



Compact interactive dual-branch network for real-time semantic segmentation

Yongsheng Dong¹ · Haotian Yang¹ · Yuanhua Pei¹ · Longchao Shen¹ · Lintao Zheng¹ · Peiluan Li²

Received: 5 December 2022 / Accepted: 20 March 2023 / Published online: 26 April 2023
© The Author(s) 2023

Abstract

The traditional complete dual-branch structure is effective for semantic segmentation tasks. However, it is redundant in some sense. Moreover, the simple additive fusion of the features from the two branches may not achieve the satisfactory performance. To alleviate these two problems, in this paper we propose an efficient compact interactive dual-branch network (CIDNet) for real-time semantic segmentation. Specifically, we first build a compact interactive dual-branch structure by constructing a compact detail branch and a semantic branch. Furthermore, we build a detail-semantic interactive module to fuse several specific stages of the two branches in the backbone network with the corresponding stages of the detail resolution branch. Finally, we propose a dual-branch contextual attention fusion module to deeply fuse the extracted features and predict the final segmentation result. Extensive experiments on Cityscapes and CamVid dataset demonstrate that the proposed CIDNet achieve satisfactory trade-off between segmentation accuracy and inference speed, and outperforms 20 representative real-time semantic segmentation methods.

Keywords Semantic segmentation · Dual-branch network · Real-time · Deep learning

Introduction

Deep learning and related theories have been developing in recent years in the research of transfer learning [1], object detection [2, 3], style transfer [4], nonlinear systems [5, 6] and so on. Semantic segmentation, as one of the most fundamental tasks in the computer vision community, aims to assign a semantic class label to each pixel in the given image. It has been extensively and deeply studied and applied in a variety of fields, such as augmented reality [7, 8], autonomous driving [9, 10], medical images [11, 12], satellite imagery [13], video surveillance [14, 15] and so on. Many mobile terminal tasks have great demand for segmentation speed, so the real-time semantic segmentation [16] method comes into being. Real-time semantic segmentation tasks require speed while ensuring accuracy, So the most challenge of real-time semantic segmentation is to achieve the optimal balance between

accuracy and efficiency. That is, it is urgent and important to build a real-time semantic segmentation method that achieves a good balance between accuracy and efficiency.

In recent years, with the development of convolutional neural networks and the proposal of fully convolutional network (FCN) [17], a series of real-time semantic segmentation methods [18–20] have been proposed. These methods have low-latency and considerable segmentation accuracy. To capture details information and semantic information, two bilateral segmentation networks were proposed by construct a dual-branch architecture [21, 22]. One pathway is designed to capture the spatial details, the other pathway is introduced to extract the categorical semantics. It's worth thinking about, complete dual-branch design has brought better segmentation accuracy at the same time also brought more computational cost. However, it is not enough to completely restore the lost spatial information by only relying on upsampling, so it is necessary to introduce high-resolution feature maps.

Some typical real-time semantic segmentation networks have recently proposed by choosing the lightweight backbones to improve real-time inference while using feature fusion or aggregation modules to compensate for the drop of accuracy [21, 23, 24]. However, Short-Term Dense Concate-

✉ Yongsheng Dong
dongyongsheng98@163.com

¹ School of Information Engineering, Henan University of Science and Technology, Luoyang 471023, China

² School of Mathematics and Statistics, Henan University of Science and Technology, Luoyang 471023, China

nate (STDC) network (STDCNet) [25] believes that these lightweight backbones borrowed from image classification task may not be the best choice for solving image segmentation problem. And it uses the characteristics of high resolution feature map as auxiliary, shown in Fig. 1a. Besides, the real-time semantic segmentation demands for an efficient inference speed. In fact, two types of methods can be used to promote the inference speed: (i) Restricting the input image size, smaller input size results in less computation cost with the same network architecture; (ii) Channel Pruning, pruning channels in early stages can improve inference speed. Although these two manners can improve the inference speed to a certain extent, they may also lead to a decrease in accuracy. To tackle this problem, as shown in Fig. 1a, BiSeNet [21] adopt dual-branch architecture to fuse the low-level details and high-level semantics. However, complete dual-branch maybe is time-consuming, and the auxiliary path is not effective enough due to the loss of detail information guidance. The method we propose is different from the method a and b in Fig. 1. The method of Fig. 1a guides the fusion of semantic information and spatial information by extracting spatial information from the low-level stage of semantic branch. Compared with Fig. 1a, our method has one more short-term spatial detail branch, which can make the network extract richer spatial detail features. Compared with Fig. 1b, our method reduces the redundant parts of the spatial detail branches, and transmits information between the two branches, so that the spatial information and semantic information can form a stepwise fusion, which makes the fusion effect better.

Inspired by the Short-Term Dense Concatenate module (STDC [25] module), we propose a real-time and dual-branch architecture named CIDNet. As illustrated in Fig. 2, CIDNet adopts the encode-decoder architecture. The compact interactive dual-branch not only maintains the high resolution of the features but also ensures a certain speed. The last and most critical step is the fast and efficient integration of the two branches of semantic information and spatial detail. We use pyramid pooling to further extend high-level semantics, while using the idea of self-attention to fuse the two branches quickly and effectively.

Our main contributions are summarized as follows:

1. We propose an efficient and effective two-pathway architecture, termed the Compact Interactive Dual-Branch network, for real-time semantic segmentation. Due to its special structure, it saves a certain number of parameters and computation, and it is faster and more accurate than common two-pathway networks.
2. We propose a Detail-Semantic Interactive Module (DSIM) to reduce the loss of two branches during fusion. It enables effective interaction between the semantic branch and the detail branch. Meanwhile, it can effectively guide the

fusion of semantic and spatial information between high resolution and low resolution.

3. We propose a Dual-Branch Contextual Attention Fusion Module (DBCAFModule). It combines pyramid pooling and self-attention mechanism to integrate semantic branch and spatial branch deeply.
4. We conduct extensive experiments to investigate the effectiveness of our methods. Our methods achieves impressive results benchmarks of Cityscapes, CamVid. Specifically, our CID1-50 achieves 75.1% mIoU on the Cityscapes *val* set at a speed of **164.1 FPS** on Tesla V100 card. Under the same experiment setting, our CID2-75 achieves **77.7%** mIoU at a speed of 92.9 FPS. At the same input image resolution, our params is 1.8 M and 5.2 M less than STDCNet, respectively.

Related work

BiSeNetV2 [22] consists of a Detail Branch and a Semantic Branch, which are merged by an Aggregation Layer (BGA Module). Each layer of the detail branch consists by a convolution layer followed by batch normalization and activation function, the semantic branch consists by several Inverted Bottleneck and Gather-and-Expansion Layers, which inspired by the lightweight recognition model, e.g., Xception [26], Mobilenet [27], ShuffleNet [28]. However, the complete dual-branch architecture may be inefficient due to repeated processing in the initial stage. STDCNet [25] is a single-stream method, which backbone consist by the Short-Term Dense Concatenate module (STDC Module). But the channel number of each stage output may be redundancy.

In this chapter, our discussion mainly focuses on the four groups of methods most relevant to our work, i.e., generic semantic segmentation methods, real-time semantic segmentation methods, and feature fusion modules.

Generic semantic segmentation

After traditional segmentation methods, e.g., threshold selection [29], region growing [30], super-pixel [31–33] and graph algorithms [34, 35], many CNN-based methods have been proposed [36]. Recently, With the development of computer hardware and deep learning, semantic segmentation has also made remarkable leap-forwards. A series methods base on FCN [17] keep improving state-of-the-art performance on various benchmarks. The Deeplabv3 [19] abandoned CRFs [37] post-processing and devises an atrous spatial pyramid pooling to capture multi-scale context. The DeepLabv3plus [38] introduce the decoder to fuse upsample feature maps with low-level feature maps. The Segnet [39] utilized the indices of max-pooling operation in encoder to upsampling operation in decoder. The PSPNet [40] adopts a pyramid

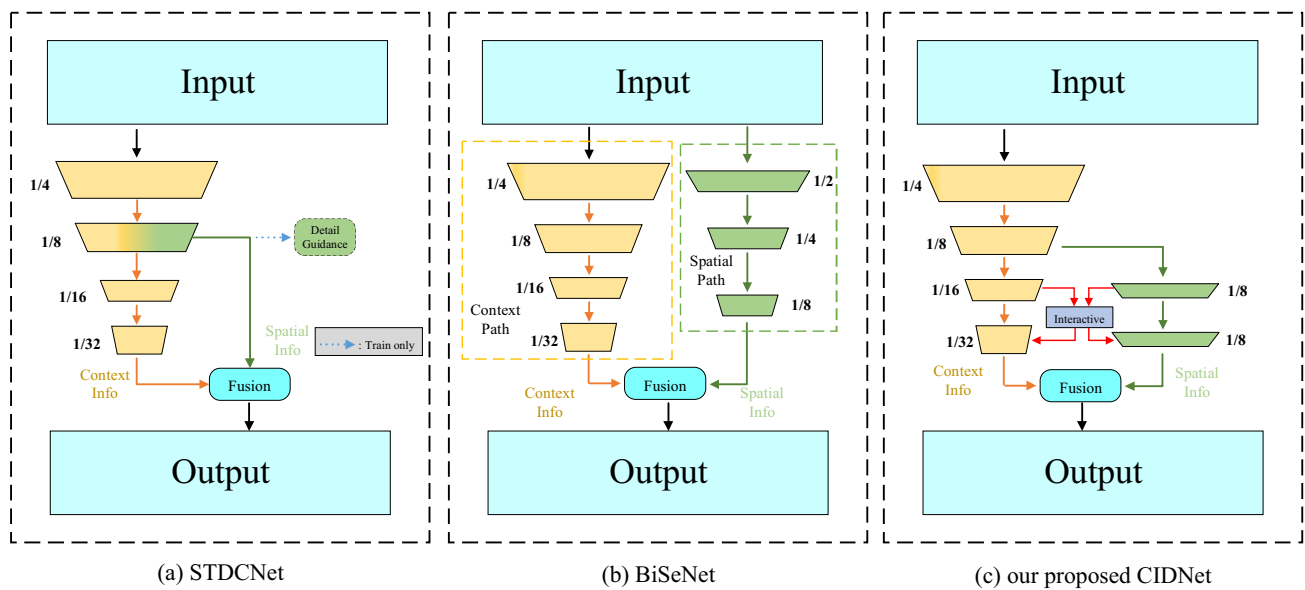


Fig. 1 Illustration of architectures of STDCNet, BiSeNet and our proposed approach. **a** Short-Term Dense Concatenate network (STDCNet), which use a Detail Guidance module to encode spatial information in one low-level features stage. **b** Bilateral Segmentation Network

(BiSeNet), which use an extra Spatial Path to encode spatial information. **c** Our proposed method, which use an interactive module to fuse high-level semantics information with low-level detail features

pooling module on the dilation backbone to capture local and global context information. Both dilation backbone and encoder–decoder structure are mainstream semantic segmentation architectures. Meanwhile, some methods introduce the attention mechanisms, e.g., OCNet [41] and DFANet [24] use self-attention, PSANet [42] use spatial attention and EncNet [20] use channel attention, to capture long-range dependencies. In this paper, we propose a novel and efficient architecture to achieves good trad-off between speed and accuracy. In this paper, we propose a novel and efficient architecture to achieves good trad-off between speed and accuracy.

Real-time semantic segmentation

Many scene parsing tasks require real-time inference and its semantic segmentation algorithms attract increasing attention. In this situation, there are two mainstream efficient and effective semantic segmentation methods, and most of them adopt a lightweight backbone. (i) *encoder–decoder architecture*. The encoder–decoder structure is a good paradigm to effectively utilize the multi-level image feature information extracted by the backbone network. The encoder usually uses a deep network to extract contextual information through convolution and downsampling operations. The decoder gradually recovers the resolution and fuses the multi-level feature maps extracted by the backbone network to guarantee dense predictions. STDCNet [25] adopts a pre-trained STDC networks as the encoder and use the context

path of BiSeNetV1 [21] to encode the context information, meanwhile, the decoder fused the learned detail features and the context features. SwiftNet [43] use lightweight lateral connections to assist with upsampling. (ii) *multi-branch architecture*. The encoder–decoder architecture saving computation cost to a certain extent, but this greatly impairs the long-range dependency information lost due to multiple downsampling processes, which cannot be recovered by upsampling or deconvolution, and it will whittle down the accuracy of semantic segmentation. To alleviate the problem, the multi-branch architecture is proposed. BiSeNetV1 [21] and BiSeNetV2 [22] proposed a dual-branch network, the two paths are used to extract spatial information and context information respectively. ICNet [44] fused multi-scale feature map in a cascaded manner to achieve a good speed-accuracy trade-off. DDRNet [45] used a deep bilateral networks and multiple bilateral fusions to improve segmentation efficiency. (iii) *lightweight encoder*. Some semantic segmentation networks used the encoder from image classification task as the backbone. But they are not specifically designed for segmentation tasks. Therefore, the channels of these encoders may be redundant or insufficient. MobileNetV1 [27] use depthwise separable convolutions to reduce parameters and computation. MobileNetV2 [46] introduced bottleneck structure, and it has very few channels. MobileNetV3 [47] combines network architecture search (NAS) and NetAdopt algorithms, and apply the Squeeze-and-Excite (SE) in the residual layer. ShuffleNet [28] utilizes the compactness of grouped convolutions and proposes a chan-

nel shuffle operation to stimulate information fusion between different groups.

Feature fusion module

The feature fusion is an important and common operation of semantic segmentation network, which can strengthen the representation of the features extracted by the encoder. The basic fusion operation is the element-wise summation or concatenation. BiSeNetV1 [21] uses the Attention Refinement Module (ARM) to facilitate multi-scale fusion in the context path, and Feature Fusion Module (FFM) fuses the features in spatial path and context path. Although STDC-Net [25] uses the same fusion method as BiSeNetV1 [21], it has only a detail path, and the spatial path is one of the stages. In BiSeNetV2 [22], the Bilateral Guided Aggregation (BGA) Layer fuse the complementary information from the detail branch (the low-level) and semantic branch (the high-level). DFANet [24] fuses features in sub-network aggregate and sub-stage aggregate ways. DDRNet [45] has two parallel deep branches with different resolutions, and features fusion through multiple bilateral fusion operations.

Our proposed method

In this section, we first introduce the architecture of our proposed Compact Interactive Dual-Branch Network (CIDNet). We then describe our proposed Spatial-Channel Interactive Module (SCIM) used for reducing the loss of two branches of CIDNet during fusion, and further give the Dual-Branch Contextual Attention Fusion Module (DBCAFM) used to fuse the two branches for semantic segmentation.

The architecture of compact interactive dual-branch network (CIDNet)

Due to that the detail branch is time-consuming, the current complete dual-branch network has a large computational cost and is not easily trained [22]. In this work, we propose a novel Compact Interactive Dual-branch Network (CIDNet). The architecture of our proposed CIDNet consists of a compact interactive detail branch and a semantic branch, shown in Fig. 2. The compact interactive detail branch shares the first three stages with the backbone network and merges with the semantic branch at the S3 stage. Our compact interactive detail branch has two stages (D4 and D5), and each stage has two layer. Each layer of the detail branch is a convolution layer followed by batch normalization and activation function. Each convolution layer has a stride $s = 1$, which can maintain high resolution. High resolution and high channel capacity make detail branch encode rich spatial location information.

Note that, an appropriate resolution size is important for segmentation. due to that the operation of downsampling will change the relative position between pixel–pixel. That is, maintaining a certain resolution to extract position information can make the segmentation effect better. In fact, if a larger resolution is selected for the detail branch, the amount of calculation will increase sharply, and the inference speed will be reduced at the same time. If a smaller resolution is selected, the function of keeping position information will be lost. As used in a large number of segmentation networks (e.g., [21, 22, 25]), 1/8 of the original input is the best choice for resolution, and width channels can retain more location details information. On the other hand, to achieve a trade-off between inference speed and efficiency, we follow the BiSeNetV2 [22] philosophy of using wide channel dimensions and shallow layers. We choose the 1/8 of the original input image resolution and 128 channels as the detail branch layers. Meanwhile, it should also be noted that our constructed detail branch is different from BiSeNetV2 [22] because our detail branch are short-term.

The semantic branch consists of six stages, stem is composed of S1–S2 stages, S3–S5 stages are stacked by group short-term dense concatenate modules (GSTDC modules), and finally S6 stage is composed of two 1×1 convolution and a 3×3 groups of convolution. To alleviate channel redundancy problem and reduce the parameters, we construct a Group-Short-Term Dense Concatenate module (GSTDC module). It is an extension of Short-Term Dense Concatenate module (STDC module) [25]. As in previous work [22, 45, 48], the number of channels in the backbone network is selected as [32, 64, 128, 256, 512]. In our GSTDC module, we use a stepped group convolution instead of the standard 3×3 convolution, add residual connections, and halve the number of input channels.

For clarity, we plot the CatBlock, GSTDC Module, and STDC module in Fig. 3. Each GSTDC module has four blocks and a fusion layer as in the STDC Module. The first block has one convolutional layer with kernel = 1×1 , one batch normalization layer and a ReLU activation function. The last three blocks are the group convolution with kernel 3×3 , and the number of groups is [4, 2, 1], and all of them have one batch normalization layer and a ReLU activation function. The output channels of these four blocks are respectively 1/2, 1/4, 1/8, and 1/8 of the entire module output channels. The last fusion layer is used to fuse the outputs of the previous four blocks in a concatenation manner. Finally, The input is added to the concatenation fusion result, which is the idea of skip connect. It can be seen from Fig. 3b, c that different from the STDC module, our two GSTDC modules adopt residual connection, and we use depthwise convolution in the downsampling module, so that the number of channels can be matched and the diversity between channels can be retained as much as possible.

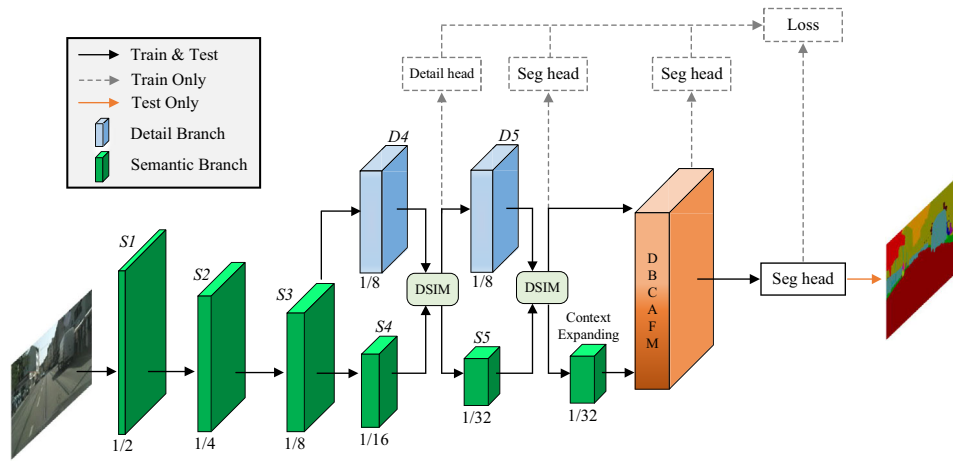
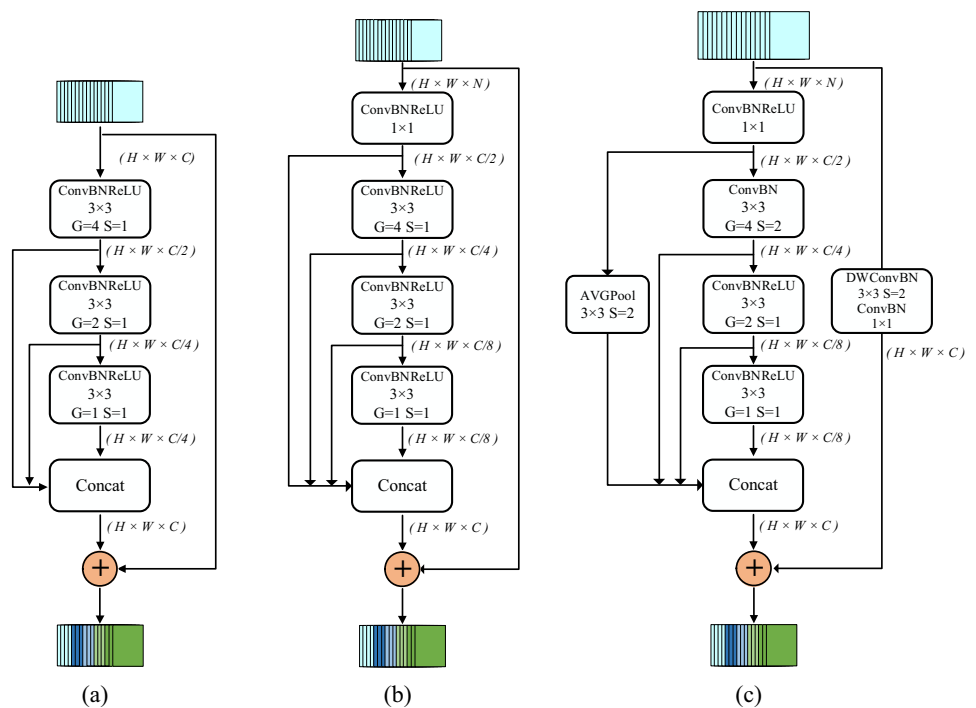


Fig. 2 Overview of the compact interactive dual-branch network. The CID network contains a compact interactive detail branch (the blue cubes) and a semantics branch (the green cubes). *DSIM* denotes the *Detail-Semantic Interactive Module*. *DBC AFM* denotes the *Dual-Branch Contextual Attention Fusion Module*. Meanwhile, the number below the cube is the ratio of the feature map size to the input resolution.

In addition, in the accelerated training part, we design three auxiliary segmentation heads to improve the segmentation performance without additional inference cost. The auxiliary segmentation header includes two CrossEntropyLoss and one DetailAggregateLoss [25]. *Seg Head* denotes the segmentation head, *Detail Head* denotes the Detail head [25]

Fig. 3 The GSTDC module and CatBlock in the semantic branch. **a** Is the catblock, which adopts a keep-resolution strategy. This block can enlarge the receptive field. **b** Short-Term Dense Concatenate module with groups (GSTDC module) used in our network. **c** The GSTDC module with stride = 2.

Notation: *Conv* is convolutional operation. *BN* is the batch normalization. *ReLU* is the ReLU activation function. *G* denotes the group convolution. *AVGPool* is the average pooling. *S* denotes the stride. *DWConvBN* is depthwise convolution operation with batch normalization. *Concat* means concatenation. “+” represents element-wise adding. Meanwhile, 1×1 , 3×3 denote the kernel size, $H \times W \times C$ means the tensor shape (height, width, depth)



Note that, to improve the inference speed and make the network more lightweight, our GSTDC module uses half of the output channels in the STDC module. Table 1 show the detailed structure of our compact interactive dual-branch. In the first two stages, we adopt a wide number of channels to maintain rich detail information, use max pooling for fast downsampling, and then use a CatBlock to expand the receptive field. In stages S3–S5 phase, we only use the GSTDC basic block stacking. In S6 stage adopts the bottleneck of

flexible structure, which can flexibly adjust output channels, also can consider whether to continue downsampling feature maps, and we call the S6 stage for the Context Expanding Module.

In the following subsection, we will describe our proposed spatial-channel interactive module for reducing the loss of two branches during the branch fusion.

Table 1 Instantiation of the compact interactive detail branch and the semantics branch

Stage	Output size	CID1		CID2	
		Input	Compact detail	Compact detail	Semantic
S1	256 × 512		Semantic Conv, 3 × 3, 32, s = 2 Conv, 3 × 3, 32, s = 1 Conv, 3 × 3, 64, s = 1	Compact detail	Semantic Conv, 3 × 3, 32, s = 2 Conv, 3 × 3, 32, s = 1 Conv, 3 × 3, 64, s = 1
S2	128 × 256		Semantic MaxPool, 3 × 3, 64, s = 2 CatBlock, 3 × 3, 64, s = 1	Compact detail	Semantic MaxPool, 3 × 3, 64, s = 2 CatBlock, 3 × 3, 64, s = 1
S3	64 × 128		Semantic GSTDC, 128, s = 2 GSTDC, 128, s = 1	Compact detail	Semantic GSTDC, 128, s = 2 GSTDC, 128, s = 1 GSTDC, 128, s = 1
S4	32 × 64		Semantic Conv, 3 × 3, 128, s = 1 Conv, 3 × 3, 128, s = 1 GSTDC, 256, s = 2 GSTDC, 256, s = 1	Compact detail	Semantic Conv, 3 × 3, 128, s = 1 Conv, 3 × 3, 128, s = 1 GSTDC, 256, s = 2 GSTDC, 256, s = 1 GSTDC, 256, s = 1 GSTDC, 256, s = 1
S5	16 × 32		Semantic Detail-semantic interactive Conv, 3 × 3, 128, s = 1 Conv, 3 × 3, 128, s = 1 GSTDC, 512, s = 2 GSTDC, 512, s = 1	Compact detail	Semantic Detail-semantic interactive Conv, 3 × 3, 128, s = 1 Conv, 3 × 3, 128, s = 1 GSTDC, 512, s = 2 GSTDC, 512, s = 1
S6	16 × 32		Semantic Detail-semantic interactive Conv, 1 × 1, 512, s = 1 Conv, 3 × 3, 512, s = 1 Conv, 1 × 1, 1024, s = 1	Compact detail	Semantic Detail-semantic interactive Conv, 1 × 1, 512, s = 1 Conv, 3 × 3, 512, s = 1 Conv, 1 × 1, 1024, s = 1

The two versions in the table are CID1 and CID2. Note that *Conv*, 3 × 3 shown in the table refers to the Conv-BN-ReLU with kernel 3 × 3. The number after the convolution operation represents the number of output channels. *s* denote stride, *MaxPool* is the max pooling operation. *CatBlock* denote our CatBlock. *GSTDC* denote our basic block

Detail-semantic interactive module

For dual-branch networks, it is important to effectively fuse the feature information extracted from both two branches. Current complete dual-branch structures [21, 22] have no interaction between the two ways, and thus the accuracy is not satisfactory. Although there is also have dual-branch with interaction structure [45], but the fusion method is to add the feature maps on both sides directly. This fusion method may have the phenomenon of “fusion loss”, because the semantic branch contains rich and accurate semantic information, and the feature map from details branch may not match the information on its corresponding position. To alleviate the above problems, in this work we propose a spatial-channel interactive module to guide the information exchange between the two branches. In this way, the resulting two branches can have certain common features at different scales. Our spatial-channel interactive module contains two paths: (i) high-resolution maps in the detail branch are integrated into low-resolution feature maps in the semantic branch.(called high-low path) (ii) low-resolution maps in the semantic branch are integrated into high resolution feature maps in the detail branch.(called low-high path)

We next describe the detailed structure of the detail-semantic interactive module, shown in Fig. 4. Firstly, the high-resolution feature map retains a lot of position and detail information. In the high-low path, we adopt the method of fast downsampling and expanding the number of channels, and directly fuse by additive the feature map after downsampling and the low-resolution feature map. In the low-high path, we used the attention guided strategy (AGS), low-resolution features contain abundant semantic information. To effectively integrate them with high-resolution feature maps, we draw on the self-attention mechanism, and first upsample the low-resolution feature maps by bilinear interpolation. Then, the upsampled feature map is element-wise multiplication with the high-resolution feature map, and the high-level feature map is obtained by fusing the channel-wise direction. The obtained high-level feature map is input into the sigmoid function, and finally the relative attention mask α is obtained. After obtaining the two attention maps, we further perform pixel-level product between mask and prediction, and then perform pixel-level sum between them to obtain the final result.

We can formulate the above procedure as follow:

$$\begin{cases} F'_s = \text{Downsample}(F_d) + F_s, \\ F'_d = \text{AGS}(F_d, F_s) + F_d, \end{cases} \quad (1)$$

where AGS is the attention guided strategy, F_s is the output of before stage of the semantic branch, and F'_s is the input to the next stage of the semantic branch. F_d is the output

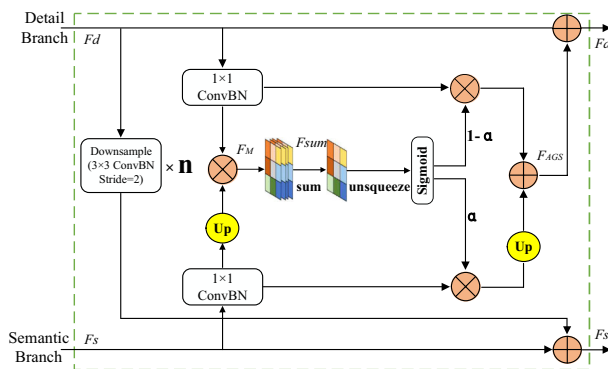


Fig. 4 Detailed design of the detail-semantic interactive module. Notation: *Conv* is convolutional operation. *BN* is the batch normalization. *Sigmoid* is the Sigmoid activation function. *sum* denotes the channel-wise summation operation. *unsqueeze* denotes inserts a dimension of size 1 at the specified location. α denotes the weight output of Sigmoid function. *Up* is the standard bilinear interpolation operation. “+” represents element-wise adding. “x” represents element-wise product. Meanwhile, $1 \times 1, 3 \times 3$ denote the kernel size. “ $\times n$ ” represents the number of times, and the value of “ n ” is determined according to the feature map downsampling size

of before stage of the compact interactive detail branch, and F'_d is the input to the next stage of the compact interactive detail branch. In short, the procedures of the attention guided strategy (AGS) can be formulated as follow:

$$\begin{cases} F_M = \text{Multiply}(F_d, \text{Upsample}(F_s)), \\ F_{\text{sum}} = \text{Sum}(F_M), \\ \alpha = \text{Sigmoid}(F_{\text{sum}}), \\ F_{\text{AGS}} = F_d \cdot (1 - \alpha) + F_s \cdot \alpha, \\ F'_d = F_{\text{AGS}} + F_d, \end{cases} \quad (2)$$

where *Multiply* is element-wise multiplication operation, *Sum* is the channel-wise summation operation, *Upsample* is the standard bilinear interpolation operation, F_{AGS} is the output of the AGS. F_d is the output of the compact interactive detail branch. α is the relative attention mask with F_d and F_s .

Dual-branch contextual attention fusion module

Large receptive fields can capture higher-level semantic information. Large convolution kernels and dilated convolutions are often used to expand the receptive field, but these operations bring large delay and high computational complexity. Pyramid pooling has been shown to be an effective method for extracting high-level semantics [19, 43] for semantic segmentation [45]. Furthermore, Deep Aggregation Pyramid Pooling Module (DAPPM) was proposed and effectively used in real-time semantic segmentation [45]. However, the structure of DAPPM is somewhat complex. In this work, we propose a simplified DAPPM used to

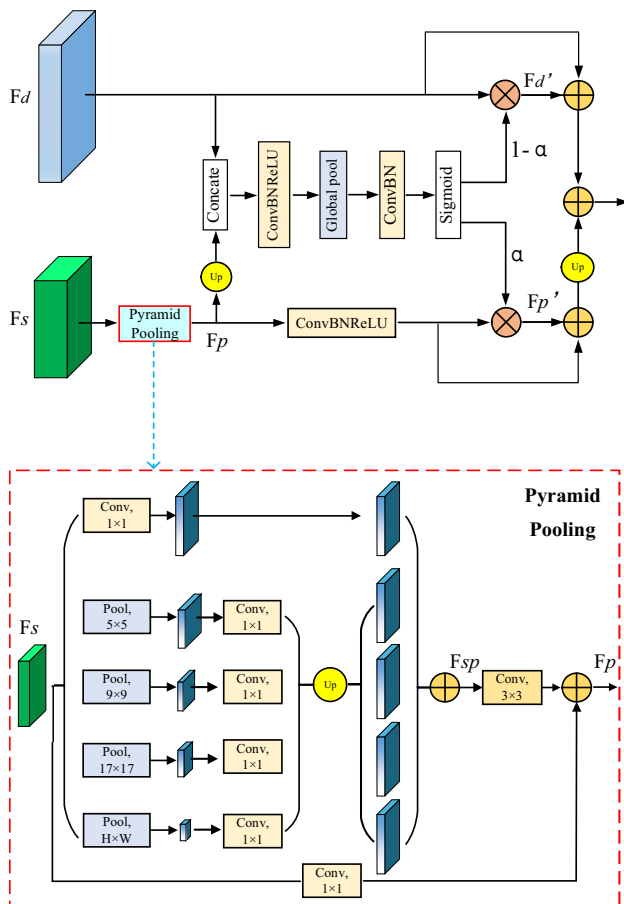


Fig. 5 The detailed architecture of dual-branch contextual attention fusion module. Notation: *Conv* is convolutional operation, and 1×1 , 3×3 denote the kernel size. *Pool* is global-average-pooling operation, and 5×5 , 9×9 , 17×17 , $H \times W$ means pooling kernel (“ H ” indicates the height of the input feature and “ W ” indicates the width of the input feature). *Up* is bilinear upsampling operation. “+” represents element-wise addition

extract contextual information more efficiently from low-resolution images in the semantic branch before fusing the two branches.

Due to the different sampling methods of the compact interactive detail branch and the semantic branch, the two branches have different depths and widths. To better integrate high-level semantic features extracted from semantic branches and spatial details extracted from detail branches, we propose a Dual-Branch Contextual Attention Fusion Module (DBCAF), which is inspired by attention fusion module of AttnNet [49].

For clarity, we plot the detailed structure of the dual-branch contextual attention fusion module in Fig. 5. Assume F_d and F_s respectively denote the output feature maps from the compact detail branch and semantic branch. The input feature maps F_s are the $1/32$ image resolution. The input feature maps F_s is first processed by the pyramid pooling. In the pyramid pooling procedure, we use different large pool-

ing kernels to generate feature maps. The pyramid pooling procedure has four glob-average-pooling operations and the pooling kernels size are 5×5 , 9×9 , 17×17 , and $H \times W$ (“ H ” indicates the height of the input feature and “ W ” indicates the width of the input feature) respectively. Afterwards, the output features are followed by the convolution with kernel size 1×1 and batch normalization. Then, all the four features are processed by upsampling operations to the same resolution size as the input resolution size. Finally, we can obtain the high-level semantics F_{sp} by fusing these four features with one without pooling feature (only have 1×1 convolution operation). The output F_{sp} of the pyramid pooling is input one 3×3 convolution layer and further add with F_s (after adjusting the channel). We get the resulting (F_p) of pyramid pooling. Then upsample the F_p to the same size as F_d and then concatenated with F_d , and then input into the attention layer. The attention layer consists of a convolution layer (kernel size is 1×1 with BN and ReLU activation function), a global average pooling layer (the size of the pooling kernel is the same as the size of the feature map), a convolution layer (kernel size is 1×1 with BN), and a sigmoid activation function. After the attention layer, mask α is obtained, and the attention map is obtained by multiplying α with F_p . and then add this attention map with F_p to get F_p' . Similarly, multiply $(1-\alpha)$ with F_d to get an attention map, and then add this attention map with F_d to get F_d' . Finally, F_p' and F_d' are fused by element-wise addition. This operation is defined as follows:

$$\begin{cases} F_p = \text{Pyramid Pooling}(F_s), \\ F_{\text{output}} = \text{Sum}[\text{Upsample}(F_p \cdot \alpha + F_p), F_d \cdot (1 - \alpha) + F_d], \end{cases} \quad (3)$$

where F_{output} denotes the dual-branch contextual attention fusion module output, *Sum* is the element-wise addition operation, *Upsample* is the standard bilinear interpolation operation, F_s is the output of the semantic branch. F_d is the output of the compact interactive detail branch. F_p is the output of the pyramid pooling. α is the relative attention mask with F_p and F_s .

Experiments

In this section, we test our proposed CIDNet on two datasets: Cityscapes [50] and CamVid [51] and compare it with 20 representative methods. First, we introduce the datasets and implementation details. We then do a comparative test for our backbone network to verify its effectiveness. Next, we investigate the validity of the compact interactive dual-branch structure and the effect of each module of our proposed method on the Cityscapes validation set with the ablation.

Finally, we report our final accuracy and speed (FPS) results on different benchmarks by comparing it with the other algorithms.

Datasets and evaluation metrics

Cityscapes

The Cityscapes [50] Dataset focuses on semantic understanding of urban street scenes, which are road scenes taken from the perspective of a car, and it is one of the most commonly used datasets for segmentation tasks. This dataset provides for a volume of 5000 images with high quality dense pixel annotations and split into training, validation and test sets, with 2975, 500 and 1525 images in our experiments to verify the effectiveness of our method. The annotated images include 30 classes, 19 of which are used for semantic segmentation task, meanwhile the resolution of the images is high to 2048×1024 . Due to the high resolution of this dataset, this poses a great challenge to real-time semantic segmentation methods.

CamVid

Cambridge-driving Labeled Video Database (CamVid) [51] is a road scene segmentation dataset with small-scale. This CamVid dataset is similar to the Cityscapes dataset, but smaller in magnitude and resolution than Cityscapes. There are 701 densely annotated frames extracted from the video sequence, in which 367 images for training, 101 images for validation and 233 images for testing. All images are at a same 960×720 resolution and 32 semantic categories, of which the subset of 11 classes are used for our segmentation experiments. To increase the number of training samples, we merge the training and validation sets for training meanwhile evaluate our method on the test set.

Evaluation metrics

On all dataset we used, we adopt the standard metric of the mean intersection of union (mIoU) and Frames Per Second (FPS) as the evaluation metrics. The mIoU is defined as the ratio of the intersection set and union set of the prediction and the ground truth of the model is summed and then averaged. The FPS is defined as: the number of frames of pictures processed by the model per second. Let i denote the true value, j denote the predicted value, p_{ij} denote the prediction of i as j , mIoU can be expressed as:

$$mIoU = \frac{1}{k+1} \sum_{i=0}^k \frac{p_{ii}}{\sum_{j=0}^k p_{ij} + \sum_{j=0}^k p_{ji} - p_{ii}}. \quad (4)$$

Implementation details

Training settings

We choose the Adam algorithm as optimizer, while utilizing the “polynomial decay” learning rate scheduler and the warm-up strategy. Since we only use a single GPU card, we use different batch sizes when training images of different resolutions. For Cityscapes dataset, resolutions 768×1536 and 512×1024 correspond to batch sizes of 10 and 24, respectively. The max iteration is 140,000, the initial learning rate is 0.005, and we set the warm-up strategy at the first 2000 iterations. For CamVid dataset, the batch size is 24 and the input resolutions is 720×960 . The max iteration is set as 60,000, the initial learning rate is set as 0.01, and we set the warm-up strategy at the first 400 iterations.

For data augmentation, we utilize random scaling, random padding crop, random horizontal flipping, random color jittering and normalization. The random scale ranges in $[0.25, 2.0]$, $[0.5, 2.5]$ for Cityscapes and CamVid respectively. The training cropped resolution of Cityscapes is 512×1024 and 768×1536 , and the training cropped resolution of CamVid is 720×960 . In all training experiments, we conduct our experiments use `palddlepaddle` [52] on NVIDIA GTX 3090 GPU, CUDA 11.2, CUDNN 8.1.

Inference settings

We do not use any speedup tricks or acceleration strategy, e.g., sliding-window evaluation and TensorRT acceleration. For Cityscapes we use a resolution of 768×1536 , 512×1024 for inference. We use the input of 1024×2048 resolution, the time of resizing is included in the inference time measurement. We first resize it to 768×1536 or 512×1024 resolution for inference and then resize the prediction to the original size of the input. For CamVid, the resolution of 960×720 is used for inference. We conduct all inference experiments under CUDA 10.1, CUDNN 7.6 on Tesla V100 GPU. We use standard metric of the mean intersection of union (mIoU) for segmentation accuracy comparison and frames per second (FPS) for inference speed comparison.

Experiments on cityscapes

Ablation study

In this section we perform ablation experiments to demonstrate the performance of our proposed CIDNet versus other network architectures, as well as the effectiveness of each component in our proposed CIDNet. In the following experiments, we train our methods on the Cityscapes [50] training set and evaluate on the Cityscapes validation set.

Table 2 Comparisons with complete dual-branch and compact interactive dual-branch of CIDNet1 and CIDNet2 on Cityscapes val set

Method	Complete	Compact interactive	mIoU (%)	FPS
CIDNet1-50	✓		72.1	152.3
CIDNet1-50		✓	75.1	164.1
CIDNet1-75	✓		73.8	90.6
CIDNet1-75		✓	76.4	104.4
CIDNet2-50	✓		74.3	128.0
CIDNet2-50		✓	75.3	139.0
CIDNet2-75	✓		74.0	78.6
CIDNet2-75		✓	77.7	92.9

Bold indicates the maximum value in the corresponding column

Table 3 Comparisons with different interactive module number of CIDNet1 and different fusion methods on Cityscapes val set

Method	Resolution	Add	DSIM		mIOU (%)
			× 1	× 2	
CIDNet1-50	512 × 1024	✓			74.3
CIDNet1-50	512 × 1024		✓		75.0
CIDNet1-50	512 × 1024			✓	75.1
CIDNet1-75	768 × 1536	✓			76.1
CIDNet1-75	768 × 1536		✓		75.7
CIDNet1-75	768 × 1536			✓	76.4

Notation: Add means element-wise addition fusion method. DSIM means detail-semantic interactive module, in which × 1, × 2 denotes use the DSIM module once or twice

Effectiveness of compact interactive dual-branch

To demonstrate the effectiveness of our compact interactive dual-branch architecture, we also compare it with the complete dual-branch network [21, 22], where we divide the input image into two paths from the beginning, such as BiSeNetV2 [22]. Experiments are performed on the same configuration and platform. Table 2 reports their comparison results.

It can be seen from Table 2 that our compact interactive dual-branch architecture outperforms the complete dual-branch architecture in both speed and performance. So, our compact interactive dual-branch architecture is effective.

Effectiveness of detail-semantic interactive module

To investigate the effectiveness of our proposed detail-semantic interactive module (DSIM), we conduct comparative trials using “Add” fusion and using “interactive” fusion, and the results are shown in Table 3. It can be seen that interactive operation has better segmentation effectiveness, and the performance based on two interactive operations is slightly better than one interactive operation.

Comparisons with state-of-the-arts

In this subsection, we demonstrate the capacity of CIDNet for semantic segmentation by comparing it with the 20 representative models on the Cityscapes dataset. For fair comparison, we evaluate our model with resolution 512 × 1024 and 768 × 1536, respectively. For clarity, we use CIDNet1 and CIDNet2 respectively to denote our proposed CIDNet based on the CID1 and CID2 in Table 1. Furthermore, we denote our proposed CIDNet1 on the two input sizes, 50% and 75% of the original images, respectively as CIDNet1-50 and CIDNet1-75. Similarly, our proposed CIDNet2 on the two input sizes respectively as CIDNet2-50 and CIDNet2-75.

Table 4 reports the comparison results of our proposed CIDNet and the 20 models. It can be observed that from Table 4 that we present the model name, the backbone name, segmentation accuracy, speed (FPS), input resolution, GFLPOs and parameters of various approaches. Specifically, we can observe CIDNet1-50 achieves 164.1 FPS and 75.1 % mIoU, and with the resolution of 768 × 1536, CIDNet2-75 achieves 92.6 FPS and 77.7% mIoU for the validation set. Our method outperforms STDCNet on both GFLOPs and Params, but is slightly inferior to STDCNet in terms of inference speed due to the presence of high-resolution branches. As can be seen from Table 4, many models pre-trained on the ImageNet, which can significantly improve the segmentation accuracy, but it is a very time-consuming process. Our models chose to train from scratch. Moreover, our CIDNet2-75 model is loaded with the training weights of CIDNet1-50. Due to the limitation of GPU memory, the training batch size is limited to a small value, while the training batch size of a small size model (such as 512 × 1024) is larger, and its training gradient is more accurate. Therefore, we load the training weight of CIDNet1-50 into the large size model CIDNet2-75. As shown in Table 4, on Cityscapes val set, our model speed is the fastest among the same performance, so our network achieves the trade-off between speed and performance. At the same input resolution, our model parameters are smaller. For example, at 512 × 1024 resolution, our CIDNet1-50 model parameters are 34.3% less than STDC1-

Table 4 Comparisons with other methods on Cityscapes

Model	Backbone	mIoU (%)		Speed (FPS)	GPU	Resolution	GFLOPs	Params
		val	test					
ENet [53]	No	–	58.3	135.4	TitanX	360 640	3.8	0.4M
ESPNet [54]	ESPNet	–	60.3	112.9	TitanX	512 × 1024	–	0.4M
ESPNetV2 [23]	ESPNetV2	66.4	66.2	–	–	512 × 1024	–	–
ERFNet [55]	No	70.0	68.0	41.7	TitanX M	512 × 1024	27.7	20M
ICNet [44]	PSPNet50	–	69.5	30.3	TitanX M	1024 × 2048	28.3	26.5M
DFANet A [24]	Xception A	–	71.3	100	TitanX	1024 × 1024	3.4	7.8M
DFANet B [24]	Xception B	–	67.1	120	TitanX	1024 × 1024	2.1	4.8M
DFANet A' [24]	Xception A	–	70.3	160	TitanX	512 × 1024	1.7	7.8M
Fast-SCNN [56]	No	68.6	68.0	123.5	TitanXp	1024 × 2048	–	1.1M
SwiftNet [57]	ResNet18	75.5	75.4	39.9	GTX 1080Ti	1024 × 2048	104.0	11.8M
BiSeNetV1 [21]	Xception 39	69.0	68.4	105.8	GTX 1080Ti	786 × 1536	14.8	5.8M
BiSeNetV1 [21]	ResNet18	74.8	74.7	65.5	GTX 1080Ti	786 × 1536	55.3	49M
BiSeNetV2 [†] [22]	No	73.4	72.6	156	GTX 1080Ti	512 × 1024	21.1	–
BiSeNetV2-L [†] [22]	No	75.8	75.3	47.3	GTX 1080Ti	512 × 1024	118.5	–
SFNet [58]	DF1	–	74.5	74	GTX 1080Ti	1024 × 2048	–	9M
SFNet [58]	DF2	–	77.8	53	GTX 1080Ti	1024 × 2048	–	10.5M
STDC1-Seg50 ^{*†} [25]	STDC1	72.2	71.9	206.9	Tesla V100	512 × 1024	24.8	8.3M
STDC2-Seg50 ^{*†} [25]	STDC2	74.2	73.4	156.6	Tesla V100	512 × 1024	38.0	12.3M
STDC1-Seg75 ^{*†} [25]	STDC1	74.5	75.3	140.7	Tesla V100	786 × 1536	55.9	8.3M
STDC2-Seg75 ^{*†} [25]	STDC2	77.0	76.8	106.2	Tesla V100	786 × 1536	85.6	12.3M
CIDNet1-50	No	75.1	73.5	164.1	Tesla V100	512 × 1024	16.3	6.5M
CIDNet2-50	No	75.3	74.4	139.0	Tesla V100	512 × 1024	17.2	7.1M
CIDNet1-75	No	76.4	73.0	104.4	Tesla V100	786 × 1536	36.7	6.5M
CIDNet2-75	No	77.7	75.2	92.6	Tesla V100	786 × 1536	38.6	7.1M

Bold indicates the maximum value in the corresponding column

We train and evaluate our models with 1024 × 2048 resolution input that is resized into 768 × 1536, 512 × 1024 in the model. *no* indicates the method do not have a backbone. Notation: *backbone* indicates the backbone models pre-trained on the ImageNet dataset. * indicates that the inference speeds for models are tested on our platform, and its GFLOPs and params are provided by our platform. The corresponding speed is measured using TensorRT acceleration if the method is marked with †. “–” represents that the methods do not report the corresponding result. The DFANet A and DFANet B are use the optimized depthwise convolutions to increase speed

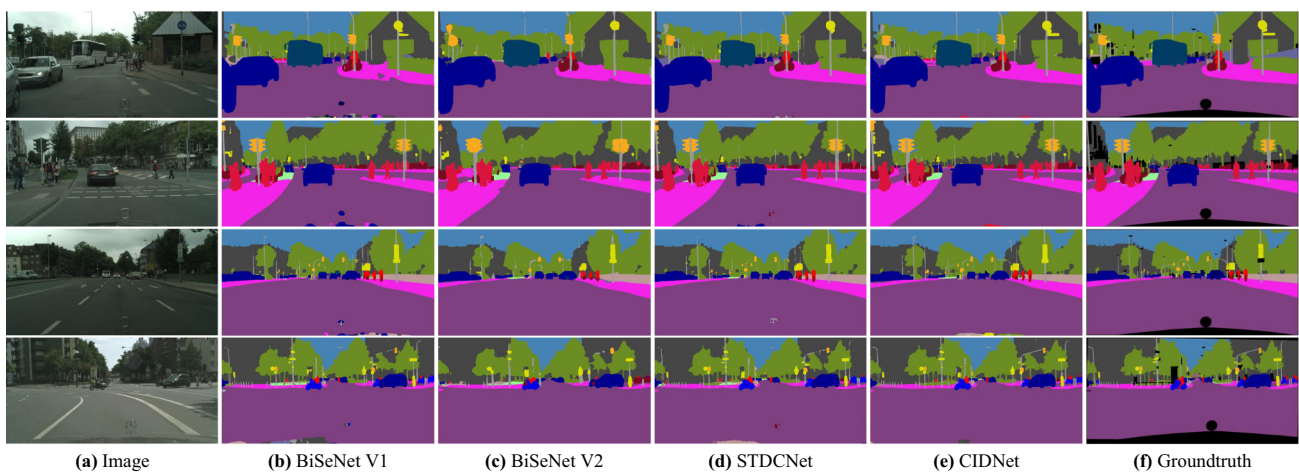


Fig. 6 Example results of the different methods on Cityscapes dataset. The first line is input images, and Lines 2–5 display the results of BiSeNet V1, BiSeNet V2, STDCNet and CIDNet. The final line is the ground-truth

Table 5 Comparisons with state-of-the-art real-time methods on CamVid test set

Model	Backbone	mIoU (%)	Speed (FPS)	GPU
ENet [53]	No	51.3	61.2	TitanX
ICNet [44]	PSPNet50	67.1	27.8	TitanX
DFANet A [24]	Xception A	64.7	120	TitanX
DFANet B [24]	Xception B	59.3	160	TitanX
SwiftNet [57]	ResNet18	72.6	–	GTX 1080Ti
BiSeNetV1 [21]	Xception 39	65.6	175	GTX 1080Ti
BiSeNetV1 [21]	ResNet18	68.7	116.3	GTX 1080Ti
BiSeNetV2 [22]	No	72.4	124.5	GTX 1080Ti
BiSeNetV2-L [22]	No	73.2	32.7	GTX 1080Ti
BiSeNetV2* [†] [22]	No	76.7	124.5	GTX 1080Ti
BiSeNetV2-L* [†] [22]	No	78.5	32.7	GTX 1080Ti
STDC1-Seg [†] [25]	STDC1	73.0	197.6	GTX 1080Ti
STDC2-Seg [†] [25]	STDC2	73.9	152.2	GTX 1080Ti
CIDNet1	No	71.3	130.7	Tesla V100
CIDNet2	No	71.5	109.6	Tesla V100
CIDNet1*	No	76.3	130.7	Tesla V100
CIDNet2*	No	77.8	109.6	Tesla V100

With 720×960 input, we evaluate the segmentation accuracy and corresponding inference speed. Notation: *backbone* means that the backbone models pre-trained on the additional datasets, e.g., the ImageNet dataset and the Cityscapes dataset. *no* indicates the method do not have a backbone. * indicates that the models are pre-trained on Cityscapes. The corresponding speed is measured using TensorRT acceleration if the method is marked with [†]

Seg50, and at 1536×768 resolution, our CIDNet1-75 model parameters are 73.2% less than STDC1-Seg75. In order to further demonstrate the superiority of our proposed method, we show the visualization results of several different methods on the cityscapes dataset in Fig. 6. From these figures, it can be seen that our method is the closest to the ground truth, that is, our method is superior to the three comparable methods.

Experiments on CamVid

We also conduct experiments on the CamVid dataset to further demonstrate the performance of CIDNet. Like other works, the input resolution for training and inference is 720×960 . Table 5 shows the comparison results with other methods.

In Table 5 CIDNet1 and CIDNet2 respectively to denote our proposed CIDNet based on the CID1 and CID2 in Table 1. In Table 5 CIDNet2* achieves the best performance, which is 77.8% mIoU with 109.6 FPS. This further demonstrates the superior performance of our method. ‘*’ means the model is loaded with the training weights of CIDNet2-75 (training in cityscapes dataset).

Conclusion

In this paper, we propose an efficient compact interactive dual-branch network (CIDNet) for real-time semantic segmentation. Considering the complete dual-branch network is time-consuming, we first refine the traditional dual-branch network, and construct a compact detail branch and semantic branch. To prevent the loss of information, we propose a detail-semantic interactive fusion module. Finally, we construct a dual-branch contextual attention fusion module to deeply fuse the extracted features, and further predict the final segmentation result. Experimental results on the Cityscapes and CamVid datasets demonstrate that the proposed CIDNet achieves satisfactory trade-off between segmentation accuracy and inference speed, and outperforms 16 representative real-time semantic segmentation methods. Our network also has shortcomings: how to make the spatial detail branch occupy less computational resources, and how to improve the fusion efficiency of the two branches. This will be further improved in future work. In practical applications, we are faced with marginal devices, mobile terminals and other devices with relatively weak computing power, which requires our algorithm to be light enough and the amount of

computation should be as small as possible. Therefore, in the future work, we will try to deploy our algorithm on mobile terminals to improve the lightweight and efficient semantic segmentation algorithm.

Acknowledgements This work was supported in part by the National Natural Science Foundation of China under Grant 62071171, and in part by Natural Science Foundation of Henan under Grant 232300421023.

Data Availability The datasets that support the findings of this study are available in Cityscapes Dataset at <https://www.cityscapes-dataset.com/>, and CamVid Dataset at <http://mi.eng.cam.ac.uk/research/projects/VideoRec/CamVid/>.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Tao H, Qiu J, Chen Y, Stojanovic V, Cheng L (2023) Unsupervised cross-domain rolling bearing fault diagnosis based on time-frequency information fusion. *J Frankl Inst* 360(2):1454–1477
- Dong Y, Jiang Z, Tao F, Fu Z (2022) Multiple spatial residual network for object detection. *Complex Intell Syst* 2022:1–16
- Dong Y, Shen L, Pei Y, Yang H, Li X (2023) Field-matching attention network for object detection. *Neurocomputing* 535:123–133
- Dong Y, Tan W, Tao D, Zheng L, Li X (2021) Cartoonlossgan: learning surface and coloring of images for cartoonization. *IEEE Trans Image Process* 31:485–498
- Zhuang Z, Tao H, Chen Y, Stojanovic V, Paszke W (2022) An optimal iterative learning control approach for linear systems with nonuniform trial lengths under input constraints. *IEEE Trans Syst Man Cybern Syst* 2022:1
- Xu Z, Li X, Stojanovic V (2021) Exponential stability of nonlinear state-dependent delayed impulsive systems with applications. *Nonlinear Anal Hybrid Syst* 42:101088
- Azuma RT (1997) A survey of augmented reality. *Presence Teleoper Virtual Environ* 6(4):355–385
- Li D, Shi G, Wu Y, Yang Y, Zhao M (2020) Multi-scale neighborhood feature extraction and aggregation for point cloud segmentation. *IEEE Trans Circuits Syst Video Technol* 31(6):2175–2191
- Siam M, Gamal M, Abdel-Razek M, Yogamani S, Jagersand M, Zhang H (2018) A comparative study of real-time semantic segmentation for autonomous driving. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp 587–597
- Wang L, Wu J, Liu X, Ma X, Cheng J (2022) Semantic segmentation of large-scale point clouds based on dilated nearest neighbors graph. *Complex Intell Syst* 8(5):3833–3845
- You H, Yu L, Tian S, Cai W (2021) Dr-net: dual-rotation network with feature map enhancement for medical image segmentation. *Complex Intell Syst* 2021:1–13
- Amin J, Sharif M, Gul E, Nayak RS (2021) 3D-semantic segmentation and classification of stomach infections using uncertainty aware deep neural networks. *Complex Intell Syst* 2021:1–17
- Dechesne C, Mallet C, Le Bris A, Gouet-Brunet V (2017) Semantic segmentation of forest stands of pure species combining airborne lidar data and very high resolution multispectral imagery. *ISPRS J Photogramm Remote Sens* 126:129–145
- Zhuang J, Wang Z, Wang B (2020) Video semantic segmentation with distortion-aware feature correction. *IEEE Trans Circuits Syst Video Technol* 31(8):3128–3139
- Tan Z, Liu B, Chu Q, Zhong H, Wu Y, Li W, Yu N (2020) Real time video object segmentation in compressed domain. *IEEE Trans Circuits Syst Video Technol* 31(1):175–188
- Dong Y, Zhao K, Zheng L, Yang H, Liu Q, Pei Y (2023) Refinement co-supervision network for real-time semantic segmentation. *IET Comput Vis* 31:1–11
- Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pp 3431–3440
- Ji J, Shi R, Li S, Chen P, Miao Q (2020) Encoder–decoder with cascaded CRFs for semantic segmentation. *IEEE Trans Circuits Syst Video Technol* 31(5):1926–1938
- Chen L-C, Papandreou G, Schroff F, Adam H (2017) Rethinking atrous convolution for semantic image segmentation. Preprint [arXiv:1706.05587](https://arxiv.org/abs/1706.05587)
- Zhang H, Dana K, Shi J, Zhang Z, Wang X, Tyagi A, Agrawal A (2018) Context encoding for semantic segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 7151–7160
- Yu C, Wang J, Peng C, Gao C, Yu G, Sang N (2018) Bisenet: bilateral segmentation network for real-time semantic segmentation. In: *Proceedings of the European conference on computer vision (ECCV)*, pp 325–341
- Yu C, Gao C, Wang J, Yu G, Shen C, Sang N (2021) Bisenet v2: bilateral network with guided aggregation for real-time semantic segmentation. *Int J Comput Vis* 129(11):3051–3068
- Mehta S, Rastegari M, Shapiro L, Hajishirzi H (2019) Espnetv2: a light-weight, power efficient, and general purpose convolutional neural network. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp 9190–9200
- Li H, Xiong P, Fan H, Sun J (2019) Dfanet: deep feature aggregation for real-time semantic segmentation. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp 9522–9531
- Fan M, Lai S, Huang J, Wei X, Chai Z, Luo J, Wei X (2021) Rethinking bisenet for real-time semantic segmentation. In: *2021 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp 9711–9720
- Chollet F (2017) Xception: deep learning with depthwise separable convolutions. In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pp 1251–1258
- Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H (2017) Mobilenets: efficient convolutional neural networks for mobile vision applications. Preprint [arXiv:1704.04861](https://arxiv.org/abs/1704.04861)
- Zhang X, Zhou X, Lin M, Sun J (2018) Shufflenet: an extremely efficient convolutional neural network for mobile devices. In: *Pro-*

- ceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 6848–6856
29. Otsu N (1979) A threshold selection method from gray-level histograms. *IEEE Trans Syst Man Cybern* 9(1):62–66
 30. Vincent L, Soille P (1991) Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Trans Pattern Anal Mach Intell* 13(06):583–598
 31. Ren X, Malik J (2003) Learning a classification model for segmentation. In: *IEEE International conference on computer vision*, vol 2, pp 10–10
 32. Achanta R, Shaji A, Smith K, Lucchi A, Fua P, Süsstrunk S (2012) Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Trans Pattern Anal Mach Intell* 34(11):2274–2282
 33. Bergh MVD, Boix X, Roig G, Capitani BD, Gool LV (2012) Seeds: superpixels extracted via energy-driven sampling. In: *European conference on computer vision*, pp 13–26
 34. Boykov YY, Jolly M-P (2001) Interactive graph cuts for optimal boundary and region segmentation of objects in nd images. In: *Proceedings 8th IEEE international conference on computer vision. ICCV 2001*, vol 1, pp 105–112
 35. Rother C, Kolmogorov V, Blake A (2004) “Grabcut” interactive foreground extraction using iterated graph cuts. *ACM Trans Graph (TOG)* 23(3):309–314
 36. Liu Q, Dong Y, Li X (2023) Multi-stage context refinement network for semantic segmentation. *Neurocomputing* 535:53–63
 37. Krähenbühl P, Koltun V (2011) Efficient inference in fully connected CRFs with gaussian edge potentials. *Adv Neural Inf Process Syst* 24:1
 38. Chen L-C, Zhu Y, Papandreou G, Schroff F, Adam H (2018) Encoder–decoder with atrous separable convolution for semantic image segmentation. In: *Proceedings of the European conference on computer vision (ECCV)*, pp 801–818
 39. Badrinarayanan V, Kendall A, Cipolla R (2017) Segnet: a deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans Pattern Anal Mach Intell* 39(12):2481–2495
 40. Zhao H, Shi J, Qi X, Wang X, Jia J (2017) Pyramid scene parsing network. In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pp 2881–2890
 41. Yuan Y, Huang L, Guo J, Zhang C, Chen X, Wang J (2018) Ocnet: object context network for scene parsing. Preprint [arXiv:1809.00916](https://arxiv.org/abs/1809.00916)
 42. Zhao H, Zhang Y, Liu S, Shi J, Loy CC, Lin D, Jia J (2018) Psanet: point-wise spatial attention network for scene parsing. In: *Proceedings of the European conference on computer vision (ECCV)*, pp 267–283
 43. Orsic M, Kreso I, Bevandic P, Segvic S (2019) In defense of pre-trained imagenet architectures for real-time semantic segmentation of road-driving images. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp 12607–12616
 44. Zhao H, Qi X, Shen X, Shi J, Jia J (2018) Icnnet for real-time semantic segmentation on high-resolution images. In: *Proceedings of the European conference on computer vision (ECCV)*, pp 405–420
 45. Hong Y, Pan H, Sun W, Jia Y (2021) Deep dual-resolution networks for real-time and accurate semantic segmentation of road scenes. Preprint [arXiv:2101.06085](https://arxiv.org/abs/2101.06085)
 46. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen L-C (2018) Mobilenetv2: inverted residuals and linear bottlenecks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pp 4510–4520
 47. Howard A, Sandler M, Chu G, Chen L-C, Chen B, Tan M, Wang W, Zhu Y, Pang R, Vasudevan V, et al (2019) Searching for mobilenetv3. In: *Proceedings of the IEEE/CVF international conference on computer vision*, pp 1314–1324
 48. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pp 770–778
 49. Song Q, Mei K, Huang R (2021) Attanet: attention-augmented network for fast and accurate scene parsing. In: *Proceedings of the AAAI conference on artificial intelligence*, vol 35, pp 2567–2575
 50. Cordts M, Omran M, Ramos S, Rehfeld T, Enzweiler M, Benenson R, Franke U, Roth S, Schiele B (2016) The cityscapes dataset for semantic urban scene understanding. In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pp 3213–3223
 51. Brostow GJ, Shotton J, Fauqueur J, Cipolla R (2008) Segmentation and recognition using structure from motion point clouds. In: *European conference on computer vision*, pp 44–57
 52. Ma Y, Yu D, Wu T, Wang H (2019) PaddlePaddle: an open-source deep learning platform from industrial practice. *Front Data Comput* 1(1):105–115
 53. Paszke A, Chaurasia A, Kim S, Culurciello E (2016) Enet: a deep neural network architecture for real-time semantic segmentation. Preprint [arXiv:1606.02147](https://arxiv.org/abs/1606.02147)
 54. Mehta S, Rastegari M, Caspi A, Shapiro L, Hajishirzi H (2018) Espnet: efficient spatial pyramid of dilated convolutions for semantic segmentation. In: *Proceedings of the European conference on computer vision (ECCV)*, pp 552–568
 55. Romera E, Alvarez JM, Bergasa LM, Arroyo R (2017) Erfnet: efficient residual factorized convnet for real-time semantic segmentation. *IEEE Trans Intell Transp Syst* 19(1):263–272
 56. Poudel RP, Liwicki S, Cipolla R (2019) Fast-scnn: fast semantic segmentation network. Preprint [arXiv:1902.04502](https://arxiv.org/abs/1902.04502)
 57. Oršic M, Krešo I, Bevandic P, Šegvic S (2019) In defense of pre-trained imagenet architectures for real-time semantic segmentation of road-driving images. In: *2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp 12599–12608
 58. Li X, You A, Zhu Z, Zhao H, Yang M, Yang K, Tan S, Tong Y (2020) Semantic flow for fast and accurate scene parsing. In: *European conference on computer vision*, pp 775–793

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.