Complex & Intelligent Systems (2023) 9:6427–6436
https://doi.org/10.1007/s40747-023-01046-y

ORIGINAL ARTICLE

DMBR-Net: deep multiple-resolution bilateral network for real-time and accurate semantic segmentation

Pengfei Meng[1] · **Shuangcheng Jia**[1] · **Qian Li**[1]

Received: 15 November 2022 / Accepted: 9 March 2023 / Published online: 12 May 2023

Abstract
It has been proved that the two-branch network architecture for real-time semantic segmentation is effectiveness. However, existing methods still can not obtain sufficient context information and sufficient detailed information, which limits the improvement of the accuracy of existed two-branch methods. In this paper, we proposed a real-time high-precision semantic segmentation network based on a novel multi-resolution feature fusion module, an auxiliary feature extracting module, an upsampling module and multi-ASPP(atrous spatial pyramid pooling) module. We designed a feature fusion module, which is integrated with sufficient features of different resolutions to help the network get both sufficient semantic information and sufficient detailed information. We also studied the effect of the side-branch architecture on the network, and made new discoveries that the role of the side-branch is more than regularization, it may either slow down the convergence or accelerate the convergence by influencing the gradient of different layers of the network, which is dependent on the parameters of the network and the input data. Based on the new discoveries about the side-branch architecture, we used a side-branch auxiliary feature extraction layer in the network to improve the performance of the network. We also designed an upsampling module, which can get better detailed information than the original upsampling module. In addition, we also re-considered the locations and number of atrous spatial pyramid pooling (ASPP) modules, and modified the network architecture according to the experimental results to further improve the performance of the network. We proposed a network based on the above study. We named this network Deep Multiple-resolution Bilateral Network for Real-time, referred to as DMBR-Net. The network proposed in the paper achieved 81.3% mIoU(Mean Intersection over Union) at 110FPS on the Cityscapes validation dataset, 80.7% mIoU at 104FPS on the CamVid test dataset, 32.2% mIoU at 78FPS on the COCO-Stuff test dataset.

Keywords Multi-resolution feature · Auxiliary feature extraction · ASPP · Upsample

Introduction

Accurate understanding of traffic scenes is very important for automatic driving. Based on the understanding of the traffic scene, automatic driving vehicles predict the track of vehicles or pedestrians. Understanding traffic scenes determines the accuracy of the predicted track, then determines the safety performance of automatic driving vehicle. Compared with the object detection of laser radar, using RGB images can complete object detection in severe weather conditions such as fog, snow, sandstorm, and the cost is low. However, object detection lose the relative position information of the scene, so accurate image semantic segmentation is very helpful for understanding traffic scenes. Semantic segmentation of the elements such as lane marks, road arrows, vehicles, pedestrians, buildings and sidewalks is a key technique to realize automatic driving and automatic high-precision map making. Since fully convolutional networks (FCN) [20] used deep learning to deal with semantic segmentation problems firstly, a series of deep learning networks have been proposed, including U-Net [27], SegNet [1], DeepLab series [4–7], RefineNet [18], PSPNet [41], DeepMask [36], OCRNet [39] and HMS [32]. These methods indicate that semantic segmentation networks must be able to obtain both sufficient semantic information and sufficient detailed information. Even though these models achieved encouraging segmentation accuracy, these networks are too complex to meet the real-time requirements.

✉ Qian Li
liqian@zhidaoauto.com

[1] Mogo Auto Intelligence and Telematics Information Technology Co., Ltd., 36 Beisanhuan Dong Road, Dongcheng District, Beijing, China

For the semantic segmentation of road scenes or street scenes, we don't want to spend too much inference time, and at the same time we want to obtain high accuracy to meet the instantaneity requirements such as automatic driving, real-time high-precision map making. To satisfy the requirements of real-time or running on mobile devices, many effective networks have been proposed. ENet [23] achieved great speed improvement using a lightweight decoder and downsampling in the early stages of the network. ICNet [42] proposed an image cascade network for semantic segmentation, which used semantic information with low resolution and detailed information with high resolution. MobileNetv2 [28] reduced the complexity of the overall model using depthwise separable convolution. Bisenetv2 [37] used two branches to obtain semantic information and detailed information respectively. Although these networks have used some methods to obtain semantic information and detailed information, the problem of insufficient semantic information or detailed information has not been completely solved. These works significantly reduced the delay and memory usage of segmentation models, but their low accuracy limits their real-world application.

To improve the accuracy of the model while maintaining speed of the model, we analyzed the dual branch networks again in this paper. To solve the problem that sufficient context information and sufficient detailed information cannot be obtained in real-time semantic segmentation networks, this paper proposed a feature fusion module. In addition to feature fusion module, we also did some other research on this issue. In a word, this paper proposed a network based on a multi-resolution feature fusion module, an auxiliary feature extraction module, an upsampling module and multi-ASPP(atrous spatial pyramid pooling) module. Compared with other lightweight real-time semantic segmentation networks, our network achieved the highest accuracy at an appropriate speed. For this paper, the main contributions are as follows:

(1) A novel multi-resolution feature fusion module that can be flexibly embedded into any network.
(2) An upsampling module that goes beyond deconvolution and it is very easy to use.
(3) A method of using multi ASPP to improve network performance.

Related work

Since the design concept of real-time networks is very different from that of ordinary networks, we group the related works into two categories, i.e., high-performance semantic segmentation and real-time semantic segmentation.

High-accuracy semantic segmentation

The early semantic segmentation networks usually used the encoder–decoder architecture [1,20,27]. The encoder gradually expands its receptive field through convolution or pooling and the decoder uses deconvolution or upsampling to recover the resolution from the outputs of the encoder. However, it is difficult to retain sufficient details during the downsampling process of the encoder–decoder network. A strategy is to utilize dilated convolutions, which could enlarge field-of-view without reducing the spatial resolution. Based on this idea, DeepLab series [4–7] achieved great improvement by using dilated convolutions with different expansion rates in the network. However, DeepLabs only used this method after the last downsampling in the network, and did not further explore this method. PSPNet [43] proposed a Pyramid Pooling Module (PPM) to process multi-scale context information. HRNet [34] adopts multi-path and bilateral connection to learn and integrate features of different scales. Inspired by the self-attention mechanism [33] used in machine translation to obtain context information, many meaningful works for semantic segmentation [10,14,34] introduced non-local operation [35] into computer vision. A critical problem for the above methods are that they are very time-consuming in the inference stage and can not meet the real-time requirements.

Real-time semantic segmentation

Real-time semantic segmentation networks generally adopts two methods: encoder–decoder method and two pathway method.

SwiftNet [22] used a lightweight decoder that used a low-resolution input to obtain semantic information and another high-resolution input to obtain detailed information. DFANet [16] used a light-weight backbone based on depthwise separable convolution, and reduced input size to speed up inference. ShuffleSeg [11] used ShuffleNet [40] as its backbone. ShuffleNet combined channel shuffling and group convolution, which can significantly reduce computing cost. However, these networks still adopted the encoder–decoder architecture, the encoder extracted contextual information by a deep network and the decoder restored the resolution by interpolation or transposed convolution to complete dense predictions. The encoder–decoder architecture loses some details during repeated downsampling, and these lost information cannot be completely recovered through upsampling, which will impairs the accuracy of semantic segmentation. With this consideration, BiSeNet [37,38] proposed a two-branch network architecture, which uses one pathway for extracting semantic information, and the other shallow pathway for extracting detailed information as a supplement. Then several works based on the two-branch network

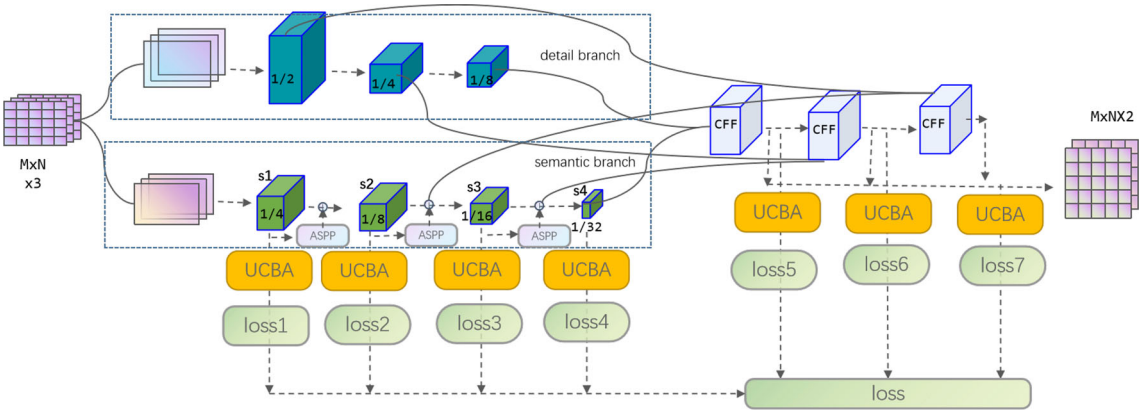

Fig. 1 The network architecture proposed in this paper, M and N represent the length and width of the image respectively

architecture were proposed to improve their presentation capability or reduce the model complexity [12,25,26,37]. However, the problem that the network cannot obtain sufficient context information or detailed information still exists. These networks only weaken the phenomenon of information loss in the process of feature extraction. To further alleviate this problem, this paper proposed a network based on a novel multi-resolution feature fusion module, an auxiliary feature extraction module, an upsampling module and multi-ASPP(atrous spatial pyramid pooling) module.

Method

As shown in the Fig. 1, the network architecture proposed in this paper mainly includes two branches, the detail branch and the semantic branch. The network module names beginning with "loss" are side-branch auxiliary feature extraction layers, which draw on the idea of the auxiliary classifier mentioned in inception-v3 [31]. However, there are new discoveries that are different from the said auxiliary classifier. A detailed description about it will be provided in part 3.1. In the feature fusion stage, we adopted hierarchical cascade fusion base on the cascade feature fusion module(CFF), the output of CFF module are further fused by cascade fusion. A detailed description about CFF module will be provided in part 3.2. Another one specific method is to use the multi ASPP in the feature extraction stage of the semantic branch, which will be described in part 3.4. The upsampling module named UCBA will be described in part 3.3.

The auxiliary feature extraction layer

The Fig. 2 is the architecture of the auxiliary feature extraction layer used in the experiment. The parameters of the first convolution operation are as follows: $kernel\_size = 3 \times 3$, $out\_channels = 2N$, $stride = 1$, $padding = 1$,

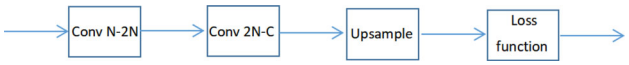

Fig. 2 The architecture of auxiliary feature extraction layer, N represents the number of channels, C represents the number of classes

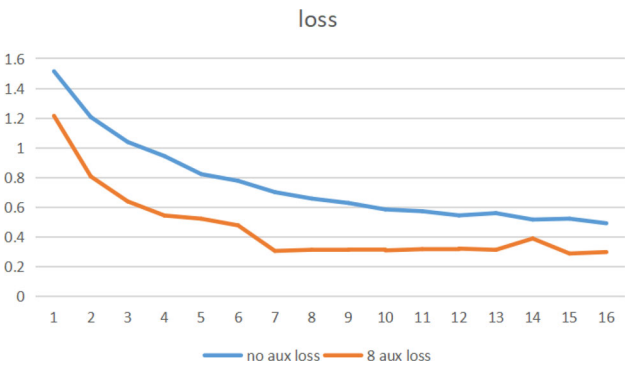

Fig. 3 Comparison of loss downward trend between using auxiliary feature extraction module and not using auxiliary feature extraction module

and the parameters of the second convolution operation are as follows: $kernel\_size = 1 \times 1$, $out\_channels = C$, $stride = 1$, $padding = 0$. Then we used an upsampling module to upsample the feature map from $H/n \times W/n (n = 2, 4, 8, 16, 32)$ size to $H \times W$ size. The upsampling module used here is UCBA, which is described in part 3.3. Finally, we used the cross-entropy to calculate the loss value. Compared with not using the auxiliary feature extraction layer, using the auxiliary feature extraction layer can obtain higher mIoU. The experimental results as shown in Table 4 in the following chapters has proved this. In the process of verifying the effect of this layer, we made a new discovery that the auxiliary feature extraction layer of the network has shown its advantages in the early stage of the training process as shown in Fig. 3. Compared with the network without any auxiliary feature extraction layer, the performance is improved remarkably at the beginning of training.

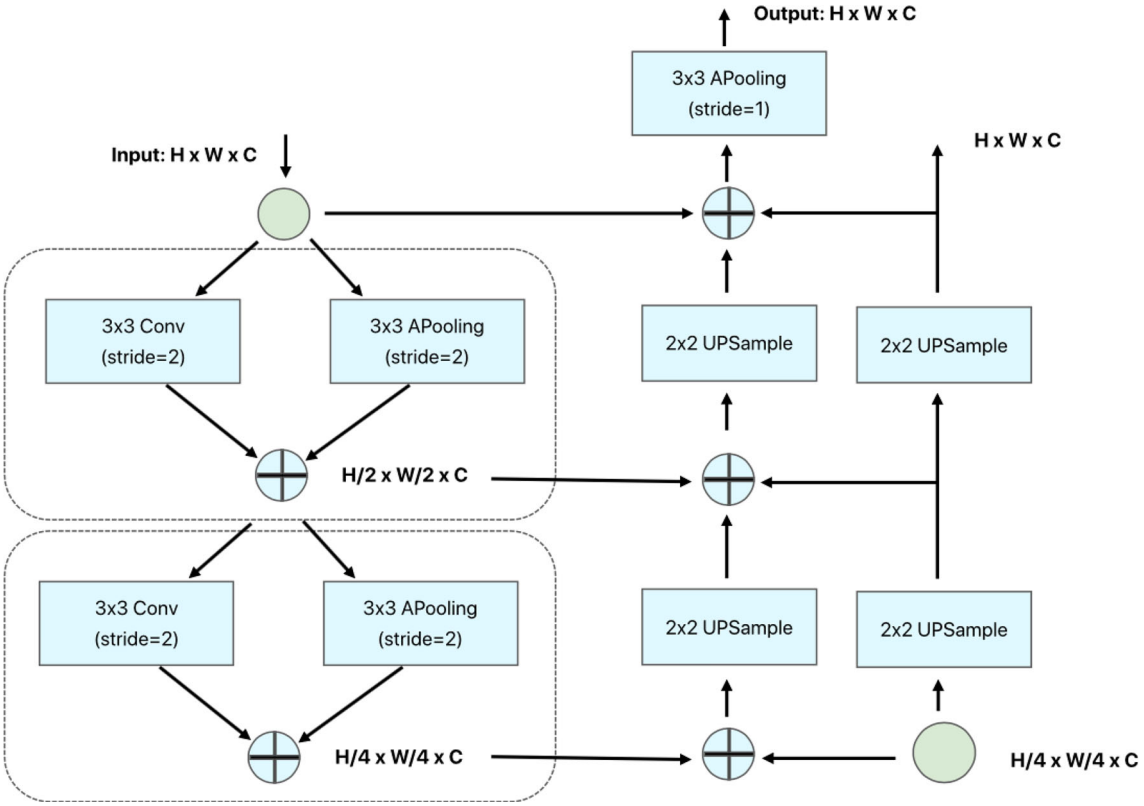

Fig. 4 Cascade feature fusion module

We all know that the side-branch auxiliary feature extraction layer has a regularization effect, which is explained in detail in Inception-v3. To be exact, it has a regularization effect essentially because the auxiliary feature extraction layer increases the gradient of the parameters of the shallow layers, and the gradient of the parameters of the deep layers is reduced relatively. The essence of training a model is to train a function with a huge amount of parameters to fit the real data. If this function is more closely related to the parameters of the shallow layers of the network, the faster adjustment of the parameters of the shallow layers will speed up the convergence of the network. If this function is more closely related to the parameters of the deep layers, the faster adjustment of the parameters of the deep layers will speed up the convergence of the network. The gradient descent method is used to adjust the parameters of the deep learning network during the training process, and the gradient descent method adjusts the parameters according to the gradient of each parameter of the network. The auxiliary feature extraction layers can control the gradient of the parameters of the networks, so it can control the adjustment speed of the parameters of the shallow layers and the deep layers respectively to speed up the convergence of the network. Therefore, the following conclusions can be inferred: the side-branch auxiliary layer may speed up the convergence of the network or slow down the convergence of the network, which is related to the location of the side-branch auxiliary layer, the weight of the side-branch auxiliary layer and the training data. The experimental results of the inception networks and the experiment of our study also further confirmed that.

Cascade feature fusion module

To obtain sufficient context information and sufficient detailed information, we designed a cascade feature fusion module, we named this module CFF (Cascade Feature Fusion Module), the Fig. 4 is the architecture of the CFF module:

For the feature I_d with a shape of $H \times W \times C$ generated by the detail branch (the detail branch in Fig. 1), the feature D_1 with a shape of $H/2 \times W/2 \times C$ is generated by the first down-sampling, and then the second down-sampling performed, a feature D_2 with a shape of $H/4 \times W/4 \times C$ is generated. For the feature I_u with a shape of $H/4 \times W/4 \times C$ generated by the semantic branch (the semantic branch in Fig. 1), the feature U_1 with the shape of $H/2 \times W/2 \times C$ is generated by the first upsampling, then the second upsampling performed, the feature U_2 with the shape $H \times W \times C$ is generated. We use F_u to represent the upsampling function, and the feature O obtained after the fusion can be expressed as Equation (1):

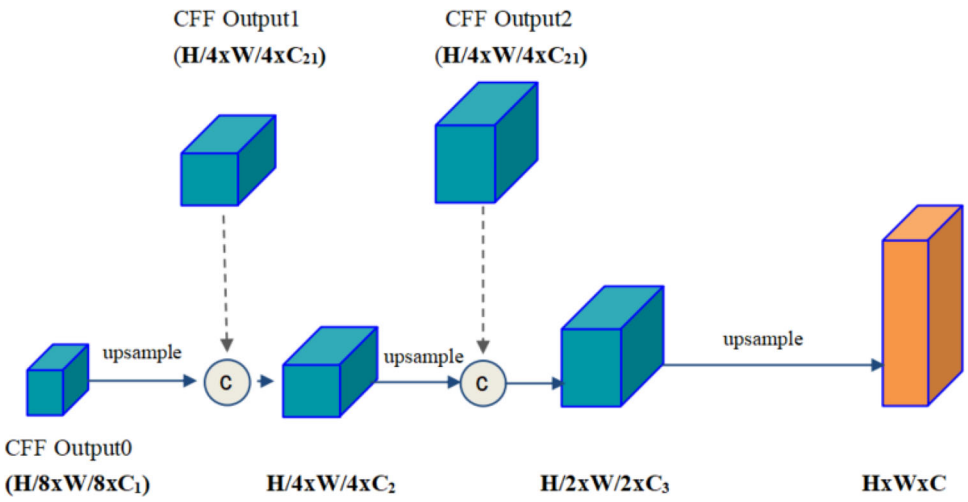

Fig. 5 Fusion mode of CFF module output

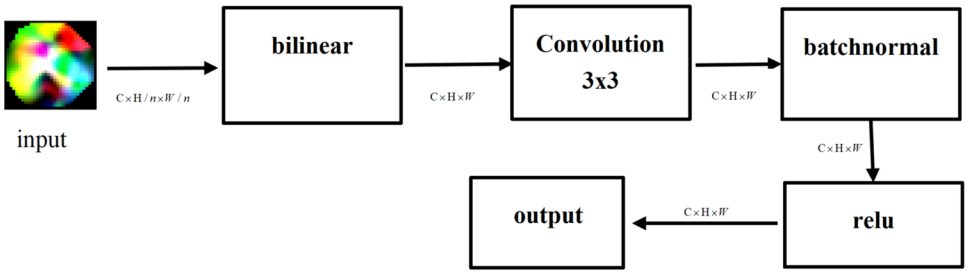

Fig. 6 The architecture of UCBA upsampling module

$$O = F_u(I_u + D_2) + F_u(U_1 + D_1) + F_u(U_2 + I_d) \quad (1)$$

For the obtained feature O, we perform another average pooling operation to obtain the final feature fusion result. This architecture can not only fully integrate features of different sizes, but also expand the receptive field of features by downsampling, enabling the network to obtain richer contextual information.

We used the CFF module mentioned above for feature fusion. From Fig. 1, we can see that we used the CFF module to fuse feature of 1/32 and 1/8, 1/16 and 1/4, 1/8 and 1/2 respectively. We denote the three fused features as O1, O2, and O3 respectively. In order to fuse the features of shallow layers and the features of deep layers, we fused O1, O2 and O3 in series. The specific architecture is shown in the Fig. 5.

CFF module not only produces high-resolution features but also integrates the information of low-resolution features into the high-resolution features, which reduced the excessive influence of noise contained in high-resolution features on the result. CFF module uses a parallel pooling layer to strengthen important information during down-sampling, and it integrates information of high-resolution features into low-resolution features before upsampling as a supplement. CFF module can obtain more richer context information and detailed information than general methods.

More effective upsampling module

Instead of using the common deconvolution [29] or a single bilinear interpolation as the upsampling module, we redesigned an upsampling module with the bilinear, convolution, batch normalization and relu. This module is named UCBA. Then we verified the effectiveness of this upsampling module through experiments. The following Fig. 6 illustrates the architecture of this upsampling module:

Suppose we have a grayscale image with a shape of 4×4, we use a convolution kernel of 3×3 size to perform convolution operations on the image, and finally get a single-channel feature with a shape of 2×2. If we use matrix operations to represent this convolution operation process, the following operations will be performed:

Flatten the 4×4 grayscale image into a matrix X with a shape of 16×1.

$$X = [x_1, x_2, \ldots, x_{16}]^T \quad (2)$$

We flatten the single-channel feature with the shape of 2×2 obtained by the convolution operation into a matrix Y with a shape of 4×1.

$$Y = [y1, y2, y3, y4]^T \quad (3)$$

Suppose we use C to represent the convolution matrix, then the convolution operation can be expressed by the mathematical Equation (4):

$$Y = CX \tag{4}$$

Just fill the 9 parameters corresponding to the convolution kernel into the corresponding positions of the matrix C, and fill the other positions with 0 to get the matrix C. Deconvolution is the inverse operation of the above matrix operation process. The process of deconvolution can be expressed by the following Equation (5):

$$X = C^T Y \tag{5}$$

According to the above deconvolution calculation formula, we can get x_1, we use C_1^T to represent the first row in the matrix C^T, x_1 can be expressed as Equation (6):

$$x_1 = C_1^T Y \tag{6}$$

Other elements in the X matrix are calculated in the same way as x_1. We can clearly find that each element of the X matrix is related to all elements of the Y matrix. We can infer that the deconvolution establishes a global relationship. Furthermore, we can consider that deconvolution has a global receptive field. Many theories and facts have proved that a large receptive field is friendly to semantic information and not friendly to detailed information. For the semantic segmentation task we want to do, the result needs to contain more detailed information for obtaining better segmentation results. The large receptive field can not get detailed information very well, and deconvolution has a global receptive field, so deconvolution for upsampling is not very suitable for semantic segmentation tasks. So the upsampling module in our network adopts the bilinear+convolution+batchnormal+relu module. The subsequent experiments also proved the effectiveness of our upsampling method.

Improved ASPP module

The widely used ASPP module is proposed in the deeplab network, which uses atrous convolution to expand the receptive field of the convolution kernel without losing resolution. The following is the calculation formula for the output size of the atrous convolution. $Input : (N, C_{in}, H_{in}, W_{in}), Output : (N, C_{out}, H_{out}, W_{out})$

$$H_1 = dilation[0] * (kernel\_size[0] - 1) \tag{7}$$
$$H_2 = H_{in} + 2 * padding[0] - H_1 \tag{8}$$
$$H_{out} = \frac{H_2}{stride[0]} + 1 \tag{9}$$
$$W_1 = dilation[0] * (kernel\_size[0] - 1) \tag{10}$$
$$W_2 = W_{in} + 2 * padding[0] - W_1 \tag{11}$$
$$W_{out} = \frac{W_2}{stride[0]} + 1 \tag{12}$$

ASPP performs feature extraction by setting different atrous rates to generate convolution kernels for different receptive fields. It also uses global average pooling to obtain the global receptive field. Intuitively speaking, the purpose of fusing features of different resolutions is to fuse features of different receptive fields. To further integrate the features of different receptive fields, we used the atrous spatial pyramid pooling (ASPP) in the semantic branch, but we used the ASPP module more than once, ASPP modules were used at different locations in the semantic branch.

Many literatures have proved that fusing features of different sizes or resolutions can improve the performance of the network, such as the deeplab series, but the general method is to use the ASPP module only in the last layer of the feature extraction network. Although fusing features of different receptive fields or different sizes can improve the performance of the network, which layer should fuse the features of different receptive fields or different resolutions? There is no such research to answer this question. We did some experimental research on this issue. We added an ASPP layer to the four stages of the semantic branch respectively, the dilation rate of ASPP was set to [5,11,17]. And then we added a shorted connection to add up the original features and the features extracted by the ASPP layer. Thereby, we can fuse the features of large receptive field extracted by ASPP module and the relatively small receptive field features. Table 2 shows that the performance is best if ASPP module and shorted connection are added in the last three stages of a semantic branch. The following Fig. 7 shows how we embed the ASPP module:

Experiment

Datasets

We verified the effectiveness of our method in the public dataset cityscapes [8], which is a semantic understanding image dataset of urban street scenes. It mainly contains street scenes from 50 different cities, and has 5000 high-quality pixel-level annotated images of driving scenes in an urban environment (2975 for train, 500 for val, 1525 for test, 19 categories in total). In addition, it has 20,000 coarsely annotated images (gt coarse). Here we use only 5000 finely labeled data to verify the effectiveness of our model, 2975 for training, and 500 for verification. During training, we first scale the 2048×1024 resolution image to a resolution of 1024×512, and then input it into the network for training.

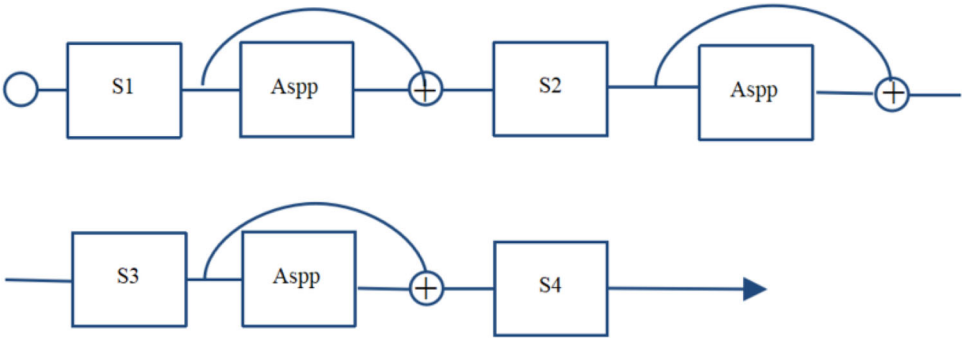

Fig. 7 ASPP module embedding. The meaning and location of *S*1, *S*2, *S*3, *S*4 in the model are shown in Fig. 1

To further verify the effectiveness and wide applicability of the proposed network in this paper, we also conduct experiments on the Cambridge-driving Labeled Video Database (CamVid) [2] and COCO-Stuff datasets [3], respectively. Cambridge-driving Labeled Video Database (CamVid) is a dataset of road scenes from the perspective of autonomous vehicles. It contains 701 images with a resolution of 960 × 720. We use 367 of these images for training and 101 for validation, 233 images were used for testing. We use only 11 of the 32 candidate categories it provides, like other methods, to facilitate comparisons with different methods. COCO-Stuff is a popular coco dataset with pixel-level labeled. It contains 10K images, 91 things classes and 91 stuff classes. We use 9K of these images for training and 1K for testing like other methods.

Super parameters

We use the stochastic gradient descent (SGD) algorithm with 0.9 momentum to train our model. We set a batchsize of 16 during training. When training with the cityscapes and Cam Vid datasets, we set the weight decay to 0.0005. When training with the COCO-Stuff dataset, we set the weight decay to 0.0001. We set the initial learning rate lr to 5e-2. The learning rate is decayed using the "pol" strategy during training, and the learning rate for each iteration is calculated as: $(1 - \frac{iter}{iters_{max}})^{power} * lr$. We train 20K iterations on each dataset separately. The inference speed is measured on the NVIDIA GeForce 1080Ti card.

Ablation study

The following Table 1 is the experimental result on the validation data of the cityscapes dataset after we added the ASPP layer to the network. The second column shows whether we use the ASPP module before extracting 1/32 feature in the semantic branch. The third column shows whether we use the ASPP module before extracting 1/16 feature in the semantic branch. The fourth column shows whether we use the ASPP module before extracting 1/8 feature in the semantic branch. The fifth column shows whether we use the ASPP module before extracting 1/4 feature in the semantic branch. From Table 1 below, we can see that the network achieved the best performance after we added ASPP layers in the last three stages of the network.

Table 1 The impact of adding different numbers of ASPP layers on the validation data of the cityscapes dataset

BisenetV2	S4	S3	S2	S1	mIoU	Gain
✓					73.36	–
✓	✓				74.76	1.4
✓	✓	✓			75.97	2.61
✓	✓	✓	✓		**76.15**	**2.79**
✓	✓	✓	✓	✓	76.07	2.71

Boldface indicates the highest accuracy value among all the models

Table 2 The impact of CFF module on the validation data of the cityscapes dataset

	BisenetV2+CFF	BisenetV2
IoU (%)	77.12	73.36
Gain	**3.76**	–

Boldface indicates the highest accuracy value among all the models

Table 3 The effect of UCBA layer on cityscapes val dataset

	BisenetV2+UCBA	BisenetV2
IoU (%)	73.76	73.36
Gain	**0.2**	–

Boldface indicates the highest accuracy value among all the models

Table 2 shows the experimental results of the network on the validation data of the cityscapes dataset after replacing the feature fusion part of Bisenetv2 into CFF module. As can be seen from Table 2 below, the model with CFF module achieved a total mIoU improvement of 3.76%.

The following Table 3 is the experimental results of the network on the validation data of the cityscapes dataset after using our UCBA module in the network. In the experiment, we set the "align_corners" parameter of bilinear to true. From

Table 4 The effect of the model when the position and number of side branches are different. Bisenetv2 backbone is the semantic branch, detail branch of Bisenetv2. Bisenetv2 backbone with CFF obtained 73.26% mIoU.We use it as a baseline in the experiment

BisenetV2 backbone+CFF	Loss7	Loss6	Loss5	Loss4	Loss3	Loss2	Loss1	mIoU	Gain
✓	✓						✓	73.84	0.58
✓	✓	✓					✓	74.48	1.22
✓	✓	✓	✓				✓	75.51	2.25
✓	✓	✓	✓	✓			✓	76.81	3.55
✓	✓	✓	✓	✓	✓		✓	77.70	4.44
✓	✓	✓	✓	✓	✓	✓	✓	**78.73**	**4.47**
✓								73.26	

Boldface indicates the highest accuracy value among all the models

Table 5 The results of different networks on the cityscapes validation dataset

Method	mIoU(%)	FPS	GPU
DF2-Seg1 [17]	75.9	67.2	GTX 1080Ti
DF2-Seg2 [17]	76.9	56.3	GTX 1080Ti
SwiftNetRN-18 [22]	75.5	39.9	GTX 1080Ti
CABiNet [15]	76.6	76.5	RTX 2080Ti
BiSeNet2 [37]	73.4	156	GTX 1080Ti
GAS [19]	72.4	108.4	TitanXp
STDC2-Seg75 [9]	77.0	97.0	GTX 1080Ti
PP-LiteSeg-B2 [24]	78.2	102.6	GTX 1080Ti
HyperSeg-S [21]	78.2	16.16	GTX 1080Ti
DDRNet-23 [12]	79.5	37.1	GTX 2080Ti
Ours	**81.3**	110	GTX 1080Ti

Boldface indicates the highest accuracy value among all the models

Table 3 below, it can be seen that the model with UCBA module achieved a total mIoU improvement of 0.2%.

The following Table 4 is the experimental results of the network on the validation data of the cityscapes dataset after using the auxiliary feature extraction layer in the network. From Table 4 below, it can be seen that the model with the auxiliary feature extraction layer achieved a total mIoU improvement of 4.47%. The network obtained the best performance after adding the auxiliary feature extraction layer at each stage of the semantic branch.

Comparison

The following Table 5 is a comparison of the experimental results of our network and the current mainstream semantic segmentation network on the validation data of the cityscapes dataset. We use the network based on bisenetv2 backbone with the multi-resolution feature fusion module, the auxiliary feature extraction module, the upsampling module and multi-ASPP proposed in the paper achieved that result. During inference, the output of the network is a single-channel image with a resolution of 1024 × 512. We then resize it to a grayscale image with a resolution of 2048 × 1024, and then use the resolution to calculate the mIoU value.

Table 6 The results of different networks on the CamVid test dataset

Method	mIoU(%)	FPS	GPU
MSFNet [30]	75.4	91.0	GTX 2080Ti
PP-LiteSeg-T [24]	75.0	154.8	GTX 1080Ti
TD2-PSP50 [13]	76.0	11.0	TITAN X
BiSeNetV2 [37]	72.4	124.5	GTX 1080Ti
HyperSeg-L [21]	79.1	16.6	GTX 1080Ti
DDRNet-23 [12]	76.3	94	GTX 2080Ti
DFANet B [16]	59.3	160	GTX 1080Ti
DFANet A [16]	64.7	120	GTX 1080Ti
Ours	80.7	104	GTX 1080Ti

Table 7 The results of different networks on the COCO-Stuff test dataset

Method	mIoU(%)	FPS	GPU
DDRNet-23 [12]	32.1	129.2	GTX 2080Ti
PSPNet50 [43]	32.6	6.60	GTX 1080Ti
ICNet [42]	29.1	35.7	GTX 1080Ti
BiSeNetV2 [37]	25.2	87.9	GTX 1080Ti
Ours	32.2	78	GTX 1080Ti

The following Table 6 is a comparison of the experimental results of our network and the current mainstream semantic segmentation network on the test data of the CamVid dataset. We use the network based on bisenetv2 backbone with the multi-resolution feature fusion module, the auxiliary feature extraction module, the upsampling module and multi-ASPP proposed in the paper achieved that result. During inference, the output of the network is a single-channel image with a resolution of 960 × 720.

The following Table 7 is a comparison of the experimental results of our network and the current mainstream semantic segmentation network on the test data of the COCO Stuff dataset. We use the network-based on bisenetv2 backbone with the multi-resolution feature fusion module, the auxil-

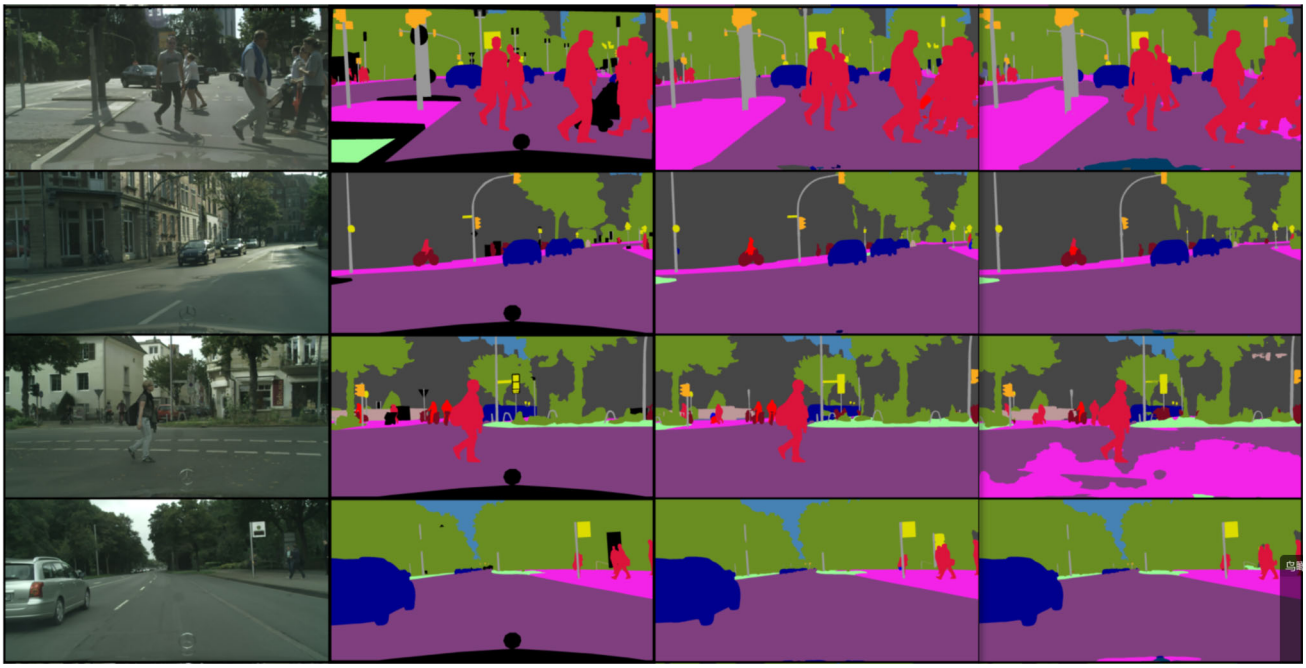

Fig. 8 The left column is the original image, the second column is Ground Truth, the third column is the predictions of the network proposed in the paper, the fourth column is the predictions of Bisenetv2

iary feature extraction module, the upsampling module and multi-ASPP proposed in the paper achieved that result. During inference, the output of the network is a single-channel image with a resolution of 640 × 640.

From the above results, we can see that the network proposed in this paper achieved 81.3% mIoU at 110FPS on Cityscapes validation dataset, 80.7% mIoU at 104FPS on the CamVid test dataset, 32.2% mIoU at 78FPS on the COCO-Stuff test dataset. Better results were obtained at a relatively fast speed. Figure 8 shows the visualization comparison on Cityscapes dataset.

Conclusion

In this paper, we conducted research and experiments on the side branch auxiliary feature extraction module, the feature fusion module, the upsampling module, and the atrous spacial pooling pyramid module. We made new discoveries that the role of side-branch is more than regularization, it may either slow down the convergence or accelerate the convergence by influencing the gradient of different layers of the network, which is dependent on the parameters of the network and the input data. We also proposed an upsampling module and a feature fusion module. We used the ASPP module to further optimize the network. We analyzed the effect of each improvement on the network in detail through experiments, and confirmed the effectiveness of the above methods. The network proposed in this paper achieved 81.3% mIoU at 110FPS on Cityscapes validation dataset, 80.7% mIoU at 104FPS on the CamVid test dataset, 32.2% mIoU at 78FPS on the COCO-Stuff test dataset.

Availability of data and materials The datasets generated during and/or analysed during the current study are available in the https://www.cityscapes-dataset.com/, http://images.cocodataset.org, http://mi.eng.cam.ac.uk/research/projects/VideoRec/CamVid/repository

Declarations

Conflict of interest The authors declared that they have no conflicts of interest to this work.

References

1. Badrinarayanan V, Kendall A, Cipolla R (2017) Segnet: a deep convolutional encoder–decoder architecture for image segmentation. IEEE Trans Pattern Anal Mach Intell 39:1

2. Brostow GJ, Fauqueur J, Cipolla R (2009) Semantic object classes in video: a high-definition ground truth database. Pattern Recogn Lett 30(2):88–97
3. Caesar H, Uijlings JRR, Ferrari V (2016) Coco-stuff: Thing and stuff classes in context. CoRR, arXiv:1612.03716
4. Chen LC, Papandreou G, Kokkinos I, Murphy K, Yuille AL (2014) Semantic image segmentation with deep convolutional nets and fully connected crfs. Comput Sci 4:357–361
5. Chen LC, Papandreou G, Kokkinos I, Murphy K, Yuille AL (2018) Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Trans Pattern Anal Mach Intell 40(4):834–848
6. Chen LC, Papandreou G, Schroff F, Adam H (2017) Rethinking atrous convolution for semantic image segmentation
7. Chen LC, Zhu Y, Papandreou G, Schroff F, Adam H (2018) Encoder-decoder with atrous separable convolution for semantic image segmentation. Springer, Cham
8. Cordts M, Omran M, Ramos S, Rehfeld T, Schiele B (2016)The cityscapes dataset for semantic urban scene understanding. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)
9. Fan M, Lai S, Huang J, Wei X, Chai Z, Luo J, Wei X (2021) Rethinking bisenet for real-time semantic segmentation. CoRR, arXiv:2104.13188
10. Fu J, Liu J, Tian H, Fang Z, Lu H (2018) Dual attention network for scene segmentation
11. Gamal M, Siam M, Abdel-Razek M (2018) Shuffleseg: real-time semantic segmentation network
12. Hong Y, Pan H, Sun W (2021) Senior Member, IEEE, and Yisong Jia. Deep dual-resolution networks for real-time and accurate semantic segmentation of road scenes
13. Hu P, Heilbron FC, Wang O, Lin ZL, Sclaroff S, Perazzi F (2020) Temporally distributed networks for fast video semantic segmentation. CoRR, arXiv:2004.01800
14. Huang Z, Wang X, Wei Y, Huang L, Huang TS (2020) Ccnet: Criss-cross attention for semantic segmentation. IEEE Trans Pattern Anal Mach Intell 99:1
15. Kumaar S, Lyu Y, Nex F, Michael YY (2020) Efficient context aggregation network for low-latency semantic segmentation, Cabinet
16. Li H, Xiong P, Fan H, Sun J (2020) Dfanet: Deep feature aggregation for real-time semantic segmentation. In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)
17. Li X, Zhou Y, Pan Z, Feng J (2019) Partial order pruning: for best speed/accuracy trade-off in neural architecture search. IEEE
18. Lin G, Milan A, Shen C, Reid I (2017) Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)
19. Lin P, Sun P, Cheng G, Xie S, Shi J (2020) Graph-guided architecture search for real-time semantic segmentation. In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)
20. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. IEEE Trans Pattern Anal Mach Intell 39(4):640–651
21. Nirkin Y, Wolf L, Hassner T (2021) Hyperseg: Patch-wise hypernetwork for real-time semantic segmentation. In Computer Vision and Pattern Recognition
22. Orsic M, Kreso I, Bevandic P, Segvic S (2019) In defense of pre-trained imagenet architectures for real-time semantic segmentation of road-driving images. CoRR, arXiv:1903.08469
23. Paszke A, Chaurasia Ak, Kim S, Culurciello E (2016) Enet: A deep neural network architecture for real-time semantic segmentation. CoRR, arXiv:1606.02147
24. Peng J, Liu Y, Tang S, Hao Y, Chu L, Chen G, Wu Z, Chen Z, Yu Z, Du Y (2022) Pp-liteseg: A superior real-time semantic segmentation model
25. Poudel RPK, Bonde U, Liwicki S, Zach C (2018) Contextnet: Exploring context and detail for semantic segmentation in real-time. CoRR, arXiv:1805.04554
26. Poudel RPK, Liwicki S, Cipolla R (2019) Fast-scnn: Fast semantic segmentation network. CoRR, arXiv:1902.04502
27. Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. Springer International Publishing, Berlin
28. Sandler M, Howard AG, Zhu M, Zhmoginov A, Chen LC (2018) Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation. CoRR, arXiv:1801.04381
29. Shi W, Caballero J, Theis L, Huszar F, Wang Z (2016) Is the deconvolution layer the same as a convolutional layer?
30. Si H, Zhang Z, Lv F, Yu G, Lu F (2019) Real-time semantic segmentation via multiply spatial fusion network
31. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2818–2826
32. Tao A, Sapra K, Catanzaro B (2020) Hierarchical multi-scale attention for semantic segmentation. CoRR, arXiv:2005.10821
33. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. CoRR, arXiv:1706.03762
34. Wang J, Sun K, Cheng T, Jiang B, Deng C, Zhao Y, Liu D, Mu Y, Tan M, Wang X, Liu W, Xiao B (2019) Deep high-resolution representation learning for visual recognition. CoRR, arXiv:1908.07919
35. Wang X, Girshick RB, Gupta A, He K (2017) Non-local neural networks. CoRR, arXiv:1711.07971
36. Xu K, Guan K, Peng J, Luo Y, Wang S (2019) Deepmask: an algorithm for cloud and cloud shadow detection in optical satellite remote sensing images using deep residual network
37. Yu C, Gao C, Wang J, Yu G, Shen C, Sang N(2020) Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation
38. Yu C, Wang J, Peng C, Gao C, Yu G, Sang N (2018) Bisenet: Bilateral segmentation network for real-time semantic segmentation. European Conference on Computer Vision,
39. Yuan Y, Chen X, Wang J (2020) Segmentation Transformer: Object-Contextual Representations for Semantic Segmentation. In European Conference on Computer Vision
40. Zhang X, Zhou X, Lin M, Jian S (2017) An extremely efficient convolutional neural network for mobile devices, Shufflenet
41. Zhao H, Shi J, Qi X, Wang X, Jia J (2016) Pyramid scene parsing network. In IEEE Computer Society,
42. Zhao H, Qi X, Shen X, Shi J, Jia J (2017) Icnet for real-time semantic segmentation on high-resolution images. CoRR, arXiv:1704.08545
43. Zhao H, Shi J, Qi X, Wang X, Jia J (2016) Pyramid scene parsing network. CoRR, arXiv:1612.01105