MFNet: Multi-Feature Fusion Network for Real-Time Semantic Segmentation in Road Scenes

Mengxu Lu, Zhenxue Chen, Chengyun Liu, Sile Ma, Lei Cai, and Hao Qin

***Abstract*— Although high-accuracy networks have been applied to semantic segmentation at present, their inference speeds remain slow. A trade-off between accuracy and speed is demanded for real-time applications. To approach this problem, we propose Multi-Feature Fusion Network (MFNet) with real-time efficient prediction capacity. MFNet adopts three branches (attention, semantic and spatial information) to capture low-level and high-level features. Additionally, MFNet exerts asymmetric factorized (AF) blocks to extract local and long-range features. As a result, without any pre-training or post-processing, MFNet using only 1.34 M parameters, achieves 72.1% mean intersection over union (mIoU) on the Cityscapes test set at a speed of 116 frames per second (FPS), with 512 × 1024 high resolution on a single Titan Xp graphics card. Our network's performance stands out from other state-of-the-art networks on four datasets (Cityscapes, CamVid, KITTI, and Gatech).**

***Index Terms*— Semantic segmentation, convolutional neural network (CNN), real-time, multi-feature fusion.**

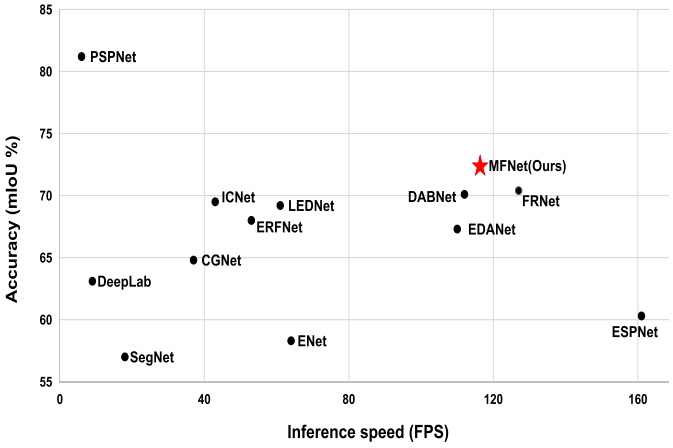

Fig. 1. Inference speeds run on a single Titan Xp and accuracy on the Cityscapes [6] test set.

I. Introduction

SEMANTIC segmentation is the classification of each pixel in an image in order to segment the image according to semantics. Segmentation through video or image is a basic problem in computer vision. In the automatic driving task, after obtaining the current road image, semantic segmentation technology is used to segment and extract road information to help an automatic driving system make decisions.

In 2015, Shelhamer *et al.* [1] proposed the Fully Convolutional Neural Network (FCN). FCN replaces the fully connected layer of the classification model, such as AlexNet [2] and Visual Geometry Group Network (VGGNet) [3], with a convolutional layer that transforms the classification task into the semantic segmentation task. Several excellent Convolutional Neural Networks (CNNs) have achieved advanced results on public benchmark datasets. For example, representative of deep and large networks, Pyramid Scene Parsing Network (PSPNet) [4] and DeepLab-v3 [5] have achieved greater than 80% mIoU on the Cityscapes [6] dataset. However, these high-accuracy networks are usually based on complicated baseline networks (like VGGNet or ResNet [7]), which have hundreds of layers and thousands of channels, resulting in low inference speed. With the application of semantic segmentation, it is found that these large networks consume a large amount of resources, and cannot meet real-time requirements. In order that semantic segmentation can be better applied to practical projects, light-weight networks such as SegNet [8], ENet [9], Efficient Residual Factorized Network (ERFNet) [10], and Efficient Spatial Pyramid Network (ESPNet) [11] have become the focus of semantic segmentation research. However, increase in speed results in a loss of accuracy. Fig. 1 illustrates the inference speed (FPS) and accuracy (mIoU) of several state-of-the-art methods. From Fig. 1, it is apparent that making a good trade-off between accuracy and speed requires further investigation.

In this paper, we address these challenges by proposing an efficient Multi-Feature Fusion Network (MFNet) for semantic segmentation in road scenes, taking into account the accuracy of the model and increasing the inference speed, all while reducing the parameters. Multiple branches are employed in the network. An attention branch with a position attention block captures the spatial dependencies between any two positions of the input feature maps. A semantic branch excavates

Manuscript received 29 July 2021; revised 17 March 2022; accepted 2 June 2022. Date of publication 25 July 2022; date of current version 7 November 2022. This work was supported in part by the National Key Research and Development Program of China under Grant 2019YFB1311001, in part by the National Natural Science Foundation of China under Grant 61876099, and in part by the Key Research and Development Project of Shandong Province under Grant 2022CXGC010503. The Associate Editor for this article was J. Alvarez. *(Mengxu Lu and Zhenxue Chen are co-first authors.) (Corresponding author: Zhenxue Chen.)*

Mengxu Lu, Zhenxue Chen, Chengyun Liu, Sile Ma, and Hao Qin are with the School of Control Science and Engineering, Shandong University, Jinan 250061, China (e-mail: 201500171046@mail.sdu.edu.cn; chenzhenxue@sdu.edu.cn; liuchengyun@sdu.edu.cn; masile@sdu.edu.cn; 202014785@mail.sdu.edu.cn).

Lei Cai is with the School of Artificial Intelligence, Henan Institute of Science and Technology, Xinxiang 453003, China (e-mail: cailei2014@126.com).

Digital Object Identifier 10.1109/TITS.2022.3182311

high-level features by deepening the network, while a spatial information branch retains the low-level features. Finally, the feature fusion block is applied to merge multiple features. To increase speed, we use depth-wise convolution instead of normal convolution in most of the network, as depth-wise convolution separates the channel and the spatial information, reducing the parameters. The reduction of parameters by a large margin brings about a decline of effect. We put some 3×3 convolutions in the place of some 1×1 convolutions to solve this problem. Dilated convolutions with a larger receptive field are applied in the network as well. Factorized convolution that decomposes a 3×3 convolution into 3×1 and 1×3 convolutions cuts down computational complexity with only a slight degradation of performance, also an important characteristic of our network.

Our main contributions are summarized as follows:

- We propose Multi-Feature Fusion Network (MFNet) which reaches 72.1% mIoU on the Cityscapes dataset at a speed of 116 FPS using high resolution images (512×1024), on a single Titan Xp graphics card, without any pre-training or post-processing.
- We create three branches in our network to fuse multiple features: the attention branch, the semantic branch, and the spatial information branch. Each of the three has proved indispensable.
- We design the AF block as the core block of our network, fusing both local and long-range features via light-weight construction.
- We evaluate MFNet on four public road scenes understanding datasets (Cityscapes [6], CamVid [12], KITTI [13], and Gatech [14]) and find that it earns a trade-off between accuracy and prediction speed. MFNet stands out among state-of-the-art semantic segmentation networks, as seen in Fig. 1.

II. Related Works

In this section, we look at some state-of-the-art methods for semantic segmentation, especially those with efficient skills comparable to MFNet.

A. CNNs in Semantic Segmentation

A milestone in the use of CNNs for semantic segmentation is the introduction of FCN with end-to-end training by adopting a fully convolutional network. With FCN as inspiration, many CNNs for semantic segmentation continue to be introduced. Even now, much research is focused on searching for methods with remarkable performance in speed and accuracy.

B. Accuracy of Networks

In reviewing networks with high performance by the metric of accuracy, a number are worthy of mention. SegNet uses a classical encoder-decoder architecture to sample and recover feature maps. U-Net [15] obtains predictions combining deep features as well as shallow information through the u-shape. DeepLab-v3 designs the Atrous Spatial Pyramid Pooling (ASPP) module to capture contextual information for better performance with different dilation rates. PSPNet utilizes pyramid pooling to unite contextual information. Dilated convolution, widely used in CNNs, can enlarge the receptive fields without increasing the parameters.

C. Real-Time Semantic Segmentation

Because of the limits of computational resources and timeliness in practice, real-time semantic segmentation models are required. ENet, with only 0.37M parameters, is the foremost light-weight semantic segmentation network. ERFNet turns the 2D convolution kernel (3×3) into two 1D convolution kernels (3×1, 1×3) thus lowering the computational cost and decreasing parameters. ESPNet with its Efficient Spatial Pyramid (ESP) module, gathers the features at different times. ContextNet [16] and Fast-SCNN [17] take advantage of two branches, a shallow network path along with a deep branch, in order to learn local and global image context at the same time. LEDNet [18] has an asymmetric encoder-decoder architecture. The two operations (channel split and shuffle) in the encoder significantly save computational resources, and the Attention Pyramid Network (APN) in the decoder further lightens the whole network. Depth-wise separable convolution, widely used in networks, reduces parameters by splitting channels and fuses features with a 1×1 convolution. FRNet [19] and FSSNet [20] use factorized and regular (FR) blocks and continuous factorized block, respectively, to gain efficient and accurate results. SwiftNet [21] employs light-weight upsampling with lateral connections on a general purpose architecture to realize real-time segmentation. Lite-HRNet [22] replaces costly pointwise (1×1) convolutions in shuffle blocks with a light-weight unit, conditional channel weighting, to break the computing bottleneck. Trans4Trans [23] proposes a segmentation architecture with transformer-based encoder and decoder with Transformer Parsing Module(TPM). SegFormer [24] unifies Transformers with light-weight multilayer perceptron (MLP) decoders. Our network comes up with asymmetric factorized (AF) block based on 3×1 and 1×3 convolutions to lighten the network.

D. Attention Modules

Attention modules have been widely used for various tasks such as natural language processing, image recognition, and speech recognition. The attention model (AM) was first used for machine translation [25]. Attention modules applied in semantic segmentation have the capacity to capture long-range dependencies. Dual Attention Network (DANet) [26] exploits both spatial and channel dimensions for semantic interdependencies. Crisscross Network (CCNet) [27] achieves better efficiency with criss-cross attention. To reduce computation complexity of attention module, FANet [28] and ECANet [29] are put forward. FANet proposes fast spatial attention by changing the order of operations to cut down the computational cost. ECANet uses a Horizontal Segment Attention (HSA) module and a Pyramidal Space Attention (PSA) module to lessen the computational burden. The attention block in our network utilizes max-pooling and shared weights based on the position attention module of DANet [26].

TABLE I
THE ARCHITECTURE DETAILS OF MFNET REFER TO THE SETTINGS AND OUTPUT OF EACH LAYER

	Stage	Type	Mode	Output size
Encoder	Downsample1	3×3 Conv	s=2,p=1	512×1024×32
	Conv1	3×3 Conv	s=1,p=1	512×1024×32
	Conv2	3×3 Conv	s=1,p=1	512×1024×32
	Concat1	-	-	512×1024×35
	Downsample2	3×3Conv+2×2Maxpool	s=2,p=1;s=2	256×512×64
	AF Block1	2×	d=2,2	256×512×64
	Conv3	1×1 Conv	s=1,p=0	256×512×32
	Concat2	-	-	256×512×131
	Downsample3	3×3 Conv	s=2,p=1	128×256×128
	Downsample4	3×3 Conv	s=2,p=1	64×128×128
	Conv+AB+Conv	1×1+attention+1×1	s=1,p=0	64×128×64
	AF Block2	5×	d=2,4,4,16,16	64×128×128
	Concat3	-	-	64×128×259
	Conv4	1×1 Conv	s=1,p=0	64×128×64
	Concat4+Conv	3×3 Conv	s=1,p=1	64×128×64
	Upsample1	bilinear	×4	256×512×64
	FFB+Conv	future fusion+1×1	s=1,p=0	256×512×64
Decoder	Conv5	1×1 Conv	s=1,p=0	256×512×c
	Upsample2	bilinear	×4	1024×2048×c

[1] The variable 'c' refers to the numbers of classes to be predicted. 's' represents stride, 'p' is padding and 'p' states dilated rate. Input images are 1024×2048 with three channels.

E. Feature Fusion

As the depth of the network and times of down-sampling, the fusion and reuse of features deserve further attention. FCN exploits the skip architecture, which combines a deep, coarse layer with a shallow, fine layer to capture accurate and detailed feature maps. RefineNet [30] puts forward long-range residual connections to extract multi-level features for high-resolution prediction, explicitly exploiting information of value along the down-sampling process. BiseNet [31] utilizes two paths (the spatial and the semantic) and a feature fusion block to acquire combined features. FSFNet [32] brings forward a feature selective fusion module (FSFM) to merge features in different levels or scales, and a multiscale context enhancement module to aggregate multiscale and global context information. RFNet [33] designs Attention Feature Complementary (AFC) module to extract and fuse features from RGB and Depth branches. Our network constructs Feature Fusion Block (FFB) to blend features from three different branches.

III. PROPOSED APPROACH

In this section, we propose the architecture of the Multi-Feature Fusion Network (MFNet) for semantic segmentation in road scenes. Each branch of MFNet (the attention, semantic, and spatial information branches) is described in detail. After that, the asymmetric factorized block, which makes the network more light-weight, is demonstrated. Finally, use of the feature fusion block to fuse features of different stages is introduced.

A. MFNet Architecture

People perceive things through the cooperation of various senses. Multi-source information acquired through smell, hearing, vision, etc. is fused by the brain to make people perceive the existence of things. Inspired by human beings, neural network can also integrate multi-stage features to improve the accuracy of prediction. Spatial features possess detailed information from a concrete perspective, and semantic features have abstract information from a big-picture perspective. In a convolutional neural network, the shallow layer, which has a large resolution feature map, learns the low spatial features. A series of operations such as convolution and pooling provide the network sufficient depth to study the features of rich semantic information. Features with low resolution can be used for tasks like image classification, while semantic segmentation requires high-resolution output. It is not enough to use only high-level features rich in semantics. It is necessary to effectively fuse semantic features of the high level with spatial features of the low level to obtain more accurate segmentation results. The multi-branch feature fusion network proposed in this paper effectively integrates the features of different levels.

We use three-branches network to simultaneously improve accuracy and speed, as shown in Fig. 2 and TABLE I. Our network has no backbone, and is trained from scratch. The down-sample blocks have two modes, as seen in Fig. 3, a one-branch mode with only a 3×3 convolution of stride 2, and a two-branch mode, where the left branch has a 3×3 convolution of stride 2, and the right branch has a 2×2 max-pooling. In order to process high-resolution images and maintain speed for real-time requirements, we take four down-samples to sample the image 1/16 of the original images. Apart from down-sample2, the other three down-sample stages are in one-branch mode. The down-sample2 stage is designed to utilize the two-branch mode in order to reserve significant information in the early stage. The other down-sample stages are just 3×3 convolutions, simplifying the network.

Firstly, we down-sample the images to 1/2 of the original images thus reducing computational pressure, and extend channels to 32. Then, two 3×3 convolutions with stride 2 and 32 filters process feature maps. To reduce loss of information, we average pool input images to 1/2, 1/4, and 1/16 of the original size with three channels for further use. The second down-sample, with 64 filters, occurs after concatenating the output and 1/2 pooled images. After that, our core block (the AF block) is used twice to explore low features. These feature maps are not only convolved by 1×1 in the spatial branch for spatial information, but also concatenate with the input of AF block1 and 1/4 pooled images to make 131 channels. Feature maps, after concatenating, are down-sampled twice by 3×3 convolutions to achieve 1/16 of the original images and 128 channels. Then, branches are divided into two, including one attention branch and one semantic information branch. The attention branch adopts the position attention block (AB), which is able to capture long-range contextual information, and a 3×3 convolution with stride 2 to decrease channels from 128 to 64. The semantic information branch incorporates

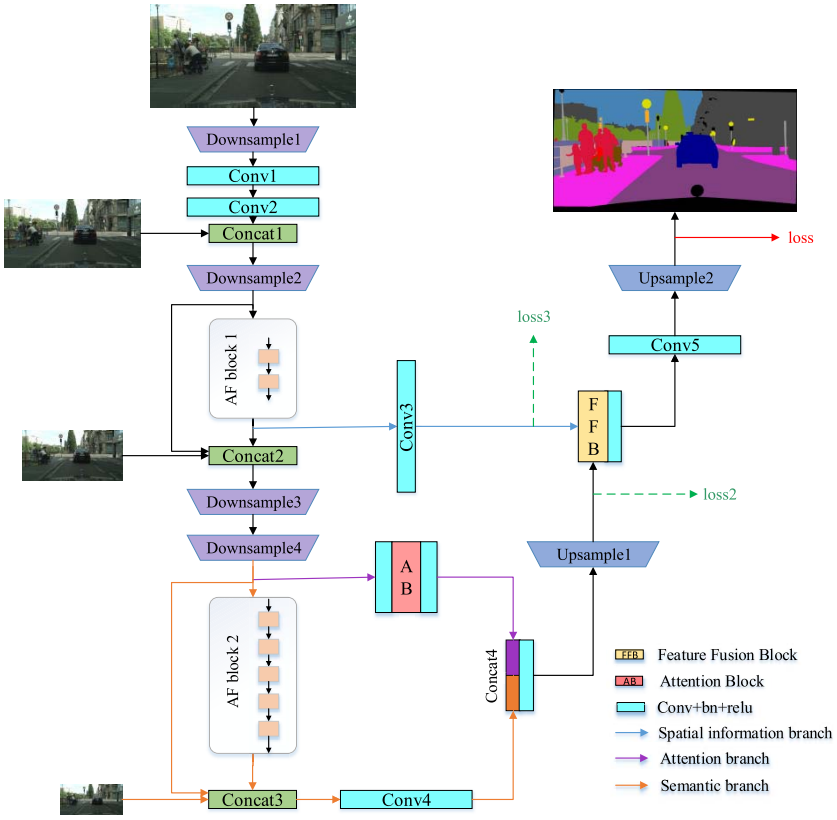

Fig. 2. An overview of MFNet. The blue line represents the spatial information branch. The purple line means the attention branch. AB in the attention branch signifies the attention block. The orange line indicates the semantic branch. FFB is the feature fusion block. Loss2 and loss3 is auxiliary loss derived from two places in the figure, which is used in ablation experiments, but not in MFNet. AF block 1 and AF block 2 employ the AF block twice and five times, respectively.

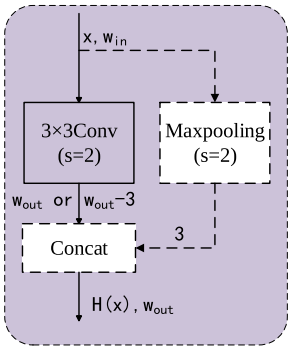

Fig. 3. Down-sample. 'x' represents input images. 'w' signifies channels.

five times AF blocks, which have huge receptive fields for learning high-level features and abundant semantic information. Concatenating the input, output of these five AF blocks and 1/16 pooled images results in 259 channels, which is relatively large. Therefore, a 1×1 convolution with 64 filters is used to prune several redundant features. The semantic information branch and the attention branch concatenate with each other first, and then up-sample four times to fuse with the spatial branch using the Future Fusion Block (FFB). At last, a 1×1 convolution is employed to obtain channels, similar to classes to be predicted. Up-sampling by bilinear interpolation is designed to gain network input size in the end.

B. Attention Branch (AB)

The attention module is divided into position attention and channel attention. Position attention is to calculate the attention of other pixels to a certain pixel and get the importance of this pixel, thus assigning a weight to each pixel. Position attention can make full use of spatial location information, which is the most valuable point to use in pixel level semantic segmentation task. Channel attention is to weight channels according to the importance of channels. Since vehicles are obvious in the marked Nos. 1, 2, 8, 16 feature maps in Fig. 4(b), large weights are assigned to these channels and small weights are assigned to other channels in order to predict vehicles. However, the attention branch of MFNet is drawn from the deep layer. The features will be like Fig. 4(c). The images are full of abstract features with equal importance, so it is difficult to assign different weights to different channels. It can also be seen from the ablation study that channel attention has not achieved a good effect in MFNet, so we only adopts positional attention in the attention branch.

Conventional convolution can only express local receptive fields. Even though dilated convolution can expand receptive fields, it is difficult to set proper dilation rates and it occupies large memory space. Therefore, we propose the position attention to fully obtain global information. The attention block, as shown in Fig. 5, is used after the last down-sampling module. The response weight obtained by considering the

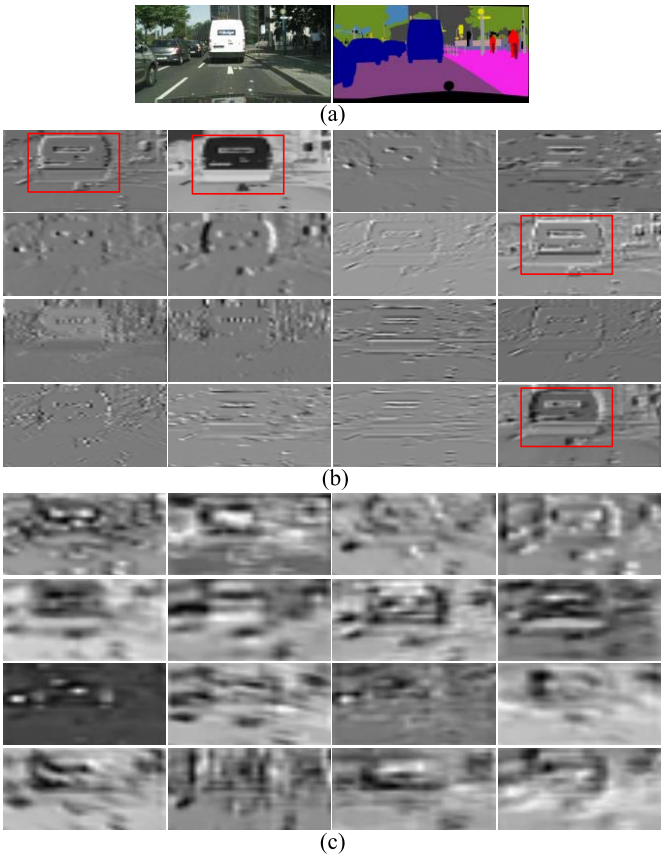

Fig. 4. (a) Input and output of semantic segmentation, (b) Visualization of shallow features, (c) Visualization of deep features.

response of all positions in the feature to the current position represents the degree of attention paid to the current position. Existing semantic segmentation networks ignore the relationship between pixels, and the segmentation images are not continuous enough. Position attention promotes the efficient expression of information between two long distance pixels by calculating the dependence between positions.

Different from the position attention in DANet, our proposed attention block utilizes max-pooling and weight sharing. Max-pooling is equivalent to local attention. It can select and retain important features by extracting the maximum value. It can also filter information to remove non-salient features. The max-pooling reduces the size of the feature map and enhances the expressiveness and generalization ability. Weight sharing reduces parameters, and ablation experiments show that reduced parameters do not affect performance of MFNet.

First, we use max-pooling to reduce the input from h×w×c to (h-1)×(w-1)×c. Then, three 1 × 1 convolutions are used in parallel, where two of them share weights, causing the structure in the red box evolves into the structure in the green box. After that, the features of the two adjusted dimensions are multiplied to obtain the feature of size (h-1)(w-1) × (h-1)(w-1), which is the response weight matrix of each position to all positions. Softmax puts the weight between 0 – 1, and the response feature is obtained by multiplying the weight by the middle side features. Finally, the resulting response feature is connected with the original feature via residual

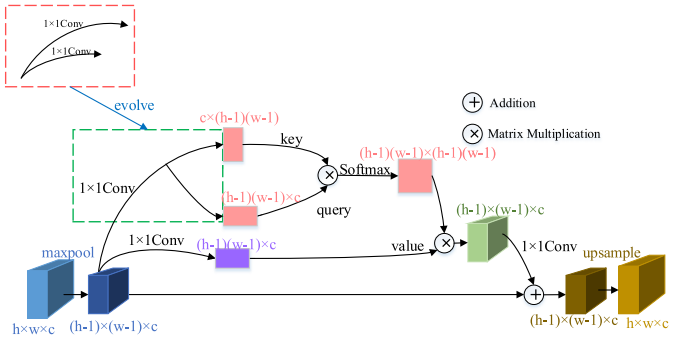

Fig. 5. The attention block. 'h', 'w' and 'c' represent height, width and channels of images, respectively.

connection, to output the optimized feature after the attention mechanism. Thereafter, convolution and upsampling are utilized to get the final output of the attention branch.

C. Semantic Branch

In the semantic branch, the output after four down-sampling is mined for deep features. Five asymmetry factorized (AF) blocks and concentration with 1/16 size of the original images sufficiently explore the deep semantic information. The dilated convolution with different sampling rates {2, 2, 4, 4, 16} is used serially in the five bottleneck blocks, so that the feature combines both the surrounding information and global information, and regions of different scales and arbitrary sizes can be accurately and effectively segmented.

D. Spatial Information Branch

In the semantic and attention branches, the features obtained are high-level features, which can be used to identify the categories contained in each region with low resolution. It is necessary to employ spatial information with low-level features to obtain results with high resolution and accurate segmentation.

In the spatial information branch, the features after the first AF block are adopted. Features in this stage have learned adequate spatial information after a series of previous operations, and the resolution is relatively high. After the convolution operation and adjustment to feature dimensions, this spatial information branch integrates with the other two branches.

E. Feature Fusion Block (FFB)

Feature fusion methods mainly include max-fusion, add-fusion, concatenation and convolution. Max-fusion and add-fusion fuse several feature maps with the same size and channels. The max-fusion maximizes and the add-fusion adds the values at the same position in several feature maps. The images fused by these two methods ignore the features in the original feature images and get the result of the interaction of several feature images. Concatenation is the fusion of several feature maps with the same size but not necessarily the same channels. Through concatenation, several feature maps are cascaded together to form fusion features with the same size and more channels that adds input channels. In this way,

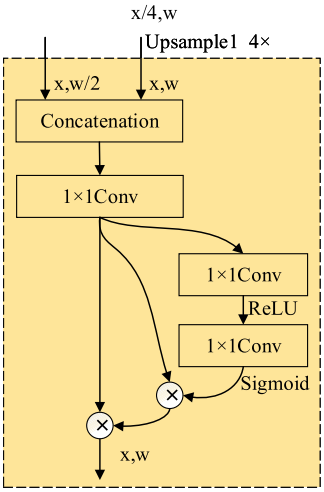

Fig. 6. The feature fusion block. 'x' represents input images. 'w' signifies input channels.

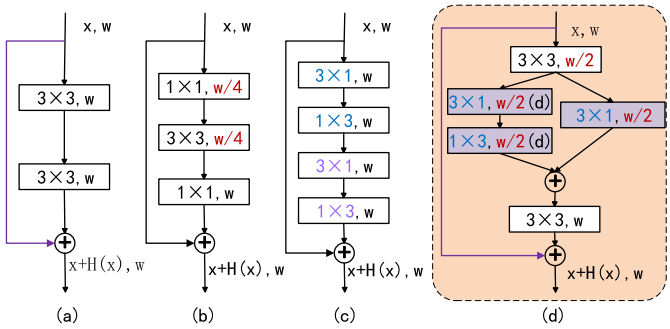

Fig. 7. (a) Non-bottleneck, (b) Bottleneck, (c) Non-bottleneck-1D, (d) Asymmetric Factorized (AF) block. ResNet proposes layers (a) (b), and ERFNet presents factorized residual layers (c) based on (a)(b). The AF block (d) is the core block of our network, inspired by them. 'w' signifies input channels.

all the features of the original feature map can be retained, but a large number of channels takes up large space and requires high computing capacity. Convolution is carried out after concatenation to further fuse features. Meanwhile, the number of channels is reduced through this step to reduce the subsequent computational burden.

In MFNet's multi-branch structure, different branches have different feature levels. The attention and semantic branches have high-level features, while the spatial information branch has low-level features. The features of the semantic branch and the attention branch are first fused by means of concatenation and convolution. Then the features of the spatial information branch also need to be fused. Due to the different feature levels of these branches, a feature fusion block, as shown in Fig. 6, has been adopted, rather than the simple concatenation method.

Features are up-sampled after the semantic branch and the attention branch by bilinear interpolation. The features of the different branches are connected in series, and then fused through Convolution-BatchNorm-ReLU. The fused features are adjusted through the attention mechanism of channel dimensions, and different weights are assigned to different channels, so as to complete the fusion of the different levels' features.

F. Asymmetric Factorized (AF) Block

The core block of our network is asymmetric factorized block, which extracts information via lightweight residual layers, as shown in Fig. 7(d).

The residual layer, as in Fig. 7(a), adds the output of multiple convolutions and the input of this layer, increasing the efficiency of the network and improving gradient dissipation in back propagation. The module in Fig. 7(b) sets middle channels as 1/4 of the input, which not only makes the structure light, but also eliminates redundant features. Factorized residual layers, as in Fig. 7(c), use convolutions with 1D filters to increase computational efficiency. AF blocks in our network gathering the merits of them can be seen in Fig. 7(d).

3×3 convolutions occur at the beginning of the AF block to acquire local information. Middle channels are reduced to 1/2 input channels instead of 1/4, since the input channels of the AF block are at most 128, and the network must retain information of value.

The two branches consist of two asymmetric paths with a left factorized depth-wise dilated asymmetric convolution (3×1, 1×3) and a right depth-wise non-dilated 3×1 convolution (thus the name "asymmetric factorized" block). The 3×1 convolution in the right branch extracts surrounding information without dilation. The factorized convolution in the left branch is a dilated one, because a dilated convolution has the capacity for a large receptive field to associate long-range information. The dilation rates are {2, 2} in the first two AF blocks, and {2, 4, 4, 16, 16} in the second five AF blocks, as shown in TABLE I. The three convolutions in these two branches are all depth-wise, reducing parameters to a great extent. Finally, a 3×3 convolution conflates features on width by 3×1, features on height by 1×3, and local features by non-dilated 3×1, and restores channels to be added with the input.

SS-nbt module in LEDNet [18] and EAC module in EACNet [34] like AF block are shown in Fig. 8(a)(b)(c). SS-nbt splits the input into two branches with half channels in each branch and shuffles channels after adding all features. AF block makes use of the 3×3 convolution at the beginning of the module to reduce channels to half as the input of two branches. Compared with channel split and shuffle, the 3×3 convolution has parameters that enable the network to learn how to reduce channels while retaining useful information and removing redundant information. EAC adds features from three branches many times resulting in an increasing number of channels and computational pressure. Different from EAC, AF adds features from two branches once, except the residual connection. AF block uses fewer convolutions and has a simpler structure than SS-nbt and EAC. In the ablation study, we replace AF block with SS-nbt or EAC, but there is no improvement, which indicates that AF block is sufficient for our network with non-redundant information. 3×1 and 1×3 convolutions are employed in pairs in LEDNet and EACNet to act as 3×3 convolutions. The 3×1 convolution appears alone in the right branch of AF block and the ablation study

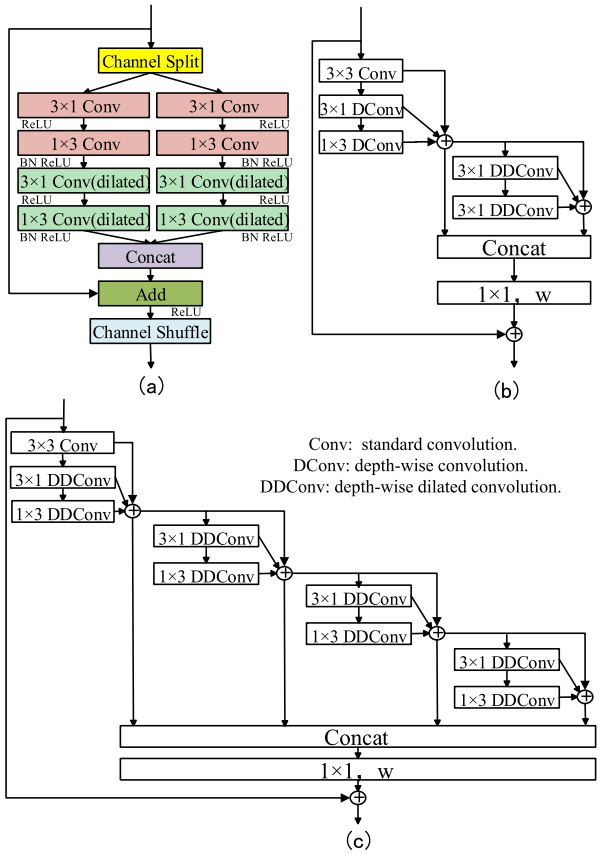

Fig. 8. (a) SS-nbt module in LEDNet [18], (b) EAC-A module in EACNet [34], (c) EAC-B module in EACNet.

verifies the effectiveness of this structure. That inspires us to use 3 × 1 or 1 × 3 convolution alone without the other in the model.

IV. Experiments

In this section, we evaluate our network MFNet on four datasets of road scenes. They are Cityscapes [6], CamVid [12], KITTI [13] and Gatech [14], all diffusely used for road semantic segmentation. First of all, the datasets information and the implementation details used in this paper are introduced. Then, to explore the potential of MFNet, we implement some ablation studies on the Cityscapes validation set. Subsequently, we provide the consequences on the Cityscapes and CamVid test sets, and compare with state-of-the-art networks to demonstrate the efficiency and value of MFNet, based on mIoU, inference speed, FLOPs, etc. Finally, some results of the KITTI and Gatech datasets are presented for further proof of the effectiveness of MFNet. Fig. 10, 11, 12, and 13 depict visual outputs of semantic segmentation.

A. Experimental Settings

1) Cityscapes Dataset: The Cityscapes dataset contains 50 cities' street scenes on a large scale. 5000 fine pixel-level annotated images are grouped into three parts, covering 2975 training images, 500 validation images, and the remaining 1525 testing images. There are also additional 20,000 weakly annotated coarse images, not used in our experiments. We use 5000 high-resolution annotated images with 19 classes and a size of 1024 × 2048. We base classes IoU on 19 classes such as "road", "car", "building", "vegetation", etc. and base Categories IoU on 8 categories containing "flat", "object", "nature", "construction", etc. Input resolution for training is 512 × 1024.

2) CamVid Dataset: The CamVid is a popular semantic road understanding dataset. 701 images are divided into 367 images and 101 images for training and validating, and 233 images for testing. Referring to the number and size, the CamVid dataset is smaller than the Cityscapes dataset. The resolution of original images is 720 × 960 with 11 labeled classes. They are randomly cropped into 512 × 512 resolution for network input.

3) KITTI Dateset: The KITTI labeled by Xu *et al.* is the smallest dataset in number of these four datasets. 107 images with the resolution of 368 × 1200 are separated into groups of a training set of 70 and a validation and testing set of 37, respectively. Input images are cropped into 256 × 800 and we train five categories combined from the initial 28 classes are used for training.

4) Gatech Dateset: The Gatech is a video dataset including 63 training videos and 38 testing sequences. One out of five frames (1249 images total) is selected as the training set, and one out of two frames (1426 in all) constitutes the testing set, as images of the adjacent frames are very similar. The resolution of this dataset is 480 × 640, with 8 classes. Image sizes are cropped to 256 × 480 for training.

5) Implementation Details: A single NVIDIA Titan Xp GPU with CUDA and cuDNN backend is used on the Pytorch framework in Python in our experiment. In order to obtain better results with limited data, we apply data augmentation by random scale with scale factors {0.75, 1.0, 1.25, 1.5, 1.75, 2.0}, mean subtraction, random crop and random horizontal flipping while training. The border expansion is executed before the random crop if the dealt images are smaller than the crop size. The max epochs are set to 1000, with no pre-training or post-processing, and results will be better with larger max epochs. Cross-entropy loss ignoring the unlabeled pixels, frequently employed in classification tasks to calculate output loss, is the loss function used in our experiment. Furthermore, different classes in the dataset are so imbalanced that each class's weight has to be figured in the training dataset. Class weight is defined as $w_{class} = 1/\log(1.1 + p_{class})$, where p_{class} is the probability of a class.

For Cityscapes, a Stochastic Gradient Descent (SGD) optimizer is adopted with momentum 0.9 and weight decay of 10^{-4}. Following DeepLab-v2 [35], the initial learning rate is 0.02, multiplied by (1-current_iter/max_iter)$^{0.9}$ later, known as 'poly' learning rate strategy. We take full advantage of the GPU by the batch size of 10.

For CamVid, KITTI, and Gatech, the Adam optimizer was employed with the original learning rate of 10^{-3}, calculated by 'poly' learning rate, parameter betas from 0.9 to 0.999, and the weight decay of 2×10^{-4}. 16 batch size is the modest during training.

6) Evaluation Metrics: Mean IoU (mIoU), global accuracy, and frames per second (FPS), extensively used in semantic

TABLE II

ABLATION STUDY ON CITYSCAPES VAL SET, DEMONSTRATING THE EFFECTIVENESS OF THE THREE BRANCHES

Model	A	S	SI	Parameters(M)↓	Speed(FPS)↑	FLOPs(G)↓	mIoU(%)↑
M1		✓	✓	1.22	120	8.9	69.8
M2	✓		✓	1.35	125	7.5	63.0
M3	✓	✓		1.21	112	9.0	66.5
MFNet	✓	✓	✓	1.34	116	9.1	**71.6**

[1] The attention, semantic, and spatial information branches are removed in turn in M1, M2, and M3, respectively.

segmentation and real-time road scenes understanding are evaluation metrics in this paper to exhibit our results. They are defined as follows:

$$t_i = \sum_j n_{ij} \tag{1}$$

$$mIoU = \frac{1}{n_{cls}} \sum_i \left(\frac{n_{ii}}{t_i + \sum_j n_{ji} - n_{ii}} \right) \tag{2}$$

$$Global\ Accuracy = \frac{\sum_i n_{ii}}{\sum_i t_i} \tag{3}$$

$$FPS = \frac{1}{t} \tag{4}$$

where n_{ij} expresses the number of pixels of class i mistakenly predicted to class j, t_i is the entire number of pixels of class i, n_{cls} represents the number of classes in each dataset and t signifies the time it takes to process one image.

B. Ablation Study

In this subsection, the multi-branch structure of MFNet is compared and analyzed on the Cityscapes validation set. Some experiments, which are shown in TABLE II, III, IV, V, VI, VII, VIII are conducted to testify the necessity and superiority of three branches and explore the potential of MFNet.

1) Ablation Study for Three Branches: As can be seen in TABLE II, employing all three branches achieves the optimal results. M1 (without the attention branch) decreases mIoU by 1.8%, which demonstrates the utility of the attention block. Although the speed of M2 (without the semantic branch) is slightly faster, the effect goes down by 8.6% overall because the semantic branch is significant for semantic segmentation. M3 (without the spatial information branch) reduces mIoU by 5.1% due to the lack of low-level features.

2) Ablation Study for the Attention Block: We select the position attention, whose attention map is (h-1)(w-1) × (h-1)(w-1), to get the relationship between different positions and realize long-range dependencies for scene understanding. As can be seen in TABLE III, to testify the performance of the position attention block, we implement the contrast experiment using channel attention with attention map c × c. M4 (with the channel attention) proves to be lower by 3.1% mIoU than MFNet, which suggests that position attention is more suitable for MFNet.

TABLE III

ABLATION STUDY ON CITYSCAPES VAL SET, DEMONSTRATING THE EFFECTIVENESS OF THE POSITION ATTENTION

Model	mode	Parameters(M)↓	Speed(FPS)↑	FLOPs(G)↓	mIoU(%)↑
M4	channel attention	1.31	116	9.1	68.5
MFNet	position attention	1.34	116	9.1	**71.6**

TABLE IV

ABLATION STUDY ON CITYSCAPES VAL SET, CHANGING SOME PARAMETERS IN THE ATTENTION MODULE

Model	mode	Parameters(M)↓	Speed(FPS)↑	FLOPs(G)↓	mIoU(%)↑
M21	FFB-Concat	1.34	139	9.1	70.7
MFNet	FFB-Bilinear	1.34	116	9.1	**71.6**

TABLE V

ABLATION STUDY ON CITYSCAPES VAL SET, DEMONSTRATING THE EFFECTIVENESS OF THE AF BLOCK STRUCTURE

Model	mode	Parameters(M)↓	Speed(FPS)↑	FLOPs(G)↓	mIoU(%)↑
M10	AF-RR	1.34	107	9.2	70.2
M11	AF-FF	1.34	109	9.1	69.8
M12	AF-FR	1.34	110	9.2	68.4
M13	AF-ff	1.33	117	9.1	69.1
M14	AF-LED	1.06	124	7.8	70.5
M15	AF-EAC1	1.74	95	11.1	69.6
M16	AF-EAC2	2.56	71	15.1	69.8
MFNet	AF-fF	1.34	116	9.1	**71.6**

[1] The structures of M10, M11, M12, M13 are shown in Fig. 9.
[2] M15 means all AF blocks in MFNet are replaced by EAC-A in Fig. 8(b).
[3] M16 replaces AF block1 with EAC-A in Fig. 8(b) and AF block2 with EAC-B in Fig. 8(c).

TABLE VI

ABLATION STUDY ON CITYSCAPES VAL SET, CHANGING THE NUMBER OF AF BLOCKS IN AF BLOCK 1 AND AF BLOCK 2

Model	mode	Parameters(M)↓	Speed(FPS)↑	FLOPs(G)↓	mIoU(%)↑
M17	AF-14	1.15	131	7.6	69.3
M18	AF-26	1.48	107	9.4	69.9
M19	AF-35	1.37	91	10.4	70.6
M20	AF-36	1.52	89	10.7	70.1
MFNet	AF-25	1.34	116	9.1	**71.6**

TABLE IV shows some changes to max-pooling and shared weights in the attention module. M5 (without max-pooling) and max-pooling module: M6 (kernel size 3, stride 1), M7 (kernel size 4, stride 1), M8 (kernel size 2, stride 2) do not reach the mIoU of MFNet (max-pooling with kernel size 2, stride 1). The mIoU of M9 (without shared weights) is 1.1% lower than MFNet. This proves that MFNet employs applicable max-pooling and shared weights to reserve crucial information and filter redundant information.

3) Ablation Study for AF Block: Our core block plays an important role in the semantic branch, we explore the

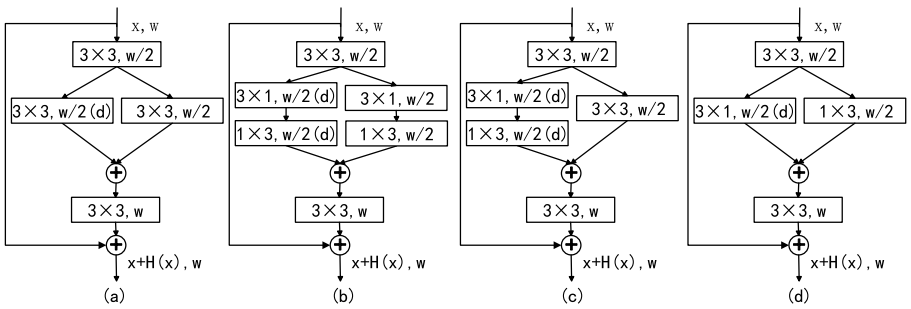

Fig. 9. (a) M10 AF-RR, (b) M11 AF-FF, (c) M12 AF-FR, (d) M13 AF-ff. 'x' represents input images. 'w' signifies input channels. 'd' means dilated convolution. R: 3 × 3, F: 3 × 1 and 1 × 3, f: 3 × 1.

TABLE VII
ABLATION STUDY ON CITYSCAPES VAL SET, DEMONSTRATING THE EFFECTIVENESS OF THE FEATURE FUSION BLOCK

Model	mode	Parameters(M)↓	Speed(FPS)↑	FLOPs(G)↓	mIoU(%)↑
M21	FFB-Concat	1.34	139	9.1	70.7
MFNet	FFB-Bilinear	1.34	116	9.1	**71.6**

TABLE VIII
ABLATION STUDY ON CITYSCAPES VAL SET, ADDING AUXILIARY LOSS

Model	mode	Parameters(M)↓	Speed(FPS)↑	FLOPs(G)↓	mIoU(%)↑
M22	+loss2	1.34	105	9.2	70.1
M23	+0.5loss2	1.34	105	9.2	69.8
M24	+0.2loss2	1.34	105	9.2	70.2
M25	+loss2+loss3	1.34	102	9.2	69.9
M26	+0.5loss2+0.5loss3	1.34	102	9.2	69.8
M27	+0.2loss2+0.2loss3	1.34	102	9.2	69.5
MFNet	loss	1.34	116	9.1	**71.6**

most efficient structure from five different AF blocks and modules in LEDNet and in EACNet, as shown in Fig. 7(d), Fig. 9(a)(b)(c)(d) and Fig. 8(a)(b)(c). As can be seen in TABLE V, M10, M11, M12, M13, M14, M15, M16 perform 1.4%, 1.8%, 3.2%, 2.5%, 1.1%, 2%, 1.8% mIoU lower than MFNet, respectively, making it apparent that a combination of factorized convolution and non-dilated 1D convolution produces better results.

TABLE VI explores the impact of different numbers of AF blocks on MFNet. MFNet (AF block 1 and AF block 2 set to 2 and 5, respectively) is more effective than AF block 1 and AF block 2 set to (1, 4) in M17, (2, 6) in M18, (3, 5) in M19 and (3, 6) in M20.

4) Ablation Study for the Feature Fusion Block (FFB): As can be seen in TABLE VII, M9 adopts simple concatenating instead of the feature fusion block to achieve 70.7% mIoU. MFNet invests the feature fusion block with bilinear interpolation, improving accuracy from 70.7% to 71.6%.

5) Ablation Study for Loss: To take advantage of different branches, we introduce loss2 and loss3 in two places in Fig. 2. The experiments in TABLE VIII prove that MFNet without auxiliary loss has a higher mIoU than adding loss at loss2 or both loss2 and loss3. This shows that the features obtained by each branch alone do not have a good predictive effect on the final result. The final segmentation requires the joint action of the three branches. The importance of the feature fusion block that fuses features of the three branches is proved from the side.

C. Comparison With Other State-of-the-Art Models

In this subsection, we compare the speed, FLOPs, required memory and accuracy of our proposed method with other state-of-the-art models. Results on the Cityscapes, CamVid, KITTI, and Gatech datasets are provided in TABLE IX, X, XI, XII respectively. MFNet is observed to perform with fast speed, low FLOPs, low required memory and high accuracy compared to other models. What counts is that, there are no testing tricks such as multi-crop and multi-scale testing at all in our network during evaluation. To ensure fairness, in CamVid, KITTI and Gatech, speed is computed under the same environment with regards to GPU and the image resolution.

1) Cityscapes: We report the comparison of speed, FLOPs, Memory and accuracy (mIoU) on the Cityscapes dataset in TABLE IX. MFNet achieves 72.1% mIoU without any pre-training or post-processing, such as CRFs, at a speed of 116 FPS by processing 512 × 1024 images. Compared to high-accuracy methods without real-time capability, we only use approximately 0.5% parameters to achieve comparable results with DeepLab and PSPNet. Compared to networks pretrained on ImageNet, such as SegNet and ICNet, our network not only has improved 15.1%, 2.6% mIoU on accuracy, respectively, but also performs better in terms of speed, FLOPs and required memory. When compared to light-weight networks trained from scratch (ENet, ERFNet, LEDNet, EDANet, DABNet and CGNet), MFNet stands out among them on both speed and accuracy. In the same environment, although ESPNet and FRNet are a little faster than MFNet, MFNet achieves a higher mIoU. FLOPs, Memory are roughly proportional to the input size. When they are converted to the same input size, FLOPs of MFNet are only higher than four networks (ENet, EDANet, ESPNet and Lite-HRNet-18) and required memory is only higher than two networks (ESPNet and SwiftNet).

TABLE IX
COMPARISONS BETWEEN MFNET AND OTHER STATE-OF-THE-ART NETWORKS ON THE CITYSCAPES TEST SET

Model	Input Size	Pretrained	Parameters(M) ↓	Speed(FPS)↑	FLOPs(G)↓	Memory(M)↓	GPU	mIoU(%)↑
DeepLab [35]	512×1024	✓	262.1	0.25	457.8	5945	-	63.1
PSPNet [4]	713×713	✓	250.8	0.78	412.2	3935	-	81.2
SegNet [8]	360×640	✓	29.5	16.7	286.0	-	GTX Titan XM	57
ICNet [36]	1024×2048	✓	26.5	30.3	28.3	-	GTX Titan XM	69.5
ENet [9]	360×640	x	0.4	135.4	3.8	-	GTX Titan XM	58.3
ERFNet [10]	512×1024	x	2.1	41.7	27.7	866	GTX Titan XM	68.0
LEDNet [18]	512×1024	x	0.9	71	11.5	761	GTX 1080Ti	69.2
EDANet [37]	512×1024	x	0.7	81.3	9.0	354	GTX Titan X	67.3
DABNet [38]	512×1024	x	0.8	104	10.4	817	GTX 1080Ti	70.1
CGNet [39]	360×640	x	0.5	35.2	6	334	GTX 2080Ti	64.8
ESPNet [11]	512×1024	x	0.4	112	4.7	-	GTX Titan X	60.3
FRNet [19]	512×1024	x	1.0	127	12.9	632	GTX Titan XP	70.4
SwiftNet [21]	1024×2048	✓	11.8	39.9	104	1672	GTX 1080Ti	75.5
Lite-HRNet-18 [22]	512×1024	x	1.1	-	2.0	-	NVIDIA V100	72.8
Trans4Trans [23]	768×768	x	13.5	-	20.7	-	GTX 1080Ti	78.2(MS)
SegFormer [24]	1024×1024	✓	3.8	15.2	125.5	-	Tesla V100	76.2
MFNet(Ours)	512×1024	x	1.34	116	9.1	496	GTX Titan XP	72.1

[1] '-' means the methods do not report the corresponding results.
[2] FLOPs: Floating point operations.
[3] Memory: the required memory footprint for the model. "Memory" is the result of our run, and the other data is from their papers.
[4] "MS" means multi-scale testing.
[5] Speed of CGNet is measured at 1024×2048 resolution.

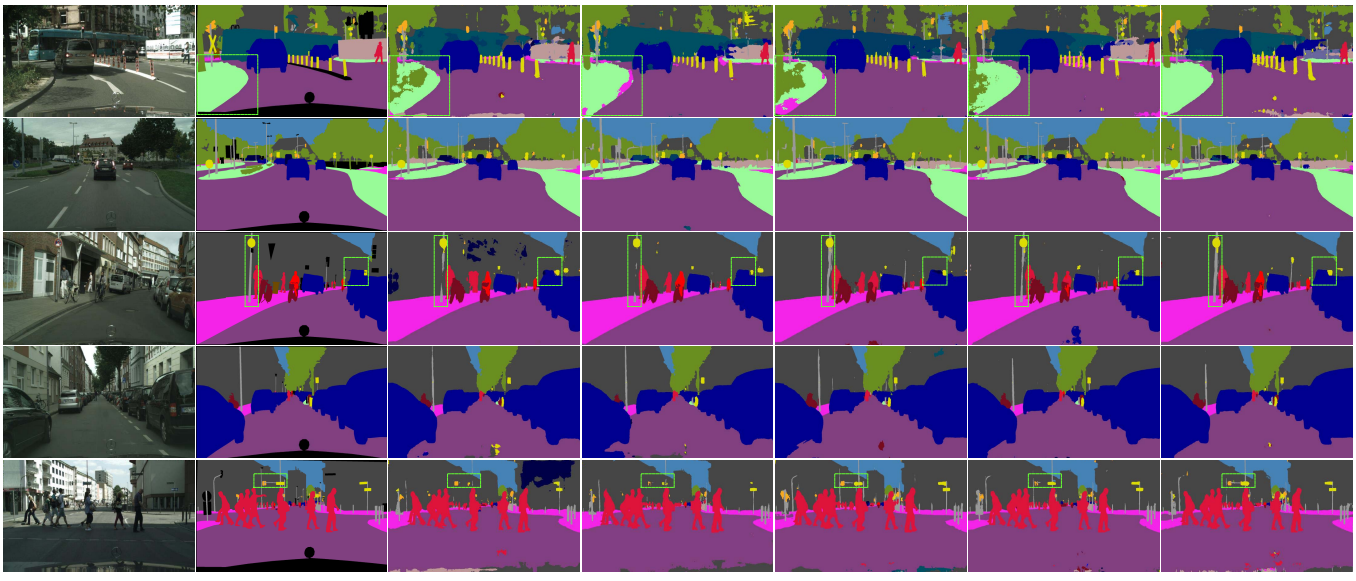

Fig. 10. Visible results on the Cityscapes validation set.

The accuracy of segmentation is related to many factors other than the performance of the network, such as: input size, pre-training, computing resources, multi-scale testing, etc. SwiftNet (large input size), Lite-HRNet-18 (8 NVIDIA Tesla V100 GPUs), Trans4Trans (multi-scale testing, large input size) and SegFormer(8 NVIDIA Tesla V100 GPUs, large input size, pre-training) attain higher mIoU than MFNet.

The segmentation images of MFNet on the Cityscapes validation set are shown in Fig. 10. It can be seen that the segmentation effect of most categories, such as car, bicycle, road, and sky is good, with accurate contour and less noise.

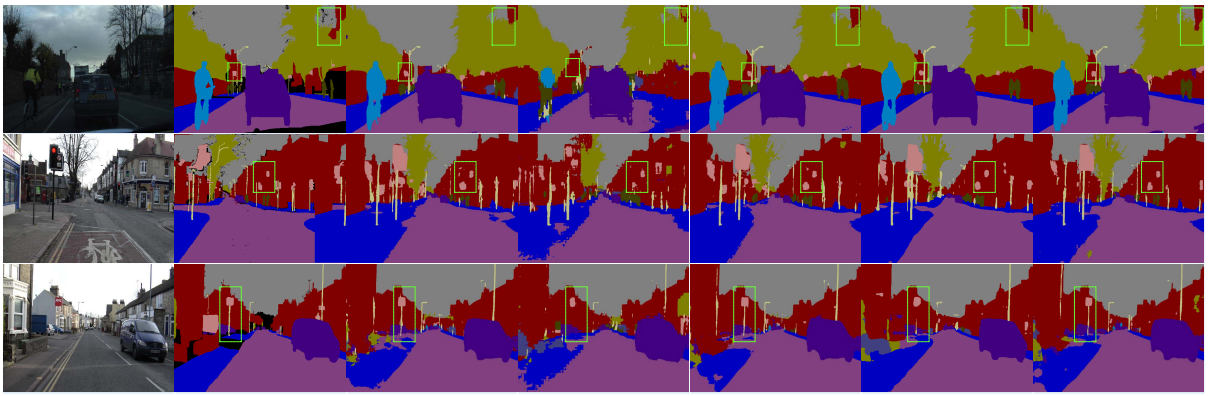

Fig. 11. Visible results on the CamVid test set.

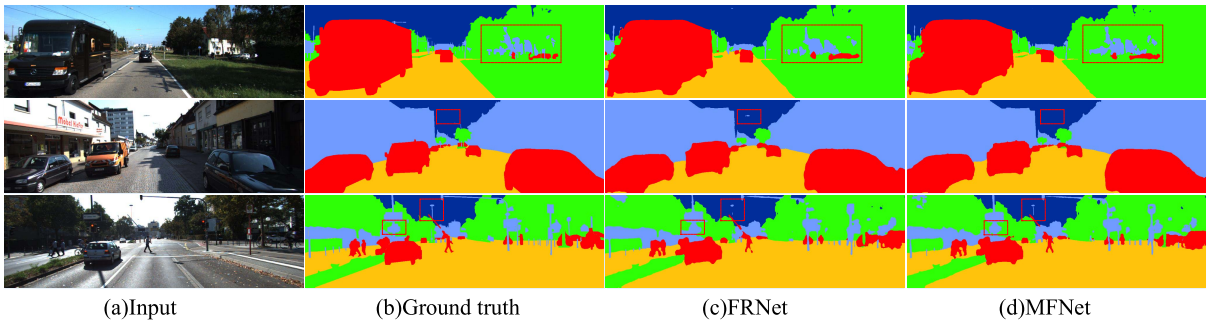

Fig. 12. Visible results on the KITTI test set.

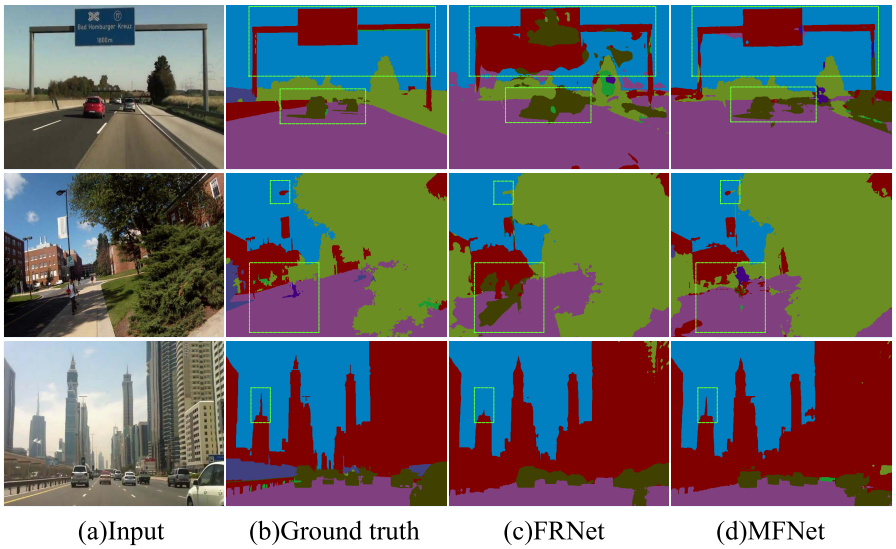

Fig. 13. Visible results on the Gatech test set.

MFNet also performs well on small object segmentation, for example signvegetation, pole and traficlight.

2) CamVid: Results on the CamVid dataset can be seen from TABLE X, in which MFNet achieves 71.5% mIoU at a speed of 145 FPS by manipulating images of resolution 512 × 512. The accuracy of MFNet comes first among the ten light-weight networks in TABLE X. The speed of MFNet is only slower than ESPNet and FRNet. The segmentation images on the CamVid dataset are in Fig. 11, which exhibit that MFNet can effectively segment road scenes.

3) KITTI: MFNet achieves 93.7% mIoU and 132 FPS with image size 368 × 1200. Although there are only 107 images for training, MFNet obtains the highest accuracy among these four datasets on account of merely 5 categories to be predicted. The results as shown in TABLE XI exhibit the excellence of our network with regard to global accuracy and speed. The experiment on KITTI demonstrates that MFNet performs excellently on small datasets. The segmentation images in Fig. 12 shows that MFNet handles details well, and the segmented objects do not stick to the surroundings.

TABLE X
EVALUATION RESULTS ON THE **CAMVID TEST SET**

Model	Parameters(M)↓	Speed(FPS)↑	mIoU(%)↑
SegNet [8]	29.5	36	55.6
ICNet [36]	7.8	49	67.1
ENet [9]	0.4	78	51.3
EDANet [37]	0.7	126	66.4
DABNet [38]	0.8	124	66.4
CGNet [39]	0.5	72	65.6
ESPNet [11]	0.4	**202**	55.6
FRNet [19]	1.0	157	67.4
SwiftNet [21]	11.8	-	63.3
MFNet(Ours)	1.34	145	**71.5**

[1] Speed is computed on a single Titan Xp with a resolution of 512×512.

TABLE XI
EVALUATION RESULTS ON THE **KITTI TEST SET**

Model	Speed(FPS)↑	Global Acc(%)↑	mIoU(%)↑
Probabilistic Fusion [13]	-	79.0	-
Evidential Fusion [13]	-	81.4	-
FRNet [19]	**147**	96.5	92.5
MFNet(Ours)	132	**97.1**	**93.7**

[1] Speed is computed on a Titan Xp with a resolution of 368×1200.

TABLE XII
EVALUATION RESULTS ON THE **GATECH TEST SET**

	Model	Speed(FPS)↑	Global Acc(%)↑	mIoU(%)↑
3D models	3D-V2V [40]	-	66.7	-
	3D-V2V-Pretrained [40]	-	76.0	-
2D models	2D-V2V [40]	-	55.7	-
	FRNet [19]	**161**	80.9	-
	MFNet(Ours)	151	**81.1**	**46.3**

[1] Speed is computed on a single Titan Xp with a resolution of 480×640.

4) Gatech: MFNet achieves 81.1% global accuracy on the Gatech dataset and is the highest among the compared networks, as can be seen in TABLE XII. With an image resolution of 480 × 640, the inference speed attains 151 FPS, which meets the real-time requirements for the scene understanding task. From visualized outputs seen in Fig. 13, small classes may sometimes be confused with surroundings, owing to some incorrect labeling. Even so, MFNet can perform fine segmentation.

V. CONCLUSION

In this paper, we have proposed a novel Multi-Feature Fusion Network (MFNet) with asymmetric factorized (AF) blocks for real-time road semantic segmentation. The efficiency and accuracy of our method, identified by results on Cityscapes, CamVid, KITTI and Gatech datasets, is owed to its three branches and AF blocks. MFNet achieves 72.1% mIoU on the Cityscapes test set at a speed of 116 FPS. In conclusion, MFNet cannot only meet real-time requirements, but also predict accurately for road semantic segmentation.

One limitation is that the data for MFNet comes from computer experiments and MFNet is not embedded in the device. In the future, we will embed MFNet into practical devices and further improve the accuracy and speed of MFNet.

REFERENCES

[1] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, Apr. 2017.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 2, pp. 84–90, Jun. 2017.

[3] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–14.

[4] 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. 6230–6239.

[5] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017, *arXiv:1706.05587*.

[6] 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.

[7] 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.

[8] 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.

[9] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "ENet: A deep neural network architecture for real-time semantic segmentation," 2016, *arXiv:1606.02147*.

[10] E. Romera *et al.*, "ERFNet: Efficient residual factorized convnet for real-time semantic segmentation," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 1, pp. 263–272, Oct. 2018.

[11] 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.

[12] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla, "Segmentation and recognition using structure from motion point clouds," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2008, pp. 44–57.

[13] P. Xu, F. Davoine, J.-B. Bordes, H. Zhao, and T. Denœux, "Multimodal information fusion for urban scene understanding," *Mach. Vis. Appl.*, vol. 27, no. 3, pp. 331–349, Apr. 2016.

[14] S. H. Raza, M. Grundmann, and I. Essa, "Geometric context from videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 3081–3088.

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

[16] R. P. Poudel, U. Bonde, S. Liwicki, and C. Zach, "ContextNet: Exploring context and detail for semantic segmentation in real-time," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2018, pp. 1–12.

[17] R. P. Poudel, S. Liwicki, and R. Cipolla, "Fast-SCNN: Fast semantic segmentation network," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2019, pp. 1–9.

[18] Y. Wang *et al.*, "LedNet: A lightweight encoder–decoder network for real-time semantic segmentation," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 1860–1864.

[19] M. Lu, Z. Chen, Q. M. J. Wu, N. Wang, X. Rong, and X. Yan, "FRNet: Factorized and regular blocks network for semantic segmentation in road scene," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 4, pp. 3522–3530, Apr. 2022.

[20] X. Zhang, Z. Chen, Q. M. J. Wu, L. Cai, D. Lu, and X. Li, "Fast semantic segmentation for scene perception," *IEEE Trans. Ind. Informat.*, vol. 15, no. 2, pp. 1183–1192, Feb. 2019.

[21] 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/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12599–12608.

[22] C. Yu *et al.*, "Lite-HRNet: A lightweight high-resolution network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 10440–10450.
[23] J. Zhang, K. Yang, A. Constantinescu, K. Peng, K. Müller, and R. Stiefelhagen, "Trans4Trans: Efficient transformer for transparent object and semantic scene segmentation in real-world navigation assistance," *IEEE Trans. Intell. Transp. Syst.*, early access, Mar. 28, 2022, doi: 10.1109/TITS.2022.3161141.
[24] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 1–14.
[25] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–15.
[26] J. Fu *et al.*, "Dual attention network for scene segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3141–3149.
[27] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, "CCNet: Criss-cross attention for semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 603–612.
[28] P. Hu *et al.*, "Real-time semantic segmentation with fast attention," *IEEE Robot. Autom. Lett.*, vol. 6, no. 1, pp. 263–270, Jan. 2021.
[29] K. Yang, J. Zhang, S. Reis, X. Hu, and R. Stiefelhagen, "Capturing omni-range context for omnidirectional segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 1376–1386.
[30] 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. 5168–5177.
[31] 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)*, 2018, pp. 325–341.
[32] Y. Pei, B. Sun, and S. Li, "Multifeature selective fusion network for real-time driving scene parsing," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–12, 2021.
[33] L. Sun, K. Yang, X. Hu, W. Hu, and K. Wang, "Real-time fusion network for RGB-D semantic segmentation incorporating unexpected obstacle detection for road-driving images," *IEEE Robot. Autom. Lett.*, vol. 5, pp. 5558–5565, 2020.
[34] Y. Li, X. Li, C. Xiao, H. Li, and W. Zhang, "EACNet: Enhanced asymmetric convolution for real-time semantic segmentation," *IEEE Signal Process. Lett.*, vol. 28, pp. 234–238, 2021.
[35] 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.
[36] 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.
[37] S.-Y. Lo, H.-M. Hang, S.-W. Chan, and J.-J. Lin, "Efficient dense modules of asymmetric convolution for real-time semantic segmentation," in *Proc. ACM Multimedia Asia*, 2019, pp. 1–6.
[38] G. Li, I. Yun, J. Kim, and J. Kim, "DABNet: Depth-wise asymmetric bottleneck for real-time semantic segmentation," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2019, pp. 1–12.
[39] T. Wu, S. Tang, R. Zhang, J. Cao, and Y. Zhang, "CGNet: A light-weight context guided network for semantic segmentation," *IEEE Trans. Image Process.*, vol. 30, pp. 1169–1179, 2021.
[40] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Deep End2End Voxel2Voxel prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2016, pp. 402–409.

Mengxu Lu was born in Jiangsu, China, in 1997. She received the B.S. degree in automation from the School of Control Science and Engineering, Shandong University, Jinan, China, in 2019, where she is currently pursuing the M.S. degree in control science and engineering. Her research interests include machine learning, deep learning, and semantic segmentation.

Zhenxue Chen was born in Shandong, China, in 1977. He received the B.S. degree in automatic from the School of Electrical Engineering and Automation, Shandong Institute of Light Industry, Jinan, China, in 2000, the M.S. degree in computer science from the School of Information Science and Engineering, Wuhan University of Science and Technology, Wuhan, China, in 2003, and the Ph.D. degree in pattern recognition and intelligent systems from the Institute of Image Recognition and Artificial Intelligence, Huazhong University of Science and Technology, Wuhan, in 2007. From 2012 to 2013, he was a Visiting Scholar with Michigan State University, East Lansing, MI, USA. He is currently a Professor with the School of Control Science and Engineering, Shandong University. He has published over 100 papers in refereed international leading journals/conferences, such as IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, *Information Sciences*, *Neurocomputing*, *Neural Computing and Applications*, and *SP-IC*. His main areas of interest include image processing, pattern recognition, and computer vision, with applications to face recognition.

Chengyun Liu was born in Henan, China, in 1975. She received the B.S. degree in communication from Huazhong Normal University, Wuhan, China, in 1999, the M.S. degree in pattern recognition and intelligent systems from the Wuhan University of Science and Technology, Wuhan, in 2005, and the Ph.D. degree in pattern recognition and intelligent systems from Shandong University, Jinan, China, in 2016. She is currently an Associate Professor with the School of Control Science and Engineering, Shandong University. Her research interests include automatic target detection and recognition, image processing, and computer vision.

Sile Ma was born in Shandong, China, in 1965. He is currently a Professor with the School of Control Science and Engineering and the Institute of Oceanography, Shandong University. He is a Doctoral Supervisor. He is engaged in distributed control systems, unmanned aerial vehicle inspection, and computer vision.

Lei Cai was born in Henan, China, in 1979. He received the B.S. degree from the Logistics Institute of Air Force, Xuzhou, China, in 2002, the M.S. degree from the Xuzhou Air Force College, Xuzhou, in 2006, and the Ph.D. degree from Air Force Engineering University, Xi'an, China, in 2009. He has been a Professor with the School of Artificial Intelligence, Henan Institute of Science and Technology. His research areas are image processing, pattern recognition, light field reconstruction, and multi-agent adaptive coordination.

Hao Qin was born in Shandong, China, in 1998. He received the B.S. degree in automation from the School of Control Science and Engineering, Shandong University, Jinan, China, in 2020, where he is currently pursuing the M.S. degree in control science and engineering. His research interests include machine learning, deep learning, and gait recognition.