Deep Multi-Branch Aggregation Network for Real-Time Semantic Segmentation in Street Scenes

Xi Weng, Yan Yan, *Member, IEEE,* Genshun Dong, Chang Shu, Biao Wang,
Hanzi Wang, *Senior Member, IEEE*, and Ji Zhang, *Senior Member, IEEE*

***Abstract*—Real-time semantic segmentation, which aims to achieve high segmentation accuracy at real-time inference speed, has received substantial attention over the past few years. However, many state-of-the-art real-time semantic segmentation methods tend to sacrifice some spatial details or contextual information for fast inference, thus leading to degradation in segmentation quality. In this paper, we propose a novel Deep Multi-branch Aggregation Network (called DMA-Net) based on the encoder-decoder structure to perform real-time semantic segmentation in street scenes. Specifically, we first adopt ResNet-18 as the encoder to efficiently generate various levels of feature maps from different stages of convolutions. Then, we develop a Multi-branch Aggregation Network (MAN) as the decoder to effectively aggregate different levels of feature maps and capture the multi-scale information. In MAN, a lattice enhanced residual block is designed to enhance feature representations of the network by taking advantage of the lattice structure. Meanwhile, a feature transformation block is introduced to explicitly transform the feature map from the neighboring branch before feature aggregation. Moreover, a global context block is used to exploit the global contextual information. These key components are tightly combined and jointly optimized in a unified network. Extensive experimental results on the challenging Cityscapes and CamVid datasets demonstrate that our proposed DMA-Net respectively obtains 77.0% and 73.6% mean Intersection over Union (mIoU) at the inference speed of 46.7 FPS and 119.8 FPS by only using a single NVIDIA GTX 1080Ti GPU. This shows that DMA-Net provides a good tradeoff between segmentation quality and speed for semantic segmentation in street scenes.**

***Index Terms*—Deep learning, real-time semantic segmentation, lightweight convolutional neural networks, multi-branch aggregation.**

I. Introduction

SEMANTIC segmentation, which predicts the semantic label of each pixel in an image, is a fundamental and challenging task in street scene understanding. During the past few decades, semantic segmentation in street scenes has attracted increasing attention, mainly due to its important role in autonomous driving systems [1]–[4]. Generally, these systems demand fast inference speed for interaction and response.

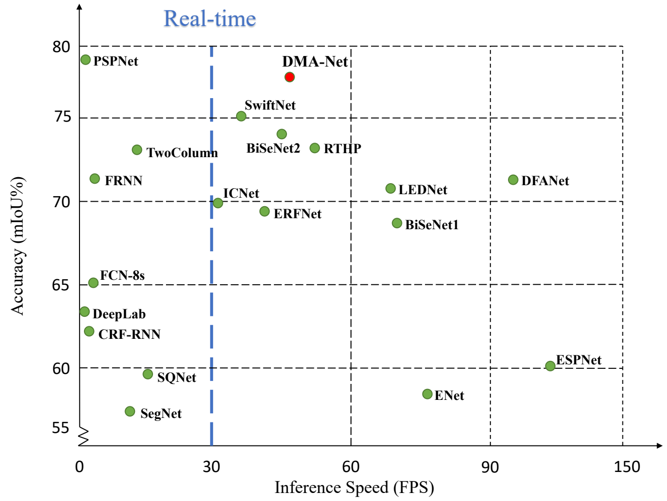


Fig. 1. Accuracy (mIoU) and inference speed (FPS) obtained by several state-of-the-art semantic segmentation methods, including SwiftNet [2], PSPNet [7], ENet [13], ERFNet [14], BiSeNet [15], ICNet [16], LEDNet [17], RTHP [18], DFANet [19], ESPNet [20], FCN-8s [24], DeepLab [25], CRF-RNN [26], SegNet [27], SQNet [28], FRRN [29], TwoColumn [30], and the proposed DMA-Net on the Cityscapes test set.

Street scene images are often captured by a surveillance camera mounted behind the windshield of a driving car. Generally, images in street scene datasets (such as Cityscapes [5] and CamVid [6]) contain different kinds of objects (e.g., road, car, and building). Compared with the objects in natural scenes, some objects in street scenes are visually similar (such as building vs. wall, and truck vs. bus). How to distinguish similar objects is of great importance for street scene understanding and plays a critical role in achieving good segmentation accuracy.

Benefiting from the outstanding performance of Deep Convolutional Neural Network (DCNN), a large number of semantic segmentation methods [7]–[10] have been proposed and shown significant performance improvements in terms of segmentation accuracy, especially for distinguishing similar objects in street scenes. The success of the above methods relies largely on sophisticated DCNN models (such as Xception [11] and ResNet-101 [12]) as the backbone networks to capture low-level spatial details and high-level semantics.

This work was partly supported by the National Natural Science Foundation of China under Grants 62071404, U21A20514, and 61872307, by the Open Research Projects of Zhejiang Lab under Grant 2021KG0AB02, by the Natural Science Foundation of Fujian Province under Grant 2020J01001, and by the Youth Innovation Foundation of Xiamen City under Grant 3502Z20206046. *(Corresponding authors: Yan Yan; Biao Wang.)*

X. Weng, Y. Yan, G. Dong, and H. Wang are with the Fujian Key Laboratory of Sensing and Computing for Smart City, School of Informatics, Xiamen University, China (e-mail: xweng@stu.xmu.edu.cn; yanyan@xmu.edu.cn; gshdong@qq.com; hanzi.wang@xmu.edu.cn).

C. Shu is with the School of Information and Communication Engineering, University of Electronic Science and Technology of China, China (e-mail: changshu@uestc.edu.cn).

B. Wang is with Zhejiang Lab, China (e-mail: wangbiao@zhejianglab.com).

J. Zhang is with University of Southern Queensland, Australia & Zhejiang Lab, China (e-mail: zhangji77@gmail.com).

Unfortunately, these DCNN models usually involve heavy computational operations and high memory consumption. As a consequence, although remarkable progress has been made by these methods, their high computational costs and memory requirements inhibit the deployment of semantic segmentation in many real-world applications with limited power resources (such as self-driving cars and driver assistance systems).

To achieve fast inference speed, a variety of real-time semantic segmentation methods [13]–[20] have been developed by leveraging lightweight networks (such as MobileNetV2 [21] and ShuffleNet [22]) as the backbone networks. However, the feature extraction capability of lightweight networks is often inferior, and thus these networks are difficult to extract feature maps with rich spatial and contextual information for pixel-level classification. Therefore, the accuracy of these methods in segmenting similar objects in street scenes is greatly affected. Fig. 1 shows the accuracy (mean Intersection over Union (mIoU)) and inference speed (Frames Per Second (FPS)) obtained by several state-of-the-art semantic segmentation methods on the Cityscapes test set. Obviously, different from the rapid development of high-quality semantic segmentation methods, research towards real-time semantic segmentation in street scenes without reducing too much accuracy is still left behind.

Recently, some methods, such as Bilateral Segmentation Network (BiSeNet) [15] and Deep Feature Aggregation Network (DFANet) [19], have been developed in pursuit of high segmentation accuracy at real-time inference speed. BiSeNet employs a two-branch DCNN model to combine the spatial and semantic information. Nevertheless, the lack of communication between two branches may weaken the learning capacity of the model. DFANet makes use of deep feature aggregation to address real-time semantic segmentation on high-resolution images, where the feature maps are concatenated at both the network-level and the stage-level. However, simple aggregation operations (such as the element-wise addition and the channel-wise concatenation) [18], [19] are not optimal since the feature maps from the encoder have a gap. These operations may cause feature interference, and thus the decoder cannot faithfully pay attention to objects at different scales, leading to a performance decrease. This issue is more pronounced in street scenes, which usually cover different scales of objects.

In the light of the above issues, we propose a novel Deep Multi-branch Aggregation Network, called DMA-Net, based on the encoder-decoder structure for real-time semantic segmentation in street scenes. Specifically, we adopt a lightweight network (i.e., ResNet-18 [12]) as the encoder and develop a Multi-branch Aggregation Network (MAN) as the decoder. In MAN, a Lattice Enhanced Residual Block (LERB) consisting of two lattice structures is designed to combine the spatial and contextual enhanced blocks in each branch of MAN. In particular, we leverage two weight learning blocks to adjust the weights of two lattice structures adaptively. Meanwhile, a Feature Transformation Block (FTB), which emphasizes the important information while ignoring the irrelevant information in the feature maps, is introduced to explicitly transform the feature map from the neighboring branch before feature aggregation. Moreover, a Global Context Block (GCB) is employed to capture the rich global contextual information, which is critical for semantic segmentation.

In summary, our main contributions of this paper are summarized as follows:

- We develop LERB to effectively enhance both spatial details and contextual information of feature maps from the encoder. In particular, the lattice structures in LERB allow the potential of various combinations of enhanced blocks, greatly enlarging the representation space of LERB in an efficient manner. Therefore, the problem of inferior feature extraction capability of the lightweight backbone network is significantly solved, improving the performance of segmenting similar objects.
- We propose FTB to generate the transformed feature maps based on a transformation tensor at a relatively small computational cost. In this way, the gap between different levels of feature maps is largely mitigated. As a result, the problem of feature interference between high-level and low-level feature maps is alleviated, and these feature maps can be appropriately aggregated.
- The key components (i.e., ResNet-18, LERB, FTB, and GCB) are tightly combined and jointly optimized in DMA-Net to achieve real-time semantic segmentation in street scenes. Our proposed DMA-Net obtains 77.0% and 73.6% mIoU on the challenging Cityscapes and CamVid test datasets at the speed of 46.7 FPS and 119.8 FPS, respectively (only a single NVIDIA GTX 1080Ti GPU is used). These results demonstrate that our proposed DMA-Net is able to make a good tradeoff between accuracy and speed for semantic segmentation in street scenes.

The rest of this paper is organized as follows. First, we review the related work in Section II. Then, we describe the proposed DMA-Net in detail in Section III. Next, we give ablation studies and show experimental results on two challenging street scene semantic segmentation datasets in Section IV. Finally, we draw our conclusion in Section V.

II. Related Work

DCNN has made great success in various computer vision tasks, since its outstanding achievement on the large-scale image classification task [31]. In recent years, a series of DCNN-based semantic segmentation methods have been developed and achieved excellent performance on the benchmark datasets. In this section, we briefly review some state-of-the-art semantic segmentation methods, including high-quality methods and real-time ones.

A. High-Quality Semantic Segmentation Methods

Fully Convolutional Network (FCN) [24] is the pioneering semantic segmentation method. FCN replaces the fully-connected layers of the classification networks with the convolutional layers, and it forms the foundation of modern semantic segmentation methods. To generate dense feature maps, FCN makes use of skip connections to combine the coarse and fine feature maps. Ronneberger *et al.* [32] propose a U-shape Network (U-Net), which consists of an encoder and a decoder. The encoder gradually increases the receptive fields to capture the contextual information, while the decoder recovers the spatial information from the outputs of the encoder in a layer-by-layer manner. DeepLab [25] introduces the atrous convolution [33] to enlarge the receptive fields of the network without increasing the number of parameters. DeepLabv3+ [10] also adopts the encoder-decoder structure, where a network similar to DeepLabv3 [34] is employed to encode the contextual information while a simple decoder is leveraged to refine the segmentation accuracy (especially near the object boundaries). Multi-path Refinement Network (RefineNet) [8] refines high-level feature maps by using fine-grained low-level feature maps based on a generic multi-path framework.

The existence of objects at multiple scales in street scenes raises a great challenge in semantic segmentation. To address this challenge, a standard way is to perform segmentation on multiple re-scaled versions of the same input image and then aggregate the output feature maps. Although such a way can boost the segmentation accuracy, it usually significantly increases the computational burden [35]. DeepLabv2 [9] develops an Atrous Spatial Pyramid Pooling (ASPP) module to robustly segment multi-scale objects. ASPP extracts feature maps in multiple parallel atrous convolution branches with different sampling rates, thus capturing objects and contextual information at different scales. Similarly, Pyramid Scene Parsing Network (PSPNet) [7] aggregates the contextual information from different regions based on a pyramid network structure. Context Encoding Network (EncNet) [36] exploits the global contextual information through a context encoding module to enlarge the receptive fields and segment multi-scale objects. To refine the outputs, some methods [25] [26] also employ the probabilistic graphical model, such as Conditional Random Fields (CRF) [37], as a post-processing step to improve the segmentation accuracy of object boundaries.

Recently, the self-attention mechanism has been adopted in several state-of-the-art methods. Dual Attention Network (DANet) [38] develops a position and channel attention module to improve the segmentation accuracy by adaptively capturing and aggregating the contextual information. Expectation-Maximization Attention Network (EMANet) [39] computes a compact basis set to reduce the computational complexity of semantic segmentation by using an expectation-maximization iteration manner.

The aforementioned methods show high segmentation accuracy on various benchmark datasets. Many methods (such as RefineNet [8] and U-Net [32]) adopt the encoder-decoder structure. Unfortunately, they generally suffer from heavy computational costs and long inference time, due to the large number of network parameters or the large-scale floating-point operations, or both. In this paper, DMA-Net is also based on the encoder-decoder structure. However, compared with symmetric encoder-decoder structures used in U-Net and RefineNet, DMA-Net is much more lightweight and specifically designed for real-time semantic segmentation in street scenes.

B. Real-Time Semantic Segmentation Methods

Real-time semantic segmentation methods aim to generate high-quality prediction at fast inference speed (e.g., more than 30 FPS). Segmentation Neural Network (SegNet) [27] is the early real-time semantic segmentation method, which removes the fully-connected layers in the network to obtain a small architecture and utilizes the max pooling indices to upsample the feature maps. Efficient Neural Network (ENet) [13] designs a compact encoder-decoder structure, where early downsampling is employed to make it suitable for the low-latency semantic segmentation task. However, ENet cannot robustly segment large objects due to the relatively small receptive fields used in the compact architecture. Efficient Spatial Pyramid Network (ESPNet) [20] proposes an efficient spatial pyramid module, where the standard convolution is decomposed into point-wise convolutions and a spatial pyramid of dilated convolutions. Hence, the computational complexity of the model is reduced. Similarly, Efficient Residual Factorized Network (ERFNet) [14] designs a novel convolutional layer, which utilizes residual connections and factorized convolutions to efficiently perform semantic segmentation. The above methods usually compromise spatial details or contextual information to achieve fast inference speed. Such a manner leads to poor segmentation results. Therefore, compared with high-quality semantic segmentation methods, the segmentation accuracy of these methods is still far from being satisfactory.

Recently, the multi-branch framework has drawn much interest. For example, Zhao *et al.* [16] propose an Image Cascade Network (ICNet) based on the simplified version of PSPNet and cascade networks. ICNet combines the semantic information from low-resolution images and the detailed spatial information from high-resolution images. BiSeNet [15] adopts a two-branch DCNN structure to respectively encode the spatial and semantic information, so as to improve both the inference speed and segmentation accuracy. Note that BiSeNet explores the spatial details and the semantic information separately. The lack of communication between branches may influence the learning ability of the DCNN model. To address the above problem, DFANet [19] employs a feature reuse strategy to make a balanced tradeoff between accuracy and speed. However, DFANet aggregates feature maps at the different levels by a simple network structure, thereby ignoring the differences between them.

In this paper, the proposed DMA-Net also takes advantage of the multi-branch framework. However, different from the above methods, DMA-Net progressively aggregates the feature maps from the high-level branch to the low-level branch based on an elaborately-designed lightweight decoder MAN (mainly consisting of LERB, FTB, and GCB). Therefore, DMA-Net is able to effectively and efficiently segment objects in complex street scenes. Moreover, DMA-Net makes use of different levels of feature maps from different stages of ResNet-18 as the inputs for multiple branches in MAN, where a principal loss or an auxiliary loss is specifically employed to supervise the output of each branch. As a result, each branch focuses on capturing the semantic information at a certain scale.

III. The Proposed Method

In this section, we present the proposed DMA-Net in detail. We first give an overview of DMA-Net in Section III-A. Then,

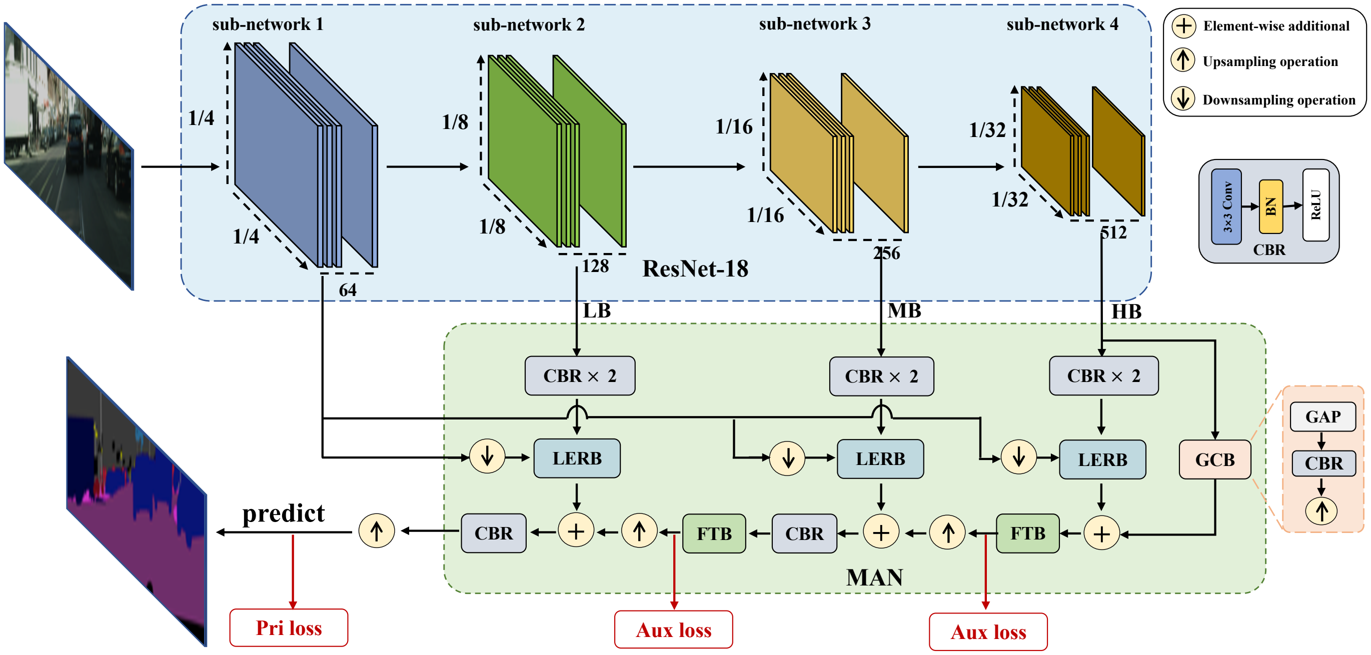


Fig. 2. The overall framework of the proposed DMA-Net. ResNet-18 extracts different levels of feature maps. MAN aggregates the feature maps from ResNet-18 to generate the final prediction. In the figure, CBR denotes the Conv-BN-ReLU module. "GAP" denotes the global average pooling operation. "Pri loss" and "Aux loss" represent the principal loss and the auxiliary loss, respectively.

we introduce each component of DMA-Net from Section III-B to Section III-C. Next, we give the joint loss in Section III-D. Finally, we present some discussions about our DMA-Net in Section III-E.

A. *Overview*

DMA-Net consists of two main parts: ResNet-18 and a Multi-branch Aggregation Network (MAN). In particular, a Lattice Enhanced Residual Block (LERB), a Feature Transformation Block (FTB), and a Global Context Block (GCB) are developed in MAN.

The overall framework of DMA-Net is illustrated in Fig. 2. An image $\mathbf{I} \in \mathbb{R}^{H\times W\times C}$ is taken as the input of ResNet-18, where H, W, and C represent the height, the width, and the number of channels of the image $\mathbf{I}$, respectively. First, ResNet-18 efficiently downsamples the input image by several consecutive convolutional blocks to generate different levels of feature maps. Then, as the core of DMA-Net, MAN takes the feature maps from different stages of ResNet-18 as the inputs, and progressively performs feature aggregation from the high-level branch to the low-level branch. In MAN, LERB effectively enhances feature representations of the network, while FTB greatly reduces the semantic gap between feature maps before feature aggregation. In addition, instead of relying on multi-scale input images or a specifically-designed multi-scale module, MAN not only exploits the multi-scale information by recursively aggregating feature maps from the high-level branch to the low-level branch, but also explicitly adopts both the principal loss and the auxiliary losses.

DMA-Net is a lightweight encoder-decoder network. On the one hand, we employ ResNet-18, which is much simpler than complex DCNN models (such as ResNet-101 and Xception) used in high-quality semantic segmentation methods, as the encoder to ensure high inference speed. On the other hand, we develop MAN as the decoder with a small amount of network parameters to efficiently and effectively combine spatial details and contextual information. Therefore, DMA-Net can achieve a good balance between accuracy and inference speed.

B. *ResNet-18*

An encoder plays a critical role in basic feature extraction of the input images. In this paper, we adopt ResNet-18 (pre-trained with ImageNet [40]) as our encoder. The input images are downsampled by using a max-pooling layer at the earlier layer of ResNet-18. Moreover, ResNet-18 is composed of a small number of layers in the network. Therefore, ResNet-18 has the distinct advantage of high efficiency with fast speed and small memory consumption.

Specifically, we remove all the network layers (including the pooling layers and the fully-connected layers, etc.) after the last residual building block of ResNet-18 to obtain a simplified version of ResNet-18. Hence, the network architecture of the simplified version of ResNet-18 consists of a standard 7×7 convolutional layer, a 3×3 max-pooling layer, and eight 3×3 residual building blocks. The eight residual building blocks can be divided into four sub-networks (i.e., sub-network 1 to sub-network 4), according to the size of the output feature maps, as shown in Fig. 2. Generally, the size of the output feature maps is reduced to one half after passing through each sub-network. Therefore, we can obtain four different levels of feature maps (whose sizes correspond to $1/4$, $1/8$, $1/16$, and $1/32$ of the original image size) from four sub-networks.

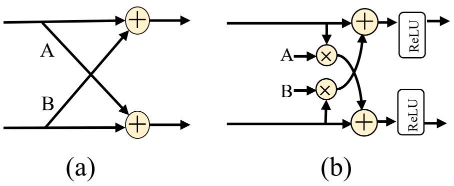


Fig. 3. The network architecture of (a) a standard lattice filter with 2-channel filters, and (b) our lattice structure.

C. Multi-branch Aggregation Network (MAN)

Compared with complex DCNN models, the feature extraction capability of ResNet-18 is inferior. To achieve a good balance between segmentation accuracy and inference speed, we resort to an elaborately-designed decoder MAN to aggregate different levels of feature maps for semantic segmentation in street scenes. Thus, the spatial and contextual information can be effectively combined in the decoder. In particular, multiple Conv-BN-ReLU modules (each Conv-BN-ReLU module includes a 3×3 convolutional layer followed by a Batch Normalization (BN) layer and a ReLU activation function) are used in MAN to reduce the number of feature channels. Such a way ensures the small computational cost of MAN.

The network architecture of MAN is shown in Fig. 2. We can see that, the four different levels of feature maps from four sub-networks of ResNet-18 are used as the inputs of MAN. To be specific, the feature maps, whose sizes are $1/8$, $1/16$, and $1/32$ of the original input image size, are respectively fed into three branches, including a Low-level Branch (LB), a Mid-level Branch (MB), and a High-level Branch (HB). For each branch, we first employ two Conv-BN-ReLU modules to reduce the dimension of the feature map. Then, LERB is designed to improve the feature representations, given the feature maps from the Conv-BN-ReLU module and the first sub-network of ResNet-18. Finally, the output feature map of LERB is combined with the transformed feature map based on FTB from the neighboring branch. Note that, in HB, the last downsampled feature maps obtained from ResNet-18 are also fed into GCB to model the global contextual dependency, which can provide the rich high-level contextual information for MAN. The outputs of these branches are progressively aggregated to obtain the final predicted results.

It is worth pointing out that our proposed MAN is able to effectively and efficiently capture the multi-scale information. As it is well known [18], [34], [35], one problem in the application of DCNN to semantic segmentation is the difficulty of using a single scale to perform pixel-level dense prediction, because of the existence of objects at multiple scales. Hence, how to accurately capture the multi-scale object information while maintaining fast inference speed of the network is a great challenge. Traditional methods either rely on multiple re-scaled versions of the input images [35] or use an additional multi-scale module (such as ASPP [34] or DASPP [18]) to tackle the multi-scale problem. However, such manners [18], [34] usually bring additional consumption in terms of both computational complexity and memory requirement.

Different from the above methods, our proposed MAN recursively aggregates the multi-level information from the different branches to obtain the segmentation results. In MAN, each branch tackles the feature map at a certain size from the sub-networks of ResNet-18 and is trained by using a principal loss or an auxiliary loss. This enables MAN to successfully deal with the multi-scale problem of semantic segmentation.

In the following, we respectively introduce three key components of MAN (i.e., LERB, FTB, and GCB) in detail.

1) Lattice Enhanced Residual Block (LERB): In this paper, inspired by the residual building blocks [12] and the lattice filter [41], we develop LERB to enhance feature representations in each branch. The structure of the lattice filter, also called as X-section, is the physical topology of an all-pass filter with the butterfly structure, which decomposes the input signal to multi-order representations [41]. Fig. 3 shows the network architecture of a standard lattice filter and the lattice structure used in our method.

The network architecture of LERB is shown in Fig. 4. LERB mainly consists of a contextual module and a spatial module to enhance the contextual information and spatial details, respectively. Specifically, the input feature map $\mathbf{X} \in \mathbb{R}^{H^l \times W^l \times C^l}$ is fed into the contextual module consisting of a contextual enhanced block, a weight learning block, and a lattice structure. The contextual enhanced block contains two 3×3 convolutional layers followed by a BN layer. Here, the atrous rates of two convolutional layers are respectively set to 2 and 4 to capture sufficient contextual information. Meanwhile, the weight learning block (consisting of a 1×1 convolutional layer and a Sigmoid activation function) is adopted to adaptively learn two weight tensors (i.e., $\mathbf{A}_c \in \mathbb{R}^{H^l \times W^l \times 1}$ and $\mathbf{B}_c \in \mathbb{R}^{H^l \times W^l \times 1}$), which are used for the lattice structure. The nonlinear function induced by the contextual enhanced block is denoted as $\mathrm{C}(\cdot)$. Therefore, the two output feature maps in the lattice structure are formulated as

$$\begin{aligned} \mathbf{P}_c &= \sigma(\mathbf{X} + \eta(\mathbf{B}_c) \otimes \mathrm{C}(\mathbf{X})), \\ \mathbf{Q}_c &= \sigma(\eta(\mathbf{A}_c) \otimes \mathbf{X} + \mathrm{C}(\mathbf{X})), \end{aligned} \tag{1}$$

where $\mathbf{P}_c \in \mathbb{R}^{H^l \times W^l \times C^l}$ and $\mathbf{Q}_c \in \mathbb{R}^{H^l \times W^l \times C^l}$ represent the intermediate feature maps. $\sigma(\cdot)$ denotes the ReLU activation function. '$\otimes$' means the element-wise multiplication operation. $\eta(\cdot)$ indicates the broadcast operation, where the weights are broadcast (copied) along the channel dimension.

The output feature map $\mathbf{F}_c \in \mathbb{R}^{H^l \times W^l \times C^l}$ from the lattice structure can be obtained as

$$\mathbf{F}_c = \mathbf{P}_c \oplus \mathbf{Q}_c, \tag{2}$$

where '$\oplus$' represents the element-wise addition operation.

Then, $\mathbf{F}_c$ is passed through a spatial module consisting of a spatial enhanced block, a weight learning block, and a lattice structure. Meanwhile, the downsampled feature map $\mathbf{M}$ from the first sub-network of ResNet-18, which has the same size as $\mathbf{F}_c$, is also used as the input of the spatial enhanced block. In the spatial enhanced block, $\mathbf{F}_c$ and $\mathbf{M}$ are first concatenated along the channel dimension. By concatenating the feature maps from the first lattice structure and the downsampled ones from the first sub-network of ResNet-18, we are able

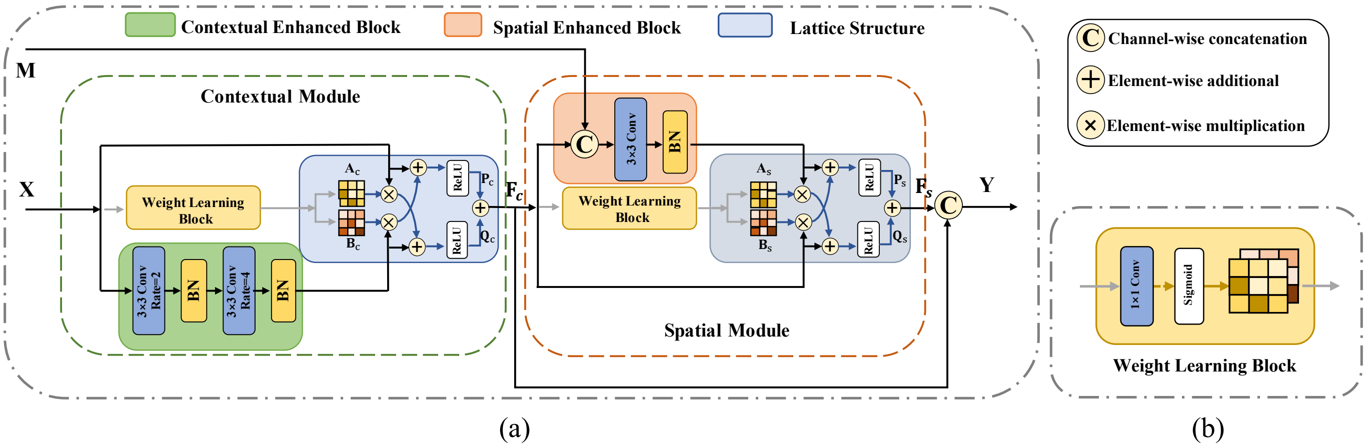


Fig. 4. The network architecture of (a) LERB and (b) Weight Learning Block. In the figure, "Rate" means the atrous rate. "BN" denotes the batch normalization layer. "ReLU" and "Sigmoid" indicate the ReLU and Sigmoid activation functions, respectively.

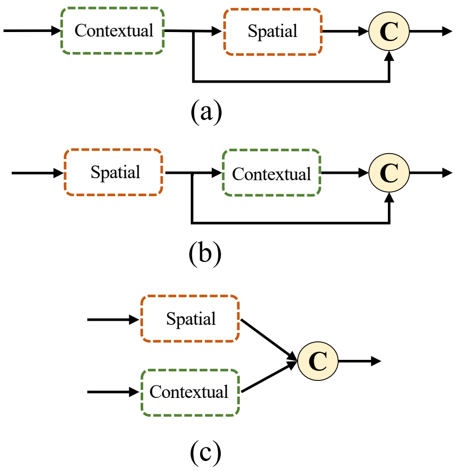


Fig. 5. Various ways of combining the contextual module and the spatial module. (a) the spatial module after the contextual module in serial (i.e., LERB), (b) the contextual module after the spatial module in serial, and (c) the spatial module and the contextual module in parallel. In the figure, "Contextual" and "Spatial" respectively represent the contextual module and the spatial module, respectively.

to enhance spatial details in the spatial module. Based on the concatenated feature map, a 3×3 convolutional layer followed by a BN layer is used to enhance feature spatial representations. The nonlinear function induced by the spatial enhanced block is denoted as $\mathrm{S}(\cdot)$. Meanwhile, $\mathbf{F}_c$ is also used to learn two weight tensors (i.e., $\mathbf{A}_s \in \mathbb{R}^{H^l \times W^l \times 1}$ and $\mathbf{B}_s \in \mathbb{R}^{H^l \times W^l \times 1}$) by a weight learning block. Therefore, the two output feature maps in the lattice structure are formulated as

$$\begin{aligned} \mathbf{P}_s &= \sigma(\eta(\mathbf{B}_s) \otimes \mathbf{F}_c + \mathrm{S}(\mathrm{concat}(\mathbf{F}_c, \mathbf{M}))), \\ \mathbf{Q}_s &= \sigma(\mathbf{F}_c + \eta(\mathbf{A}_s) \otimes \mathrm{S}(\mathrm{concat}(\mathbf{F}_c, \mathbf{M}))), \end{aligned} \tag{3}$$

where $\mathbf{P}_s \in \mathbb{R}^{H^l \times W^l \times C^l}$ and $\mathbf{Q}_s \in \mathbb{R}^{H^l \times W^l \times C^l}$ represent the intermediate feature maps. $\mathrm{concat}(\cdot,\cdot)$ represents the channel-wise concatenation operation.

The output feature map $\mathbf{F}_s \in \mathbb{R}^{H^l \times W^l \times C^l}$ from the lattice structure can be obtained as

$$\mathbf{F}_s = \mathbf{P}_s \oplus \mathbf{Q}_s. \tag{4}$$

Finally, the output feature map $\mathbf{Y} \in \mathbb{R}^{H^l \times W^l \times 2C^l}$ from LERB is represented as

$$\mathbf{Y} = \mathrm{concat}(\mathbf{F}_c, \mathbf{F}_s). \tag{5}$$

We should point out that there are various ways of combining the contextual module and the spatial module, as shown in Fig. 5. In general, the receptive fields of the input feature map are enlarged in the contextual module, while those do not change in the spatial module. As a consequence, when the contextual module and the spatial module are combined as given in Figs. 5(b) and 5(c), the feature maps used for concatenation have different receptive fields. Such a manner is not only detrimental for feature aggregation, but also increases the learning difficulty of the network. In contrast, the feature maps used for concatenation have the same receptive fields for LERB (i.e., Fig. 5(a)). Obviously, this benefits feature aggregation in the decoder.

Note that the contextual enhanced block and the spatial enhanced block developed in LERB are different from the basic block that was firstly proposed in [12] to address the degradation problem in deep networks. In particular, the contextual enhanced block takes advantage of two atrous convolutions (instead of standard convolutions used in the basic block) to enlarge the receptive fields and thus encodes the contextual information. The spatial enhanced block makes use of the downsampled feature map from the first sub-network of ResNet-18 to exploit spatial details. By integrating the contextual enhanced block and the spatial enhanced block into the lattice structures, LERB effectively enhances both the spatial and contextual information. Moreover, we leverage two weight learning blocks to adaptively adjust the weights of two lattice structures. *Such a way generates various combinations of enhanced blocks, which can enlarge the feature representation* *space very efficiently*. Hence, compared with the basic block, LERB provides much better feature extraction capability.

2) Feature Transformation Block (FTB): For semantic segmentation, it is of great importance to encode both the spatial and contextual information for predicting score maps. On the one hand, with the increase of network depth, the high-level feature maps mainly encode the sufficient contextual information while lacking spatial details. On the other hand, the low-level feature maps capture the rich spatial information. To exploit multi-level feature maps, many modern methods use element-wise addition [8], [24], [42] or channel-wise concatenation [10], [19], [32] to aggregate the semantic and spatial feature maps. However, such ways might not be beneficial for semantic segmentation, due to the gap between different levels of feature maps. Therefore, simply aggregating feature maps without taking the differences between feature maps into consideration may not only cause feature interference, but also decrease the segmentation accuracy.

To address the above problem, motivated by the Spatial and Channel Squeeze Excitation (scSE) block [43], we develop FTB to transform the feature map before aggregation. In particular, a transformation tensor is generated to indicate the importance of a feature map, and then is used to weigh each channel and spatial location of a feature map. Therefore, it can be used to emphasize the important information while ignoring the irrelevant information in the input feature map, so that an effective transformed feature map is obtained. In this way, the differences between multi-level feature maps can be greatly alleviated.

The network architecture of FTB is shown in Fig. 6. FTB is comprised of two main sub-branches to perform attention operations along the channel and spatial dimensions. Meanwhile, a weight learning sub-branch is used to adaptively learn the weights for the channel sub-branch and the spatial sub-branch. Roughly, FTB only consists of several convolutional layers and linear operations. Furthermore, the intermediate feature maps in the spatial sub-branch have a small number of channels (i.e., 1) and those in the channel sub-branch have a small resolution (i.e., 1×1). Hence, FTB is a lightweight module.

More specifically, FTB first employs a Conv-BN-ReLU module to generate a feature map $\mathbf{X}_f \in \mathbb{R}^{H' \times W' \times C'}$. For the spatial sub-branch, the feature map $\mathbf{X}_f$ is fed into a 1×1 convolutional layer and a Leaky ReLU activation function to obtain the attention tensor $\mathbf{X}_s \in \mathbb{R}^{H' \times W' \times 1}$. Meanwhile, the feature map $\mathbf{X}_f$ is also fed into a Global Average Pooling (GAP) layer to obtain the tensor $\mathbf{X}_g \in \mathbb{R}^{1 \times 1 \times C'}$ encoding the global information, which can be used in both the channel sub-branch and weight learning sub-branch. Then, in the channel sub-branch, $\mathbf{X}_g$ is sequentially fed into a 1×1 convolutional layer, a BN layer, a ReLU activation function and a linear layer to obtain the attention tensor $\mathbf{X}_c \in \mathbb{R}^{1 \times 1 \times C'}$. In the weight learning sub-branch, $\mathbf{X}_g$ is fed into a linear layer followed by a softmax activation function to adaptively learn two weights v and w. Therefore, the transformation tensor $\mathbf{T} \in \mathbb{R}^{H' \times W' \times C'}$ can be computed as

$$\mathbf{T} = \beta(\eta(v\mathbf{X}_s) + \eta(w\mathbf{X}_c)), \tag{6}$$

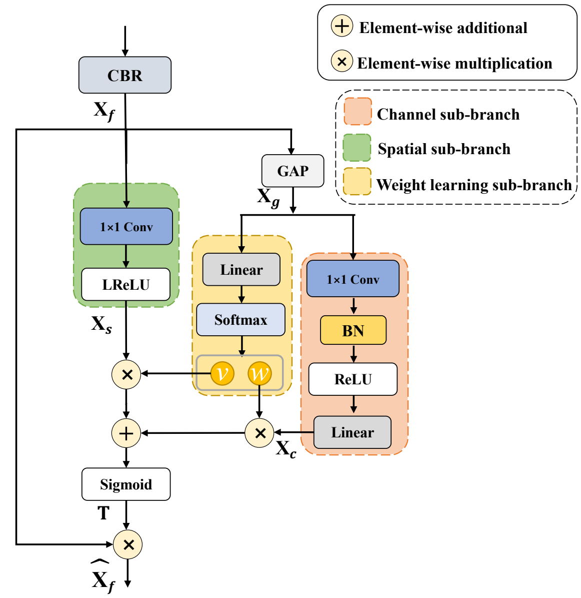


Fig. 6. The network architecture of FTB. In the figure, "LReLU" represents the Leaky ReLU activate function.

where $\beta(\cdot)$ denotes the Sigmoid activation function. Before the addition operation, the channel attention values are broadcast (copied) along the spatial dimension, while the spatial attention values are broadcast along the channel dimension.

Finally, the element-wise multiplication is performed between $\mathbf{X}_f$ and $\mathbf{T}$, which is formulated as

$$\widehat{\mathbf{X}}_f = \mathbf{T} \otimes \mathbf{X}_f, \tag{7}$$

where $\widehat{\mathbf{X}}_f$ represents the transformed feature map.

It is worth noting that the scSE block also learns the attention on both spatial and channel dimensions. However, different from the scSE block, the proposed FTB not only applies the attention sub-branches on both spatial and channel dimensions to obtain attention tensors, but also adaptively learns weights by designing a weight learning sub-branch. Hence, the attention tensors from the spatial and channel sub-branches can be effectively combined to obtain a transformation tensor. Furthermore, the scSE block uses the squeeze operator and the excitation operator on the channel dimension. However, the squeeze operator may lead to information loss since the channel number is reduced. In contrast, FTB removes these operators to model the dependencies between channels more accurately.

3) Global Context Block (GCB): Currently, most semantic segmentation methods are based on the DCNN models that are originally designed for the image classification task. Such a task relies largely on the high-level semantic information (such as object-level or category-level evidence). These DCNN models, however, may not accurately identify and locate the objects due to the lack of global contextual information, thus leading to a negative impact on the accuracy of semantic segmentation. Therefore, the global contextual information plays a critical role in street scene segmentation.

Based on the above observations, similar to BiSeNet [15], we append a GCB at the end of ResNet-18 to exploit the contextual information of the image. The network architecture of GCB is shown in Fig. 2. GCB first performs the GAP operation on the feature map (whose size is $1/32$ of the original input image size) from sub-network 4 of ResNet-18 to obtain a 1×1 feature map with the largest receptive fields. Then, the feature map is passed through a Conv-BN-ReLU module. Finally, the bilinear interpolation is used to restore the feature map back to $1/32$ of the original input image size. In fact, compared with the pooling features with multiple window sizes used in RefineNet [8], GCB has smaller memory consumption and less floating-point operations.

D. Joint Loss

In DMA-Net, both the auxiliary loss and principal loss are employed to optimize the training of the network. In particular, the auxiliary losses are used to supervise the training of the MB and HB of MAN, and the principal loss is employed to supervise the output of the whole network (i.e., the output from the LB of MAN). To be specific, the joint loss is formulated as

$$\begin{aligned}\mathcal{L}_{joint} =& \mathcal{L}_{principal}(\mathbf{O}^p, \mathbf{O}) + \lambda[\mathcal{L}_{auxiliary}(\mathbf{O}^p_{mid}, \mathbf{O}) \\ &+ \mathcal{L}_{auxiliary}(\mathbf{O}^p_{high}, \mathbf{O})],\end{aligned} \tag{8}$$

where $\mathcal{L}_{joint}$, $\mathcal{L}_{principal}$, and $\mathcal{L}_{auxiliary}$ represent the joint loss, the principal loss, and the auxiliary loss, respectively. λ denotes the balance weight. $\mathbf{O}^p$ denotes the predicted output from the whole network. $\mathbf{O}^p_{mid}$ and $\mathbf{O}^p_{high}$ denote the resized predicted outputs (having the same size as the input image) from the MB and HB of MAN, respectively. $\mathbf{O}$ denotes the ground-truth semantic labels.

All the loss functions adopt the pixel-wise cross entropy, whose form is defined as follows:

$$\mathcal{L}(\mathbf{Z}^p; \mathbf{Z}) = -\frac{1}{N}\sum_{i}^{K}\sum_{j}^{N} z_{i,j}\log(z^p_{i,j}), \tag{9}$$

where $\mathbf{Z}^p$ is the predicted output given by the softmax function and $\mathbf{Z}$ is the ground-truth semantic labels. $z^p_{i,j}$ and $z_{i,j}$ denote the probability value of the i-th category at the j-th pixel location of the output and its corresponding ground-truth label, respectively. N is the total number of pixels and K is the total number of semantic categories.

E. Discussions

Both our proposed DMA-Net method and some recent real-time semantic segmentation methods [2], [15], [18] take advantage of the encoder-decoder structure to improve the segmentation accuracy. However, there are significant differences between DMA-Net and these methods.

First, we propose LERB to address the problem of inferior feature extraction capability of the lightweight backbone network. Specifically, LERB enhances the spatial detail and context information in the feature maps by two feature enhancement blocks (i.e., a contextual enhanced block and a spatial enhanced block). In particular, LERB can expand the representation space of features by introducing the lattice structures. Hence, LERB effectively and efficiently improves the feature representations of the network. In contrast, previous methods either use additional branches for feature enhancement (such as BiSeNet [15], RTHP [18]), or employ a parallel network structure to enlarge the receptive fields of the network (such as SwiftNet [2]). Although these methods can enhance feature maps to a certain extent, additional branches or parallel networks will also bring high computational costs.

Second, we leverage FTB to reduce the gap between different levels of feature maps. Specifically, we use a weight learning sub-branch in FTB to adaptively enhance the important information and suppress the irrelevant information. Therefore, the problem of feature interference between different levels of feature maps is greatly alleviated, so that the spatial and contextual information in these feature maps can be properly aggregated. On the contrary, many methods [18], [19] adopt simple aggregation operations (such as the element-wise addition and the channel-wise concatenation) to aggregate different levels of feature maps. Hence, they ignore the differences between feature maps, resulting in a performance decrease.

IV. Experiments

In this section, we evaluate the performance of the proposed DMA-Net on the challenging street scene benchmarks (including the Cityscapes and CamVid datasets). We first introduce the datasets and evaluation metrics in Section IV-A. Then, we describe the implementation details in Section IV-B. Next, we conduct ablation studies to analyze the effectiveness of each key component of DMA-Net in Section IV-C. We compare DMA-Net with several state-of-the-art real-time semantic segmentation methods on the Cityscapes and CamVid datasets in Section IV-D and Section IV-E, respectively. Finally, we discuss the limitations of DMA-Net in Section IV-F.

A. Datasets and Evaluation Metrics

The Cityscapes dataset consists of 25,000 high-resolution (with the size of 1024×2048) street scene images that were collected from 50 different cities in Germany. These images are divided into two parts: 5,000 fine-annotated images and 20,000 weakly-annotated images. In this paper, we only use the fine-annotated images in our experiments. These fine-annotated images can be classified into 30 categories and split into three datasets: a training dataset (including 2,975 images), a validation dataset (including 500 images), and a test dataset (including 1,525 images). Similar to state-of-the-art semantic segmentation methods [14], [16], we only use 19 common semantic categories (such as sidewalk, road, and car) in our experiments. For the test dataset, we evaluate our method by using the online service provided by Cityscapes, which do not release the ground-truth images to users.

The CamVid dataset is another challenging semantic dataset for street scene understanding. It consists of 701 high-resolution (with the size of 720×960) video frames collected from five video sequences and 11 semantic categories. For a fair comparison, we split the whole dataset into training,

TABLE I
THE ACCURACY , SPEED, AND PARAMS ANALYSIS OF DIFFERENT BACKBONE NETWORKS: MOBILENETV2, RESNET-101, AND RESNET-18 ON THE CITYSCAPES VALIDATION DATASET.

Backbone Network	mIoU (%)	Speed (FPS)	Params (M)
FCN+MobileNetV2	61.7	28	**2.04**
FCN+ResNet-101	**65.2**	9	51.95
FCN+ResNet-50	64.1	19	32.95
FCN+ResNet-18	63.6	**54**	11.77

TABLE II
THE INFLUENCE OF GCB, LERB, AND FTB ON THE CITYSCAPES VALIDATION DATASET.

Method	mIoU (%)	Speed (FPS)	Params (M)
Baseline	72.9	55.3	12.99
Baseline+GCB	74.1	55.0	13.13
Baseline+GCB+FTB	75.7	54.4	13.50
Baseline+GCB+LERB	76.3	47.4	14.23
DMA-Net	76.8	46.7	14.60

validation, and test datasets, which respectively contain 367 images, 101 images, and 233 images, as done in [27].

For evaluation metrics, we adopt mean Intersection over Union (mIoU) and Frames Per Second (FPS), which measure the segmentation accuracy and latency, respectively. Moreover, we also use the number of parameters (Params) and floating-point operations (FLOPs) to evaluate the memory consumption and computational complexity of the model, respectively.

B. Implementation Details

For training, we employ the horizontal flipping, random scaling (the scale ratio ranges from 0.5 to 2.0), and random cropping on all the images to augment the dataset. The final image resolution for Cityscapes is 768×1536 and that for CamVid is 640×640. All the network parameters of the convolutional layers in ResNet-18 are initialized from the publicly available ResNet-18 [12] pretrained on the ImageNet [40]. The network parameters of MAN and GCB are randomly initialized by using the Kaiming normal initialization [44].

To optimize the whole network, we adopt Stochastic Gradient Descent (SGD) [45] with the batch size of 16, the momentum of 0.9, and the weight decay of 0.0005 to update the network parameters for Cityscapes. Moreover, we utilize the online hard example mining [46] to mitigate the influence of class imbalance. Similar to state-of-the-art semantic segmentation methods, we use the popular "poly" learning rate strategy $(1 - \frac{iter}{total\_iters})^{power}$ with the power of 0.9 to update the learning rate, where the initial learning rate is set to 0.005. For CamVid, the batch size and learning rate are set to 4 and 0.001, respectively.

The whole training process contains 60,000 iterations for Cityscapes and 80,000 iterations for CamVid. Codes are implemented by the PyTorch framework. All experiments on speed analysis are performed by using a single NVIDIA GTX 1080Ti GPU.

C. Ablation Studies

In this subsection, we investigate the effectiveness of each key component of DMA-Net (including ResNet-18, MAN, LERB, FTB, and GCB) step-by-step. In the following experiments, we evaluate these components on the Cityscapes validation dataset [5].

1) Effectiveness of ResNet-18: In this paper, we employ ResNet-18 (a lightweight version of ResNet) as our backbone network (the encoder of DMA-Net). As we mentioned above, the backbone network provides the basic feature extraction for the whole network, and it can affect both the segmentation accuracy and the inference speed of semantic segmentation. Complicated backbone networks have a large number of network parameters and floating-point operations, leading to serious degradation of the inference speed. Therefore, lightweight networks are usually adopted as backbone networks for real-time semantic segmentation.

To evaluate the effectiveness and efficiency of ResNet-18, we compare it with three widely used backbone networks (including MobileNetV2 [21], ResNet-50 [12], and ResNet-101 [12]). For simplicity, all the backbone networks are pretrained on the ImageNet dataset and use FCN [24] as the base structure. The comparison results are shown in Table I.

We can see that FCN+ResNet-101 achieves the highest segmentation accuracy (about 65.2% mIoU), which is about 3.5% and 1.6% higher than FCN+MobileNetV2 and FCN+ResNet-18, respectively. However, the number of network parameters of FCN+ResNet-101 is significantly high (about 51.95M), and its inference speed is the slowest (about 9 FPS) among all the competing methods. Although FCN+MobileNetV2 has the smaller number of parameters than the other three backbone networks, it achieves the lowest mIoU. FCN+ResNet-50 achieves 64.1% mIoU in terms of segmentation accuracy and the inference speed of 19 FPS. Note that FCN+ResNet-18 achieves worse segmentation accuracy than FCN+ResNet-101 and FCN+ResNet-50, but its number of parameters is much smaller. Moreover, FCN+ResNet-18 has much faster inference speed than the other competing methods. This shows that ResNet-18 can achieve a good balance between accuracy and inference speed. In the following, we will fix ResNet-18 as our encoder.

2) Effectiveness of MAN: To demonstrate the effectiveness of MAN (the decoder of DMA-Net), we evaluate the influence of different combinations of key components on the accuracy, speed, and memory consumption, as shown in Table II. The Baseline method adopts the encoder-decoder structure and it consists of ResNet-18 and a simplified version of MAN, where LERB, FTB, and GCB are not used. The Baseline+GCB, Baseline+GCB+FTB, Baseline+GCB+LERB, and DMA-Net methods share the same network architectures as the Baseline method, except that GCB, GCB+FTB, GCB+LERB, and GCB+LERB+FTB are respectively employed in MAN.

By comparing Table I and Table II, the Baseline method achieves 72.9% mIoU, which is much higher than FCN+ResNet-18 (about 9.3% mIoU higher). This demon-

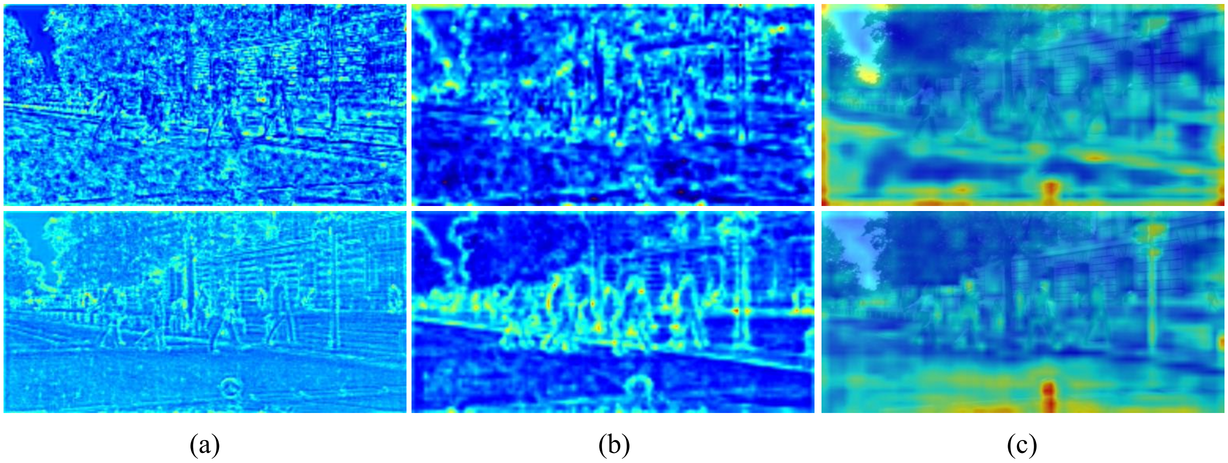


Fig. 7. Visualization results of feature maps. The images from the first column to the last column represent the feature maps from the (a) LB, (b) MB, and (c) HB, respectively. The upper panel and the lower panel show the feature maps before and after LERB, respectively.

strates the superiority of the encoder-decoder structure. Compared with the Baseline method, the Baseline+GCB method achieves better segmentation accuracy (about 1.2% mIoU higher). This result validates the importance of GCB.

Incorporating FTB or LERB into the Baseline+GCB method can further boost the segmentation accuracy. By taking into account both LERB and FTB, our DMA-Net method achieves the highest mIoU (about 76.8%). The above results show that both LERB and FTB are beneficial to improve the performance of semantic segmentation. This is because the joint learning of LERB and FTB enables the network to effectively aggregate hierarchical feature maps.

From the perspective of inference speed, the speed of Baseline+GCB is only slightly slower than that of Baseline. Hence, GCB brings only a small computational cost. By combining LERB with Baseline+GCB, the speed of the Baseline+GCB+LERB method is only about 7.6 FPS slower than that of the Baseline+GCB method. This shows the efficiency of LERB. Meanwhile, FTB has a subtle influence on the inference speed, since the speed of the Baseline+GCB+FTB method is only slightly slower than that of the Baseline+GCB method. Similarly, the inference speed of DMA-Net is almost the same as that of Baseline+GCB+LERB. In terms of the number of network parameters, the differences between all the competing methods are not significant. Thus, the memory consumption of these methods is relatively small (< 15M).

In summary, the above experimental results show that by incorporating GCB, LERB, and FTB into MAN, our method is able to achieve a good tradeoff between speed and accuracy.

3) Effectiveness of LERB: In this subsection, we evaluate the effectiveness of our proposed LERB. We replace the LERB in DMA-Net with the Basicblock and Bottleneck used in ResNet [12], respectively. The comparison results are shown in Table III.

We can observe that the mIoU obtained by DMA-Net is improved by about 1.0% in comparison with DMA-Net (Basicblock). Moreover, compared with DMA-Net (Bottleneck), DMA-Net also achieves higher accuracy (about 1.2% mIoU higher). With regards to speed, DMA-Net is only about 3 FPS and 4.2 FPS slower than DMA-Net (Basicblock) and DMA-Net (Bottleneck), respectively. These results demonstrate that LERB can enhance feature representations of our network more effectively than the other residual blocks for real-time semantic segmentation in street scenes.

TABLE III
THE ACCURACY, SPEED, AND PARAMS COMPARISON BETWEEN LERB, LERB-ADDITION, AND RESIDUAL BUILDING BLOCKS: BASICBLOCK, BOTTLENECK ON THE CITYSCAPES VALIDATION DATASET.

Method	mIoU (%)	Speed (FPS)	Params (M)
DMA-Net (Basicblock)	75.8	49.7	15.05
DMA-Net (Bottleneck)	75.6	50.9	13.59
DMA-Net (LERB-addition)	76.1	47.9	14.60
DMA-Net (LERB-b)	76.4	46.7	14.60
DMA-Net (LERB-c)	76.3	46.7	14.60
DMA-Net	76.8	46.7	14.60

In order to further investigate the effectiveness of the lattice structure in LERB on the final performance, we also replace the lattice structure in LERB with the element-wise addition operation, named DMA-Net (LERB-addition). As we can see, compared with the simple element-wise addition operation, adopting the lattice structure in LERB improves the segmentation accuracy by about 0.7% mIoU with a slight drop in terms of inference speed. This indicates that feature maps with different combinations in the lattice structure can effectively improve the representation capability of the network in an efficient manner.

Then, we compare LERB with its two variants. The two variants are denoted as DMA-Net (LERB-b) and DMA-Net (LERB-c) according to the structures given in Figs. 5(b) and 5(c), respectively. We can see that the accuracy obtained by DMA-Net (LERB-b) and DMA-Net (LERB-c) is lower than that obtained by DMA-Net. This is because the feature maps used for concatenation have different receptive fields, which have an adverse effect on the feature aggregation, thereby

TABLE IV
THE ACCURACY, SPEED, AND PARAMS COMPARISON BETWEEN DIFFERENT TRANSFORMATION BLOCKS ON THE CITYSCAPES VALIDATION DATASET.

Method	mIoU (%)	Speed (FPS)	Params (M)
DMA-Net (scSE)	76.5	46.6	14.25
DMA-Net (FTB_WLB)	76.4	46.8	14.60
DMA-Net	76.8	46.7	14.60

TABLE V
THE ACCURACY, SPEED, AND PARAMS COMPARISON BETWEEN DIFFERENT GLOBAL CONTEXTUAL MODULES ON THE CITYSCAPES VALIDATION DATASET.

Method	mIoU (%)	Speed (FPS)	Params (M)
DMA-Net (GAP)	76.5	46.8	14.60
DMA-Net	76.8	46.7	14.60

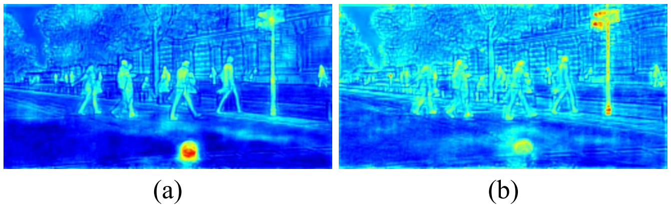


Fig. 8. Visualization results of feature maps obtained by (a) Baseline+GCB and (b) Baseline+GCB+FTB, respectively.

reducing the final segmentation performance.

Finally, we give some visualization results to show the importance of LERB, as shown in Fig. 7. To be specific, we visualize the feature maps before and after LERB in three branches of MAN for DMA-Net. We can observe that LERB enables the network to enhance spatial details and context information of feature maps in three branches. For example, the feature map after LERB pays more attention to edge details in the LB, while it focuses more on the context information in the HB.

4) Effectiveness of FTB: In this subsection, we further study the importance of FTB. The results are listed in Table IV. DMA-Net (scSE) denotes the method that has the same network architecture as DMA-Net, except that FTB is replaced with the scSE block [43].

From Table IV, we can see that the segmentation accuracy obtained by DMA-Net is higher than DMA-Net_scSE (about 0.3% mIoU improvement). This is because FTB adaptively combines the spatial and channel sub-branches with a weight learning sub-branch. Moreover, FTB preserves more information than scSE in the channel dimension (note that the squeeze operator used in scSE is not adopted in FTB). Therefore, FTB is able to obtain informative transformed feature maps. As a result, different levels of feature maps can be effectively aggregated. From the perspective of speed, DMA-Net and DMA-Net (scSE) obtain almost the same inference speed. In terms of network parameters, DMA-Net is only slightly higher than DMA-Net (scSE).

We further investigate the effect of adaptive weights on the final segmentation performance. We denote DMA-Net without using the weight learning sub-branch in FTB as DMA-Net (FTB_WLB). DMA-Net achieves better accuracy than DMA-Net (FTB_WLB). This shows the importance of the weight learning sub-branch in FTB.

Finally, we also visualize the feature maps (the output of MAN) obtained by Baseline+GCB and Baseline+GCB+FTB, respectively. Some visualization results are shown in Fig. 8. Compared with the feature map obtained by Baseline+GCB, the feature map obtained by Baseline+GCB+FTB not only preserves finer edge details, but also better focuses on objects at different scales. This validates that FTB can effectively reduce the gap between different levels of feature maps and thus facilitates the combination of spatial details and semantic information.

5) Effectiveness of GCB: In this subsection, we further verify the effectiveness of GCB. We compare GCB with GAP. The comparison results are as shown in Table V, where DMA-Net (GAP) shares the same network architecture as DMA-Net, except that GCB is replaced with GAP.

Compared with the DMA-Net (GAP) method, the DMA-Net method increases about 0.3% mIoU, which indicates the superiority of GCB. GCB can effectively capture the global contextual information. This is due to the fact that we use the convolutional operation after the GAP operation, which enables the network to extract more compact global feature representations, thus improving the final segmentation performance. Meanwhile, GCB has little influence on the inference speed, since DMA-Net is only slightly slower than DMA-Net (GAP).

6) Influence of Auxiliary Loss: In this subsection, we evaluate the influence of auxiliary loss on the final performance. In the experiments, we change the balance weight λ from 0 to 1.2. All the results are shown in Fig. 9.

In Fig. 9, we can observe that the accuracy obtained by DMA-Net is only slightly different when the values of λ are within the range of $[0.2, 1.2]$. This shows that the network is not very sensitive to the value of λ. When $\lambda = 1$, our proposed method achieves the best performance (77.4% mIoU). Therefore, employing the auxiliary loss in MAN is beneficial to improve the segmentation performance. When $\lambda = 0$, only the principal loss is used to supervise the training. In this case, the mIoU obtained by our method drops to 76.8%. The above results show that the proposed auxiliary loss enables our method to explicitly supervise the training of the MB and HB of MAN, thus improving the segmentation performance.

D. Comparisons with State-of-the-Art Methods

To evaluate the effectiveness and efficiency of DMA-Net, we first compare it with the simplified PSPNet [7] and SwiftNet [2] on the Cityscapes validation dataset. The simplified PSPNet is obtained by compressing the kernel keeping rate of PSPNet and SwiftNet is the current representative real-time semantic segmentation method. All the results are reported in Table VI.

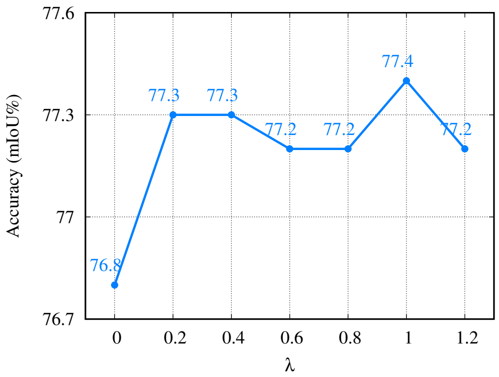


Fig. 9. The accuracy obtained by the proposed DMA-Net with different values of the parameter λ on the Cityscapes validation dataset.

TABLE VI
THE ACCURACY AND SPEED COMPARISON BETWEEN THE PROPOSED METHOD AND PSPNET, ICNET ON THE CITYSCAPES VALIDATION DATASET.

Item	PSPNet	SwiftNet	DMA-Net (ours)
mIoU (%)	67.9	75.4	**77.4**
Time (ms)	170	25	**21.4**
Speed (FPS)	5.9	39.9	**46.7**
Image size	713 × 713	1024 × 2048	1024 × 2048

It can be seen that DMA-Net outperforms the two competing methods and achieves an overwhelming performance with 77.4% mIoU accuracy at the inference speed of 46.7 FPS. Specifically, the segmentation accuracy of our proposed DMA-Net exceeds that of the simplified PSPNet and ICNet by about 10% and 2%, respectively. Meanwhile, the inference speed of our method is much faster than that of the two competing methods. The results demonstrate that DMA-Net provides excellent inference speed and high segmentation accuracy on the Cityscapes validation dataset.

Then, we compare our proposed method with several state-of-the-art real-time semantic segmentation methods on the Cityscapes test dataset, as given in Table VII. In Table VII, the inference speed, segmentation accuracy, FLOPs, and Params are included. The FLOPs and Params indicate the number of floating-point operations and the parameters of the network, respectively. Note that our method is also compared with the accuracy-oriented DeepLab and PSPNet methods.

When using the original image (with the size of 1024 × 2048) as the input, our proposed DMA-Net achieves 77.0% mIoU at the inference speed of 46.7 FPS. Moreover, DMA-Net has only 94.2G FLOPs and 14.60M Params, which are substantially better than some real-time semantic segmentation methods (including SegNet and SQNet). More specifically, DMA-Net is about 32 FPS faster and 20.9% mIoU higher than SegNet. Although ESPNet achieves the fastest inference speed and the lowest memory consumption, its mIoU is about 16.7% lower than DMA-Net. Compared with BiSeNet2, DMA-Net not only performs better in terms of accuracy and speed, but also has fewer Params. Although DMA-Net obtains slower inference speed than DFANet, it improves the segmentation accuracy by about 5.7% mIoU while maintaining the real-time performance. Compared with our previous method RTHP, DMA-Net adopts higher resolution images as the inputs, and achieves better accuracy (about 3.4% mIoU higher) and similar inference speed. Furthermore, DMA-Net even achieves better performance than an accuracy-oriented semantic segmentation method. For example, the proposed DMA-Net is about 185 times faster, and about 14% mIoU higher than DeepLab.

When using a low-resolution image (with the size of 768 × 1536) as the input, our method (denoted as DMA-Net (small)) achieves 75.6% mIoU at the inference speed of 76.8 FPS. Compared with SwiftNet, our method not only achieves higher mIoU, but also is nearly 2 times faster. Therefore, our method achieves a good balance between accuracy and inference speed.

Similar to BiSeNet, DFANet, and ICNet, DMA-Net also adopts the multi-branch framework. However, compared with BiSeNet that employs a feature fusion module to combine the feature maps from the spatial and context branches, DMA-Net progressively aggregates the feature maps from the high-level branch to the low-level branch. Different from DFANet that performs deep feature aggregation through sub-network and sub-stage cascade, DMA-Net leverages a multi-branch aggregation network (i.e., MAN) based on LERB and FTB. Unlike ICNet that takes the cascade image inputs for different branches, DMA-Net exploits different levels of feature maps from four stages of ResNet-18 as the inputs for multiple branches. Moreover, DMA-Net takes advantage of an elaborately-designed MAN, which not only aggregates different levels of feature maps, but also captures the multi-scale information.

The per-class, mean-class, and category accuracy values of the Cityscapes test dataset are given in Table VIII. Here, the results obtained by BiSeNet2 are based on the open source codes[1] and the input image resolution of 1024×2048. It can be seen that our proposed method achieves the best performance on most classes, especially the similar objects (building vs. wall, truck vs. bus). In particular, our method obtains much higher mIoU than other methods on some classes (such as truck and bus). Although our method obtains the second best performance on some classes (such as vegetation and sky), the difference is trivial (less than 1% IoU). Meanwhile, our method achieves the lowest mIoU variance, which further shows the effectiveness of our method.

It is worth noting that the Cityscapes dataset was collected from 50 different cities in Germany, where the training set, the validation set, and the test set consist of the images captured in different cities. Although these subsets show different scene changes, our method is still able to achieve good segmentation performance at real-time inference speed. Some qualitative segmentation results are shown in Fig. 10. Generally speaking, DMA-Net can correctly assign the labels to different scales of objects in street scenes, such as the pedestrians in the second row and the cars in the third row of Fig. 10.

All our experiments are based on an NVIDIA GTX 1080Ti GPU on the desktop platform, which is also employed by state-

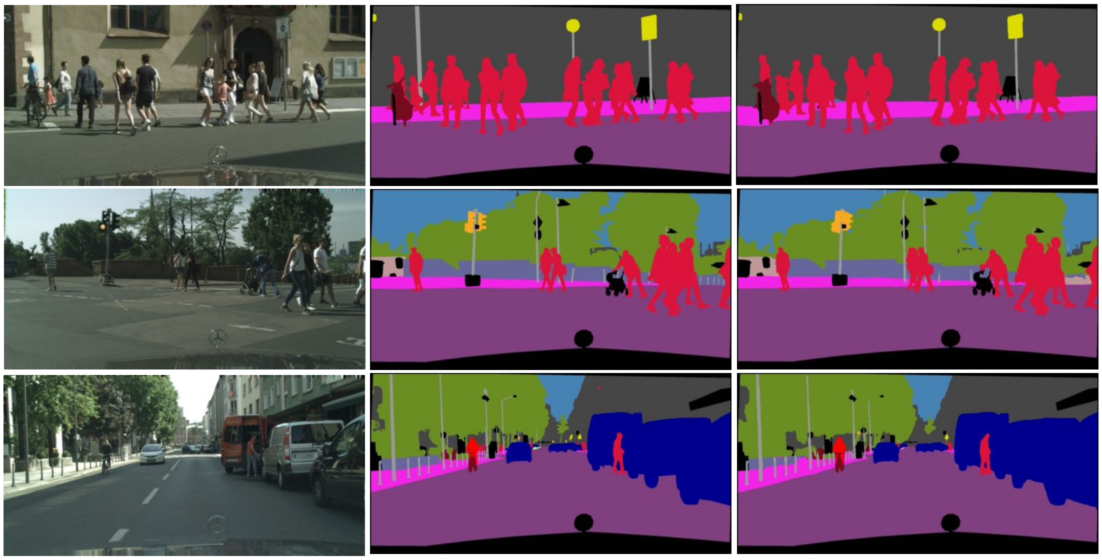
[1] https://github.com/CoinCheung/BiSeNet

TABLE VII
COMPARISONS BETWEEN THE PROPOSED METHOD AND OTHER STATE-OF-THE-ART METHODS ON THE CITYSCAPES TEST DATASET. "-" INDICATES THAT THE CORRESPONDING RESULT IS NOT PROVIDED BY THE METHOD.

Method	Input Size	FLOPs (G)	Params (M)	Speed (FPS)	mIoU (%)
DeepLab [25]	512 × 1024	457.8	262.1	0.25	63.1
PSPNet [7]	713 × 713	412.2	250.8	0.78	78.4
SegNet [27]	640 × 360	286	29.5	14.6	56.1
ENet [13]	630 × 630	4.4	0.4	76.9	58.3
ESPNet [20]	512 × 1024	4.7	0.4	112	60.3
SQNet [28]	1024 × 2048	270	-	16.7	59.8
CRF-RNN [26]	512 × 1024	-	-	1.4	62.5
FCN-8S [24]	512 × 1024	136.2	-	2.0	65.3
FRRN [29]	512 × 1024	235	-	2.1	71.8
ERFNet [14]	512 × 1024	-	2.1	41.7	68.0
ICNet [16]	1024 × 2048	29.8	26.5	30.3	69.5
TwoColumn [30]	512 × 1024	57.2	-	14.7	72.9
DFANet [19]	1024 × 1024	3.4	7.8	100.0	71.3
LEDNet [17]	512 × 1024	-	0.94	71	70.6
RTHP [18]	448 × 896	49.5	6.2	51.0	73.6
BiSeNet1 [15]	768 × 1536	14.8	5.8	72.3	68.4
BiSeNet2 [15]	768 × 1536	55.3	49	45.7	74.7
SwiftNet [2]	1024 × 2048	104	11.8	39.9	75.5
DMA-Net (small)	768 × 1536	53.0	14.60	76.8	**75.6**
DMA-Net	1024 × 2048	94.2	14.60	46.7	**77.0**

TABLE VIII
THE PER-CLASS, CLASS AND CATEGORY IOU(%) ON THE CITYSCAPES TEST DATASET FOR DMA-NET COMPARED TO OTHER METHODS. LIST OF CLASSES (FROM LEFT TO RIGHT): ROAD, SIDE-WALK, BUILDING, WALL, FENCE, POLE, TRAFFIC LIGHT, TRAFFIC SIGN, VEGETATION, TERRAIN, SKY, PEDESTRIAN, RIDER, CAR, TRUCK, BUS, TRAIN, MOTORBIKE AND BICYCLE. "CLA" DENOTES MIOU (19 CLASSES), "VAR" DENOTES THE VARIANCE.

Method	Roa	Sid	Bui	Wal	Fen	Pol	TLi	TSi	Veg	Ter	Sky	Ped	Rid	Car	Tru	Bus	Tra	Mot	Bic	Cla	Var
SegNet [27]	96.4	73.2	84.0	28.4	29.0	35.7	39.8	45.1	87.0	63.8	91.8	62.8	42.8	89.3	38.1	43.1	44.1	35.8	51.9	57.0	5.09
ENet [13]	96.3	74.2	75.0	32.2	33.2	43.4	34.1	44.0	88.6	61.4	90.6	65.5	38.4	90.6	36.9	50.5	48.1	38.8	55.4	58.3	4.61
FCN-8s [24]	97.4	78.4	89.2	34.9	44.2	47.4	60.1	65.0	91.4	69.3	93.9	77.1	51.4	92.6	35.3	48.6	46.5	51.6	66.8	65.3	4.11
ERFNet [14]	97.9	82.1	90.7	45.2	50.4	59.0	62.6	68.4	91.9	69.4	94.2	78.5	59.8	93.4	52.3	60.8	53.7	49.9	64.2	69.7	2.85
LEDNet [17]	98.1	79.5	91.6	47.7	49.9	62.8	61.3	72.8	92.6	61.2	94.9	76.2	53.7	90.9	64.4	64.0	52.7	44.4	71.6	70.6	2.82
BiSeNet2 [15]	98.2	82.9	91.7	44.5	51.1	**63.5**	**71.2**	75.0	92.9	71.1	94.9	83.6	65.4	94.9	60.5	68.7	56.8	61.5	72.7	73.8	2.40
SwiftNet [2]	98.3	83.9	92.2	46.3	52.8	63.2	70.6	**75.8**	**93.1**	70.3	**95.4**	84.0	64.5	95.3	63.9	78.0	71.9	61.6	**73.6**	75.5	2.15
DMA-Net	**98.5**	**85.5**	**92.2**	**53.3**	**55.3**	62.5	70.9	74.9	93.0	**71.2**	95.0	**84.0**	**66.6**	**95.6**	**68.2**	**82.8**	**76.6**	**64.5**	73.2	**77.0**	**1.82**



(a) (b) (c)

Fig. 10. Segmentation results of the proposed DMA-Net on the Cityscapes validation dataset. The images from the first column to the last column respectively denote (a) input images, (b) ground-truth images, and (c) our predicted results.

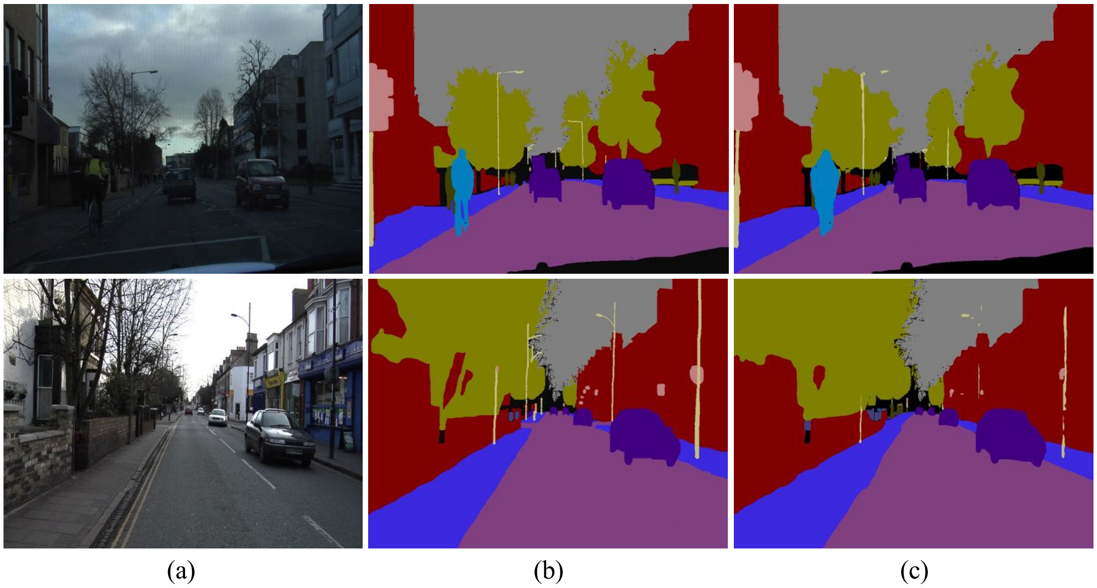


Fig. 11. Segmentation results of the proposed DMA-Net on the CamVid test dataset. The images from the first column to the last column respectively denote (a) input images, (b) ground-truth images, and (c) our predicted results.

of-the-art real-time semantic segmentation methods (such as LEDNet [17], BiSeNet [15], and SwiftNet [2]). In this way, we can compare our method with these state-of-the-art methods by using the same platform. As shown in Table VII, our method achieves better segmentation accuracy than other methods at the competitive inference speed. Meanwhile, the number of parameters obtained by our method is only 14.60M. Note that the main differences between the embedded platform in autonomous driving and the desktop platform are the computing power of graphics cards and memory resources. Therefore, our method is still able to outperform these competing methods when applied to real-world autonomous driving applications.

E. Results on the CamVid Dataset

To further illustrate the superiority of our method, we perform experiments on the CamVid dataset. The evaluation results are reported in Table IX. In this experiment, we also fine-tune the model (pre-trained by Cityscapes) on the CamVid dataset to verify the transfer properties of our model. We denote the fine-tuned model as DMA-Net†.

We can observe that our proposed DMA-Net method obtains competitive results (i.e., 73.6% mIoU at the inference speed of 119.8 FPS) among all the methods. In particular, DMA-Net obtains much faster inference speed than most methods (such as SegNet, ENet, and ICNet). Compared with BiSeNet2, DMA-Net not only achieves better accuracy (about 4.9% mIoU higher), but also gives a faster inference speed. Moreover, our DMA-Net also obtains better segmentation performance (about 1% mIoU higher) than SwiftNet. In a word, our method achieves a balanced tradeoff between accuracy and speed. DMA-Net† achieves the best segmentation accuracy of 76.2% mIoU, which is about 2.6% mIoU higher than DMA-Net. This is because the Cityscapes dataset involves a large number of training samples, enabling us to obtain a powerful pre-trained model. As a result, the pre-trained model can be easily fine-tuned to classify different classes on the small dataset.

TABLE IX
THE ACCURACY AND SPEED COMPARISON BETWEEN THE PROPOSED METHOD AND OTHER METHODS ON THE CAMVID TEST DATASET. †THE CITYSCAPES DATASET IS USED FOR PRETRAINING.

Method	Input Size	mIoU (%)	Speed (FPS)
SegNet [27]	360 × 480	46.4	46
ENet [13]	360 × 480	51.3	61.2
ICNet [16]	720 × 960	67.1	27.8
CGNet [47]	360 × 480	65.6	-
BiSeNet1 [15]	720 × 960	65.6	175
BiSeNet2 [15]	720 × 960	68.7	116.3
DFANet [19]	720 × 960	64.7	120
SwiftNet [2]	720 × 960	72.6	-
DMA-Net (ours)	720 × 960	**73.6**	**119.8**
DMA-Net† (ours)	720 × 960	**76.2**	**119.8**

Note that the images in the CamVid dataset are captured from video sequences. Different from the Cityscapes dataset, there exist severe illumination variations on CamVid. However, our method still obtains good segmentation results. Some segmentation results are shown in Fig. 11. Therefore, our method is robust to scene changes and is applicable to real-world applications requiring real-time inference speed.

F. Limitations

In this subsection, we discuss the limitations of our proposed DMA-Net. DMA-Net is able to effectively and efficiently perform semantic segmentation. However, it still suffers from the following two challenges.

1) Severe occlusions between objects. An object can easily be occluded by other objects in street scenes. In particular, when the target object and occluded objects have similar colors and shapes, our proposed method is prone to give wrong

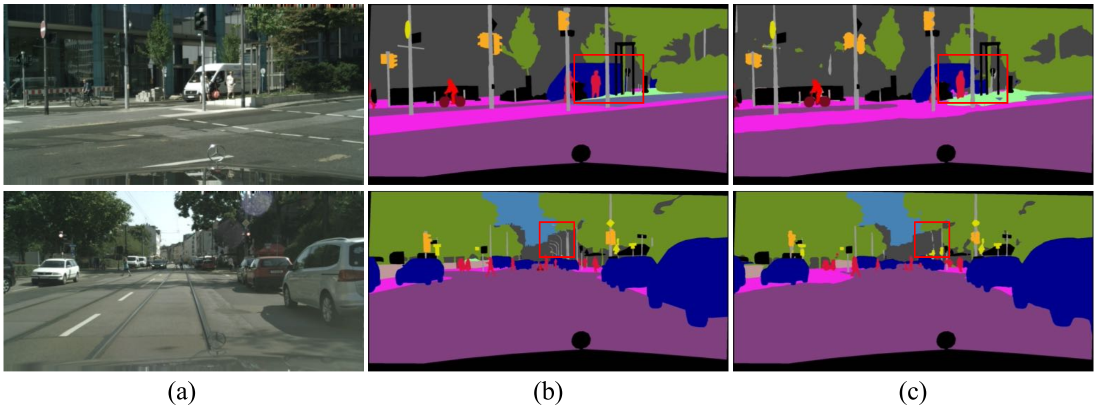


Fig. 12. Some failure cases of the proposed DMA-Net on the Cityscapes validation dataset. The images from the first column to the last column respectively denote (a) input images, (b) ground-truth images, and (c) our predicted results.

segmentation results. This is because the similar appearance of different objects makes the network difficult to determine which category the pixel belongs to in the occluded areas. A failure segmentation result is shown in the first row of Fig. 12. In the future, the object depth information can be exploited to address the occlusion problem.

2) Small objects in the scenes. Semantic segmentation performs pixel-level classification, where both spatial details and contextual information play an important role in achieving good performance. In our method, the encoder (i.e., ResNet-18) generates different levels of feature maps encoding the spatial and contextual information, while our MAN gradually aggregates the feature maps from the encoder to perform pixel inference. In DMA-Net, in order to improve the speed of the network, we do not design a branch to deal with the high-resolution feature maps from the first sub-network of ResNet-18. Therefore, the detailed spatial information of small objects may lose to some extent during feature aggregation. In this way, MAN may not be able to recover the spatial information, thus leading to the misclassification of some small objects in the final segmentation results. As illustrated in Table VIII, our method achieves a high IoU for some large objects (such as road and building). In contrast, our method gets a low IoU for some small objects (such as fence and pole). A failure segmentation result is shown in the second row of Fig. 12. In future, more powerful lightweight networks can be designed to provide a good tradeoff between model capacity and inference speed.

Note that the above two challenges also exist in other real-time semantic segmentation methods (such as ICNet [16] and DFANet [19]).

V. Conclusion

In this paper, we have presented a novel DMA-Net method for real-time semantic segmentation in street scenes. DMA-Net consists of two main parts: ResNet-18 and MAN. ResNet-18 generates different levels of feature maps, while MAN takes advantage of LERB, FTB, and GCB to aggregate these feature maps and capture the multi-scale information. In particular, LERB makes use of lattice structures to effectively enhance feature representations while FTB adaptively generates the transformed feature maps for feature aggregation. Furthermore, GCB encodes the rich global contextual information. These components are tightly coupled and jointly trained to ensure high-quality segmentation results while running at real-time. Experimental results on two challenging street scene benchmarks (including the Cityscapes and the CamVid datasets) have demonstrated the effectiveness and efficiency of our proposed DMA-Net.

References

[1] L. Li, B. Qian, J. Lian, W. Zheng, and Y. Zhou, “Traffic scene segmentation based on RGB-D image and deep learning,” *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 5, pp. 1664–1669, May 2018.

[2] M. Orsic, I. Kreso, P. Bevandic, and S. Segvic, “In defense of pre-trained ImageNet architectures for real-time semantic segmentation of road-driving images,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12 607–12 616.

[3] B. Chen, C. Gong, and J. Yang, “Importance-aware semantic segmentation for autonomous vehicles,” *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 1, pp. 137–148, Jan. 2019.

[4] W. Zhou, J. S. Berrio, S. Worrall, and E. Nebot, “Automated evaluation of semantic segmentation robustness for autonomous driving,” *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 5, pp. 1951–1963, May 2020.

[5] M. Cordts *et al.*, “The cityscapes dataset for semantic urban scene understanding,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3213–3223.

[6] G. Brostow, J. Fauqueur, and R. Cipolla, “Semantic object classes in video: A high-definition ground truth database,” *Pattern Recognit. Lett.*, vol. 30, no. 2, pp. 88–97, Jan. 2009.

[7] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2881–2890.

[8] G. Lin, A. Milan, C. Shen, and I. Reid, “RefineNet: Multi-path refinement networks for high-resolution semantic segmentation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1925–1934.

[9] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[10] L. C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 801–818.

[11] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1251–1258.

[12] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[13] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "ENet: A deep neural network architecture for real-time semantic segmentation," Jun. 2016, *arXiv:1606.02147*. [Online]. Available: https://arxiv.org/abs/1606.02147

[14] E. Romera, J. M. lvarez, L. M. Bergasa, and R. Arroyo, "ERFNet: Efficient residual factorized ConvNet for real-time semantic segmentation," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 1, pp. 263–272, Jan. 2018.

[15] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "BiSeNet: Bilateral segmentation network for real-time semantic segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 325–341.

[16] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia, "ICNet for real-time semantic segmentation on high-resolution images," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 405–420.

[17] Y. Wang *et al.*, "LEDNet: A lightweight encoder-decoder network for real-time semantic segmentation," 2019, *arXiv:1905.02423*. [Online]. Available: https://arxiv.org/abs/1905.02423

[18] G. Dong, Y. Yan, C. Shen, and H. Wang, "Real-time high-performance semantic image segmentation of urban street scenes," *IEEE Trans. Intell. Transp. Syst.*, pp. 1–17, Jan. 2020.

[19] H. Li, P. Xiong, H. Fan, and J. Sun, "DFANet: Deep feature aggregation for real-time semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9522–9531.

[20] S. Mehta, M. Rastegari, A. Caspi, L. Shapiro, and H. Hajishirzi, "ESPNet: Efficient spatial pyramid of dilated convolutions for semantic segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 552–568.

[21] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 4510–4520.

[22] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 6848–6856.

[23] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *Int. J. Comput. Vision*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[24] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[25] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, May 2015.

[26] S. Zheng *et al.*, "Conditional random fields as recurrent neural networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1529–1537.

[27] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[28] M. Treml *et al.*, "Speeding up semantic segmentation for autonomous driving," in *Proc. MLITS, NIPS Workshop*, 2016, pp. 1–7.

[29] T. Pohlen, A. Hermans, M. Mathias, and B. Leibe, "Full-resolution residual networks for semantic segmentation in street scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4151–4160.

[30] Z. Wu, C. Shen, and A. van den Hengel, "Real-time semantic image segmentation via spatial sparsity," Dec. 2017, *arXiv:1712.00213*. [Online]. Available: https://arxiv.org/abs/1712.00213

[31] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.

[32] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assisted Intervent. (MICCAI)*, Oct. 2015, pp. 234–241.

[33] M. Holschneider, R. Kronland-Martinet, J. Morlet, and P. Tchamitchian, "A real-time algorithm for signal analysis with the help of the wavelet transform," in *Wavelets, Springer, Berlin, Heidelberg*, 1990, pp. 286–297.

[34] L. C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," Jun. 2017, *arXiv:1706.05587*. [Online]. Available: https://arxiv.org/abs/1706.05587

[35] L. C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille, "Attention to scale: Scale-aware semantic image segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3640–3649.

[36] H. Zhang *et al.*, "Context encoding for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 7151–7160.

[37] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected CRFs with Gaussian edge potentials," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Dec. 2011, pp. 109–117.

[38] J. Fu *et al.*, "Dual attention network for scene segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3146–3154.

[39] X. Li, Z. Zhong, J. Wu, Y. Yang, Z. Lin, and H. Liu, "Expectation-maximization attention networks for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9167–9176.

[40] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[41] A. Gray and J. Markel, "Digital lattice and ladder filter synthesis," *IEEE Trans. Audio Electroacoust.*, vol. 21, no. 6, pp. 491–500, Dec. 1973.

[42] V. Nekrasov, C. Shen, and I. Reid, "Light-weight refinenet for real-time semantic segmentation," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, Sep. 2018, pp. 125–139.

[43] A. G. Roy, N. Navab, and C. Wachinger, "Concurrent spatial and channel squeeze & excitation in fully convolutional networks," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent. (MICCAI)*, Sep. 2018, pp. 421–429.

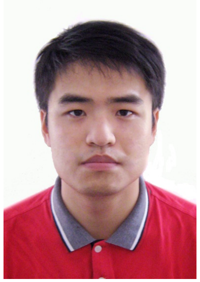
[44] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.

[45] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proc. 19th Int. Conf. Comput. Statist. (ICCS)*, 2010, pp. 177–186.

[46] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 761–769.

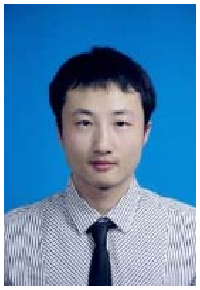
[47] T. Wu, S. Tang, R. Zhang, and Y. Zhang, "CGNet: A light-weight context guided network for semantic segmentation," Nov. 2018, *arXiv:1811.08201*. [Online]. Available: https://arxiv.org/abs/1811.08201

Xi Weng is currently a master student in the School of Informatics at Xiamen University, China. His main research interests include deep learning, semantic segmentation, autonomous driving and related fields.



Yan Yan is currently a professor in the School of Informatics at Xiamen University, China. He received the Ph.D. degree in Information and Communication Engineering from Tsinghua University, China, in 2009. He worked at Nokia Japan R&D center as a research engineer (2009-2010) and Panasonic Singapore Lab as a project leader (2011). He has published around 100 papers in the international journals and conferences including the IEEE T-PAMI, IJCV, IEEE T-IP, IEEE T-MM, IEEE T-CYB, IEEE T-CSVT, IEEE T-ITS, IEEE T-AC, PR, CVPR, ICCV, ECCV, ACM MM, AAAI. His research interests include computer vision and pattern recognition.

Genshun Dong is currently a master student in the School of Informatics at Xiamen University, China. His main research interests include deep learning and semantic segmentation.



Chang Shu is currently a lecturer in the School of Information and Communication Engineering at University of Electronic Science and Technology of China, China. He received the Ph.D. degree in Information and Communication Engineering from Tsinghua University, China, in 2011. His research interests include computer vision, machine learning, and pattern recognition.

Biao Wang received his B.E. degree in Electronic Engineering from Dalian University of Technology, in 2012, and the Ph.D. degree in Computer Science from Shanghai Jiao Tong University, in 2020. Currently, he is working as a postdoc in Artificial Intelligence Research Institute in Zhejiang Lab. His research interests include large-scale graph data mining and social network analysis.

Hanzi Wang is currently a Distinguished Professor of “Minjiang Scholars” in Fujian province and a Founding Director of the Center for Pattern Analysis and Machine Intelligence (CPAMI) at Xiamen University in China. He received his Ph.D. degree in Computer Vision from Monash University. His research interests are concentrated on computer vision and pattern recognition including visual tracking, robust statistics, object detection, video segmentation, model fitting, optical flow calculation, 3D structure from motion, image segmentation and related fields.

Ji Zhang is an IET Fellow, IEEE Senior Member, Australian Endeavour Fellow, Queensland International Fellow (Australia) and Izaak Walton Killam Scholar (Canada). He is a Full Professor at USQ and a Visiting Professor at Zhejiang Lab, Tsukuba University, Nanyang Technological University (NTU) and Michigan State University (MSU). Prof. Zhang’s research interests include data science, big data analytics, data mining and health informatics. He has published over 230 papers, many appearing in top-tier international journals and conferences, including TKDE, TCYB, TDSC, TKDD, TIST, Information Sciences, KAIS, PRL, WWWJ, JIIS, Bioinformatics, AAAI, IJCAI, SIGKDD, VLDB, ICDE, ICDM, CIKM, WWW Conference, CVPR and COLING. Prof. Zhang is the recipient of Australian Endeavour Award, USQ Research Excellence Award, Head of Department Research Award and three international conference best paper awards.