution, MobileNetV2 [24] mitigates the strong regularization issue of depth-wise separable convolution by employing inverted residual blocks. ShuffleNet [25], based on MobileNet [23] and Xception [26], achieves information fusion through channel shuffling, ensuring that the input to the subsequent group convolution comes from different groups, thus facilitating information flow across different groups. Channel shuffling can significantly reduce computational complexity while maintaining a certain level of accuracy. ShuffleSeg [27] adopts ShuffleNet [25] as its backbone, resulting in faster execution speed and lower computational cost. In

Peng's work [28], they give up the downsampling in the final stage to achieve faster inference speed, which makes the receptive field of the model insufficient to cover large target objects. Although the encoder-decoder structure reduces computational workload, the downsampling process during feature extraction may lead to the loss of some detailed features, which cannot be easily recovered through simple upsampling, thereby adversely affecting the accuracy of semantic segmentation. Hence, the dual-branch architecture is proposed to mitigate this issue.

The dual-branch network architecture typically employs two branches with distinct resolutions to capture a variety of feature information. One branch focuses on extracting semantic details by applying multiple down-sampling operations, while the other maintains high-resolution feature maps to retain rich spatial information. In the BiSeNet [29] framework, a bilateral segmentation network is proposed, which integrates a spatial path to preserve original spatial details and a context path to rapidly acquire a broad receptive field, yielding superior segmentation performance. BiSeNetV2 [30] enhances this approach by incorporating global average pooling to optimize contextual embeddings. Similarly, Fan et al. [31] introduced a Short-Term Dense Connection module, which improves the extraction of deep features by expanding the receptive field and incorporating multi-scale information. The BiAttnNet architecture [32] uses a distinctive bilateral attention mechanism, concentrating all attention modules in the detail branch to facilitate precise semantic selection. Fast-SCNN [33] leverages a learning-based down-sampling module, processing low-level features across multiple resolution branches to share computational load and boost runtime efficiency. DDRNet [34] enhances real-time segmentation by employing bilateral connections that facilitate information exchange between the context and detail branches. More recently, Xu et al. [35] proposed the use of a Proportional-Integral-Derivative (PID) controller in multi-branch networks, achieving improved accuracy, although at the cost of increased inference resource consumption.

In the semantic segmentation task, it is necessary to preserve the spatial information of the image while continuously extracting high-level semantic features. In order to achieve this, many papers need to insert many new modules, which will consume a lot of computing resources while achieving the goal. The use of residual structure in the network can better preserve feature information. Building upon this foundation, we propose a novel residual mechanism and develop a dual-branch model that strikes a balance between performance and inference speed.

3 Method

Our network architecture consists of semantic and detail branches, employing a newly designed dual-residual structure and facilitating information exchange between the dual branches via feature fusion modules. Subsequently, we will provide a detailed exposition of the network structure.

3.1 Overall Architecture for Semantic Segmentation Network

The overall structure of the Bilateral Dual-Residual Linkage Network designed in this paper is shown in Fig. 2. We adopt a dual-branch structure to achieve a balance between performance and inference speed while reducing the number of parameters. To achieve this goal, improvements are made on a completely separated dual-branch structure network. First, the original input image is quickly downsampled to 1/8 of the original image resolution, and then separate detail branches and semantic branches are created to process specific images. The detail branch is the high-resolution branch. After receiving an image with 1/8 of the original resolution, the feature map is no longer downsampled in the detail branch. The feature image always maintains 1/8 of the original image resolution. Convolution and other operations are performed on this feature map to further extract detailed features. After obtaining the feature map with a resolution of 1/8 of the original image, the semantic branch will continue to perform downsampling operations on the feature

map to further extract the semantic feature map. At different resolution stages of the semantic branch, feature fusion operations are performed with the detail branch, that is, when downsampling to 1/16 and 1/32 of the original image resolution, feature fusion is performed between the detail branch and the semantic branch, using the semantic branch information improves the detail branch feature map and superimposes the detail branch information into the semantic branch. Finally, the final feature fusion is performed at the ends of the two network branches to obtain the prediction results.

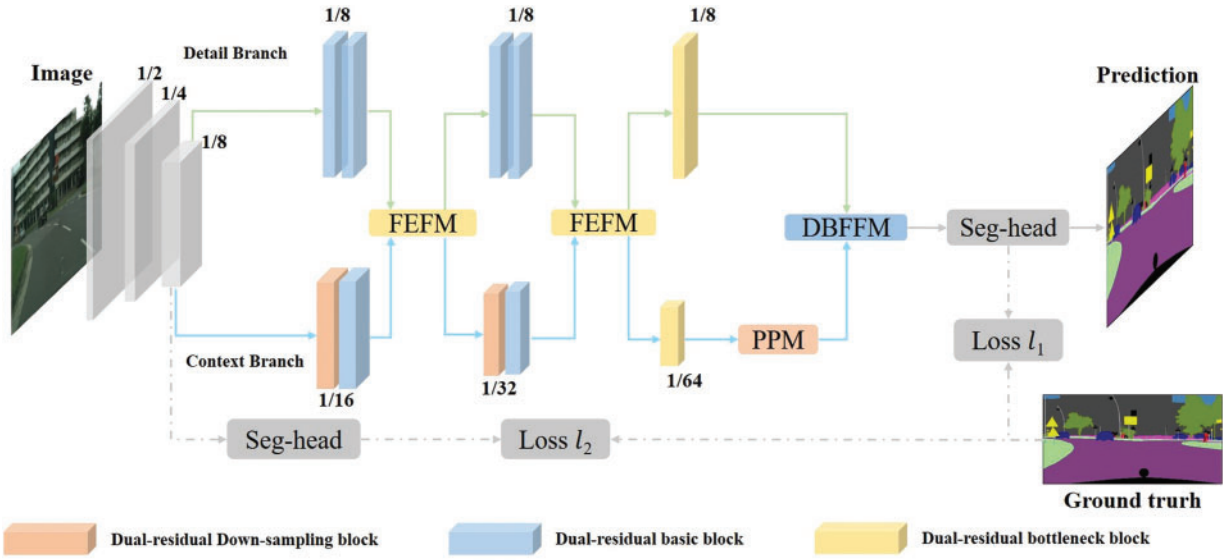

Figure 2: An overview of the architecture of dual-residual linkage network

The loss function in the network uses a binary cross-entropy loss function. The loss function used by the network mainly consists of two parts. The main loss function comes from the final prediction result after the fusion of the two branch features at the end of the network. This result is compared with the Ground truth, and the cross-entropy error is calculated to obtain the main loss function, which is represented by l_0. In addition, there is an additional auxiliary loss function in the network. After the original image is quickly down-sampled to 1/8 of the original resolution, and before entering the semantic branch and detail branch respectively, this feature map is used to compare with the ground truth, and the cross-entropy error is calculated as the auxiliary loss function, represented by l_1. The final loss function is as (1):

$$Loss = l_0 + l_1 \tag{1}$$

3.2 Dual-Residual Modules

The use of residual blocks and residual links in ResNet [7] improves the network's ability to retain original features and makes the network's learning smoother and more stable. During the training process, the problems of gradient disappearance and gradient explosion can be avoided and the network convergence process can be accelerated. However, the network requires a large amount of computing resources for training and inference, especially when the network is deep, this problem is particularly obvious. For example, networks using ResNet101 as backbone require a large amount of computing resources during calculation, and it is difficult to obtain inference results quickly. In real-time semantic segmentation tasks, in order to ensure the speed of network inference, the depth and complexity of the network need to be limited. In order to improve the accuracy of network inference as much as possible within the limited network depth and

preserve as much original feature information as possible during fast downsampling processes, this paper designs dual-residual modules as shown in Fig. 3.

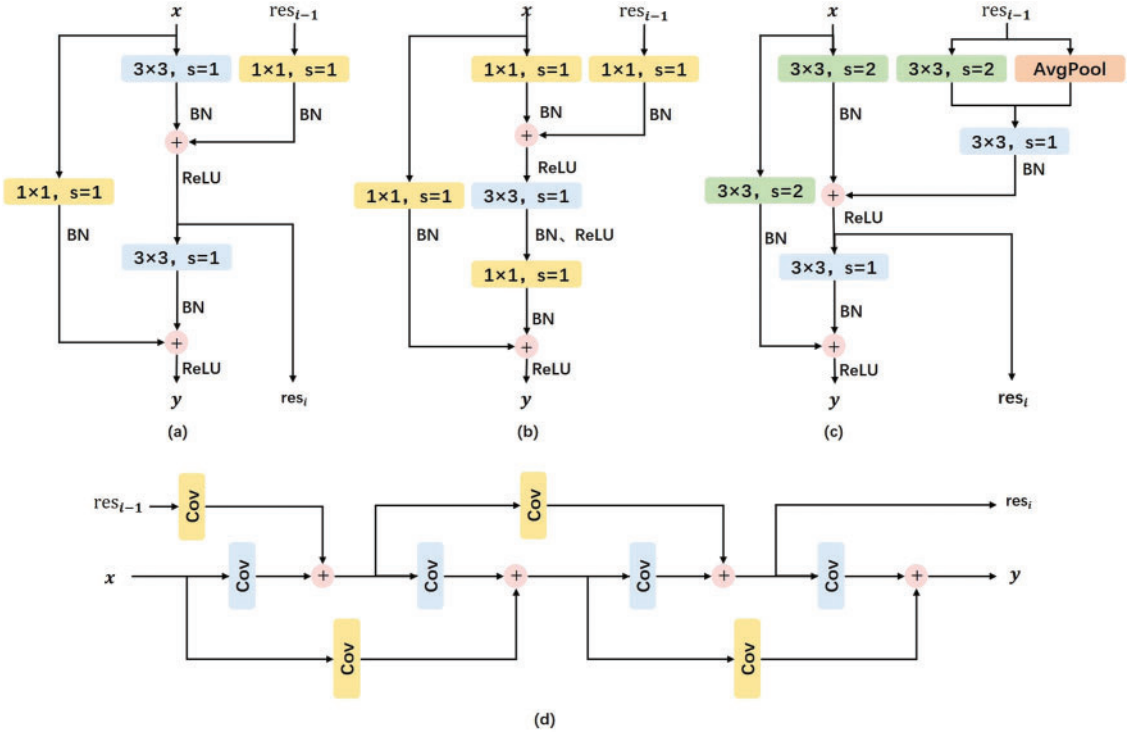

Figure 3: Illustration of three dual-residual structures, called Dual-residual basic block (a), Dual-residual bottleneck block (b), Dual-residual Down-sampling block (c) and Schematic of the network formed by the connection of dual-residual blocks (d)

In Fig. 3, (a) represents the basic dual-residual module, (b) represents the dual-bottleneck residual module, and (c) represents the dual-residual downsampling module. The input and output of each module can be expressed as shown in (2):

$$\begin{cases} y_a = C_1(x) + C_3(C_3(x) + C_1(res_{i-1}))) \\ res_i a = C_3(x) + C_1(res_{i-1}) \\ y_b = C_1(x) + C_1(C_3(C_1(x) + C_1(res_{i-1}))) \\ y_c = C_3(x) + C_3(C_3(x) \\ \qquad + C_3(C_3(res_{i-1}) + Avg(res_{i-1}))) \\ res_c = C_3(x) + C_3(C_3(res_{i-1}) + Avg(res_{i-1})) \end{cases} \tag{2}$$

In the expression, x represents the main input of the module, res_{i-1} represents the auxiliary residual input of each module, y represents the main output of the module, res_i represents the auxiliary residual output of the module, and this output is used as the auxiliary residual input of the subsequent module. C_1 is a 1×1 convolution, C_3 is a 3×3 convolution, and Avg represents the Average-pooling operation. The Batch Normalization (BN) and Rectified Linear Unit (ReLU) operations used in the structure are not written in the expression. The specific positions of BN and ReLU can be viewed in the schematic diagram. The dual-residual structure designed in this paper introduces an auxiliary residual link between the upper and lower

convolutional block in the original residual module. This link captures and transmits information from the previous module and adds it to the output of the first convolutional block in the residual module. When these modules are interconnected to form a network, any two convolutional block can be regarded as having residual connections (as shown in Fig. 3d), rather than being linked by relatively independent modules as in traditional residual module. This network structure effectively preserves important information that may be overlooked during network inference, and when we build a network with fewer layers, it speeds up the extraction process of image feature information. This plays a significant role in achieving a balance between network performance and real-time efficiency.

3.3 Dual Branch Feature Fusion Module

Accurately identifying object edges during the segmentation process is crucial for achieving high segmentation accuracy, as edges often contain important semantic information that distinguishes objects from the background. In our approach, we employed a dual-branch network structure to capture both fine-grained details and global semantic context. Specifically, the high-resolution detail feature map and the low-resolution semantic feature map are obtained at the ends of the detail branch and context branch, respectively. However, directly combining the outputs of these two branches might fail to fully exploit the complementary information between the two feature maps, potentially overlooking the diversity between low-level details and high-level semantics. This can lead to suboptimal performance, especially in terms of accurately capturing object boundaries and fine structures.

To address this limitation, we propose a dual-branch feature fusion module (DBFFM) that integrates features from both branches in a more effective manner. The key insight behind our design is that the detail branch, which preserves more granular spatial information, should serve as the foundation for feature fusion. This allows us to better retain fine details in the final segmentation output. Simultaneously, we incorporate semantic information from the context branch, which captures global and high-level context, into the feature map of the detail branch. This fusion process enables the network to combine the strengths of both low-resolution semantic features and high-resolution detail features, leading to a more accurate segmentation of object edges.

The theoretical foundation of this design is based on the observation that multi-scale feature fusion and weight-based fusion techniques consistently yield promising results in enhancing object boundary recognition and preserving local details [36,37]. By using the detailed information from the high-resolution feature map as the base, and then enriching it with the semantic context from the low-resolution feature map, our approach ensures that both the spatial accuracy and semantic coherence are preserved. This strategy effectively enhances the network's ability to capture fine boundaries while maintaining a comprehensive understanding of the overall scene. The output of the feature fusion process is formally represented as shown in Eq. (3) and depicted in Fig. 4. This approach provides a more robust framework for handling the diverse and complex nature of object boundaries in real-time semantic segmentation tasks.

$$\text{Out} = C_3\left(C_1\left(C_3\left(\vec{v}_d\right) \cdot \text{Sig}\left(C_3\left(\vec{v}_c\right)\right) + \text{Avg}\left(C_3\left(\vec{v}_d\right)\right) \cdot \text{Sig}\left(C_1\left(C_3\left(\vec{v}_c\right)\right)\right)\right)\right) \tag{3}$$

In the expression, $\vec{v}_d$ represents the output feature map of the detail branch, $\vec{v}_c$ represents the output feature map of the context branch, C_3 is a 3×3 convolution, C_1 is a 1×1 convolution, Avg represents the Average-pooling operation, Sig represents the Sigmoid function. To address the issue of mismatched feature map resolutions between detailed and semantic information, we introduced an upsampling process for the semantic feature map, achieved via bilinear interpolation. This module facilitates the fusion of features at different scales, maximizing the utilization of output features from both branches. Compared with simply adding the feature maps, this module has certain advantages in network performance.

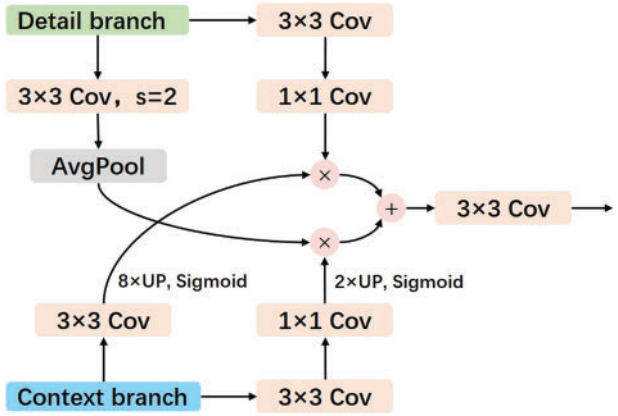

Figure 4: Illustration of dual branch feature fusion module

3.4 Feature Extraction and Fusion Module

In our network design, the context branch captures high-level semantic information, while the detail branch focuses on fine object boundaries. Initially, these branches operated independently, with information exchanged only at the final fusion stage [29]. However, research shows that timely interaction between the branches can refine the feature maps and improve overall performance [34]. To enhance this synergy, we introduce the Feature Extraction and Fusion Module (FEFM), which allows for dynamic information exchange between the branches (Fig. 5).

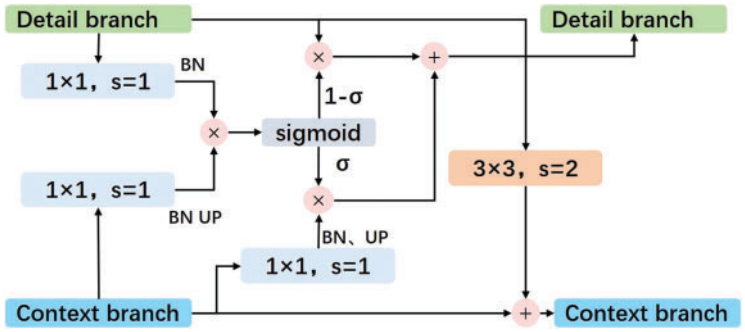

Figure 5: Illustration of feature extraction and fusion module

The FEFM works by integrating the semantic features from the context branch into the detail branch, enriching its feature maps. At the same time, it feeds back the object contour information from the detail branch into the context branch to compensate for any information lost during downsampling. This process helps both branches benefit from each other's strengths, improving the accuracy of edge detection and semantic understanding.

The theoretical basis behind this design is to ensure effective feature refinement by enabling information flow between high-level semantic features and low-level spatial details. This interaction ensures that both fine details and semantic context are preserved, leading to better segmentation results.

Additionally, we enhance the fusion process using an attention mechanism. By element-wise multiplying the feature maps of both branches and applying the sigmoid function, we can approximate the probability that each pixel belongs to a specific object. This attention mechanism refines the feature maps, improving

their accuracy. The corrected detail map is then obtained by adjusting the detail branch output based on the refined context branch features.

In summary, the FEFM improves the interaction between branches, enabling better feature refinement and attention to relevant details, which leads to more accurate semantic segmentation.

3.5 Lightweight Pyramid Pooling Module

In PSPNet [36], the introduction of the Pyramid Pooling Module (PPM) aims to better capture features at different scales. By employing multiple scale pooling layers to process feature maps, PPM addresses the issue of information loss that may occur with traditional fixed-size pooling operations. This module effectively captures features at various scales, thereby enhancing feature representation stability. By integrating features from different scales, the model gains a more accurate understanding of the overall structure and semantics of the input data. The proposal of the Deep Aggregation Pyramid Pooling Module (DAPPM) [34] further enhances the capability of extracting contextual features, leading to superior performance. To maintain real-time performance while utilizing the pyramid module, we have modified PPM by reducing the number of connections and channels, thereby reducing computational burden. Additionally, we parallelized the summation operation of intermediate feature maps to improve computational speed, as illustrated in Fig. 6.

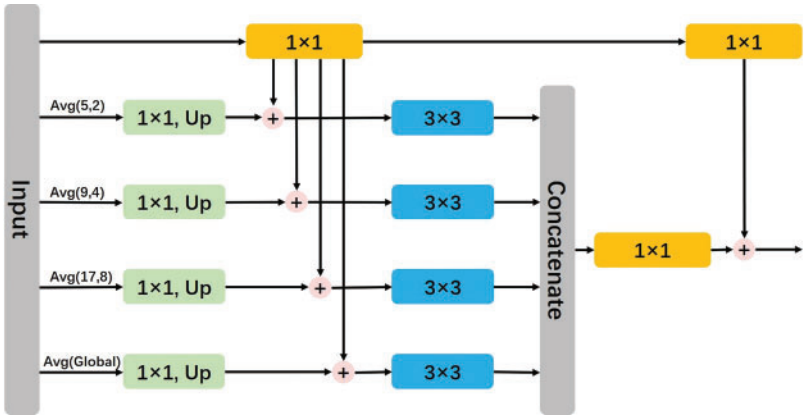

Figure 6: Illustration of lightweight pyramid pooling module

4 Experiment

In this section, we will demonstrate the training details of the model and validate the performance of each module described in the paper. The performance evaluation of individual modules of the model will be conducted on the Cityscapes dataset. Additionally, the model will undergo performance testing on the Cityscapes, COCO-Stuff, and CamVid datasets.

4.1 Datasets

The Cityscapes dataset [38] is a renowned dataset in the field of semantic segmentation, focusing on urban street scenes. It comprises 2975 pixel-level annotated images for training, 500 images for validation, and 1525 images for testing, encompassing a total of 19 semantic classes. The original image resolution is 2048×1024, which allows for better evaluation of network performance due to its higher resolution.

However, employing such high resolutions for real-time semantic segmentation tasks poses a significant challenge in terms of segmentation speed.

The COCO-Stuff dataset [39] offers a collection of complex images comprising 10,000 samples, each densely annotated across 182 categories, encompassing 91 object classes and 91 stuff classes. It is worth noting that there exist 11 classes devoid of any segment annotations. To ensure equitable comparisons, we adhere to the segmentation protocol outlined in [32], which entails a split of 9000 samples for training and 1000 samples for testing purposes.

The CamVid dataset [40] consists of 701 road scene images, each with a resolution of 960 × 720 pixels and densely annotated. It is divided into three subsets: 367 images for training, 101 images for validation, and 233 images for testing. The dataset includes 32 distinct classes, with 11 of these classes chosen for the training and testing phases.

4.2 Parameter Details

During training, we employed a strategy of training from scratch, utilizing the stochastic gradient descent algorithm (SGD) with a momentum of 0.9 to train our model. Due to certain differences between the data sets, the weight decay coefficient of the training process set for the three data sets is also different. For the Cityscapes and CamVid datasets, a weight decay coefficient of 0.0005 was set, while for the COCO-Stuff dataset, it was set to 0.0001. Weight decay regularization was applied only to the parameters of convolutional layers. The initial learning rate was set to 0.005, employing a "poly" learning strategy, where the real-time learning rate was computed iteratively as the initial rate multiplied by $(1 - \frac{Iter}{MaxIter})^n$, with n set to 0.9. The number of iterations for the Cityscapes, CamVid, and COCO-Stuff datasets was set to 300, 200, and 360 K, respectively. During training, data augmentation techniques such as random horizontal flipping, random scaling, and random cropping were applied to images in the datasets. Additionally, the resolutions of images in the datasets were adjusted to the required resolutions: 1024 × 1024 for Cityscapes, 960 × 720 for CamVid, and 640 × 640 for COCO-Stuff. The specific parameter settings can be seen in Table 1.

Table 1: Detailed parameters used for training

Dataset	Cityscapes	COCO-Stuff	CamVid
Batch size	16	8	16
Learning rate	0.005	0.005	0.005
Iteration number	300 K	200 K	360 K
Resolution	1024 × 1024	640 × 640	960 × 720
Optimizer	SGD (stochastic gradient descent), momentum: 0.9		
Weight decay	0.0005	0.0001	0.0001
Platform information	Ubuntu20.04, pytorch 1.11.0, CUDA 11.3		
CPU and GPU	Intel(R) Core(TM) i5-13600KF, Nvidia GeForce RTX 4060Ti		

The accuracy of the network is evaluated using the Mean Intersection over Union (mIoU), which is computed according to the following Eq. (4).

$$mIOU = \frac{1}{k}\sum_{i=0}^{k}\frac{TP}{FN + FP + TP} \tag{4}$$

The training was conducted on a system equipped with an NVIDIA GeForce GTX 4060Ti GPU, utilizing the PyTorch 1.11 framework with CUDA 11.3 and cuDNN 8.0 environments. Network inference rate calculations were performed on a system featuring an NVIDIA GeForce GTX 3090 GPU.

4.3 Ablation Study

This section presents ablation experiments conducted on the Cityscapes dataset, where models are trained and evaluated to assess the efficacy of various modules within the network architecture.

4.3.1 Efficiency of Two-Branch Networks

The impact of dual-branch networks on improving the performance of semantic segmentation networks was evaluated by comparing the training results of individual detail branches, individual context branches, and the training results of using dual-branch networks. During the training process, the process of down-sampling an image to 1/8 of its original resolution was kept unchanged. In single-branch training, only the corresponding network branch was retained, and the Dual Branch Feature Fusion Module and Feature Extraction and Fusion Module were removed. When training the dual-branch network, the feature fusion modules were also removed to eliminate its impact on performance. Additionally, the pyramid pooling module was not used in the three training processes. The results of mean intersection over union for network inference on the Cityscapes dataset are shown in Table 2. When inferring with individually trained detail and context branches, the mIOU values were only 61% and 64%, such training results are not ideal. However, when both context and detail branches were employed simultaneously, the mIoU exceeded 70%, demonstrating a significant improvement compared to single-branch approaches.

Table 2: Performance validation of the dual-branch network on the Cityscapes dataset

Detail	Context	mIOU (%)
✓	–	61.3
–	✓	64.3
✓	✓	71.0

4.3.2 Efficiency of Dual-Residual Linkage, DBFFM and FEFM

The effectiveness of the Dual-Residual Linkage (DRL) structure, Dual Branch Feature Fusion Module (DBFFM), and Feature Extraction and Fusion Module (FEFM) is validated in this section, as presented in Table 3. In the table, "DRL" denotes dual-residual linkage. Contrasting the proposed dual-residual linkage with the original basic residual blocks and bottleneck residual blocks, the utilization of dual-residual linkage enhances the precision of semantic segmentation predictions, resulting in an approximately 1% increase in mean intersection over union.

Table 3: Performance validation of Dual-Residual Linkage, DBFFM and FEFM on the Cityscapes dataset. DRL refers to Dual-Residual Linkage, None means there is no connection between detail branch and context branch, and Add means simply adding elements together

DRL	Modification			Fusion		mIOU (%)
	None	Add	FEFM	Add	DBFFM	
–	✓	–	–	✓	–	68.92
–	–	✓	–	✓	–	76.50
✓	–	✓	–	✓	–	78.35
✓	–	–	✓	✓	–	77.98
✓	✓	–	–	–	✓	74.84
✓	–	✓	–	–	✓	78.34
✓	–	–	✓	–	✓	78.41

The table employs the term "Modification" to denote the manner of feature interaction at different resolution stages during the forward propagation of the dual-branch network. "Fusion" indicates the fusion approach employed for integrating the detail feature maps and semantic feature maps at the end of the dual branches. "None" signifies the absence of information exchange between the two branches, while "Add" denotes direct addition as the method of feature interaction between them. When performing addition operations, if there exist discrepancies in resolution or channel numbers between the feature maps, a convolutional layer is utilized to adjust the channel numbers, bilinear interpolation is applied for image upsampling, and convolution with a stride of 2 is employed for downsampling.

Contrasting scenarios where no information exchange occurs between the detail and context branches, the introduction of Add operations or the utilization of the network modules proposed in this paper both result in improved network performance, indicating that various forms of information interaction between context branch and detail branch can enhance network accuracy. Specifically, incorporating feature interaction between the two branches leads to an approximate 3% increase in mean intersection over union. Moreover, compared to cases where lateral information exchange and feature fusion at the branch ends both utilize Add operations, the simultaneous use of Feature Extraction and Fusion Module (FEFM) and Dual Branch Feature Fusion Module (DBFFM) exhibits certain advantages in enhancing network performance.

4.3.3 Efficiency of Pyramid Pooling Module

The pyramid pooling module (PPM) addresses information loss from traditional pooling by capturing contextual features at multiple scales. However, standard PPMs can add significant computational overhead, which may affect real-time performance. To overcome this, we introduced a lightweight pyramid pooling module (LPPM) that simplifies the structure while retaining the benefits of multi-scale context aggregation. As shown in Table 4, our LPPM results in a noticeable improvement in performance, with the mean intersection over union (mIoU) increasing by approximately 2%, while the computational delay only increased by around 10 ms. This demonstrates that the lightweight pyramid pooling module effectively enhances the network's performance, improving segmentation accuracy without compromising its real-time inference speed.

Table 4: Functional verification of the feature pyramid module

PPM		Params	GFLOPs	FPS	mIoU (%)
None	LPPM				
✓	–	24.17 M	59.04	77.66	76.22
–	✓	28.86 M	59.55	74.77	78.41

4.4 Comparison

Cityscapes. On the Cityscapes dataset, we conducted a comparison of inference accuracy and inference speed across various models, including both real-time semantic segmentation networks and non-real-time semantic segmentation networks. The results of this comparison are presented in Table 5. Additionally, Fig. 7 provides visualized segmentation results of our model on the Cityscapes dataset.

Table 5: The performance comparison of various models on the Cityscapes dataset. "_" indicates that the corresponding result is not reported by the respective method

Model	Resolution	Params	GFLOPs	GPU	FPS	mIoU (%)
PSPNet [36]	1024 × 2048	65.7 M	1065.4	GTX 1080Ti	1	79.2
DeepLabV3 [17]	1024 × 2048	–	–	GTX 1080Ti	1	79.6
DF2-Segl [41]	1536 × 768	–	–	GTX 1080Ti	67.2	75.9
DF2-seg2 [41]	1536 × 768	–	–	GTX 1080Ti	56.3	76.9
BiSeNet(Res18) [29]	1536 × 768	49 M	55.3	GTX 1080Ti	65.5	74.8
BiSeNetV2 [30]	1024 × 512	–	118.5	GTX 1080Ti	47.3	75.8
SFNet [42]	1024 × 2048	12.87 M	247	GTX 1080Ti	18	78.9
MTAENet [28]	1024 × 2048	1.82 M	5.37	GTX 1080Ti	–	71.03
RegSeg [43]	1024 × 2048	3.34 M	39.1	T4	37	78.3
CABiNet [44]	1024 × 2048	2.6 M	12.0	GTX 2080Ti	76.5	76.6
STDC1-Seg75 [31]	1536 × 768	–	–	RTX 3090	74.8	74.5
HyperSeg-M [45]	1024 × 512	10.1 M	8.4	RTX 3090	59.1	76.2
HyperSeg-S [45]	1536 × 768	10.2 M	17.0	RTX 3090	45.7	78.2
DDRNet-23-S [34]	1024 × 2048	5.7 M	36.3	RTX 3090	108.1	77.8
PIDNet [35]	1024 × 2048	36.9 M	275.8	RTX 3090	31.1	80.9
BCRNet (ours)	1024 × 2048	28.86 M	59.55	RTX 3090	74.5	78.4

In comparison with several state-of-the-art models on the Cityscapes dataset, the proposed BCRNet demonstrates an excellent balance between performance and efficiency. Although PIDNet and DeepLabV3 achieve slightly higher mean Intersection over Union (mIoU) scores of 80.9% and 79.6%, respectively, BCRNet still performs competitively with an mIoU of 78.4%. More notably, BCRNet significantly outperforms these models in terms of inference speed, achieving 74.5 FPS with only 28.86 M parameters and 59.55 GFLOPs. In contrast, PIDNet, with 36.9M parameters, operates at 31.1 FPS, while PSPNet, known for its high accuracy, has over twice the number of parameters (65.7 M) and a much higher computational cost (1065.4 GFLOPs), resulting in a mere 1 FPS. The results on the RTX 3090 further validate the efficiency of BCRNet on GPU, showing the optimal balance between computational load and real-time processing capabilities, making it particularly suitable for applications requiring both precision and speed.

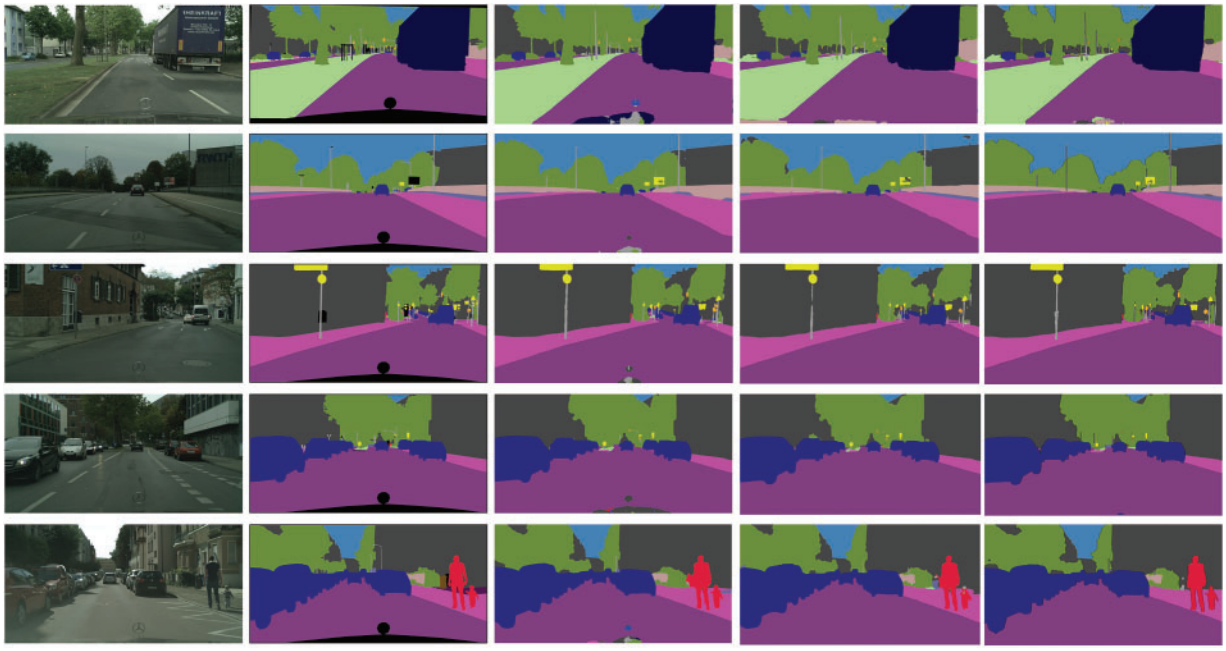

Figure 7: The feature visualization results of various models on the Cityscapes dataset. From left to right, each column displays the following: the original input image, the ground truth label, the segmentation result of BiSeNet, the segmentation result of BiSeNetV2, and the segmentation result of our model

The visualization results in 7 reveal that, in a high-resolution dataset with a rich set of semantic categories, our network achieves higher accuracy in segmenting large objects that occupy substantial portions of the image. Compared to other dual-branch semantic segmentation networks, the proposed model demonstrates superior performance in segmenting object boundaries and small targets, showing a clear performance advantage. Experimental results indicate that the improvements to the residual structure and the incorporation of inter-branch feature fusion in the dual-branch network positively contribute to enhancing the model's performance. performance.

COCO-Stuff. In Table 6, we present the performance of our network validated on the COCO-Stuff dataset. During the inference stage, dataset images are padded to 640 × 640 without employing pre-training or multi-scale training methods, which presents a unique challenge to our experiments. Despite these constraints, our network achieves 29.86% mIoU at 116.9 FPS, as shown in Table 5. This performance is competitive, especially when compared to other models like BiSeNetV2 (87.9 FPS, 25.2% mIoU) and BiSeNetV2-L (42.5 FPS, 28.7% mIoU).

Table 6: Comparison with state-of-the-art on COCO-Stuff

Model	Backbone	FPS	mIoU (%)
FCN [13]	VGG16	5.9	22.7
DeepLab [15]	VGG16	8.1	26.9
PSPNet [36]	ResNet50	6.6	32.6
ICNet [46]	PSPNet50	35.7	29.1
BiSeNetV2 [30]	–	87.9	25.2

(Continued)

Table 6 (continued)

Model	Backbone	FPS	mIoU (%)
BiSeNetV2-L [30]	–	42.5	28.7
BCRNet (ours)	–	116.9	29.86

CamVid. The CamVid dataset, which consists of high-resolution video frames for semantic segmentation with fewer categories (11 categories), is tested under conditions that simulate practical real-time usage. As shown in Table 6, our network achieves 60.38% mIoU at 118.78 FPS (Table 7). This result demonstrates a significant improvement in FPS over models like TD2-PSP50 (12 FPS, 76% mIoU), but with some room for improvement in handling lower-resolution images due to multiple downsampling processes applied in the network.

Table 7: Comparison with state-of-the-art on CamVid

Model	Backbone	FPS	mIoU (%)
TD2-PSP50 [47]	PSPNet50	12	76
DenseDecoder [48]	ResNeXt101	–	70.9
VideoGCRF [49]	ResNet101	–	75.2
MSFNet [50]	–	91.0	75.4
Enet [51]	–	61.2	51.3
SwiftNet [52]	ResNet18	–	72.58
RELAXNet [53]	–	79	71.2
BCRNet(ours)	–	118.78	60.38

4.5 Discussion

Based on the results from our comparison experiments, we observe that our real-time semantic segmentation network performs exceptionally well on high-resolution image datasets, such as Cityscapes and CamVid, achieving a high mean Intersection over Union (mIoU) and maintaining real-time inference speed. This demonstrates the effectiveness of our dual-residual structure and inter-branch feature fusion module in preserving critical features, which contributes to both high accuracy and fast processing. However, when evaluated on datasets with lower original image resolutions, such as COCO-Stuff, although the lightweight network is still able to achieve fast inference speeds, there is a noticeable drop in mIoU performance. We attribute this decline to the loss of significant feature information during the initial downsampling process, where the images are reduced to one-eighth of their original resolution before entering the dual-branch network. This feature loss, particularly evident in low-resolution images, has a more pronounced negative impact on segmentation accuracy compared to higher-resolution images. Consequently, the ability of the network to effectively capture fine-grained details in low-resolution images is compromised, which limits its overall performance on such datasets.

These findings suggest that while our network performs robustly on high-resolution datasets with simpler label categories, it still faces challenges when handling datasets with many categories and lower resolution. To address this issue, we plan to optimize the downsampling process in future work, aiming to reduce the loss of critical feature map information. One potential improvement could involve incorporating advanced techniques such as adaptive downsampling or learnable downsampling filters that dynamically

adjust the resolution reduction based on the complexity of the input image. Additionally, exploring the use of multi-scale feature aggregation could allow the network to better capture information from different levels of detail, which may mitigate the effects of downsampling and help retain important semantic features. By improving these aspects, we anticipate enhancing the network's performance on low-resolution images, thereby improving the overall balance between inference speed and segmentation accuracy, particularly for datasets with a larger number of categories and lower image resolutions.

This approach would not only improve segmentation accuracy on challenging datasets like COCO-Stuff but also enhance the generalizability of our network across a broader range of real-world scenarios, where images with varying resolutions and complexities are commonly encountered.

5 Conclusion

In this paper, we introduce a lightweight, real-time semantic segmentation network based on a dual-branch architecture, which integrates novel dual-residual connections and feature fusion modules. This network strikes an optimal balance between segmentation accuracy and inference speed, making it particularly well-suited for real-time, high-precision semantic segmentation tasks, such as those in road scene analysis. By maintaining superior inference accuracy while ensuring rapid processing times, the network demonstrates its effectiveness through ablation experiments, thereby validating the performance of the proposed dual-residual module. As a foundational component, this module can be widely adopted in the design of other networks to enhance their capacity to preserve critical feature information during downsampling operations.

Acknowledgement: I would like to express my sincere gratitude to Professor Dong Zhou for his invaluable support and guidance throughout my research.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Shijie Xiang, Dong Zhou, Dan Tian; data collection: Shijie Xiang, Zihao Wang; analysis and interpretation of results: Shijie Xiang, Dong Zhou, Dan Tian; draft manuscript preparation: Shijie Xiang, Zihao Wang. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The datasets generated and analyzed during the current study are available in the github repository, https://github.com/moser12138/BDRNet.git (accessed on 28 October 2024).

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Erisen S. Sernet-former: semantic segmentation by efficient residual network with attention-boosting gates and attention-fusion networks. arXiv preprint arXiv:2401.15741. 2024.
2. Wang W, Dai J, Chen Z, Huang Z, Li Z, Zhu X, et al. Internimage: exploring large-scale vision foundation models with deformable convolutions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2023. p. 14408–19.
3. Jain J, Li J, Chiu MT, Hassani A, Orlov N, Shi H. Oneformer: one transformer to rule universal image segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2023. p. 2989–98.
4. Depeng W, Huabin W. MFFLNet: lightweight semantic segmentation network based on multi-scale feature fusion. Multimed Tools Appl. 2024;83(10):30 073–93. doi:10.1007/s11042-023-16782-z.

5. Liu C, Gao H, Chen A. A real-time semantic segmentation algorithm based on improved lightweight network. In: 2020 International Symposium on Autonomous Systems (ISAS); 2020; IEEE. p. 249–53.
6. Chen J, Yu J, Wang Y, He X. Elanet: an efficiently lightweight asymmetrical network for real-time semantic segmentation. J Electron Imaging. 2024;33(1):013 008. doi:10.1117/1.JEI.33.1.013008.
7. 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; 2016. p. 770–8.
8. Wang Y, Zhou Q, Liu J, Xiong J, Gao G, Wu X, et al. Lednet: a lightweight encoder-decoder network for real-time semantic segmentation. In: 2019 IEEE International Conference On Image Processing (ICIP); 2019; IEEE. p. 1860–4.
9. Chen H, Qin Y, Liu X, Wang H, Zhao J. An improved DeepLabv3+ lightweight network for remote-sensing image semantic segmentation. Comp Intell Syst. 2024;10(2):2839–49. doi:10.1007/s40747-023-01304-z.
10. Jin J, Zhou W, Yang R, Ye L, Yu L. Edge detection guide network for semantic segmentation of remote-sensing images. IEEE Geosci Remote Sens Lett. 2023;20:1–5. doi:10.1109/LGRS.2023.3234257.
11. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556. 2014.
12. Ronneberger O, Fischer P, Brox T. U-Net: convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference; 2015 Oct 5–9; Munich, Germany: Springer. p. 234–41.
13. Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2015. p. 3431–40.
14. Badrinarayanan V, Kendall A, Cipolla R. Segnet: a deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans Pattern Anal Mach Intell. 2017;39(12):2481–95. doi:10.1109/TPAMI.2016.2644615.
15. Chen L-C, Papandreou G, Kokkinos I, Murphy K, Yuille AL. Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Trans Pattern Anal Mach Intell. 2017;40(4):834–48. doi:10.1109/TPAMI.2017.2699184.
16. Chen L-C, Papandreou G, Schroff F, Adam H. Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587. 2017.
17. Chen L-C, Zhu Y, Papandreou G, Schroff F, Adam H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European Conference on Computer Vision (ECCV); 2018. p. 801–18.
18. Strudel R, Garcia R, Laptev I, Schmid C. Segmenter: transformer for semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2021. p. 7262–72.
19. Xie E, Wang W, Yu Z, Anandkumar A, Alvarez JM, Luo P. Segformer: simple and efficient design for semantic segmentation with transformers. Adv Neural Inform Process Syst. 2021;34:12 077–90.
20. Wang Z, Ni X, Shang Z. Autonomous driving semantic segmentation with convolution neural networks. Opt Precis Eng. 2019;27(11):2429–38. doi:10.3788/ope.20192711.2429.
21. Lv T, Zhang Y, Luo L, Gao X. Maffnet: real-time multi-level attention feature fusion network with RGB-D semantic segmentation for autonomous driving. Appl Opt. 2022;61(9):2219–29. doi:10.1364/AO.449589.
22. Mahmood T, Cho SW, Park KR. DSRD-Net: dual-stream residual dense network for semantic segmentation of instruments in robot-assisted surgery. Expert Syst Appl. 2022;202:117420. doi:10.1016/j.eswa.2022.117420.
23. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, et al. MobileNets: efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861. 2017.
24. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen L-C. MobileNetV2: inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2018. p. 4510–20.
25. Zhang X, Zhou X, Lin M, Sun J. ShuffleNet: an extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2018. p. 6848–56.
26. Chollet F. Xception: deep learning with depthwise separable convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2017. p. 1251–8.

27. Gamal M, Siam M, Abdel-Razek M. ShuffleSeg: real-time semantic segmentation network. arXiv preprint arXiv:1803.03816. 2018.
28. Jiansheng Peng YH, Yang Q. A lightweight road scene semantic segmentation algorithm. Comput Mater Contin. 2023;77(2):1929–48. doi:10.32604/cmc.2023.043524.
29. 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); 2018. p. 325–41.
30. Yu C, Gao C, Wang J, Yu G, Shen C, Sang N. BiSeNet V2: bilateral network with guided aggregation for real-time semantic segmentation. Int J Comput Vis. 2021;129:3051–68. doi:10.1007/s11263-021-01515-2.
31. Fan M, Lai S, Huang J, Wei X, Chai Z, Luo J, et al. Rethinking bisenet for real-time semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2021. p. 9716–25.
32. Li G, Li L, Zhang J. BiAttnNet: bilateral attention for improving real-time semantic segmentation. IEEE Signal Process Lett. 2021;29:46–50. doi:10.1109/LSP.2021.3124186.
33. Poudel RP, Liwicki S, Cipolla R. Fast-SCNN: fast semantic segmentation network. arXiv preprint arXiv:1902.04502. 2019.
34. 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.
35. Xu J, Xiong Z, Bhattacharyya SP. Pidnet: a real-time semantic segmentation network inspired by pid controllers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2023. p. 19 529–39.
36. 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; 2017. p. 2881–90.
37. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. In: 31st Conference on Neural Information Processing Systems (NIPS 2017); 2017; Long Beach, CA, USA.
38. Cordts M, Omran M, Ramos S, Rehfeld T, Enzweiler M, Benenson R, et al. The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2016. p. 3213–23.
39. Caesar H, Uijlings J, Ferrari V. COCO-Stuff: thing and stuff classes in context. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2018. p. 1209–18.
40. Brostow GJ, Fauqueur J, Cipolla R. Semantic object classes in video: a high-definition ground truth database. Pattern Recognit Lett. 2009;30(2):88–97. doi:10.1016/j.patrec.2008.04.005.
41. Li X, Zhou Y, Pan Z, Feng J. Partial order pruning: for best speed/accuracy trade-off in neural architecture search. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2019. p. 9145–53.
42. Li X, You A, Zhu Z, Zhao H, Yang M, Yang K, et al. Semantic flow for fast and accurate scene parsing. In: Computer Vision-ECCV 2020: 16th European Conference; 2020 Aug 23–28; Glasgow, UK: Springer. p. 775–93.
43. Gao R. Rethinking dilated convolution for real-time semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2023. p. 4675–84.
44. Kumaar S, Lyu Y, Nex F, Yang MY. Cabinet: efficient context aggregation network for low-latency semantic segmentation. In: 2021 IEEE International Conference on Robotics and Automation (ICRA); 2021; IEEE. p. 13 517–24.
45. Nirkin Y, Wolf L, Hassner T. Hyperseg: patch-wise hypernetwork for real-time semantic segmentation. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2020. p. 4060–9.
46. Zhao H, Qi X, Shen X, Shi J, Jia J. Icnet for real-time semantic segmentation on high-resolution images. In: Proceedings of the European Conference on Computer Vision (ECCV); 2018. p. 405–20.
47. Hu P, Caba F, Wang O, Lin Z, Sclaroff S, Perazzi F. Temporally distributed networks for fast video semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020. p. 8818–27.
48. Bilinski P, Prisacariu V. Dense decoder shortcut connections for single-pass semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2018. p. 6596–6605.
49. Chandra S, Couprie C, Kokkinos I. Deep spatio-temporal random fields for efficient video segmentation. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition; 2018. p. 8915–24.

50. Pei M. Msfnet: multi-scale features network for monocular depth estimation. arXiv preprint arXiv:2107.06445. 2021.
51. 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.
52. Wang H, Jiang X, Ren H, Hu Y, Bai S. Swiftnet: real-time video object segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2021. p. 1296–305.
53. Liu J, Xu X, Shi Y, Deng C, Shi M. RELAXNet: residual efficient learning and attention expected fusion network for real-time semantic segmentation. Neurocomputing. 2022;474:115–27. doi:10.1016/j.neucom.2021.12.003.