Academic Journal of Science and Technology
ISSN: 2771-3032 | Vol. 13, No. 2, 2024

Attention-Refined Two-Branch Networks for Real-Time Semantic Segmentation

Shize Xu, Yongsheng Dong

School of Information Engineering, Henan University of Science and Technology, Luoyang 471023, China

Abstract: In real-time demanding scenarios as autonomous driving, real-time semantic segmentation is becoming more and more crucial. BiSeNetV2 has been shown to be an effective model, but its performance in improving speed is limited, especially while maintaining high accuracy. Furthermore, feature map detail loss results from combining high-level semantic and detail information, which is especially crucial for real-time semantic segmentation tasks. In this paper, an efficient Attention Refined Two-Branch Real-Time Semantic Segmentation Network (ARTRNet) is designed to alleviate the above challenges. Specifically, the whole network adopts a two-branch structure: a spatial detail branch and a lightweight dense connectivity context refinement branch, and the lightweight dense connectivity context refinement branch is composed of a novel downsampling module (DSM) and a lightweight dense feature module, which achieves high efficiency in terms of reduced computational cost and model size. In addition, the attention vector of each feature map is computed by residual linking of the Attention Refinement Module (ARM) to highlight the features. A low-resolution context aggregation module (LRCAM) consisting of lightweight Ghost modules is also proposed to enhance the spatial information processing capability of the lightweight densely connected context refinement branch. In the final fusion stage, the Deformed Convolutional Attention Refinement Fusion Module (DCARFM) is proposed, which can enhance the feature expression of the branch and improve the final segmentation results by performing the attention refinement operation on the dual branches separately. Finally, experiments on Cityscape and CamVid datasets show that ARTRNet achieves a good balance between segmentation accuracy and inference speed. On the Cityscapes dataset, we achieved 75.7% mIoU at 132 FPS and 76.9% mIoU at 96 FPS on higher resolution images.

Keywords: Real-time Semantic Segmentation, Dual Attention, Two-branch.

1. Introduction

Semantic segmentation techniques offer significant potential and opportunities for several key application areas. It aims to achieve accurate classification and segmentation of images by dividing and labelling pixels within an image by setting rules. It has been widely used in augmented reality [1], self-driving cars[2] , medical image analysis[3], remote sensing image interpretation[4], and security surveillance[5]. Deep convolutional neural networks have advanced significantly in semantic segmentation tasks in recent years, progressively taking the lead as the industry standard technology. The development of semantic segmentation algorithms[6-8] has been facilitated since Fully Convolutional Networks (FCN)[9] was proposed. These algorithms not only improve the accuracy of segmentation, but also preserve important details of the image. With the continuous development of the mobile terminal industry, many real-time semantic segmentation models [10-14] have emerged to satisfy the needs of the industry. Despite the success of these state-of-the-art models in improving segmentation quality, they also bring higher computational demands, which is a major challenge for real-time application scenarios, such as autonomous driving[2] and mechanically assisted surgery[15], as these scenarios require models that are both highly accurate and have to satisfy the speed requirements of real-time processing.

In order to adapt to the demand for real-time interactive performance in these domains, a large number of researchers have developed semantic segmentation models featuring fewer parameters and rapid reasoning capabilities[16-18]. These models fall into two main categories: one is the single-branch encoder-decoder architecture, whose representative studies [19-20] follow the line of development since FCN. The alternative category encompasses multi-branch architectures [16-17] that are precisely tailored to address the specific requirements of real-time semantic segmentation. The main difference between the two categories is reflected in their approach to multiscale semantic features. Encoder-decoder architectures typically capture the semantic information of an image through layer-by-layer downsampling and feature fusion techniques, and this process is usually done in a single processing path. In contrast, multi-branch architectures offer a distinct viewpoint, advocating that spatial detail information and high-level semantic information can be independently extracted to capture the multi-scale features of an image more effectively. BiSeNetV2 [18], a two-branch network, has become a prime example in the field of real-time semantic segmentation due to its excellent performance. Compared with the traditional single-branch structure, the two-branch structure not only performs better in boundary and small target segmentation, but also achieves a significant improvement in inference speed. In most two-branch networks, features are extracted independently on paths with different resolutions in order to speed up downsampling and reduce the cost of memory access, which usually requires the network to perform complex feature fusion operations at a later stage. In addition, some networks, including BiSeNetV2, still rely on hand-designed lightweight backbones, which limits their performance upper bound to some extent.

The study introduces a novel, dual-branch architecture for real-time processing, designated as the ARTRNet. The aim is to improve segmentation accuracy, structure interpretability, and performance that can compete with existing methods. The specific structure is shown in Figure 1, ARTRNet adopts a

coder-decoder architecture. The entire network uses a dual branching structure: a spatial detail branch and a lightweight densely connected contextual refinement branch, which achieves high efficiency with reduced computational cost and model size. During the final fusion stage, the feature representation of each branch can be enhanced by independently applying the attention refinement operation to the dual branches, and at the same time, it can replace some complex attention computations.

Our main contributions can be outlined as follows:

(1) We propose a lightweight dense connectivity context refinement branch consisting of a novel DownSampling Module (DSM) and a lightweight dense feature module, which achieves high efficiency with reduced computational cost and model size.

(2) We propose an Attention Refinement Module (ARM). to compute the attention vector of each feature map as a way to highlight features. We also propose a Low Resolution Context Aggregation Module (LRCAM) consisting of lightweight Ghost modules to enhance the spatial information processing capability of lightweight densely connected context refinement branches.

(3) We propose a Deformed Convolutional Attention Refinement Fusion Module (DCARFM), which is able to enhance the feature expression of the branch by performing separate attention refinement operations on the dual branches in the final fusion stage.

(4) Based on the above efforts, we construct a Real-Time Two-Branch Segmentation Network architecture called ARTRNet and achieved competitive results on standard benchmark tests.

2. Related Work

2.1. Single-Branch Real-Time Semantic Segmentation

Conventional semantic segmentation approaches primarily rely on established techniques and methods in computer vision and image processing, and usually use techniques of threshold segmentation[21], region segmentation[22], image features[23] or graph models[24] for pixel-level classification to achieve semantic segmentation. In recent years, within the field of real-time semantic segmentation research, some approaches have adopted the single-branch encoder-decoder architecture[9, 25-26] as their core framework. These methods capture the semantic features of an image through layer-by-layer downsampling and feature fusion techniques, while capturing both low-level detail information and high-level semantic information. ESPNet[19] enhances segmentation performance by capturing multi-scale feature information using dilated convolution at various scales. Dilated convolution enables the network to cover a larger receptive field with fewer parameters and less computation, which is very efficient when processing high-resolution images. EDANet[20] employs asymmetric convolution, dilated convolution, and dense concatenation to reduce parameters and computation, maintaining high efficiency and accuracy. This network is ideal for high-speed processing applications, such as autonomous driving, and is able to achieve fast segmentation performance while maintaining high-resolution inputs. DFANet[27] efficiently combines feature information from different layers through deep feature aggregation techniques to improve segmentation accuracy. The lightweight architecture design of the network allows it to handle high-resolution images swiftly, meeting real-time application demands effectively.

2.2. Two-Branch Real-Time Semantic Segmentation

The two-branch architecture effectively preserves the high-resolution details of an image by processing features at different scales independently compared to the single-branch architecture. The BiSeNet series[17-18] is a good example in this regard. BiSeNet[17] introduces parallel spatial and contextual paths as well as feature fusion techniques, which achieves the fast acquisition of deep semantic information while preserving the image details. It significantly improves the performance of real-time semantic segmentation. BiSeNetV2[18] simplifies and deepens the network structure, improves the operation efficiency, introduces a bilateral bootstrap aggregation layer, and combines details and semantic features more effectively. However, since BiSeNetV2 adopts a fast downsampling strategy, some important detail information may be lost to some extent, which may affect the accuracy of segmentation. To solve this problem, Fast-SCNN[13] and DDRNet[28] adopt a single-branch architecture with a shared backbone to increase multiple interactions by sharing parameters early in the network.

2.3. Feature Fusion Module

In semantic segmentation network architecture, the feature fusion module plays a critical role in integrating features from multiple levels to enhance the model's capability to understand both global context and local details of scenes. This integration boosts accuracy and segmentation effectiveness. In addition to the basic element-by-element addition or feature splicing operations, current feature fusion techniques are increasingly implemented using the attention mechanism. The attention mechanism is equivalent to an intelligent information filtering process, which can dynamically focus on the key information in the image according to the task requirements to optimise the feature representation. DANet[29] employs a dual attention mechanism to enhance feature identification by guiding attention across spatial and channel dimensions, thereby boosting segmentation accuracy. CCNet[30] enhances the feature representation through a contrast compression module to enhance feature representation, utilising contrast learning to improve feature discrimination and reducing computation through compression operations to achieve more efficient real-time semantic segmentation performance.

3. Our Proposed Method

In this section, we begin by describing the architecture of the proposed Attention Refined Two-Branch Real-Time Semantic Segmentation Network (ARTRNet). Subsequently, we elaborate on the design specifics of its key components: the Lightweight Densely-Connected Context Refined Branch and the Deformed Convolutional Attention Refined Fusion Module (DCARFM).

3.1. Attention-Refined Two-Branch Real-Time Semantic Segmentation Networks(ARTRNet)

In this section, the ARTRNet network architecture is described in detail. The overall network architecture is shown

in Figure 1. The network architecture comprises two branches: the spatial detail branch and the lightweight densely-connected context refinement branch. These branches are responsible for extracting low-level fine-grained information and deep semantic information, respectively. The spatial detail branch, characterized by wide channels and shallow layers, sufficiently captures spatial information. Inspired by the detail branch of BiSeNetV2 [18], our own high-resolution branch is designed. It uses three convolutional layers consisting of 3×3 convolutions for channel expansion, each consisting of an integrated module of 3×3 convolutions, batch normalisation and activation functions, and maximum pooling after each layer to quickly downsample the input image to a scale of 1/8. The spatial detail branch is only the processing of local image details, which cannot take up too much computational resources, so only the resolution reduction operation is performed in this process, while preserving detailed object edge information, focusing the main feature extraction capability on the lightweight densely connected contextual refinement branch.

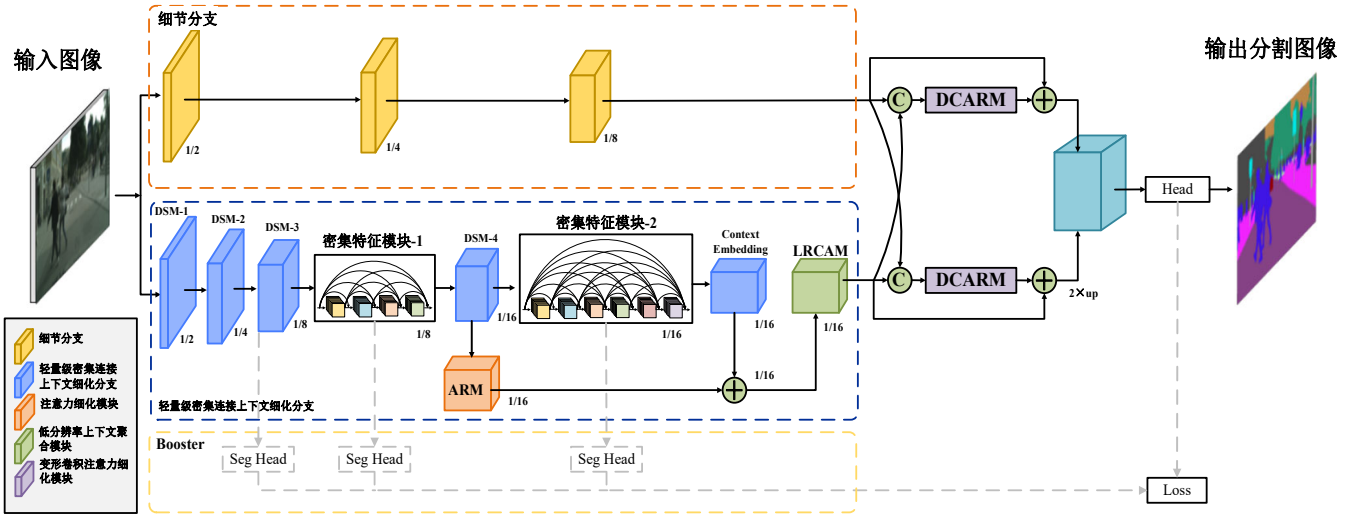

Figure 1. The structural diagram of Attention-Refined Two-Branch Real-Time Semantic Segmentation Networks (ARTRNet).

The basic construction of the lightweight dense connection context refinement branch mainly consists of a lightweight dense feature module consisting of a DownSampling Module (DSM) and a lightweight Ghost module[31], and the specific structure of the DSM module and the Ghost module is shown in Figure 2. Specifically, on the lightweight dense connectivity context refinement branch firstly goes through three stages, DSM-1, DSM-2 and DSM-3, which rapidly downsample the feature map to 1/8 of the original map while widening the number of channels, and after that, it will enter into the dense feature module-1 consisting of 5 Ghost modules, and after that, after going through one DSM-4 module for downsampling, it enters into the Dense Feature Module-2 consisting of 7 Ghost modules, and finally the sense field is expanded by a context embedding block to capture the high-level semantics, and the feature map is downsampled to 1/32. The features of Dense Feature Module-1 are selected to be output to the Attention Refinement Module (ARM) after downsampling the feature map to 1/16. The ARM module calculates feature map weights using pooling and 1×1 convolution, which are then multiplied with the input feature maps to compute channel attention. The specific structure is shown in Figure 2. The 1/32 feature maps from the context embedding block are up-sampled and operated and then summed with the feature maps passing through the ARM and passed into the Low Resolution Context Aggregation Module (LRCAM) consisting of the lightweight Ghost module, and outputs the final result of the semantic segmentation. In the final fusion stage, Deformed Convolutional Attention Refinement Fusion Module (DCARFM) is proposed, which can enhance the feature expression of the branch by cross-fertilising the dual branches with the attention refinement operation respectively, and at the same time, it can replace some complex attention computations to reduce the contextual differences between the high-level semantic information and the underlying spatial detail information.

3.2. Lightweight Dense Connection Context Refinement Branch

The spatial detail branch combines convolutional and nonlinear mapping layers to capture detailed information from local regions. Typical semantic branching focuses on delivering deep semantic information to distinguish between various object types. This process involves a more intricate branch and is also more time-intensive. To accelerate segmentation, a lightweight densely connected contextual refinement branch is employed to lower the computational cost of semantic branching. This chapter adopts a similar connectivity strategy as EDANet[20], incorporating a new DSM module and a dense connection block using the lightweight Ghost module within the context branch. Firstly three DSM modules perform continuous downsampling to 1/8 of the original image, three paths will be divided in each DSM module, two paths first go through a 3×3 convolution for 2-fold downsampling, and the other path undergoes maximum pooling for downsampling, and after that the three paths are channel-level summed up and then go through the BN and RELU activation functions, and maximum pooling is used instead of part of the convolution, which reduces the cost of computation. The specific structure is shown in Figure 2. After that, it will pass through a dense link block-1 consisting of five lightweight Ghost modules, inspired by DenseNet[32], which replaces the convolutional blocks in the dense link with lightweight Ghost modules with smaller parameter counts and faster computation speed. Each Ghost module first undergoes a 1×1 convolution, BN and RULU activation functions, and after that, after a 3×3 group convolution, BN and RULU

activation functions, it outputs the result after channel-level summation with the residual link. The specific structure is shown in Figure 2.

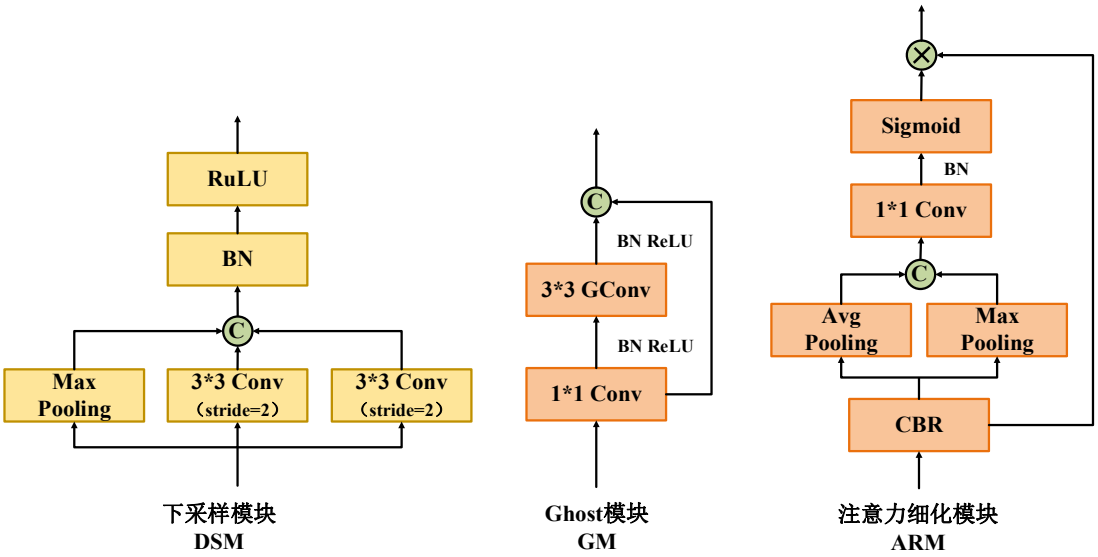

Figure 2. Architectural details of the DSM, GM and ARM.

After that, it passes through a DSM module to downsample the feature map to 1/16, and then passes through a dense connection block-2 composed of seven lightweight Ghost modules and then passes into the context embedding block to get 1/32 feature map, and the feature map from the dense connection block-1 passes through the ARM module for attention refinement operation, and the ARM module first passes through a convolution block composed of convolution, BN and RULU activation function, and then passes through a convolution block composed of 3×3 group convolution, BN and RULU activation function, and then passes through the residual link to output the result after channel-level summation. The ARM module first passes through a convolution block consisting of convolution, BN and RULU activation functions, after which it performs average pooling and maximum pooling operations for pixel-level summation, and then passes through a 1×1 convolution, BN and Sigmiod activation functions, and then performs pixel-level multiplication with residual connections after the convolution block to output the result.Following the ARM module immediately after upsampling to 1/16 of the original image, and after the context embedding block after the up-sampling of the feature map for pixel-level summation and input to the low-resolution context aggregation module, the addition of the low-resolution context aggregation module at the end of the feature extractor is mainly to enhance the spatial information processing ability of lightweight dense connection context refinement branch anywhere. In this way, the network is able to process images with complex spatial structures more efficiently, while allowing better articulation with spatial detail branches. Mimicking pyramid pooling without adding parameters, the convolution is substituted with a lightweight Ghost module, and the whole module is divided into five layers, with only one Ghost module in the first layer to get the feature map F1, two consecutive Ghost modules in the second, third, and fourth layers to get the feature maps F2, F3, and F4, and the fifth layer to first perform an average pooling operation and then enter a Ghost module to obtain the feature map F5. After that, F5 and F4 are pixel-level summed to obtain F44, and so on, to obtain the feature maps F33, F22, and F11, and finally the results are output by channel-level summing of F11, F22, F33, F44, and F5, and residual connections from the input image. The detailed structure is illustrated in Figure 3.

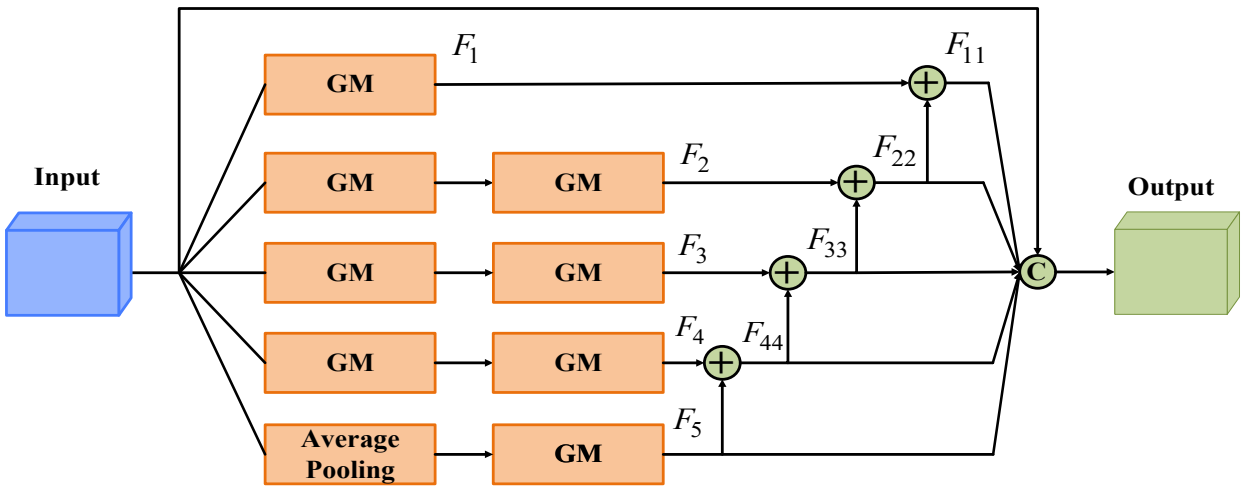

Figure 3. Architectural details of the LRCAM.

3.3. Deformed Convolutional Attention Refinement Fusion Module(DCARFM)

Low-resolution features contain rich semantic information, while high-resolution features better preserve spatial details. To effectively integrate the deep semantic information from the lightweight densely connected contextual refinement branch with the spatial detail information from the spatial detail branch, we draw inspiration from D-LKA Attention[33] to propose a Deformed Convolutional Attention Refinement Fusion Module. This module primarily consists of a Deformed Convolutional Attention Refinement Module. The detailed structure is depicted in Figure \ref{fig4_}. Firstly, in the fusion stage, a cross-fertilisation approach is adopted, specifically, the output results of the lightweight dense connectivity context refinement branch are first up-sampled and the results of the spatial detail branch are channel-level summed and then passed into the Deformed Convolutional Attention Refinement Module (DCARM) to carry out the process. DCARM is used for refinement processing, primarily filtering the features from the spatial detail branch after summation to emphasize its feature representation. After that, the spatial detail branch is downsampled and channel-level summed with the lightweight dense connection context refinement branch, and then passed into the deformed convolutional attention refinement module, where the refinement process is mainly to filter the features of the lightweight dense connection context refinement branch after summing, and highlight the feature representation of the lightweight dense connection context refinement branch. The results from these two refinements are fed into the feature fusion module, which then produces the final output.

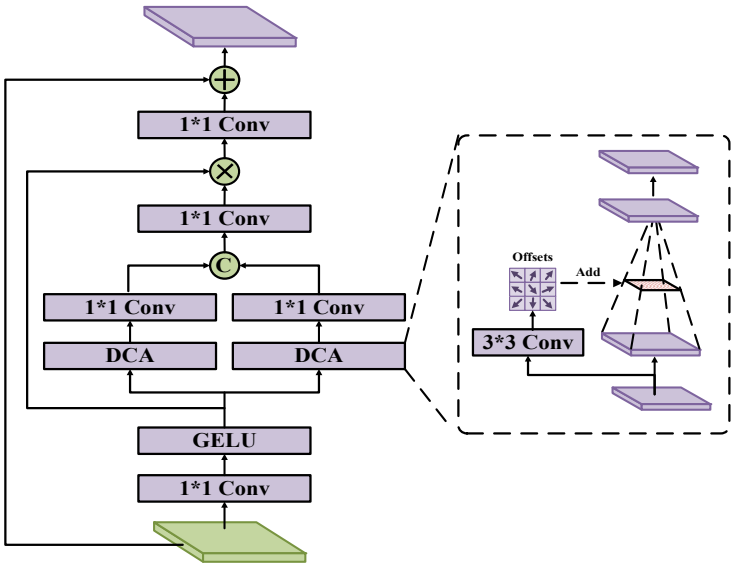

Figure 4. Architectural details of the DCARFM.

4. Experiments

In this section, we will evaluate the proposed ARTRNet on two datasets: Cityscapes[34] and CamVid[35]. We will compare its performance with other notable real-time semantic segmentation methods to demonstrate its advantages. In the following subsections, we will outline the implementation details of the datasets and training parameters. Next, we analyze the effectiveness of the Attention Refined Two-Branch Real-Time Semantic Segmentation network structure and conduct thorough ablation experiments on the Cityscapes dataset to validate each module's efficacy in our method. Finally, we compare our final accuracy and speed (FPS) results with other algorithms across various benchmarks.

4.1. Datasets and Evaluation Metrics

Cityscapes: The Cityscapes dataset[34] is a widely utilized large-scale dataset for semantic segmentation of urban scenes, serving as a prominent benchmark in computer vision research and algorithm evaluation. The dataset consists of high-resolution images from 50 cities in Germany, covering different weather conditions, different seasons and various urban scenes. The images contain rich semantic information such as roads, pedestrians, vehicles, buildings, etc., making them ideal for deep learning models for semantic segmentation studies in urban scenes. The Cityscapes dataset consists of high-resolution images, each meticulously annotated at the pixel level with labels covering 30 different categories. The dataset can be divided into two subsets with two levels of annotation: fine and coarse. The finely annotated Cityscapes dataset includes 5000 high-resolution images, with 2975 images allocated for training, 500 for validation, and 1525 for testing (annotations for the test set are available on the official website for evaluation). The roughly labelled dataset, on the other hand, contains 20,000 images. We usually use the finely labelled dataset, the resolution of these images is up to 2048×1024, which provides detailed and rich scene information for research and development.

Camvid: The CamVid dataset[35] is a specialized dataset for semantic segmentation tasks in road scenes, developed by the Computer Vision Group at the University of Cambridge. It serves as a valuable resource for research in areas like autonomous driving and traffic monitoring. It is the first video collection to integrate semantic labelling of target categories. The dataset consists of 32 semantic labels and was originally sourced from video sequences of the Cambridge city area, with a total of 701 image sequences containing hundreds to thousands of frames each, which were subsequently generated by manually selecting more than 700 images to be annotated.

The CamVid dataset comprises 701 images depicting urban streetscapes, with 367 images used for training, 101 for validation, and the remaining 233 for testing. The remaining 233 images were used as a test set. This division helps researchers to train, validate and evaluate the algorithm to better understand its performance in different scenarios. The images have a resolution of 960 × 720 and contain 11 commonly used semantic labels.

4.2. Implementation Details

Training Strategy: During training, we utilized the Adam optimizer along with a polynomial decay learning rate scheduler and a warm-up strategy. For training, a single RTX 3090 GPU is utilised.Considering the need to process images with different resolutions, the input resolutions are adjusted to 512×1024 and 768×1536, the batchSize is adjusted to 20 and 10, the maximum number of loops are both set to 140k, and the initial learning rate is set to 1e-3 on Cityscapes dataset.For data enhancement aspect, random scaling, random fill cropping, and random horizontal flipping techniques are used. On the CamVid dataset, the input resolution is adjusted to 720 × 960, the batchSize is adjusted to 16, and the maximum number of loops is 80k. For data enhancement, only random cropping is used. In addition, when training on the CamVid dataset, the trained pre-training weights on the Cityscapes dataset are added. In the inference phase, this experiment does not use any acceleration trick over strategy and uses the test code provided by the PaddlePaddle deep learning framework for speedup. In order to ensure the effectiveness of the model in practical applications, images with different resolutions were used for inference, and the processing speed and segmentation accuracy were comprehensively evaluated. Ultimately, the standard metric of concurrent average intersection and frames per second were used to compare the performance of different models.

Inference settings: During inference, without employing acceleration techniques such as sliding window evaluation or tension acceleration strategies, for the Cityscapes dataset, we use 768×1536 and 512×1024 resolutions for inference. For the CamVid dataset, inference was performed using a resolution of 960×720. We used NVIDIA GTX 3090 GPU and performed all inference experiments on CUDA 11.2 and CUDNN 8.1 environments. We employed the standard metrics of mean Intersection over Union (mIoU) for segmentation accuracy comparisons and Frames Per Second (FPS) for inference speed comparisons.

4.3. Experiments on Cityscapes

4.3.1. Ablation study

In this section, ablation experiments are carried out on each component in the lightweight densely connected contextual refinement branch of ARTRNet as well as on the deformed convolutional attention refinement fusion module. The method of control variables in physical experiments is taken and experiments are conducted module by module to see their effects on the experimental results, thus proving that the proposed model achieves significant improvements. All ablation experiments were conducted on the Cityscapes dataset.

In order to verify the effectiveness of the downsampling module (DSM), the experiments will be conducted on the basis of input images with resolutions of 512×1024 and 768 ×1536, respectively. From Table 1(a), it can be seen that when the attention refinement module is used for both by default, the mIoU of the model is not too good at 512×1024 resolution when normal convolution is used. Whereas when the DSM module is used, it can be seen that the models all have better predictions with a slightly improved mIoU of 73.2% than when the DSM module is not used. This could mean that the DSM provides a better representation of the features during the downsampling process. A similar trend is observed at a higher resolution of 768×1536. With ARM, the mIoU of ARTRNet with DSM only is 73.2% at 512×1024 resolution, while at 768×1536 resolution, the mIoU improves to 74.2%. This further confirms the effectiveness of DSM in processing higher resolution images. Next, in Table 1(b), the effect of ARM on ARTRNet performance at different resolutions is explored. Using DSM by default for all, the mIoU of ARTRNet with ARM is 73.2% at 512 × 1024 resolution. However, at 768×1536 resolution, the mIoU of ARTRNet using ARM reaches 74.2%. This shows that ARM plays an equally important role in attention refinement, especially when dealing with higher resolution images. The experimental results in this section highlight the effectiveness of DSM and ARM in ARTRNet.

Table 1. Comparison of DAM and ARM in ARTRNet on the Cityscapes validation set.

Model	Resolution	None	DSM	None	ARM	mIoU
(a) DSM						
ARTRNet	512×1024	√			√	72.3
ARTRNet	512×1024		√		√	73.2
ARTRNet	768×1536	√			√	72.8
ARTRNet	768×1536		√		√	74.2
(b) ARM						
ARTRNet	512×1024		√	√		72.6
ARTRNet	512×1024		√		√	73.2
ARTRNet	768×1536		√	√		72.9
ARTRNet	768×1536		√		√	74.2

To assess the impact of the Low Resolution Context Aggregation Module (LRCAM) and Deformed Convolutional Attention Refinement Fusion Module (DCARFM) on semantic segmentation performance across various resolutions. In this subsection, ablation experiments are performed on the Low Resolution Context Aggregation Module (LRCAM) and Deformed Convolutional Attention Refinement Fusion Module (DCARFM). Using the DSM module and ARM module by default, Table 2(c) demonstrates the experimental results of the low-resolution context aggregation module (LRCAM), from which it can be seen that the model with the introduction of the LRCAM improves the

mIoU from 74.5% to 75.7% compared to the model without the LRCAM at a resolution of 512 × 1024. This result indicates that at lower resolutions, LRCAM can effectively aggregate contextual information, thus improving segmentation accuracy. At a higher resolution of 768×1536, LRCAM also demonstrates its advantages, resulting in an increase in mIoU from 75.3% to 76.9%. This further confirms the effectiveness of LRCAM in capturing richer contextual information. Table 2(d) demonstrates the experimental results of the Deformed Convolutional Attention Refinement Fusion Module (DCARFM), from which it can be seen that the introduction of the DCARFM improves the mIoU of the model from 74.4% to 75.7% at 512 × 1024 resolution. However, at 768×1536 resolution, the model with DCARFM outperforms the model without DCARFM by 76.9% mIoU. The experimental results confirm the effectiveness of LRCAM and DCARFM in ARTRNet.

Table 2. Comparison of ablation experiments of LRCAM and DCARFM in ARTRNet on Cityscapes validation set.

Model	Resolution	None	LRCAM	None	DCARFM	mIoU
(a) LRCAM						
ARTRNet	512×1024	√			√	74.5
ARTRNet	512×1024		√		√	75.7
ARTRNet	768×1536	√			√	75.3
ARTRNet	768×1536		√		√	76.9
(b) DCARFM						
ARTRNet	512×1024		√	√		74.4
ARTRNet	512×1024		√		√	75.7
ARTRNet	768×1536		√	√		75.4
ARTRNet	768×1536		√		√	76.9

4.3.2. Comparison with SOTA Methods

In this subsection, we present performance results of ARTRNet on the Cityscapes and CamVid datasets, followed by comparisons with other state-of-the-art real-time semantic segmentation methods to validate its effectiveness. The evaluation is conducted using a resolution of 768×1536 for Cityscapes and 720 × 960 for CamVid. The models are assessed based on four key metrics: Floating-Point Operations per Second (FLOPs), number of parameters, mean Intersection over Union (mIoU), and inference speed (FPS). A comprehensive analysis of these metrics is provided below.

Table 3 compares ARTRNet with nine other models on the Cityscapes dataset. Special attention is given to the performance comparison between ARTRNet and BiSeNetV2. BiSeNetV2, a lightweight network, is designed to prioritize fast inference speed without compromising on accuracy. Without the use of a pre-trained backbone network, BiSeNetV2 achieves 73.4% mIoU and 156 FPS at 512×1024 resolution without using accelerated processing, which shows that it is competitive in real-time applications. In comparison, ARTRNet achieved 76.9% mIoU and 96 FPS at a higher resolution of 768×1536. This result suggests that ARTRNet has improved in accuracy despite being slightly slower than BiSeNetV2. This enhancement may be attributed to the specific structure and optimisation strategies that ARTRNet employs in its network design, which help to capture finer image details and thus improve the accuracy of segmentation. Table 3 shows that most of the models are pre-trained on ImageNet, which is a time-consuming process but can be traded off for a relatively high segmentation accuracy. In contrast, the training of the models in this chapter chooses to start from zero. In addition, due to the limitation of GPU memory, in order to be able to make the training gradient more accurate, the ARTRNet model in this section is loaded with the pre-training weights of ARTRNet with a resolution of 512×1024 size on top of the resolution of 768×1536 size. As can be seen from Table 3, the ARTRNet in this section achieves a balance between accuracy and speed. In terms of mIoU, the method in this section is significantly better than other more advanced methods such as BiseNetV2[18]. In Table 3, this paper uses no to denote that the method has no backbone. backbone denotes in the backbone model. "*" indicates that the inference speed, GFLOPs and parameters of the model are tested and provided on the platform in this section. If "†" the method is labelled with, the accelerated processing is performed using TensorRT. "-" indicates that the methods do not report the corresponding results.

Table 3. Comparison with other methods on the Cityscapes dataset.

Model	Backbone	Resolution	GFLOPs	Params	mIoU	FPS
DFANet[27]	43-layer CNN	1024×1024	3.4	7.8M	71.3	100
SwiftNet[37]	ResNet18	1024×2048	104.0	11.8M	75.5	39.9
LRNNet[38]	55-layer CNN	512×1024	8.58	0.68M	72.2	71
RTHP[39]	MobileNetV2	448×896	49.5	6.2M	73.6	51
PP-LiteSeg-T1[40]	STDC1	512×1024	-	-	73.1	273.6
PP-LiteSeg-T2[40]	STDC1	768×1536	-	-	76.0	143.6
BiSeNetV1[17]	Xception 39	768×1536	14.8	5.8M	69.0	105.8
BiSeNetV1[17]	ResNet18	768×1536	55.3	49M	74.8	65.5
STDC1-Seg50*†[36]	STDC1	512×1024	24.8	8.3M	72.2	206.9
STDC2-Seg50*†[36]	STDC2	512×1024	38.0	12.3M	74.2	156.6
STDC1-Seg75*†[36]	STDC1	768×1536	55.9	8.3M	74.5	140.7
STDC2-Seg75*†[36]	STDC2	768×1536	85.6	12.3M	77.0	106.2
BiSeNetV2†[18]	no	512×1024	21.1	-	73.4	156
BiSeNetV2-L†[18]	no	512×1024	118.5	-	75.8	47.3
ARTRNet	no	512×1024	19	6.6M	75.7	132
ARTRNet	no	768×1536	38.6	6.6M	76.9	96

4.3.3. Experiments on CamVid

To further validate the generalization of ARTRNet, experiments were also conducted on the CamVid dataset with an input resolution of 720×960 using the same configuration. The specific results are detailed in Table 4. ARTRNet performs the best, achieving 76.5% mIoU and 110.4 FPS, which is higher than that of STDC2-Seg[36] but slightly lower than that of the accelerated BiSeNetV2[18]. Meanwhile, ARTRNet achieves a good balance of speed and accuracy, which further proves the superior performance of the method. In Table 4, this section uses "*" to denote the pre-training of the model loaded with ARTRNet under 3/4 graph. If the method is labelled with "†", the accelerated processing was performed using TensorRT.

Table 4. Comparison with other methods on the CamVid dataset.

Model	Backbone	GPU	mIoU	FPS
ENet[41]	43-layer CNN	TitanX	51.3	61.2
ICNet[16]	PSPNet50	TitanX	67.1	27.8
DFANet A[27]	Xception A	TitanX	64.7	120
DFANet B[27]	Xception B	TitanX	59.3	160
SwiftNet[37]	ResNet18	GTX 1080Ti	72.6	-
BiSeNetV1[17]	Xception39	GTX 1080Ti	65.6	175
BiSeNetV1[17]	ResNet18	GTX 1080Ti	68.7	116.3
STDC1†[36]	STDC1	GTX 1080Ti	73.0	197.6
STDC2†[36]	STDC2	GTX 1080Ti	73.9	152.2
BiSeNetV2[18]	no	GTX 1080Ti	72.4	124.5
BiSeNetV2-L[18]	no	GTX 1080Ti	73.2	32.7
BiSeNetV2*†[18]	no	GTX 1080Ti	76.7	124.5
ARTRNet	no	RTX 3090	76.5	110.4

4.3.4. Visualization Experiments on Cityscapes

To visually highlight the significant advantages of our proposed method, Figure 5 compares our approach with several other methods on the Cityscapes dataset. Through these comparison plots, it can be clearly observed that our method is the closest to the real-world scenarios in terms of presenting results, i.e., our method exhibits a superior performance in comparison with the three comparable methods.

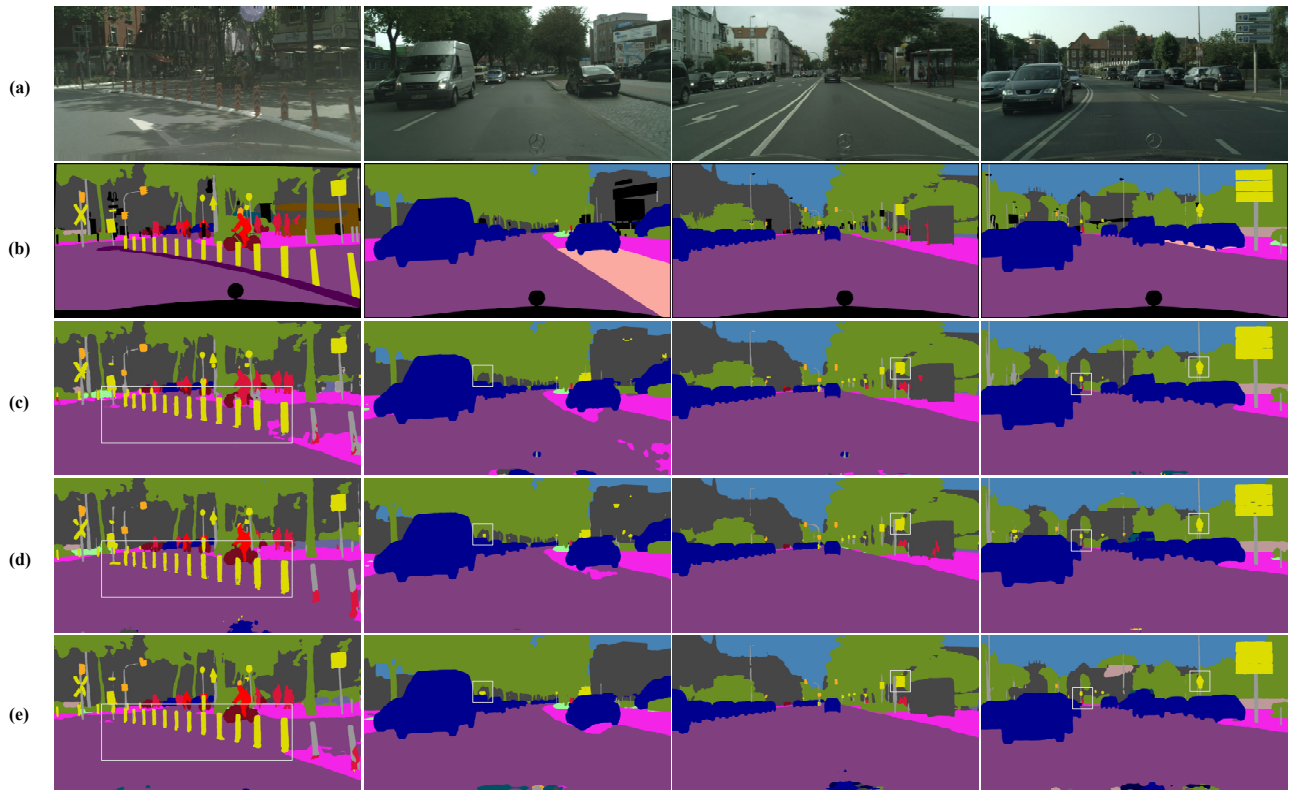

Figure 5. Visualisation of segmentation results of BiSeNetV1, BiSeNetV2 and ARTRNet on Cityscape dataset. (a) Input image; (b) Ground Truth; (c) BiSeNetV1; (d) BiSeNetV2; (e) ARTRNet.

5. Conclusion

Real-time semantic segmentation is increasingly crucial in demanding scenarios like autonomous driving. However, many existing methods prioritize accuracy over speed. Although BiSeNetV2 is effective, its speed improvement is limited while maintaining high accuracy. Therefore, achieving a balanced trade-off between speed and accuracy is

a key focus of current research. In this paper, we introduce an efficient Attention Refined Two-Branch Real-Time Semantic Segmentation Network. We propose lightweight densely connected contextual refinement branches to reduce computational load and improve speed while ensuring accuracy. Additionally, to address the challenge of feature map detail loss during branch fusion, we propose the Deformed Convolutional Attention Refinement Fusion Module. This module refines feature map details through deformed convolutional attention refinement operations, enhancing the segmentation capability of the model. Experimental results on Cityscapes and CamVid datasets demonstrate that our proposed ARTRNet achieves a favorable balance between segmentation accuracy and inference speed, surpassing other representative real-time semantic segmentation methods.

References

[1] Azuma R T. A survey of augmented reality [J]. Presence: Teleoperators and Virtual Environments, 1997, 6(4): 355-385.

[2] Siam M, Gamal M, Abdel-Razek M, et al. A comparative study of real-time semantic segmentation for autonomous driving[C]. IEEE Conference On Computer Vision and Pattern Recognition, 2018: 587-597.

[3] You H, Yu L, Tian S, et al. DR-Net: Dual-rotation network with feature map enhancement for medical image segmentation [J]. Complex and Intelligent Systems, 2021: 1-13.

[4] Dechesne C, Mallet C, Le Bris A, et al. Semantic segmentation of forest stands of pure species combining airborne lidar data and very high resolution multispectral imagery [J]. ISPRS Journal of Photogrammetry and Remote Sensing, 2017, 126: 129-145.

[5] Zhuang J, Wang Z, Wang B. Video semantic segmentation with distortion-aware feature correction [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2020, 31(8): 3128-3139.

[6] Chen L C, Zhu Y, Papandreou G, et al. Encoder-decoder with atrous separable convolution for semantic image segmentation [C]. European Conference on Computer Vision, 2018: 801-818.

[7] Nirkin Y, Wolf L, Hassner T. Hyperseg: Patch-wise hypernetwork for real-time semantic segmentation [C]. IEEE Conference on Computer Vision and Pattern Recognition, 2021: 4061-4070.

[8] Yuan Y, Huang L, Guo J, et al. OCNet: Object context for semantic segmentation [J]. International Journal of Computer Vision, 2021, 129(8): 2375-2398.

[9] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation [C]. IEEE Conference On Computer Vision and Pattern Recognition, 2015: 3431-3440.

[10] Hung S W, Lo S Y, Hang H M. Incorporating luminance, depth and color information by a fusion-based network for semantic segmentation [C]. IEEE International Conference on Image Processing, 2019: 2374-2378.

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

[12] Li X, You A, Zhu Z, et al. Semantic flow for fast and accurate scene parsing [C]. European Conference on Computer Vision, 2020: 775-793.

[13] Poudel R P K, Liwicki S, Cipolla R. Fast-scnn: Fast semantic segmentation network [J]. arXiv preprint arXiv:1902.04502, 2019.

[14] Dong Y, Zhao K, Zheng L, et al. Refinement co-supervision network for real-time semantic segmentation [J]. IET Computer Vision, 2023, 17(6): 652-662.

[15] Shvets A A, Rakhlin A, Kalinin A A, et al. Automatic instrument segmentation in robot-assisted surgery using deep learning [C]. IEEE International Conference on Machine Learning and Applications, 2018: 624-628.

[16] Zhao H, Qi X, Shen X, et al. Icnet for real-time semantic segmentation on high-resolution images [C]. European Conference on Computer Vision, 2018: 405-420.

[17] Yu C, Wang J, Peng C, et al. Bisenet: Bilateral segmentation network for real-time semantic segmentation [C]. European Conference on Computer Vision, 2018: 325-341.

[18] Yu C, Gao C, Wang J, et al. Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation [J]. International Journal of Computer Vision, 2021, 129: 3051-3068.

[19] Mehta S, Rastegari M, Caspi A, et al. Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation [C]. European Conference on Computer Vision, 2018: 552-568.

[20] Lo S Y, Hang H M, Chan S W, et al. Efficient dense modules of asymmetric convolution for real-time semantic segmentation [C]. ACM International Conference on Multimedia in Asia, 2019: 1-6.

[21] Otsu N. A threshold selection method from gray-level histograms [J]. Automatica, 1975, 11(285-296): 23-27.

[22] Vincent L, Soille P. Watersheds in digital spaces: an efficient algorithm based on immersion simulations [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1991, 13(06): 583-598.

[23] Ren, Malik. Learning a classification model for segmentation [C]. IEEE International Conference on Computer Vision, 2003: 10-17 vol. 1.

[24] Barbu A. Training an active random field for real-time image denoising [J]. IEEE Transactions on Image Processing, 2009, 18(11): 2451-2462.

[25] Badrinarayanan V, Kendall A, Cipolla R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(12): 2481-2495.

[26] Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation [C]. Medical Image Computing and Computer-Assisted Intervention, 2015: 234-241.

[27] Li H, Xiong P, Fan H, et al. Dfanet: Deep feature aggregation for real-time semantic segmentation [C]. IEEE Conference On Computer Vision and Pattern Recognition, 2019: 9522-9531.

[28] Hong Y, Pan H, Sun W, et al. Deep dual-resolution networks for real-time and accurate semantic segmentation of road scenes [J]. arXiv preprint arXiv:2101.06085, 2021.

[29] Fu J, Liu J, Tian H, et al. Dual attention network for scene segmentation [C]. IEEE Conference On Computer Vision and Pattern Recognition, 2019: 3146-3154.

[30] Huang Z, Wang X, Huang L, et al. Ccnet: Criss-cross attention for semantic segmentation [C]. IEEE International Conference on Computer Vision, 2019: 603-612.

[31] Han K, Wang Y, Tian Q, et al. Ghostnet: More features from cheap operations [C]. IEEE Conference On Computer Vision and Pattern Recognition, 2020: 1580-1589.

[32] Huang G, Liu Z, Van Der Maaten L, et al. Densely connected convolutional networks [C]. IEEE Conference On Computer Vision and Pattern Recognition, 2017: 4700-4708.

[33] Azad R, Niggemeier L, Hüttemann M, et al. Beyond self-attention: Deformable large kernel attention for medical image segmentation [C]. IEEE Winter Conference on Applications of Computer Vision, 2024: 1287-1297.

[34] Cordts M, Omran M, Ramos S, et al. The cityscapes dataset for semantic urban scene understanding [C]. IEEE Conference On Computer Vision and Pattern Recognition, 2016: 3213-3223.

[35] Brostow G J, Shotton J, Fauqueur J, et al. Segmentation and recognition using structure from motion point clouds [C]. European Conference on Computer Vision, 2008: 44-57.

[36] Fan M, Lai S, Huang J, et al. Rethinking bisenet for real-time semantic segmentation [C]. IEEE Conference On Computer Vision and Pattern Recognition, 2021: 9716-9725.

[37] Wang H, Jiang X, Ren H, et al. Swiftnet: Real-time video object segmentation [C]. IEEE Conference On Computer Vision and Pattern Recognition, 2021: 1296-1305.

[38] Jiang W, Xie Z, Li Y, et al. Lrnnet: A light-weighted network with efficient reduced non-local operation for real-time semantic segmentation [C]. IEEE International Conference on Multimedia and Expo Workshops, 2020: 1-6.

[39] Dong G, Yan Y, Shen C, et al. Real-time high-performance semantic image segmentation of urban street scenes [J]. IEEE Transactions on Intelligent Transportation Systems, 2020, 22(6): 3258-3274.

[40] Peng J, Liu Y, Tang S, et al. PP-liteseg: A superior real-time semantic segmentation model. arXiv 2022[J]. arXiv preprint arXiv:2204.02681.

[41] Paszke A, Chaurasia A, Kim S, et al. Enet: A deep neural network architecture for real-time semantic segmentation [J]. arXiv preprint arXiv:1606.02147, 2016.