Applied Intelligence (2022) 52:580–594
https://doi.org/10.1007/s10489-021-02446-8

Joint pyramid attention network for real-time semantic segmentation of urban scenes

Xuegang Hu[1] · Liyuan Jing[2] · Uroosa Sehar[3]

Accepted: 20 April 2021 / Published online: 6 May 2021

Abstract

Semantic segmentation is an advanced research topic in computer vision and can be regarded as a fundamental technique for image understanding and analysis. However, most of the current semantic segmentation networks only focus on segmentation accuracy while ignoring the requirements for high processing speed and low computational complexity in mobile terminal fields such as autonomous driving systems, drone applications, and fingerprint recognition systems. Aiming at the problems that the current semantic segmentation task are facing, it is difficult to meet the actual industrial needs due to its high computational cost. We propose a joint pyramid attention network (JPANet) for real-time semantic segmentation. First, we propose a joint feature pyramid (JFP) module, which can combine multiple network stages with learning multi-scale feature representations with strong semantic information, hence improving pixel classification performance. Second, we built a spatial detail extraction (SDE) module to capture the shallow network multi-level local features and make up for the geometric information lost in the down-sampling stage. Finally, we design a bilateral feature fusion (BFF) module, which properly integrates spatial information and semantic information through a hybrid attention mechanism in spatial dimensions and channel dimensions, making full use of the correspondence between high-level features and low-level features. We conducted a series of experiments on two challenging urban road scene datasets (Cityscapes and CamVid) and achieved excellent results. Among them, the experimental results on the Cityscapes dataset show that for 512 × 1024 high-resolution images, our method achieves 71.62% Mean Intersection over Union (mIoU) with 109.9 frames per second (FPS) on a single 1080Ti GPU.

Keywords Attention mechanism · Encoder-decoder network · Feature pyramid module · Lightweight network · Real-time semantic segmentation

1 Introduction

Deep learning is cost-effective in translation invariance and automatic extraction of the in-depth features of target input. However, traditional image processing methods require much cost for this. Therefore, deep learning has been widely used in many fields of digital image processing. Semantic segmentation using deep learning is one of the most popular research topics. It aims to group pixels according to different semantics expressed in the image, and has a wide range of applications in drones, autonomous driving systems, wearable devices, and medical image analysis [1–3].

Hitherto, most advanced semantic segmentation networks [4–6] use backbone networks with more layers as the model encoder, which helps to improve the segmentation accuracy of the network. Nevertheless, they ignore the unique requirements for low storage overhead and high processing speed of edge devices in industrial production. First, a high-precision segmentation network usually reaches hundreds of layers, which contains many weight parameters, thus posing a severe challenge to the storage capacity of edge devices. Second, there are two ways to achieve millisecond-level processing speeds in practical applications: improve the processor performance or reduce

This work was supported by the National Natural Science Foundation of China (under Grant 52076044) and the key project of the Natural Science Foundation of Chongqing, China (under Grant cstc2017jcyjBX0037).

✉ Liyuan Jing
S190131229@stu.cqupt.edu.cn

Extended author information available on the last page of the article.

the computational complexity of the model. Due to the influence of manufacturing process, it is not easy to significantly improve the processing units like Graphic card or so on. Therefore, reducing the computational complexity of the neural network is the most effective method at present.

To reduce the large number of parameters redundancy in the deep neural network and the model computational complexity, the main method is to compress the pre-trained model and transform it into an efficient small model. Recently, the commonly used compression model methods include network pruning [7], knowledge distillation [8], and low-rank approximation [9]. The operation process of network pruning is first to measure each neuron importance after training, remove some unimportant neurons, then fine-tune the network, and finally return to the first step for the next round of pruning. The low-rank approximation uses several small-scale matrices to reconstruct a dense matrix, hence it effectively reduce computation and storage costs. For example, LRNNet [10] uses singular value decomposition to simplify non-local networks and reduce the weight matrix parameter. The basic idea is to perform singular value decomposition on the weight matrix. Since, the singular vector corresponding to a larger singular value contains more matrix information, the first k largest items in the singular value matrix and the corresponding singular value vector are retained. So we need to reconstruct a weight matrix similar to the original matrix. The extraction of knowledge distillation is transfer learning. Its purpose is to transfer the knowledge learned by a complex model to a simplified small model through specific technical means, so that the small model can achieve similar performance as the large model.

Due to the of the problem that semantic segmentation networks are difficult to deploy to terminal devices because of excessive parameters and computational costs in practical application scenarios. Some researchers use lightweight image classification models as the backbone network of real-time semantic segmentation models. Although these real-time semantic segmentation algorithms [11–13] based on lightweight backbone networks can obtain deep-level semantic information, they ignore the impact of the network shallow geometric details on the segmentation results to pursue faster inference speed. Therefore, they have not designed a suitable decoder, resulting in unsatisfactory segmentation accuracy. So, balancing inference speed, segmentation accuracy, and network scale are still problems that researchers need to address.

Based on the above analysis, we design three plug-and-play modules: Joint Feature Pyramid (JFP) Module, Spatial Detail Extraction (SDE) Module, and Bilateral Feature Fusion (BFF) Module. The JFP module is used to extract rich semantic information in the deep layer of the network to enhance feature recognition capabilities. The SDE module is used to extract rich spatial contour information in the shallow layer of the network. Finally, the feature information captured by the JFP module in the spatial and channel dimensions passes through the BFF module fusion with the spatial contour information captured by the SDE module. Based on the JFP module, SDE module, and BFF module, we efficiently construct a real-time semantic segmentation model called JPANet. It can select different backbone networks as encoder according to different scenarios to achieve the trade-off between computing costs, inference speed, and segmentation accuracy.

In conclusion, our main contributions are as follows:

- We propose a new JFP module to extract strong semantic feature representations in the network, which helps JPANet accurately obtain high-level semantic information of the target object and improve segmentation accuracy.
- The SDE module for extracting multi-level local features of the shallow network is proposed. This module can make up for the geometric information lost in the down-sampling process, hence, improving the ability to segment small target objects.
- In view of the information complementary characteristics of spatial location and high-level semantics in semantic segmentation tasks, we propose a BFF module that captures the self-dependence of each category of channels and spatial locations in the middle layer of the network.
- Based on the above three modules, we designed a real-time semantic segmentation network called JPANet. JPANet makes full use of the information of high-level semantics and low-level details and satisfies the perception of high-level semantic information of low-level details and the understanding of low-level details characteristics of high-level semantic similarity. It solves the problems of the current semantic segmentation model, that are mainly cannot achieve high processing speed and low storage overhead due to its huge parameter amount and computational cost.
- Experiments on the Cityscapes dataset show that even if a 512 × 1024 high-resolution image is input, JPANet can still achieve 71.62% mIoU at 109.9 FPS. On the CamVid dataset with an input resolution of 360 × 480, JPANet can achieve 67.45% mIoU with 294 FPS.

2 Related work

This section mainly introduces the three parts most relevant to our work: lightweight backbone network, attention mechanism, and multi-scale contextual information.

2.1 Lightweight backbone network

To improve the image and video processing capabilities of embedded and mobile terminal devices, it is usually necessary to meet the requirements of low power consumption, low storage, and high real-time. Therefore, the main idea of designing a lightweight neural network is to design a more efficient convolution operation mode to reduce the redundant information in the network.

The basic component of MobileNet [14] is deep separable convolution, which can be divided into depthwise convolution and pointwise convolution. Depthwise convolution uses different convolution kernels for the input feature channels, and pointwise convolution uses 1 × 1 standard convolution kernels to perform feature maps upgrading or reducing dimensionality, restore to the target size. The depthwise separable convolution formed by the combination of deepwise convolution and pointwise convolution has much lower parameters and calculations than standard convolution, and it will not cause excessive precision loss. ShuffleNet V2 [15] uses channel split instead of group convolution in ShuffleNet V1 [16]. Each block end uses a channel shuffle operation to ensure information flow between the two branches. It is because of the fact that ShuffleNet V2 follows the four principles of efficient network design. Therefore ShuffleNet V2 is more advanced than most lightweight networks in terms of speed and accuracy. The traditional concept in lightweight neural network design process believes that there is redundant feature information in neural networks, and it is necessary to avoid the generation of these highly similar feature information. GhostNet [17] believes that the strong feature extraction ability and linear invariance of convolutional neural network are positively related to these rich feature information. So GhostNet uses a series of cheap linear transformations to generate an internal map that fully reveals the feature information.

2.2 Attention mechanism

The attention mechanism can give different weights to image pixels to focus on essential areas, results in improving the network processing capacity. Hence it has been widely used in many computer vision tasks. The method to realize the attention mechanism is mainly divided into two steps: First, calculate the given input feature information attention to weight probability. Second, extract relevant feature information based on the attention weight probability. According to the way the attention weight is applied, the attention model can be divided into spatial attention model, channel attention model, mixed attention model, etc.

In the field of image semantic segmentation, CCNet [18] replaces the traditional non-local operation by Recurrent Criss-Cross Attention block (RCCA). After passing through the RCCA module, each pixel can capture its horizontal and vertical context information, maintaining long-distance spatial dependence. While significantly reducing the model space complexity, good results have been obtained on multiple datasets. To better integrate the information of spatial detail branches and high-level context branches, BiSeNet [12] proposed a Feature Fusion Module (FFM). The FFM converts the feature information into a weight vector and then re-weights the features. Through this operation, the global context information can be integrated without too much computational cost. DANet [19] uses the position attention module to capture the spatial dependence between any two positions in the feature map, which take advantage of encoding context information into local features. The channel attention module is used to establish the semantic dependency between each channel mapping explicitly. SANet [3] introduced an attention convolution channel to strengthen important features and weaken unimportant features, and it was making the feature more directive, thus effectively considering spatial-channel interdependence. TSNet [20] introduced a self-attention mechanism in the cross-modal distillation stream, and then refined the intermediate feature maps of the depth stream and RGB stream through the cross-modal distillation stream, to further optimized the segmentation results.

2.3 Multi-scale feature fusion

The latest progress made by real-time semantic segmentation networks mainly comes from merging multi-scale context information to improve the model feature expression ability. The so-called multi-scale is to sample images with different granularities. The deep layer of the semantic segmentation network based on deep learning can represent powerful semantic information, but the resolution of the feature map is low, and the spatial detail information is scarce, which is suitable for processing large target objects. On the other hand, the shallow receptive field of the network is relatively small, the ability to express spatial detail information is strong, and the corresponding semantic features are less, which is suitable for processing small target objects. Therefore, fusing the deep and shallow features of the network is beneficial to enhance the model segmentation ability.

There are two common multi-scale feature fusion methods: the first is to use parallel multi-branch networks, such as the DeepLab series [21–23] of Atrous Spatial pyramid pooling (ASPP) module and PSPNet [24] Pyramid Pooling Module (PPM). The second is the skip connection structure. This fusing multi-scale feature is very common in image segmentation tasks, such as FCN [25], UNet [26].

3 Joint pyramid attention network

Figure 1 shows the overall architecture of JPANet. Then we will introduce our proposed SDE module, JFP module, and BFF module.

3.1 Spatial detail extraction module

The current real-time semantic segmentation algorithm mainly uses convolution factorization and continuous downsampling to reduce the calculation cost and improve the inference speed. However, the image spatial position information will gradually be lost in the process of multiple downsampling, causing irreversible adverse effects on small objects in the image. The dilated convolution [27] can increase the receptive field of the model without reducing the image resolution, and captures the surrounding and local features of the pixel. Although this method effectively extracts high-level semantic information, it does not consider how to extract spatial detail information.

Due to the shallow high-resolution images of the network contain rich location information, while the deep low-resolution images lack spatial information. To solve this problem, we propose an SDE module to extract the image spatial features in the first three stages of the backbone network. As shown in Fig. 1, the module is composed of three down-sampling units, where each down-sampling unit is composed of a standard convolution with a step of 2 and maximum pooling in parallel. The input resolution of each downsampling unit is $H \times W \times C_{in}$, and the output resolution is $\frac{H}{2} \times \frac{W}{2} \times C_{out}$, where H and W represent the height and width of the input image, C_{in} is the number of input channels, and C_{out} is the number of output channels. The number of channels through the maximum pooling is C_{in}, and the number of channels through the step convolution is $C_{out} - C_{in}$. For the 1/2 resolution image, we downsample twice. For the 1/4 resolution image, we downsample once. Then perform the residual connection (Note: the $1/x$ resolution images mentioned in this article are relative to the original input image). This construction method not only integrate the location information between different layers in the backbone network, but also strengthens the flow of spatial information of the image between the network layers. Moreover, it improves the perception of the shallow location information of the high-level semantic information.

3.2 Joint feature pyramid module

Recently most real-time semantic segmentation model based on lightweight backbone networks usually only use simple decoders to obtain higher inference speed, which results in the segmentation accuracy of the models often not satisfactory. Therefore, we have carefully design the JFP module to capture multi-scale feature information and produce better segmentation performance in the decoder part.

Since 1/8 resolution pixels are four times more than 1/16 resolution, 16 times more than 1/32 resolution. When performing the same convolution operation, the computational cost of 1/8 resolution is four times that of 1/16

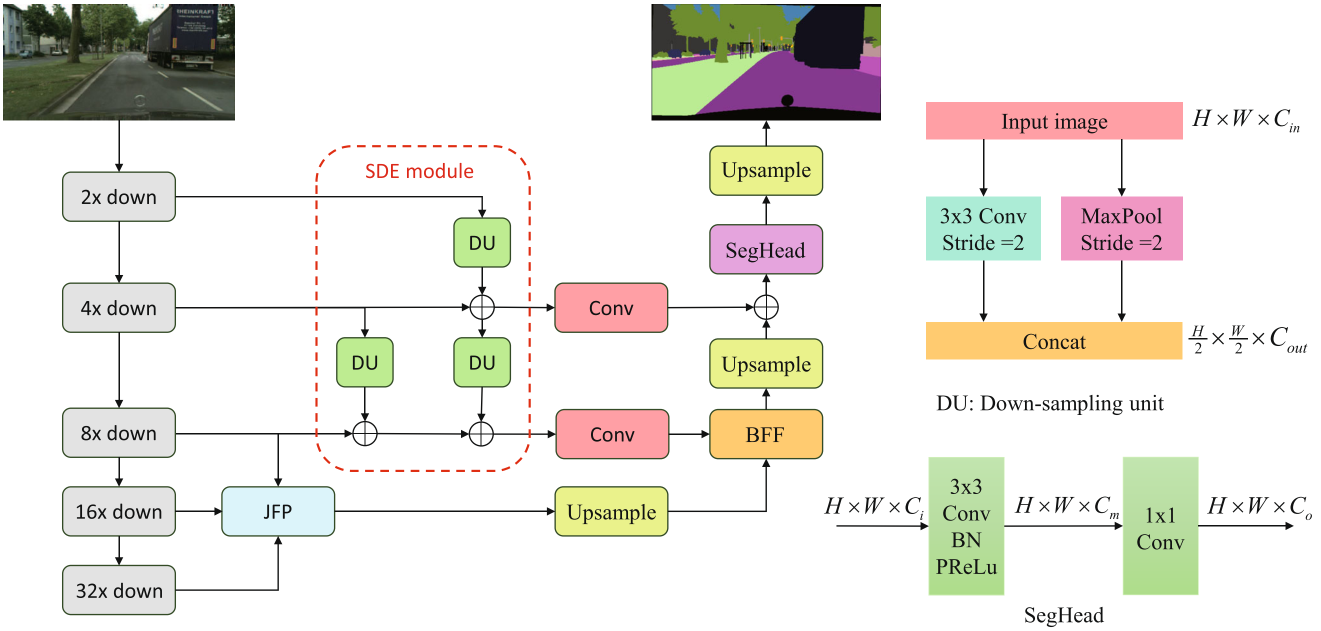

Fig. 1 The detailed structure of JPANet. JFP is a joint feature pyramid module, SDE is a spatial detail extraction module, BFF is a bilateral feature fusion module, and SegHead represents a segmentation head. In SegHead, C_i is 128, C_m is 128, and C_o is 20. ⊕ denotes the element-level addition of the feature map

resolution and 16 times that of 1/32 resolution. Although the multi-scale context information extraction on the 1/8 resolution image can greatly improve accuracy, it will also greatly increase the model computational cost. Even if extracting multi-scale context information from an image with 1/32 resolution can greatly improve the computational efficiency, it will also reduce the accuracy. To achieve the best trade-off between segmentation accuracy and segmentation efficiency, our proposed JFP module is performed on the features of 1/16 resolution.

In Fig. 2a, we use 3 × 3 standard convolution to process feature maps with 1/8 resolution and 1/16 resolution. It is just because the 1/32 resolution image has more channels, standard convolution will increase many parameters, so we use the depthwise separable convolution to process the 1/32 resolution image. Then the 1/8 resolution image is downsampled to 1/16 resolution using maximum pooling, and the high-level semantic feature map of 1/32 resolution is bilinearly upsampled to 1/16 resolution. Finally the channels are concatenated to obtain f_a.

Figure 2b is the feature pyramid structure using the split-transform-concatenate operation. First, channel shuffle is carried out for f_a, and it is divided into four parts. The feature map after division is $f_a^i, i \in \{1, 2, 3, 4\}$. The number of channels for f_a^i is $C/4$, where C is the number of channels for f_a. Then f_a^i is parallelized through 3 × 3 dilated convolution, and its specific operation is defined as follows:

$$F_a^i = \begin{cases} D(f_a^i), & i = 1, \\ D(F_a^{i-1} + f_a^i), & i = 2, 3, 4. \end{cases} \tag{1}$$

Where D represents dilated convolution, and F_a^i represents the output of the i-th dilated convolution. Finally, concatenate $F_a^i, i \in \{1, 2, 3, 4\}$ and f_a in the channel dimension to obtain f_b.

As shown in Fig. 2c, f_b contains many channels, and the direct use of standard convolution will bring more parameters, which will bring a heavy computational burden to edge devices with limited computing resources.

The formula for calculating the parameters of the unbiased depthwise separable convolution is:

$$K_h \times K_w \times C_i + C_i \times C_o \tag{2}$$

Among them, K_h and K_w are the height and width of the convolution kernel, C_i is the number of input channels, and C_o is the number of output channels. The parameter calculation formula for standard unbiased convolution is:

$$K_h \times K_w \times C_i \times C_o \tag{3}$$

It is not difficult to find that when C_o is much larger than $K_h \times K_w$, the parameter amount of the depthwise separable convolution is only $1/(K_h \times K_w)$ times that of the standard convolution. Therefore, we use 3 × 3 depth separable convolution for f_b to obtain a new feature representation f_c, which can reduce the parameter amount of this link by about nine times.

3.3 Bilateral feature fusion module

In the backbone network, shallow features receptive field is small, contains rich geometric details, and is suitable for processing small targets. Deep features have a large receptive field and strong semantic information representation

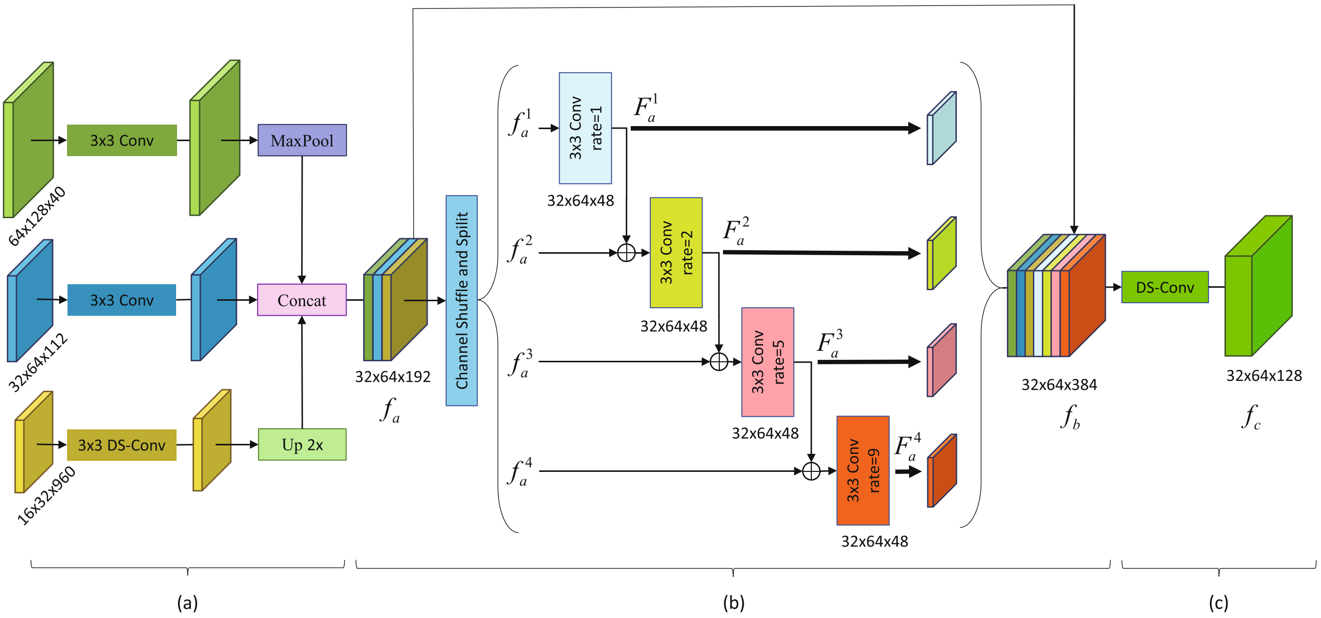

Fig. 2 Joint feature pyramid module structure

ability, suitable for processing large objects. The purpose of feature fusion is to merge the different features extracted from the image into a more discriminative feature. It can fuse the most differentiated information among different features and eliminate redundant information generated by the correlation between different features. Therefore, fusing features of different scales in semantic segmentation has become an important means to improve the accuracy of segmentation.

The most classic feature fusion method currently uses channel concatenation and element-level addition, but these two methods ignore the spatial dependence and channel dependence between pixels, leading to sub-optimal segmentation results. Therefore, we propose the BFF module (as shown in Fig. 3), which uses a channel attention mechanism and spatial attention mechanism to enhance the global correlation between feature information in parallel.

First, we concatenate the geometric detail features generated by the SDE module and the deep semantic features generated by the JFP module. Then use standard convolution to balance the scale of the feature information to get the feature map. Next, the feature map is equally divided into m_1 and m_2.

Second, we use a similar operation to CBAM [28]. The above branch uses adaptive average pooling and adaptive maximum pooling to obtain feature vectors $f_{Avg}^{C\times1\times1}$ and $f_{Max}^{C\times1\times1}$, then calculates the weight vector $V_{channel}^{C\times1\times1}$, and finally uses $V_{channel}^{C\times1\times1}$ to re-weight the features m_1 to obtain f_1. The specific operation is defined as follows:

$$V_{channel}^{C\times1\times1} = \sigma(F(Avg(m_1)) + F(Max(m_1))) \quad (4)$$

$$f_1 = V_{channel}^{C\times1\times1} \times m_1 \quad (5)$$

Here $\sigma(\cdot)$ represents the sigmoid activation function, Avg is the adaptive global average pooling, Max is the adaptive global maximum pooling, and F is the combination function, which includes two 1×1 convolutions and Parametric Rectified Linear Unit (PReLU).

Third, for the following branches, we use adaptive global average pooling and adaptive global maximum pooling in the channel dimension to obtain m_2 spatial information $S_{Avg}^{1\times H\times W}$ and $S_{Max}^{1\times H\times W}$. Then usage concatenation, standard convolution, and the activation function to get a two-dimensional spatial attention map $M_{spatial}^{1\times H\times W}$. At last we use $M_{spatial}^{1\times H\times W}$ to re-weight the features m_2 to obtain f_2. The specific operation is defined as follows:

$$M_{spatial}^{1\times H\times W} = \sigma(conv(concat(Avg(m_2), Max(m_2)))) \quad (6)$$

$$f_2 = M_{spatial}^{1\times H\times W} \times m_2 \quad (7)$$

Here $\sigma(\cdot)$ stands for sigmoid activation function, $conv$ stands for standard convolution, and $concat$ stands for channel concatenation.

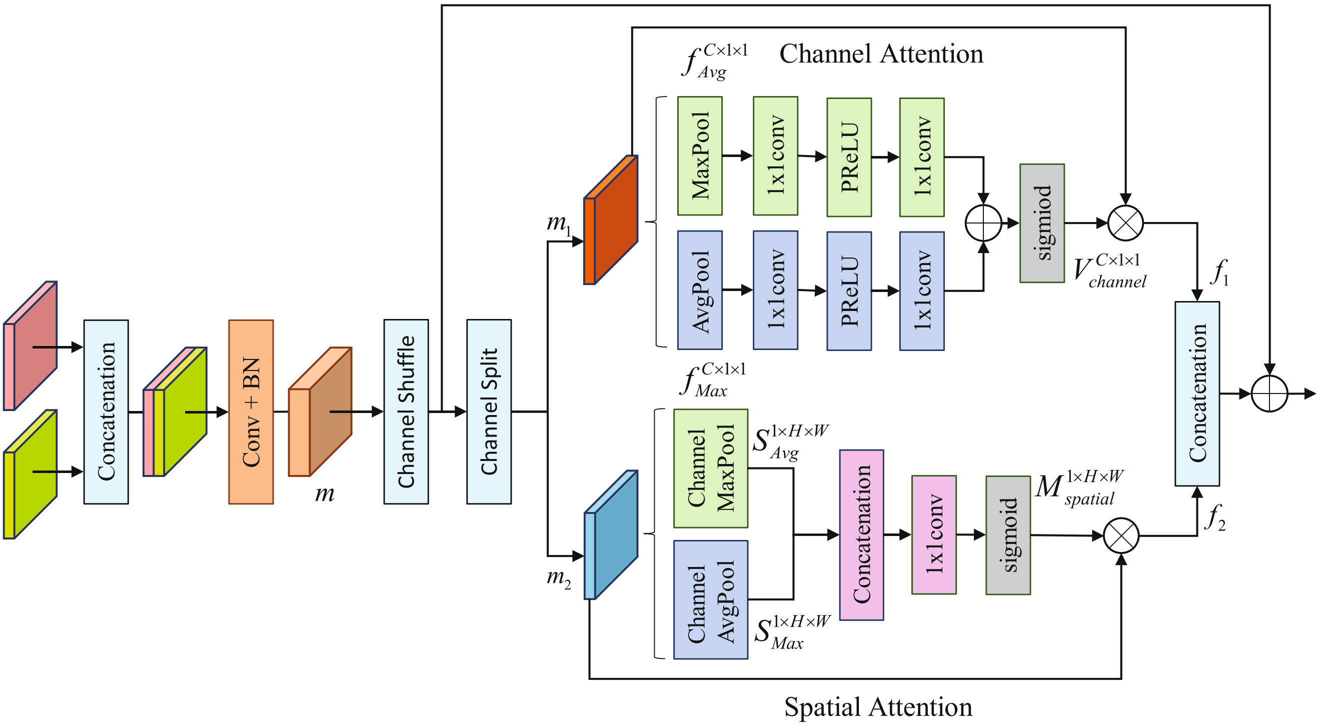

Fig. 3 Feature fusion module

Finally, the feature maps generated by the channel attention path and the spatial attention path are concatenated, and then the residual connection is made with the feature map m.

As channel attention pays attention to what it is, spatial attention pays attention to where it is. The BFF module uses a hybrid attention mechanism based on the above two. Therefore, the BFF module achieves a more effective fusion of information with complementary geometric spatial details and high-level semantic information.

4 Experiments

This section uses JPANet to perform performance evaluation experiments on two representative urban road datasets, Cityscapes and CamVid. First, introduce two datasets and implementation details and then analyze each component effectiveness in JPANet. Finally, compare JPANet with the current state-of-the-art real-time semantic segmentation model in terms of Mean Intersection over Union (mIoU), giga-floating point operations (GFLOPs), and parameters (Params).

4.1 Datasets

4.1.1 Cityscapes

Cityscapes is a large dataset for semantic understanding of urban street scenes, with a resolution of up to 1024 × 2048. It contains 5000 finely labeled pictures, of which 2975 are used for training, 500 are used for validation, and 1525 are used for testing. It also contains about 20,000 coarse labeled pictures, which can be used to pre-train the model. Cityscapes have 30 types of labeled objects, while only 19 types are used for semantic segmentation. Since it contains many semantically similar categories (for example, Car and Bus, Motorcycle and Bicycle), it poses a huge challenge to real-time semantic segmentation.

4.1.2 CamVid

CamVid is another well-known dataset for understanding complex road scenes in cities. It contains 701 high-resolution pictures extracted from video sequences. In an image with a resolution of 720 × 960, there are 11 categories for semantic segmentation. According to the previous division method [29, 30], 367 pictures are used for training, 101 pictures are used for validation, and 233 pictures are used for testing.

4.2 The experimental details

4.2.1 The experiment platform

We have performed our experiment using system with AMD R5 3600 @ 3.6GHz, NVIDIA GeForce GTX 1080Ti GPU, and 16GB RAM. The software environment specification we used during our experiment is PyTorch1.5.0, CUDA10.1, cudnn7.6.5.

4.2.2 The experimental details

In order to make full use of GPU memory, we use Apex mixed precision developed by NVIDIA to accelerate model training. The Adam optimizer is used to train the model, and the weight decay is set to 2×10^{-4}. Following the methods in [1] and [11], we also adopt the "poly" learning rate adjustment strategy:

$$lr = init\_lr \times \left(1 - \frac{epoch}{max\_epoch}\right)^{power} \tag{8}$$

Here $init\_lr$ represents the initial learning rate, and max_epoch is the maximum number of iterations. We set max_epoch to 450 and $power$ to 0.9. While we were doing experiment on the Cityscapes dataset, $init\_lr$ is 5×10^{-4} and batchsize is 10. Moreover for doing experiment on the CamVid dataset, $init\_lr$ is 1×10^{-3} and batchsize is 32.

With reference to ENet [29] and SegNet [30], we use category weights to improve the problem of category imbalance in the CamVid dataset, which is defined as:

$$W_{class} = \frac{1}{\ln(c + p_{class})} \tag{9}$$

Here c is an additional hyperparameter, we set it to 1.10, p_{class} represents each category probability.

We adopted random horizontal flipping, mean subtraction, and multi-scale methods for the input image during training for data augmentation strategies. The multi-scale include {0.75, 1.0, 1.25, 1.5, 1.75, 2.0}. In the process of training, validation, and testing, we adjusted the resolution of the input images of Cityscapes and CamVid to 512×1024 and 360 × 480 respectively. To further improve the segmentation performance of the model, we also adopted the online hard example mining algorithm [31] on the Cityscapes dataset.

The auxiliary loss function only needs a very low computational cost to improve the feature expression of model ability in the training stage and removed in the forward inference process of model. In addition to obtain the loss function $loss_1$ at the end of JPANet, we also

Table 1 Choice of weight coefficient of loss function

λ_1	λ_2	mIoU(%)
0	0	69.74
0.2	0.4	70.80
0.4	0.6	71.62
0.6	0.8	71.27
0.8	1	70.96

obtain two auxiliary functions $loss_2$ and $loss_3$ at the end of the backbone network and the end of the JFP module respectively. Therefore, the loss function in the training stage is:

$$loss = loss_1 + \lambda_1 \cdot loss_2 + \lambda_2 \cdot loss_3 \tag{10}$$

Here λ is the weight of auxiliary loss. As shown in Table 1, when λ_1 is 0.4 and λ_2 is 0.6, the best results are obtained, which is 1.88% higher than when the auxiliary loss function is not used.

4.3 Ablation studies

In this section, we conducted a series of ablation experiments to prove the JPANet model effectiveness and its three components. All ablation experiments are done on the Cityscapes dataset, where mIoU results on the test set.

4.3.1 Ablation experiments on different lightweight backbone networks

In the down-sampling process, the ability of the network to extract features and realize model translation invariance, rotation invariance, and scale invariance is crucial for real-time semantic segmentation. In order to explore the impact of different lightweight backbone networks on the comprehensive performance of JPANet, we use three different lightweight backbone networks ShuffleNet [14], MobileNet [16], GhostNet [17] to construct JPANet-S, JPANet-M and JPANet-G. The experimental results in Table 2 show that although JPANet-S has the lowest number of parameters and the fastest inference speed under the same input resolution, its segmentation accuracy is the lowest among the three networks, i.e., only 66.69%. From Table 2, we can also see that the parameter amount of JPANet-G (JPANet) is only 0.44M higher than JPANet-M, but the computational complexity of JPANet-G is 12.81% lower than JPANet-M, and the inference speed is 17.54% higher. At the same time, the segmentation accuracy has also increased by 2.01%. It can be seen that the comprehensive performance of JPANet-G is the best among the above three networks, so we choose GhostNet as our lightweight backbone network in subsequent experiments.

Table 2 Evaluate the impact of different lightweight backbone networks on our model

Backbone	Input Size	Params(M)	GFLOPs	FPS	mIoU(%)
JPANet-S	512 × 1024	2.30	12.37	172.4	66.69
JPANet-M	512 × 1024	3.05	12.49	93.5	69.61
JPANet-G	512 × 1024	3.49	10.89	109.9	71.62

Table 3 Evaluate the impact of different dilation rates on the Cityscapes test set

Dilated rate	mIoU(%)
1, 2, 4, 8	70.27
1, 2, 5, 9	71.62
1, 3, 6, 9	71.37
1, 6, 12, 18	71.41

4.3.2 Ablation experiment for dilated rate

We used four different dilated rates in the JFP module to obtain the image multi-scale information, namely {1, 2, 5, 9}. To verify this dilated sequence validity, we set up three other dilated sequence schemes in the JFP module for comparison. As shown in Table 3, when using the {1, 2, 5, 9} dilated sequence, JPANet reached 71.62% mIoU in the Cityscapes test set. When we change the dilated sequence to {1, 2, 4, 8}, the performance drops by 1.35%, which shows the necessity of increasing the dilated rate in the JFP module. When the dilated sequence continues to increase, the model performance drops by about 0.2%, so we conclude that when the dilated sequence {1, 2, 5, 9} is used, the model achieves the optimal result.

4.3.3 Ablation experiment on each component

In this section of the experiment, we use different combinations of the JFP module, SDE module, and BFF module to verify each module impact on segmentation performance. As shown in Table 4, when only the lightweight backbone network is used, and the modules we

Table 4 Evaluate the impact of different components on the Cityscapes test set

Backbone	JFP	SDE	BFF	mIoU(%)
√				60.90
√	√			66.89
√	√	√		70.20
√		√	√	68.34
√	√	√	√	71.10

propose are not used, the backbone network only achieves 60.90% mIoU. When the JFP module is connected behind the backbone network, the model segmentation accuracy is improved by 5.99%. This is because the backbone network directly performs 32 times upsampling, and the high-level semantic information lacks the perception of low-level spatial information, resulting in unsatisfactory segmentation of category boundaries. The JFP module integrates the features of three different stages. The network high-level semantic information has a certain perception of the image geometric information. So, the segmentation effect is obviously increased. It can be seen from the last two rows of Table 4 that when only the SDE module and the BFF module are used, and the JFP module is not used, the segmentation accuracy of the model is only 68.34%. After using the JFP module, the accuracy of the model increased by 2.76%. Suppose only the element-level addition method is used to fuse the deep semantic information extracted by the JFP module and the shallow detail information extracted by the SDE module. In that case, the model segmentation accuracy is only 70.20%. If the BFF module establishes the pixel channel and position dependency for feature fusion, the model segmentation performance increases by 70.20% to 71.10%.

4.3.4 Ablation experiment on different context modules

Context modules such as ASPP [23], PPM [24] and their variant modules are widely used to capture feature representations of different scales in the network. To explore the effectiveness of the JFP module relative to other context modules, we used ASPP module, PPM module, JPU module [32] to replace the JFP module in JPANet, thus constructing three heterogeneous JPANet variant networks. It can be seen from Table 5 that the three JPANet heterogeneous networks constructed using ASPP, PPM, and JPU modules not only decrease the segmentation accuracy by 0.48%, 1.02%, and 0.84% respectively, but also increase the number of parameters by 96.27%, 54.44%, and 51.28%. The computational complexity of the three heterogeneous networks composed of ASPP, PPM, and JPU modules is 10.74%, 2.20%, and 65.74% higher than that of JPANet using JFP modules. It can be seen that the JFP module we proposed can achieve higher performance at a lower computational cost, which proves the effectiveness of the JFP module.

Table 5 Evaluate the impact of different context modules on the Cityscapes test set

Method	Params(M)	GFLOPs	FPS	mIoU(%)
ASPP [23]	6.85	12.06	103.1	70.07
PPM [24]	5.39	11.13	109.9	69.53
JPU [32]	5.28	18.05	85.5	69.71
JFP	3.49	10.89	109.9	**70.55**

Bold entries highlight that our method achieves better results than other methods for the same metrics

4.3.5 Ablation experiment on different feature fusion methods

Feature fusion is a commonly used method in semantic segmentation, which can compensate for the serious loss of high-level feature space information and low-level feature semantic categories with poor prediction results. Given the complementary characteristics between high-level features and low-level features, the most common approach is to use simple channel concatenation, pixel-wise addition, and other methods to fuse these two types of information. To verify our proposed BFF module effectiveness, we use different feature fusion methods to replace the BFF module and then compare it. As shown in Table 6, the accuracy obtained by using the BFF module is 0.96% higher than the concatenation method, the computational complexity is almost reduced by 1/4, and the parameters are only 0.12M more. This is because the concatenation method merges high-level semantic information and low-level spatial information on the channel, and does not consider the interdependence of pixels in the channel and spatial position. So, its segmentation results in inferior effects as the BFF module. Since, the BFF module focuses on the internal correlation information between pixels from the channel and spatial position dimensions, FFM only focuses on the channel dimensions between pixels and ignores pixel positions relationship. Therefore, it can be seen from Table 6 that the parameters of the BFF module, and the FFM module are almost the same, but the mIoU obtained by the BBF module is 1.44% higher than that of the FFM module.

4.4 Performance comparison analysis

Our proposed JPANet has achieved very good results on the two challenging urban road scene datasets, Cityscapes

Table 6 Evaluate the impact of different feature fusion methods on the Cityscapes test set

Method	Params(M)	GFLOPs	FPS	mIoU(%)
Add	3.20	8.47	117.6	70.20
Concatenation	3.37	14.22	97.1	70.14
FFM [12]	3.45	9.88	107.5	69.66
CBAM [28]	3.64	12.1	106.4	70.72
BFF	3.49	10.89	109.9	**71.10**

Bold entries highlight that our method achieves better results than other methods for the same metrics

Table 7 Comparison of segmentation performance between the most advanced methods on the Cityscapes test set

					FPS		
Method	Input Size	Pretrain	Params(M)	GFLOPs	Others GPU	1080Ti GPU	mIoU(%)
SQ [33]	1024 × 2048	ImageNet	–	270	16.7	–	59.8
ESPNet [34]	512 × 1024	No	0.36	–	112.9	–	60.3
NDNet45-FCN5-LF [35]	512 × 1024	No	1.1	8.4	111.1	–	61.6
EFSNet [36]	512 × 1024	No	0.17	–	107	–	61.9
ThunderNet [37]	512 × 1024	No	4.7	–	96.2	–	64.0
ADSCNet [38]	512 × 1024	No	–	8.2	–	76.9	67.5
LiteSeg [11]	360 × 640	ImageNet	4.38	4.9	–	161	67.8
RPNet [39]	512 × 1024	No	1.89	20.71	–	123	67.9
ERFNet [40]	512 × 1024	No	2.1	–	83	–	68.0
BiSeNet [12]	1024 × 2048	ImageNet	5.8	14.8	105.8	–	68.4
DSNet [41]	512 × 1024	ImageNet	11.9	–	–	68	69.1
AGLNet [42]	512 × 1024	No	1.12	13.88	–	52	70.1
ICNet [43]	1024 × 2048	Coarse	26.5	28.3	30.3	–	70.6
DFANet [4]	1024 × 1024	ImageNet	7.8	3.4	100	–	71.3
MSFNet [44]	512 × 1024	ImageNet	–	24.2	117	–	71.3
JPANet(Ours)	512 × 1024	ImageNet	3.49	10.9	–	109.9	71.62

“-” means that the original paper did not give corresponding data. Because different models use different GPUs when measuring inference speed, we divide the GPU for inference speed measurement into GTX 1080Ti and other types of GPUs based on the JPANet experimental platform

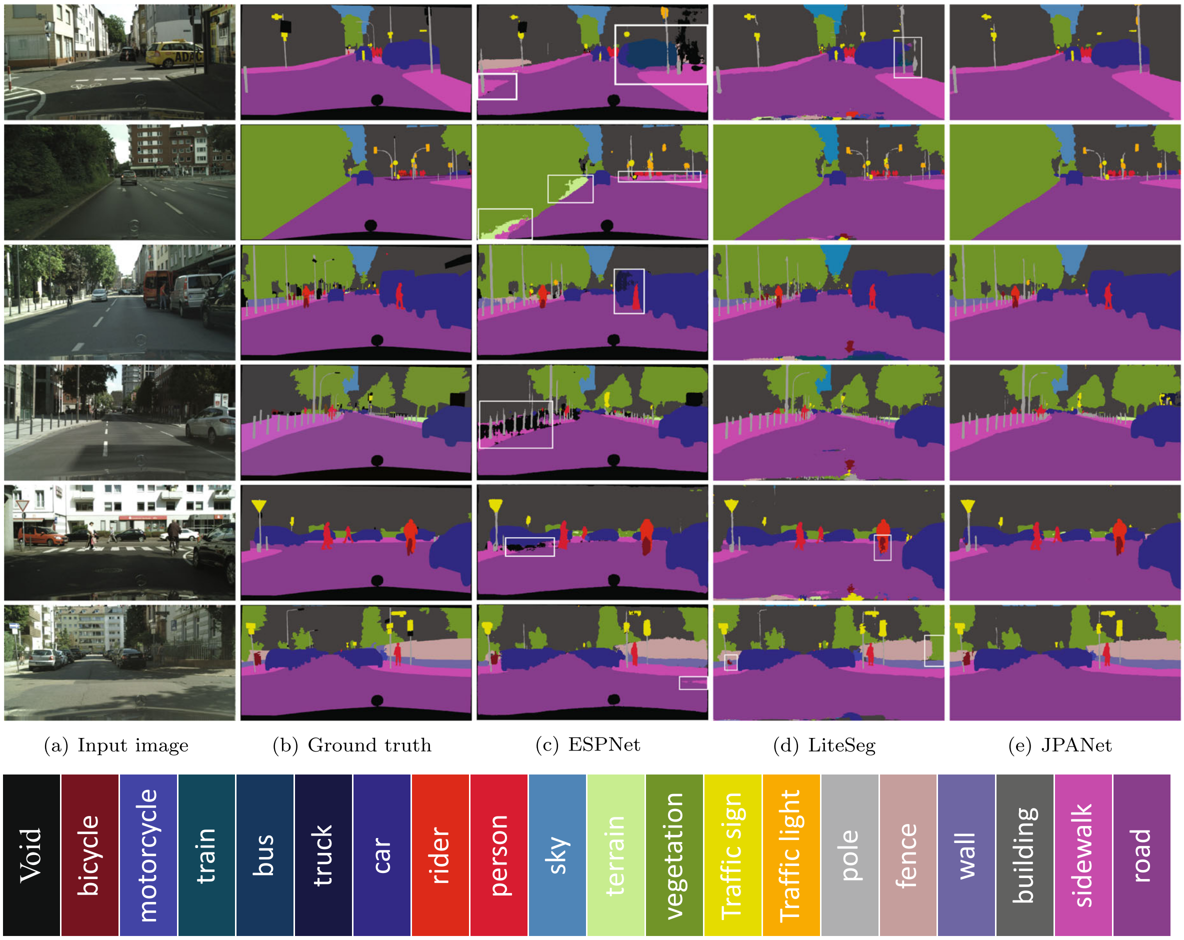
(a) Input image (b) Ground truth (c) ESPNet (d) LiteSeg (e) JPANet

Fig. 4 Visual comparisons in terms of the cityscapes validation set. From left to right are input images, ground truth, segmentation outputs from ESPNet, LiteSeg, and our JPANet

Table 8 Class mIoU scores on Cityscapes test set for the per-class category

Method	Road	Sidewalk	Building	wall	Fence	Pole	Traffic light	Traffic sign	Vegetation	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	Motorcycle	Bicycle	Class IoU	Category IoU
ENet [29]	96.3	74.2	75.0	32.2	33.2	43.4	34.1	44.0	88.6	61.4	90.6	65.5	38.4	90.6	36.9	50.5	48.1	38.8	55.4	58.3	80.4
SQ [33]	96.9	75.4	87.8	31.6	35.7	50.9	52.0	61.7	90.9	65.8	93.0	73.8	42.6	91.5	18.8	41.2	33.3	34.0	59.9	59.8	84.0
ESPNet [34]	97.0	77.5	76.2	35.0	36.1	45.0	35.6	46.3	90.8	63.2	92.6	67.0	40.9	92.3	38.1	52.5	50.1	41.8	57.2	60.3	82.2
EFSNet [36]	96.6	74.9	86.4	37.5	39.6	48.0	49.8	55.1	89.7	64.9	92.8	70.3	51.5	90.2	43.0	49.7	41.6	41.5	53.5	61.9	82.8
CGNet [45]	95.5	78.7	88.1	40.0	43.0	54.1	59.8	63.9	89.6	67.6	92.9	74.9	54.9	90.2	44.1	59.5	25.2	47.3	60.2	64.8	85.7
ERFNet [40]	97.7	81.0	89.8	42.5	48.0	**56.3**	59.8	65.3	91.4	68.2	94.2	76.8	57.1	92.8	50.8	60.1	51.8	47.3	61.7	68.0	86.5
JPANet(Ours)	**98.1**	**82.7**	**90.6**	**47.1**	**48.2**	53.1	**60.2**	**66.1**	**91.6**	**69.2**	**94.5**	**78.6**	**60.5**	**94.0**	**64.2**	**75.9**	**66.5**	**63.7**	**65.9**	**71.6**	**86.7**

Bold entries highlight that our method achieves better results than other methods for the same metrics

and CamVid. This section compares the segmentation accuracy, model parameters, and computational complexity with the most advanced models on Cityscapes and CamVid, respectively. We did not use any testing techniques in the evaluation process, such as multi-crop test and multi-scale test.

4.4.1 Comprehensive performance comparison on the cityscapes dataset

It can be observed from Table 7, that the inference speed of JPANet is comparable to the current state-of-the-art methods, but our model is simpler and more efficient, ensuring comparability in terms of parameters, computational complexity, and accuracy. The results on the Cityscapes test set show that our method achieves 71.62% mIoU with an FPS of 109.9. Below we will compare with JPANet latest method in terms of inference speed and segmentation accuracy.

Compared with BiSeNet, which has somehow same inference speed to ours. BiSeNet has 5.8M parameters, while JPANet only has 3.49M, which is 40% lower than BiSeNet. The computational complexity of BiSeNet is 14.8G, and JPANet is 26% lower than it, only 10.9G. Simultaneously, the segmentation accuracy of JPANet is 3.22% higher than BiSeNet, reaching a staggering 71.62%, which is a very considerable performance gain. Compared with our MSFNet in segmentation accuracy. Although, the accuracy of MSFNet is only 0.32% lower than ours, its computational cost is extremely expensive, and its computational complexity is as high as 24.2G. The

Table 9 Comprehensive performance comparison on the CamVid test set

Method	Input Size	Params(M)	FPS	mIoU(%)
ENet [29]	360 × 480	0.36	227	51.3
SegNet [30]	360 × 480	29.5	46	55.6
FSSNet [46]	360 × 480	0.2	179	58.6
EFSNet [36]	360 × 480	0.17	332	61.1
ERFNet [40]	360 × 480	2.06	164	63.7
DFANet [4]	720 × 960	7.8	120	64.7
RPNet [39]	360 × 480	1.89	149	64.8
BiSeNet [12]	720 × 960	5.8	–	65.6
SwiftNet [13]	720 × 960	12.9	–	65.7
EDANet [47]	360 × 480	0.68	163	66.4
DABNet [48]	360 × 480	0.81	117	66.4
ICNet [43]	720 × 960	26.5	27.8	67.1
JPANet-S	360 × 480	2.30	434	63.80
JPANet-M	360 × 480	3.05	256	68.29
JPANet-G	360 × 480	3.49	294	67.45

computational complexity of our JPANet is 55% lower than it. This huge performance improvement is more favorable to deploy our method on edge devices with limited computing resources.

Figure 4 is the JPANet visualization result on the Cityscapes validation set. To facilitate comparison, we use white boxes in Fig. 4 to mark areas where segmentation errors are more obvious in ESPNet and LiteSeg. For example: In the first row of Fig. 4, ESPNet and LiteSeg have obvious mis-segmentation for the car in the white box, while JPANet segmentation is almost perfect. In the second row, LiteSeg and JPANet divide the boundary between vegetation and road, while ESPNet divides the vegetable boundary into terrain.

We can see from Table 8 that JPANet achieved the highest scores in 18 of the 19 classification categories. It is because, JPANet emphasizes the importance of shallow spatial information, the improvement of JPANet on small object samples is the most obvious. For example, the JPANet accuracy on the traffic light and traffic sign are 24.6% and 19.8% higher than ESPNet, respectively. Besides, JPANet also pays attention to extracting multi-scale semantic information. Thus JPANet also improves the segmentation results of large targets to a certain extent. For example, the accuracy of JPANet on sidewalk and car is 1.7%, and 1.2% higher than the state-of-the-art ERFNet, respectively.

4.4.2 Comprehensive performance comparison on the CamVid dataset

We show in Table 9 the comparative data of JPANet composed of three different lightweight backbone networks and other models on the camvid test set. JPANet can not only achieve 67.45% mIoU but also obtain 294 FPS when we input 360 × 480 low-resolution images. The data in Table 9 once again proves the effectiveness of the JPANet model. Figure 5 shows the visual comparison effect of JPANet on the CamVid test set.

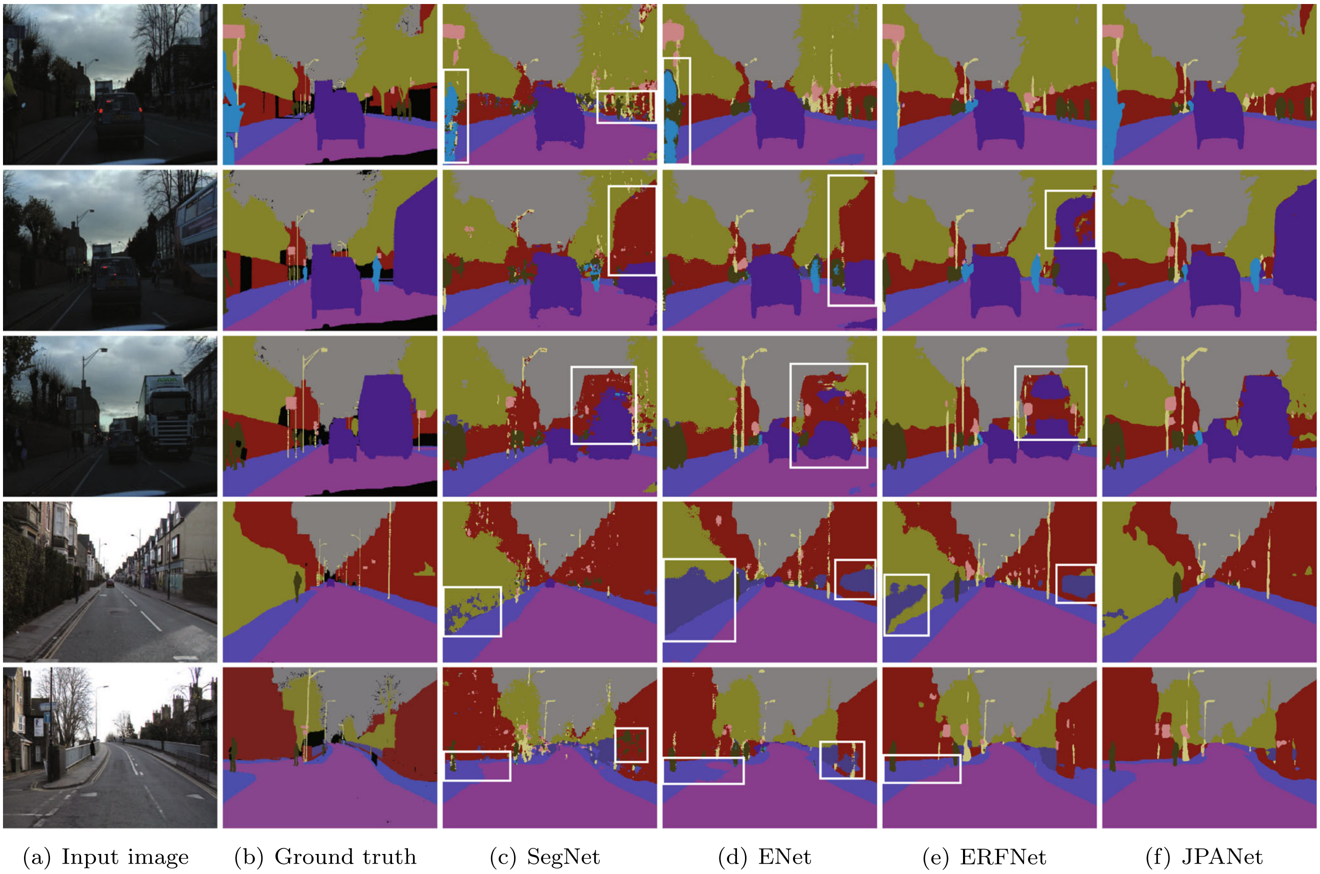

Fig. 5 Visual comparisons in terms of the CamVid test set. From left to right are input images, ground truth, segmentation outputs from SegNet, ENet, ERFNet, and our JPANet

5 Conclusion

We proposed a JPANet based on the JFP module, the SDE module, and the BFF module for real-time semantic segmentation in urban scenes. Among them, the JFP module effectively captures deep semantic information at different scales by combining three different stages of the deep network to obtain a more accurate representation of feature information. The SDE module uses the shallow dense texture information and position information of the network to capture multi-level spatial detail information. Finally, we used the BFF module to fuse the high-level semantic features and low-level spatial features with information complementarity by establishing the dependency of the feature information in the channel dimension and the location dimension. Our experimental results on two datasets show that JPANet has achieved the best performance on two extremely challenging and complex urban road scene datasets (Cityscapes and CamVid).

References

1. Hu X, Jing L (2020) LDPNEt: A lightweight densely connected pyramid network for real-time semantic segmentation. IEEE Access 8:212647–212658
2. Yu C, Wang J, Gao C, Yu G, Shen C, Sang N (2020) Context prior for scene segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 12416–12425
3. Zhong Z, Lin ZQ, Bidart R, Hu X, Daya IB, Li Z, Zheng W, Li J, Wong A (2020) Squeeze-and-attention networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 13065–13074
4. Li H, Xiong P, Fan H, Sun J (2019) Dfanet: Deep feature aggregation for real-time semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 9522–9531
5. Zhang B, Li W, Hui Y, Liu J, Guan Y (2020) MFENEt: Multi-level feature enhancement network for real-time semantic segmentation. Neurocomputing 393:54–65
6. Hu P, Perazzi F, Heilbron FC, Wang O, Lin Z, Saenko K, Sclaroff S (2020) Real-time semantic segmentation with fast attention. IEEE Robot Autom Lett 6(1):263–270
7. Molchanov P, Tyree S, Karras T, Aila T, Kautz J (2017) Pruning convolutional neural networks for resource efficient inference. In: Proceedings of international conference on learning representations (ICLR), pp 1–17
8. Luo P, Zhu Z, Liu Z, Wang X, Tang X (2016) Face model compression by distilling knowledge from neurons. Proc AAAI Conf Artif Intell (AAAI) 30(1):3560–3566
9. Denton EL, Zaremba W, Bruna J, LeCun Y, Fergus R (2014) Exploiting linear structure within convolutional networks for efficient evaluation. Adv Neural Inform Process Syst 27:1269–1277
10. Jiang W, Xie Z, Li Y, Liu C, Lu H (2020) LRNNET: A light-weighted network with efficient reduced non-local operation for real-time semantic segmentation. In: 2020 IEEE international conference on multimedia & expo workshops (ICMEW), pp 1–6
11. Emara T, Abd El Munim HE, Abbas HM (2019) LiteSeg: A Novel Lightweight ConvNet for Semantic Segmentation. In: 2019 Digital image computing: Techniques and applications (DICTA), pp 1–7
12. Yu C, Wang J, Peng C, Gao C, Yu G, Sang N (2018) Bisenet: Bilateral segmentation network for real-time semantic segmentation. In: Proceedings of the european conference on computer vision (ECCV), pp 325–341
13. Orsic M, Kreso I, Bevandic P, Segvic S (2019) In defense of pre-trained imagenet architectures for real-time semantic segmentation of road-driving images. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 12607–12616
14. Howard A, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H (2017) Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861
15. Ma N, Zhang X, Zheng HT, Sun J (2018) Shufflenet v2: Practical guidelines for efficient cnn architecture design. In: Proceedings of the European conference on computer vision (ECCV), pp 116–131
16. Zhang X, Zhou X, Lin M, Sun J (2018) Shufflenet: An extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 6848–6856
17. Han K, Wang Y, Tian Q, Guo J, Xu C, Xu C (2020) GhostNet: More features from cheap operations. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 1580–1589
18. Huang Z, Wang X, Huang L, Huang C, Wei Y, Liu W (2019) Ccnet: Criss-cross attention for semantic segmentation. In: Proceedings of the IEEE international conference on computer vision (ICCV), pp 603–612
19. Fu J, Liu J, Tian H, Li Y, Bao Y, Fang Z, Lu H (2019) Dual attention network for scene segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 3146–3154
20. Zhou W, Yuan J, Lei J, Luo T (2020) TSNet: three-stream self-attention network for RGB-D indoor semantic segmentation. IEEE Intelligent Systems
21. Chen L, 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–C848
22. Chen L, Papandreou G, Schroff F, Adam H (2017) Rethinking atrous convolution for semantic image segmentation. arXiv:1706.05587
23. Chen L-C, Zhu Y, Papandreou G, Schroff F, Adam H (2018) Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European conference on computer vision (ECCV), pp 801–C818
24. Zhao H, Shi J, Qi X, Wang X, Jia J (2017) Pyramid scene parsing network. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 2881–C2890
25. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 3431–C3440
26. Ronneberger O, Fischer P, Brox T (2015) U-net: Convolutional networks for biomedical image segmentation. In: International conference on medical image computing and computer-assisted intervention, pp 234–241
27. Yu F, Koltun V (2015) Multi-scale context aggregation by dilated convolutions. arXiv:1511.07122
28. Woo S, Park J, Lee JY, Kweon IS (2018) Cbam: Convolutional block attention module. In: Proceedings of the European conference on computer vision (ECCV), pp 3–19
29. Paszke A, Chaurasia A, Kim S, Culurciello E (2016) Enet: A deep neural network architecture for real-time semantic segmentation. arXiv:1606.02147

30. Badrinarayanan V, Kendall A, Cipolla R (2017) Segnet: a deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans Pattern Anal Mach Intell 39(12):2481–2495
31. Shrivastava A, Gupta A, Girshick R (2016) Training region-based object detectors with online hard example mining. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 761–769
32. Wu H, Zhang J, Huang K, Liang K, Yu Y (2019) Fastfcn: Rethinking dilated convolution in the backbone for semantic segmentation. arXiv:1903.11816
33. Treml M, Arjona-Medina J, Unterthiner T, Durgesh R, Friedmann F, Schuberth P, Mayr A, Heusel M, Hofmarcher M, Widrich M, Nessler B, Hochreiter S (2016) Speeding up semantic segmentation for autonomous driving. In: MLITS NIPS Workshop 2(7)
34. Mehta S, Rastegari M, Caspi A, Shapiro L, Hajishirzi H (2018) Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation. In: Proceedings of the european conference on computer vision (ECCV), pp 552–568
35. Yang Z, Yu H, Feng M, Sun W, Lin X, Sun M, Mao Z, Mian A (2020) Small object augmentation of urban scenes for Real-Time semantic segmentation. IEEE Trans Image Process 29:5175–5190
36. Hu X, Wang H (2020) Efficient fast semantic segmentation using continuous shuffle dilated convolutions. IEEE Access 8:70913–70924
37. Xiang W, Mao H, Athitsos V (2019) ThunderNet: A turbo unified network for real-time semantic segmentation. In: 2019 IEEE winter conference on applications of computer vision (WACV), pp 1789–1796
38. Wang J, Xiong H, Wang H, Nian X (2020) ADSCNEt: Asymmetric depthwise separable convolution for semantic segmentation in real-time. Appl Intell 50(4):1045–1056
39. Chen X, Lou X, Bai L, Han J (2019) Residual pyramid learning for single-shot semantic segmentation. IEEE Trans Intell Transp Syst 21(7):2990–3000
40. Romera E, Alvarez JM, Bergasa LM, Arroyo R (2017) Erfnet: Efficient residual factorized convnet for real-time semantic segmentation. IEEE Trans Intell Transp Syst 19(1):263–272
41. Chen PR, Hang HM, Chan SW, Lin JJ (2020) DSNEt: An efficient CNN for road scene segmentation. APSIPA Trans Signa Inform Process 9:1–14
42. Zhou Q, Wang Y, Fan Y, Wu X, Zhang S, Kang B, Latecki L (2020) AGLNEt: Towards real-time semantic segmentation of self-driving images via attention-guided lightweight network. Appl Soft Comput 96:106682
43. Zhao H, Qi X, Shen X, Shi J, Jia J (2018) Icnet for real-time semantic segmentation on high-resolution images. In: Proceedings of the European conference on computer vision (ECCV), pp 405–420
44. Si H, Zhang Z, Lv F, Yu G, Lu F (2019) Real-time semantic segmentation via multiply spatial fusion network. arXiv:1911.07217
45. Wu T, Tang S, Zhang R, Gao J, Zhang Y (2020) Cgnet: A light-weight context guided network for semantic segmentation. IEEE Trans Image Process 30:1169–1179
46. Zhang X, Chen Z, Wu QMJ, Cai L, Lu D, Li X (2018) Fast semantic segmentation for scene perception. IEEE Trans Indust Inform 15(2):1183–1192
47. Lo SY, Hang HM, Chan SW, Lin JJ (2019) Efficient dense modules of asymmetric convolution for real-time semantic segmentation. In: Proceedings of the ACM Multimedia Asia, pp 1–6
48. Li G, Jiang S, Yun I, Kim J, Kim J (2020) Depth-Wise Asymmetric bottleneck with Point-Wise aggregation decoder for Real-Time semantic segmentation in urban scenes. IEEE Access 8:27495–27506

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

Xuegang Hu was born in Chongqing, China in 1965. He received the B.S. degree in applied mathematics from China West Normal University, China, in 1988, the M.S. and the Ph.D. degrees in applied mathematics from Sichuan University, China, in 1995 and 2006, respectively. From 2002 to 2008, he was an associate professor with Chongqing University of Posts and Telecommunications, China. Since then, he has been a professor with the University. He is a the author of 3 books, more than 80 articles and 3 inventions.

His research interests include digital image processing and analysis and partial differential equations and their applications.

Liyuan Jing was born in Sichuan, China in 1997. He received the B.S. degree in internet of things engineering from Chengdu College of University of Electronic Science and Technology of China, in 2019. He is currently pursuing the M.S. degree in electronics and communication engineering at Chongqing University of Posts and Telecommunications. His research interests include computer vision, deep learning, panoramic segmentation and semantic segmentation.

Uroosa Sehar Currently, she is pursuing master's degree in Software Engineering with major in Artificial Intelligence and Deep Neural Networks from Northeastern University (NEU), China. She received Bachelor of Engineering in Information Technology from University of Engineering and Technology (UET), Pakistan in 2016. Now a days her research interests include new topics in Computer Vision, Image Processing, Artificial Intelligence, and Robotics.

Affiliations

Xuegang Hu[1] · Liyuan Jing[2] · Uroosa Sehar[3]

Xuegang Hu
huxg@cqupt.edu.cn

Uroosa Sehar
1828053@stu.neu.edu.cn

[1] Key Lab of Intelligent Analysis and Decision on Complex Systems, Chongqing University of Posts and Telecommunications, Chongqing, 400065, China

[2] Multimedia Communications Research Laboratory, Chongqing University of Posts and Telecommunications, Chongqing, 400065, China

[3] Cross Media Artificial Intelligence Laboratory, Northeastern University, Shenyang, 110000, China