# Lightweight Semantic Segmentation Network based on Attention Feature Fusion

Xianyan Kuang, Ping Liu, Yixi Chen, Jianhua Zhang

***Abstract*—With the popularization of intelligent mobile devices, the lightweight semantic segmentation networks for terminal-oriented have made great progress. However, the low and middle-level features are not be fully used in the networks, and It's often overlooked that low-level noises on high-level features are superimposed, which can cause the important information to be blurred. In this paper, a lightweight semantic segmentation of Attention Feature Fusion Network (AFFNet) is proposed. In this network, a multi-branch structure is adopted to utilize feature information of different stages. An attention feature fusion module is designed, in which the feature information in each stage is weighted with the channel attention and spatial attention to avoid noise superimposition and information coverage in the fusion stage. The loss function of Object Weighted Focus Loss and Cross Entropy Loss (OWFL+CEL) is introduced in the training process to suppress the class imbalance problem. The results show that, with the operation of a single RTX2080 GPU, the mean intersection over union (mIoU) of the method on the Cityscapes dataset is 70.8%, and it reaches 35.7 frames per second(FPS) under high resolution input, which proves that this method has better performance and practical value than similar networks in the tasks of real-time semantic segmentation.**

***Index Terms*—lightweight network, semantic segmentation, attention mechanism, feature fusion, Cityscapes**

## I. Introduction

THE purpose of semantic segmentation is to label the image at the pixel level and decompose a whole scene into independent entities, which will help to reason different behaviors of the target, so as to realize higher-level visual problems. Since the Full Convolutional Networks (FCN) [1] was proposed, the image segmentation algorithm based on Convolutional Neural Networks (CNNs) has been applied and shows better performance than conventional algorithms, but there are still some problems to be addressed.

On the one hand, although many networks [2], [3], [4], [5], [6] have achieved high segmentation accuracy, they often run slowly as the cost of deeper networks and more parameters. In some applications with relatively lower computational power, such as intelligent mobile devices, there are higher requirements for the network model, algorithm complexity and response speed. However the above networks are difficult to meet these requirements. On the other hand, the continuous down-sampling operation in CNNs is easy to lose the spatial information of features, which results in inaccurate target positioning. In general, the low-level features with high resolution contain more finer spatial information, while the high-level features with low resolution contain rich semantic information. Some studies have noted these situations and solutions have been put forward. For instance, Encoder-Decoder structure [6] uses skip connection to fuse the feature information extracted by the encoder to the decoder part with corresponding resolution, which ensures that the decoding network has richer information to recovery the images. Authors in [7], [8] use various resolutions as inputs for multiple paths of the network. They extract more low-level features from high-resolution images and more high-level features from low-resolution images, combining the two methods to improve performance. However, the parameter sharing structure of this multi-path input network usually needs more complex training strategies to match it. Some improved schemes proposed in [9], [10], [11], combine the extracted low-level features with the high-level features before obtaining information, which make the network structure simpler while ensuring the network's performance. Stage-Pooling module (SPM) proposed in [12] can further strengthen the reuse of the characteristics of the network middle-layer, thus can improve the performance of network segmentation.

However, the following problems have not been solved by the above algorithms. In the feature fusion stage, the uncertain semantic information contained in the low-level features is superimposed on the high-level features as noise. The simple feature fusion method is easier to blur or even cover the important information in different stages. In addition, the imbalance of data classes will be inherited to the network during the training process, which can reduce the generalization ability.

Based on above observation, a network named Attention Feature Fusion Network (AFFNet) is proposed in this paper. A multi-branch architecture in AFFNet is adopted to strengthen the reuse of middle and low-level features in the network. In order to achieve better feature fusion effect, we design an Attention Feature Fusion Module, which uses attention mechanism to weight feature information, and

Manuscript received October 25th, 2021; revised July 9th, 2023. This work was supported by the National Natural Science Foundation of China (Grant No. 51268017)

Xianyan Kuang is a Professor in the Intelligent Traffic and Intelligent Computing Lab, School of Electrical Engineering and Automation, Jiangxi University of Science and Technology, Ganzhou 341000, China. (e-mail: xianyankuang@163.com).

Ping Liu is a postgraduate student in the Intelligent Traffic and Intelligent Computing Lab, School of Electrical Engineering and Automation, Jiangxi University of Science and Technology, Ganzhou 341000, China. (e-mail: 634084502@qq.com).

Yixi Chen is a postgraduate student in the Intelligent Traffic and Intelligent Computing Lab, School of Electrical Engineering and Automation, Jiangxi University of Science and Technology, Ganzhou 341000, China. (e-mail: 18873269615@163.com).

JianHua Zhang is a postgraduate student in the Intelligent Traffic and Intelligent Computing Lab, School of Electrical Engineering and Automation, Jiangxi University of Science and Technology, Ganzhou 341000, China. (e-mail: 2393320048@qq.com).

highlights important information in features to prevent them from being blurred and covered. In the network training stage, we use the Object Weighted Focal Loss (OWFL) [22] and Cross Entropy Loss (CEL) as the loss functions to suppress the imbalance of data classes.

## II. Related Work

Semantic Segmentation Network based on convolutional neural networks has been able to show the high segmentation accuracy close to human level. Except for the expansion in depth, these networks also put forward many effective methods to improve accuracy. PSPNet [18] adopted Spatial Pyramid Pool (SPP) to merge feature maps into different sizes and then connect them through up-sampling. The Deeplab series [2], [3], [4], [5] proposed that Atrous Convolutions can enhance the network receptive field. RefineNet is proposed in [39] in which the Refine module takes a feature map and its lower proportion in the encoder and fuses it into a feature map in the decoder.

Lightweight semantic segmentation network needs to achieve a good balance between reasoning speed and segmentation accuracy. In some studies, the deep convolution network with outstanding performance is pruned, and the sparse matrix is used for storage after eliminating the unimportant parameters in the network [13], [14]. The quantitative training is carried out on the network proposed in [15], [16], [17], in which the low-precision floating-point numbers are used instead of high precision. Meanwhile, reducing network parameters is one of the feasible ways to improve efficiency. The idea is to use some special methods instead of standard convolution with high computational cost, such as factorized convolution [19], group convolution [20] and deep separable convolution [21]. It is proved that these methods can effectively reduce the amount of network parameters and ensure the same performance as standard convolution. In addition, the bottleneck design of the module is also an important means to reduce network parameters [21], [22], [23]. The number of characteristic channels of input and output is compressed and restored by using point-wise convolution, thus the number of parameters of the intermediate convolution layer can be reduced.

Attention mechanism is added to the segmentation network to improve the accuracy significantly [24], [25], [26]. The Context Encoding Module [27] weights the feature channels by establishing the relationship between channels, highlighting the feature layers that are beneficial to segmentation. Convolutional Block Attention Module (CBAM) [28] can refine the feature map and the spatial information in each feature map, which has better effect than the attention mechanism of simple attention channel. Dual attention network [29] is a channel and spatial attention mechanism different from CBAM, it connects channel and spatial attention in parallel, which also shows excellent performance.

## III. Method

The proposed method consists of three parts. Firstly, a lightweight attention feature fusion semantic segmentation network AFFNet is designed. Considering the segmentation accuracy and reasoning speed, the low-level and middle-level features of the network can be reused by multi branch architecture without additional computation. Secondly, we design an attention feature fusion module based on attention mechanism, which can achieve better feature fusion effect and improve the ability of network to understand the whole scene. Finally, the Object Weighted Focus Loss and Cross Entropy Loss functions are introduced for network training, which can effectively suppress the class imbalance problem and improve the generalization ability of the network.

### A. *AFFNet Architecture*

AFFNet is designed and optimized based on the Fast-SCNN [11] network architecture. The network structure diagram is shown in Fig. 1, which includes Learning to Down-sample, Deep Branch, Shallow Branch, feature attention fusion module (AFFM) and classifier. In this section, we introduce the specific improvements for Fast-SCNN.

Firstly, Init-unit [23] and downsample unit [20] are used to replace the original down-sampling method in the Learning to Downsample. Convolution and pooling downsampling method are used together to make the low-level features of the network have finer spatial details, which is beneficial to the subsequent feature reuse.

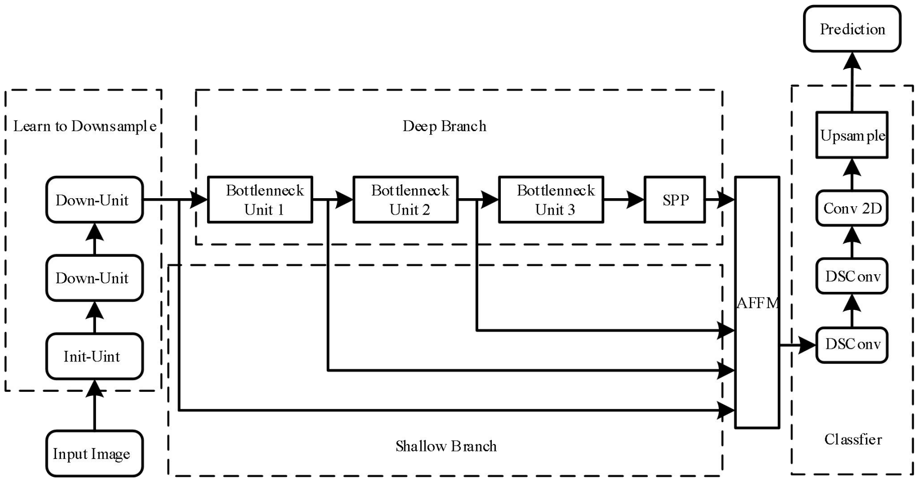
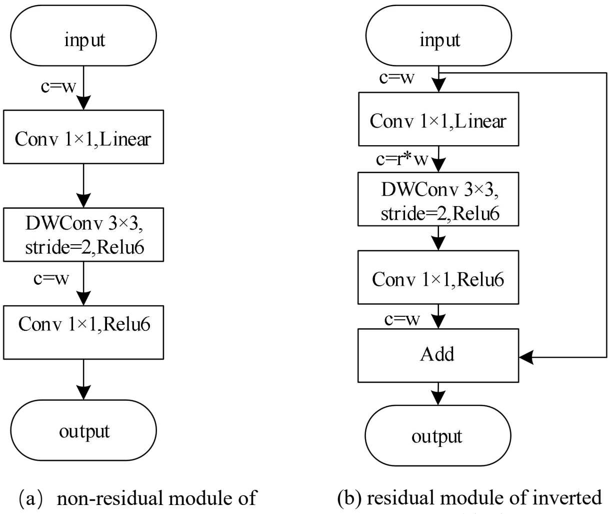


Fig. 1. Architecture of Attention Feature Fusion Network (AFFNet): each Bottleneck Unit contains three inverted residual modules, AFFM indicates the attention feature fusion module, DSConv is a depth-wise separable convolutions whose convolution kernel size is 3, Conv2D is a 3×3 standard convolution, Upsample indicates 8× upsampling of feature maps using bilinear interpolation.

Secondly, we add two branches in the shallow branch of Fast-SCNN to reuse the middle layer features. Many studies have shown the important influence of low-level features on network reasoning, because the fine spatial information in the low-level features and rich semantic information in high-level features can complement each other. However, the low-level features are relative to high-level, and the middle-level features are also meaningful for improving accuracy, which has been confirmed in [12]. Therefore, the multi-stage features are transmitted to the fusion module by adding branches. In order not to increase the computational burden of the network, there is no additional operation on the branches.

It should also be noted that the Inverted Residuals [21] are adopted to extract the feature information in the deep branch. The biggest advantage of this module is that deep convolution is used to save 8-9 times of computing resource consumption under the condition that the accuracy is only slightly reduced, which meets the requirements of building a lightweight network. The deep branch contains 9 Inverted Residuals modules, expansion ratio of the modules is set to 6, and the 1/2 down-sampling operation is carried out in the 1st and 5th modules, and the characteristic channels are raised to 96 and 128 in 4st and 6st modules. The module structure is shown in Fig. 2(a), the remaining modules are connected by residuals to strengthen the information flow between modules, as shown in Fig. 2(b).

Finally, SPP [18] is connected in series at the end of the deep branch to aggregate contextual semantic information based on different regions.



(a) non-residual module of inverted residual

(b) residual module of inverted residual

Fig. 2. Two inverted residual structures used in our paper: “c” represents the number of feature channels, “r” represents the expansion ratio is set to 6, "DWConv" indicates depth-wise convolutions, “stride” is the convolution step size and set to 2 when performing the down-sampling operation.

## B. Attention feature fusion module

Many multi-scale input network [7],[8], or multi-branch network [9],[10],[11] models, note that the fusion in different stage is of great significance to the network performance, but in order to ensure the high operation efficiency of the network, complex feature fusion methods will not be selected. On the other hand, the semantic information of features closer to the output are improved by the convolution operation. But the information in different stages is different, and it is difficult to judge which part is more important, so some important information may be blurred or even be covered.

We design the attention feature fusion module (AFFM) shown in Fig. 3. Firstly, the four input features highlight its semantic and spatial information through an attention branch and a pooling branch respectively. Then, the four attention branches are superimposed on the spatial dimension by Concat, and then channel dimension is reduced to 128 by a 3 × 3 convolution. The outputs of the four pooling branches are superimposed on the layer by Element-wise Add (EWA), a feature fusion method. Finally, the two fusion results are superimposed together to obtain a feature map with rich semantic and spatial information. The layer-by-layer superposition strategy, such as Yolo, is not adopted in this paper, instead, all input features are processed in parallel by the same operation and then fused. The details are discussed in IV.Section B.

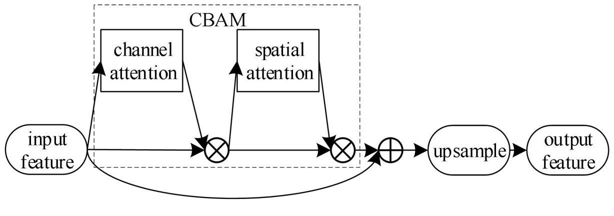
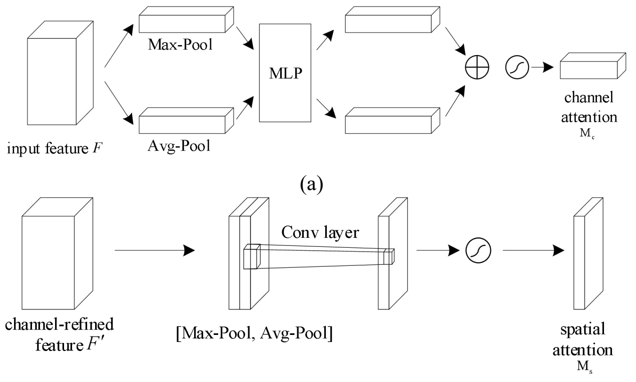


Fig. 3. The structure of an attention branch: “ ⊗ ” indicates dot multiplication and “ ⊕ ” indicates Element-wise Add.



(a)

(b)

Fig. 4. Diagram of channel attention and spatial attention: (a) The channel attention first enters the two different spatial context descriptors obtained from the input feature F after global Max-Pool and global Avg-Pool of the spatial dimension into the multi-layer perceptron(MLP), and then adds the two outputs of the MLP to the Sigmoid activation operation to generate the final Channel attention weight matrix $M_c$ .(b) The spatial attention first performs global Max-Pool and global Avg-Pool operations based on the channel, and then reduces the dimensionality to a single-channel feature after a Convolution layer, and finally generates spatial attention weight through the sigmoid function activation $M_s$ .

In the attention branch (Fig. 3.), the CBAM [28] module is used to weight the feature information, and bilinear interpolation is used to raise the feature map to the required resolution size for fusion. Compared with some channel attention [27],[30], the CBAM module can simultaneously weight the feature map and the feature channel to highlight important information, so that the network can understand which channel's feature map and which areas on the feature map are favorable for segmentation. As shown in Fig. 4, for an input image, channel attention focuses on “what” is meaningful while spatial attention focuses on “where” is an informative part. They both use Max-pooling and Avg-Pooling described in

[28] to improve the representation power of the model, which are expressed in Equation (1), (2). The channel attention and spatial attention mechanisms of CBAM module is shown in Fig. 3 and Equations (1), (2). The Max-pooling operation is adopted by the pooling branch to highlight the edge and texture information in the feature information, and the structure is shown in Fig. 5.

$$M_c(F) = \sigma\{MLP[AvgPool(F)] + MLP[MaxPool(F)]\} \quad (1)$$

$$\mathrm{M}_s(F`) = \sigma\{f^{3\times3}[AvgPool(F`); MaxPool(F`)]\} \quad (2)$$

Where σ is the sigmoid function, $F`$ is the result of multiplying the input feature and the channel weight $\mathrm{M}_c$, $f^{3\times3}$ is a convolutional layer with a convolution kernel of 3, [;] represents the feature after the channel is superimposed.

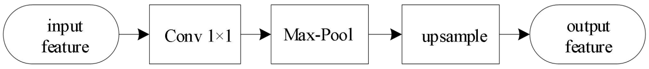


Fig. 5. The structure of the pooling branch: Point convolution (Conv 1 × 1) is used to increase the feature channel to 128, the Max-Pool window is set to 5, adopt the bilinear interpolation to upsample the feature map to 1/8 of the network input image.

### C. Loss Function

In the training process, the major classes of the dataset appear in high frequency, so it is easy to identify, and the proportion of minor classes is small, so it is difficult to classify. This imbalance problem will be inherited into the network, which will make the network under-predict the minor classes, and then affect the generalization ability of the network. The method to solve that problem of class imbalance can be summarized into data level methods and classifier level methods. The former uses under-sampling to decrease sample volume of secondary class, such as using class-aware sampling [34] to control the uniform distribution of categories in each training batch. The latter gives minor class higher weight by defining a new loss function to provide more information for network training, such as Weighted Cross Entropy Loss [12],[19],[23], Focus Loss [32] and Gradient Harmonizing Mechanism [33].

In the paper, the object weighted focal loss function [22] (OWFL) to suppress the class imbalance inheritance of data set from being inherited into the network, and Cross Entropy Loss (CEL) is used as the auxiliary loss function to improve the network training effect. The auxiliary loss weight is set to 0.5, and the final loss function is as follows.

$$L = OWFL + 0.5 \times CEL \quad (3)$$

The normalized object frequency weight and object magnitude weight together are used in OWFL, which can make the minor classes and hard objects provide more information to the loss function without affecting other objects. The Semantic Encoding Loss (SEL) [7], [22], which needs to add extra calculation cost in the network, is replaced by CEL. To construct OWFL, we first need to obtain the objects frequency weight according to (4).

$$\omega_i = \frac{1}{\ln(f_i + c)} \quad (4)$$

Where $f_i$ is the frequency of the with object appeared in the dataset, we set $c$=1.02 following Enet [23]. Then normalize the weights to [0,1] by dividing the maximum as (5).

$$\alpha_i = \frac{\omega_i}{\max(\omega_i)} \quad (5)$$

Next, the object order-of-magnitude weight is calculated by (6).

$$\gamma_i = \mathrm{OM}\left(\frac{f_i}{\min f_i}\right) \quad (6)$$

Where OM is a function to calculate the order of magnitude for a given number.

Finally, the OWFL can be calculated by (7).

$$OWFL(p_i) = -\alpha_i(1-p_i)^{\gamma_i}\ln(p_i) \quad (7)$$

Where $p_i$ is the probability of a sample belonging to the with object predicted by the network.

## IV. Experiments

The experiments include ablation, comparison, and test experiments. The feasibility and reasonableness of the proposed module structure and network architecture will be verified in the following experiments, before which the experimental data set and the specific implementation details as well as the performance evaluation criteria used in this paper are first introduced.

### A. Dataset and Implementation details

In the paper, Cityscapes [34] dataset is used as network training and test data. This dataset is a picture dataset focusing on the semantic understanding of urban street scenes, which contains a variety of stereo video sequences recorded in street scenes from 50 different cities. Besides 20000 weak annotation frames, it also contains 5000 high-quality pixel-level annotations, including a total of 2975 training set pictures, 500 validation sets, and 1525 testing sets. In the experiment, the paper only selected the image with fine label as the training, but in the final training of the model, each picture and corresponding label in the training set of Cityscapes are crop into eight 880× 880 patches with partial overlapping, obtained an extended dataset containing 23800 images for network pre-training. The overlapping clipping strategy can not only ensure that every region in the image is accessed, but also effectively reduce the cost and difficulty of network training.

The Pytorch framework is used to build the proposed AFFNet model and is run on single NVIDIA GeForce RTX 2080 under CUDA9.0.176 and CUDNN7.4.2. During the training, online data augmentation operations of random horizontal flipping and random pixel conversion are applied, and Adam optimizer is used to optimize the training. The initial learning rate is set to $5\times10^{-4}$, the momentum is set to 0.9 and the weight decay is set to $1\times10^{-4}$, the Batch size of each training is 10, and the Epoch is 150 times.

In the performance evaluation, the Intersection over Union (IoU) is adopted as the evaluation standard of segmentation accuracy. Frames Per Seconds (FPS) is adopted as the evaluation standard of model reasoning speed. The Memory Access Cost (MAC) and parameter quantity are used as auxiliary reference indexes.

$$\mathrm{IoU} = \frac{TP}{TP + FP + FN} \quad (8)$$

Where $TP$, $FP$ and $FN$ are respectively the number of true positive, false positive, and false negative at pixel level.

### B. Ablation Experiments

In this section, ablation experiments of AFFM module are conducted to explore its functions and advantages. The

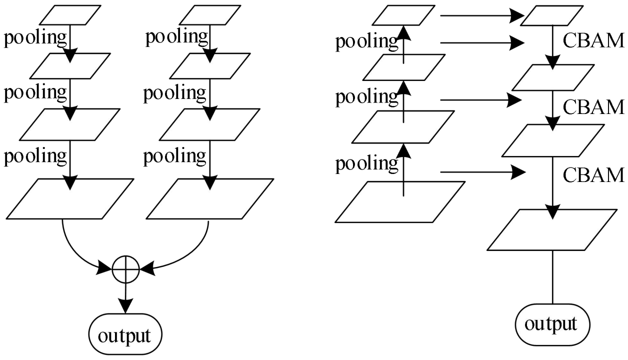
Cityscapes validation set is used as the default benchmark in the experiments. Firstly, the effect of different feature fusion methods applied to the two branches of AFFM are explored. At present, the feature fusion methods used in CNNs consist of Element-wise Add (EWA) [35],[36] and Concat [37], which are used to superimpose the feature map and channel dimension respectively. In AFFM, "Concat + EWA" method is used, in which Concat is used to fuse the features on the attention branch, and EWA is used to fuse the features on the pooling branch. This is because in the attention branch, after a series of operations, the attention layers and channels are weighted to highlight the important information, Concat is applied on the channel dimension, and then a convolution is used to fuse the weighted information, which helps preserve these weights as much as possible.. On the pooling branch, the edge, texture characteristic and other information on the feature layer are highlighted by Max-pooling, and the important information is further highlighted by EWA method. The experimental results showed in Table. I, which prove the correctness of our method.

TABLE I
COMPARISON OF FOUR DIFFERENT FEATURE FUSION COMBINATIONS ON CITYSCAPES VALIDATION SET

| Method | Parameters/×$10^6$ | mIoU/% |
|---|---|---|
| EWA | 1.46 | 67.28 |
| Concat | 1.70 | 67.71 |
| EWA+Concat | 1.62 | 68.77 |
| Concat+EWA | 1.62 | 69.48 |

before and after "+" respectively correspond to the fusion modes used on the attention branch and pooling branch, and the parameters are the total parameters of the network model.

According to the data in Table I, the difference between EWA and Concat method is not significant, which only is 0.26M, but the segmentation accuracy is improved significantly by the combination of the two methods. the segmentation accuracy of "Concat+EWA" method is 0.71% higher than that of "EWA+Concat" method under the same number of parameters, meanwhile is also the highest among the four combination methods.



(a) (b)

Fig. 6. Two different structures of AFFM: (a) Structure of "Concat+EWA", (b) Structure of FPN+PAN

Meanwhile, the structure of FPN+PAN (a feature fusion module) in YoloV4 [38] is designed to verify the rationality of AFFM's structure in parallel processing of feature information with two branches.

As shown in Fig. 6, the structure of pooling and CBAM in the two models is similar in processing the input features. Nevertheless, in Fig. 6(a), EWA method is used for feature fusion in the left branch, and the Concat method is used in the right branch. This structure has been verified in the above experiments. But in Fig. 6(b), all fusions follow the Concat method. We carry out the two modules for training, record their respective memory usage and mIoU on the Cityscapes validation set, the results are presented in table II.

Our results show that, although a simple structure occupies less computational cost, AFFM has a better feature fusion effect. We figure that this is due to the dual-branch structure of the model, in which the task of feature fusion can be better realized.

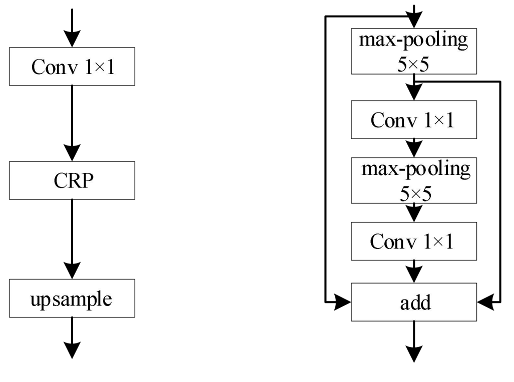
TABLE II
COMPARISON OF FEATURE FUSION MODULES WITH DIFFERENT STRUCTURES ON THE CITYSCAPES VALIDATION SET

| Module | Parameters/×$10^6$ | Mac/G | mIoU/% |
|---|---|---|---|
| FPN+PAN | 1.56 | 3.443 | 62.37 |
| AFFM | 1.62 | 5.256 | 69.48 |

FPN+PAN only represents the structure shown in Fig 6(b)

### C. Comparative experiments

In this section, two comparative experiments are introduced. Firstly, we compare the semantic segmentation performance of AFFNet and SPSSNet [12]. Both models are improved based on the Fast-SCNN architecture. However, in SPSSNet, four Stage-pooling modules (SPM) are used to build a bridge from deep branch to shallow branch, which is different from our AFFNet model. As shown in Fig.7. the high-level semantic information extracted by the deep branch is processed by a chained residual pooling (CRP) [39] after the point convolution ascending channel, then these features are up-sampled and directly superimposed on the shallow branch.



(a) (b)

Fig. 7. The structure of SPM: (a) The overall structure of SPM (b) The structure of CRP in SPM.

TABLE III
COMPARISON OF PERFORMANCE OF AFFNET AND SPSSNET ON CITYSCAPES VALIDATION SET

| Network | Parameter/×$10^6$ | Mac/G | mIoU/% | FPS/s-1 |
|---|---|---|---|---|
| SPSSNet | 1.42 | 27.281 | 68.21 | 40.0 |
| AFFNet | 1.62 | 5.256 | 69.48 | 41.7 |

Table III shows that, the mIoU and FPS of AFFNet are improved by 1.27% and 1.7, respectively, compared with SPSSNet under the same training environment, and the overall performance of AFFNet is better than SPSSNet. Benefit from the introduction of attention mechanism, AFFNet effectively avoids the problems of noise addition and information coverage in the feature fusion stage, which improves the accuracy. Although the parameters of the feature fusion module increase due to additional convolution operation, the memory access cost of the network is much lower than SPSSNet, and the final speed is also better.

In order to discuss the difference between two loss functions, OWFL+CEL and OWFL+SEL, Table IV lists the IoU of each specific class of AFFNet and DSNet in the Cityscapes test set. The results show that both have good IoU and performance in different data classes, but the overall performance of AFFNet is better than DSNet. The improvement of OWFL's performance for small objects segmentation will be discussed in the next section. As mentioned above, SEL requires additional calculations, so we think it is more appropriate to choose a loss function in the form of OWFL+CEL.

TABLE IV
CLASS-WISE IOU AND MIOU ON CITYSCAPES TEST SET OF AFFNET AND DSNET

| Network | Roa | Sid | Bui | Wal | Fen | Pol |
|---|---|---|---|---|---|---|
| DSNet | 96.6 | 77.2 | 89.8 | 50.3 | 53.5 | 57.7 |
| AFFNet | 96.9 | 83.2 | 90.4 | 78.3 | 66.7 | 40.3 |

| TLi | Tsi | Veg | Ter | Sky | Per | Rid |
|---|---|---|---|---|---|---|
| 45.9 | 65.3 | 89.9 | 57.7 | 92.1 | 73.6 | 55.9 |
| 48.5 | 61.4 | 90.1 | 40.3 | 92.6 | 71.5 | 52.4 |

| Car | Tru | Bus | Tra | Mot | Bic | mIoU |
|---|---|---|---|---|---|---|
| 90.8 | 64.4 | 77.7 | 72.8 | 52.5 | 69.4 | 69.3 |
| 92.1 | 81.5 | 83.9 | 81.5 | 60.7 | 62.0 | 70.4 |

### *D. Test experiments*

In this section, the proposed model is first trained 150 epochs on the extended data set, and then trained 150 epochs on the Cityscapes training set. The initial learning rate is adjusted to $2\times10^{-4}$ when training with Cityscapes, and the finally trained model is used to compare with the other models to test the advantages of our method.

Several single-branch semantic segmentation networks and multi-branch networks are used to compare the segmentation accuracy as shown in Table V. Results show that the segmentation accuracy increases with the increase of parameters in single-branch networks or multi-branch networks. However, Fast-SCNN can save a lot of network parameters after adding only one network branch, and the segmentation accuracy is same as ERFNet which performed best in single-branch. It proves that broadening the network width can indeed reduce the number of parameters while maintain or improve the segmentation performance. as a consequence, our AFFNet still has a 1.4% improvement in mIoU compared to SPSSNet in the multi-branch network.

TABLE V
COMPARISON RESULTS OF MIOU BETWEEN AFFNET AND CURRENT MAINSTREAM NETWORKS IN THE CITYSCAPES TEST SET

| Network | parameters/$\times10^6$ | mIoU/% |
|---|---|---|
| Enet [23] | 0.36 | 58.3 |
| ESPNet [40] | 0.4 | 60.3 |
| ERFNet [19] | 2.1 | 68.0 |
| Fast-SCNN [11] | 1.11 | 68.0 |
| SPSSNet [12] | 1.42 | 69.4 |
| AFFNet(ours) | 1.62 | 70.8 |

The reasoning speed of AFFNet, ERFNet and SPSSNet under different input sizes is also compared as shown in Table VI. It can be observed that the reasoning speed of AFFNet is much faster than ERFNet and SPSSNet when the input resolution is 512×512, and even under the high-resolution input of 1024×2048, AFFNet still guarantees an FPS of 35.7, which can meet the real-time requirements.

TABLE VI
COMPARISON RESULTS OF SEGMENTATION SPEED BETWEEN AFFNET AND CURRENT MAINSTREAM NETWORKS

| Network | 512×512 | | 512×1024 | | 1024×2048 | |
|---|---|---|---|---|---|---|
| | ms | FPS | ms | FPS | ms | FPS |
| ERFNet | 12 | 83.3 | 24 | 41.7 | 89 | 11.2 |
| SPSSNet | 7 | 135.5 | 20 | 50.0 | 32 | 31.25 |
| AFFNet | 6 | 166.6 | 16 | 62.5 | 28 | 35.7 |

The 19 classes are divided into three group according to the weight calculated by OWFL. The γ=0 group contains 10 concrete classed, which are the minor classes with less frequency in the dataset. The γ=2 group is the three major classes with the most frequency. The γ=1 group is between the above and contains 6 concrete classes. Meanwhile, we calculate the mIoU columns of the three networks in each group listed in Table VII. The segmentation accuracy of each class of AFFNet, ERFNet and SPSSNet in the Cityscapes is shown in Table VIII. It can be seen in Table VII that our OWFL compared with the Weighted Cross Entropy Loss (WCEL) used in ERFNet and SPSSNet has a slight decrease in the γ=1 group and the γ=2 group, but the mIoU of the γ=0 group has an increase of more than 10%, it can be infer that OWFL can suppress class imbalance problems with a slight reduced accuracy of the main class but a significantly improved accuracy of the secondary class. Specifically, Table VIII shows that the accuracy of most classes in the γ=0 group have a significant increase, except for a certain decline of the four classes of Traffic Light, Traffic Sign, Rider, and Bicycle. The performance decline is mainly due to the different distribution of the train and test datasets, and large number of small targets. The model may be difficult to learn the generalized features well from limited data. Therefore, OWFL can improve the overall segmentation performance of small targets and minor classes groups, but it cannot guarantee that the IoU of each specific class is higher than WCEL.

TABLE VII
THE MIOU OF Γ=0,1,2 GROUPS OF AFFNET, ERFNET AND SPSSNET

| Network | mIoU<br>（γ=0） | mIoU<br>（γ=1） | mIoU（γ=2） |
|---|---|---|---|
| ERFNet | 51.2% | 78.0% | 92.7% |
| SPSSNet | 57.3% | 77.6% | 93.0% |
| AFFNet | 67.7% | 75.6% | 92.5% |

### *E. Visualization*

The visualization results on the Cityscapes validation set are shown in Fig. 8. The segmentation effect of AFFNet is compared with that of Ground Truth and SPSSNet. Due to the introduction of attention modules at multiple levels, AFFNet can still identify distant targets in open road scenes, and the edges of near targets in complex scenes are also smoother than those of SPSSNet. Moreover, the stacked objects are usually difficult to segment, such as the motorcycle rider in the third row, but AFFNet performs better than others.

TABLE VIII
COMPARISON RESULTS OF VARIOUS CATEGORIES OF IOU IN CITYSCAPES BETWEEN AFFNET, ERFNET AND SPSSNET

| Network | IoU (%) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Wal | Fen | TLi | TSi | Rid | Tru | Bus | Tra | Mot | Byc |
| | γ=0 | | | | | | | | | |
| ERFNet | 41.6 | 45.3 | 60.5 | 64.6 | 56.4 | 45.7 | 60.6 | 27.0 | 48.7 | 61.8 |
| SPSSNet | 43.9 | 46.5 | 58.8 | 64.7 | 59.0 | 53.5 | 71.0 | 59.2 | 52.9 | 63.8 |
| AFFNet | 78.3 | 66.7 | 48.5 | 61.4 | 52.4 | 81.5 | 83.9 | 81.5 | 60.7 | 62.0 |

| Network | IoU (%) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Sid | Pol | Ter | Sky | Per | Car | Roa | Bui | Veg |
| | γ=1 | | | | | | γ=2 | | |
| ERFNet | 80.0 | 56.4 | 68.7 | 94.2 | 76.1 | 92.4 | 97.2 | 89.5 | 91.4 |
| SPSSNet | 80.8 | 53.1 | 68.7 | 94.2 | 76.2 | 92.7 | 97.7 | 89.8 | 91.5 |
| AFFNet | 83.2 | 40.3 | 74.1 | 92.6 | 71.5 | 92.1 | 96.9 | 90.4 | 90.1 |

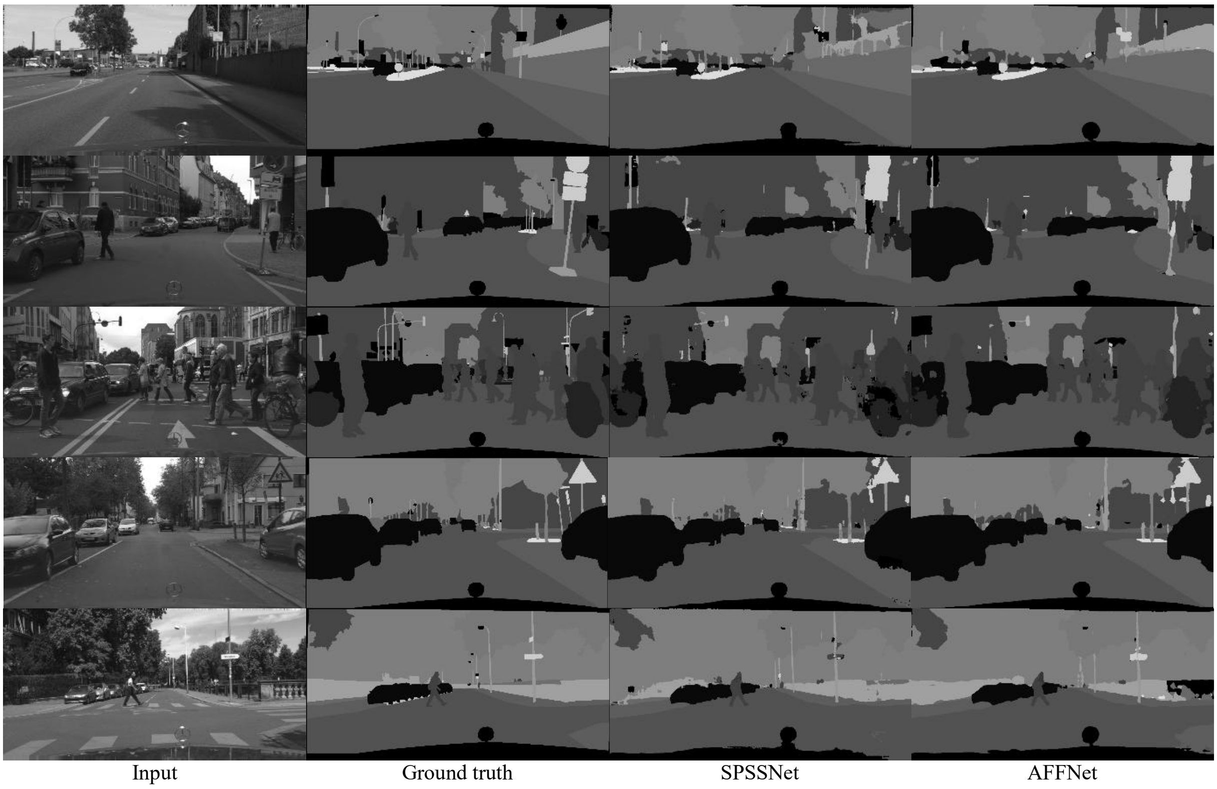


Fig. 8. Visual results of AFFNet on the Cityscapes validation set

## V. CONCLUSIONS

In this study, a real-time semantic segmentation network AFFNet is proposed, which uses a multi-branch architecture to effectively reuse the feature information of each stage. Aiming at the problem that noise superposition is neglected in the existing multi-branch network fusion stage, an attention feature fusion module is designed. The important information of features is highlighted by attention weighting and Max-pooling, a loss function in the form of OWFL+CEL is also introduced to suppress class imbalance in training. Experiments on the Cityscapes dataset show that the attention feature fusion module has a reasonable structure and helps to improve the network's ability to understand the whole scene. The OWFL+CEL loss function can effectively improve the overall segmentation performance of minor classes without increase in additional computational cost. Compared with AFFNet, the AFFNet method has better segmentation performance, and achieves the reasoning speed of 35.7 FPS, which can meet real-time requirements for the lower computational power devices with high-resolution input.

## REFERENCES

[1] J. Long, E. Shelhamer, T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, vol. 39, no. 4, pp. 640-651.

[2] L. C. Chen, G. Papandreou, I. Kokkinos, et al., "Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs," Computer Science, 2014, vol. 4, pp. 357-361.

[3] L. C. Chen, G. Papandreou, I. Kokkinos, et al., "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018, vol. 40, no. 4, pp. 834-848.

[4] L. C. Chen, G. Papandreou, F. Schroff, H. Adam, et al., "Rethinking Atrous Convolution for Semantic Image Segmentation," arXiv preprint arXiv:1706.05587, 2017.

[5] L. C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 801-818.

[6] V. Badrinarayanan, A. Kendall, R. Cipolla, "Segnet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, vol. 39, no. 12, pp. 2481-2495.

[7] R. P. K. Poudel, U. Bonde, S. Liwicki, C. Zach, et al., "Contextnet: Exploring Context and Detail for Semantic Segmentation in Real-Time," arXiv preprint arXiv:1805.04554, 2018.

[8] H. Zhao, X. Qi, X. Shen, J. Shi, J. Jia, "ICNet for Real-Time Semantic Segmentation on High-Resolution Images," in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 405-420.

[9] C. Yu, J. Wang, C. Peng, et al., "Bisenet: Bilateral Segmentation Network for Real-Time Semantic Segmentation," in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 325-341.

[10] H. Li, P. Xiong, H. Fan, J. Sun, "DFANet: Deep Feature Aggregation for Real-Time Semantic Segmentation," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 9522-9531.

[11] R. P. K. Poudel, S. Liwicki, R. Cipolla, "Fast-SCNN: Fast Semantic Segmentation Network," arXiv preprint arXiv:1902.04502, 2019.

[12] S. Mamoon, M. A. Manzoor, F. E. Zhang, Z. Ali, L. U. Jian-Feng, "SPSSNet: A Real-Time Network for Image Semantic Segmentation," Frontiers of Information Technology & Electronic Engineering, 2020, vol. 21, no. 12, pp. 1770-1782.

[13] S. Han, H. Mao, W. J. Dally, “Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding,” arXiv preprint arXiv:1510.00149, 2015.

[14] C. Li, C. J.R. Shi, “Constrained optimization based low-rank approximation of deep neural networks,” in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp732-747.

[15] I. Hubara, M. Courbariaux, D. Soudry, et al., "Binarized Neural Networks," Advances in Neural Information Processing Systems, 2016, vol. 29.

[16] I. Hubara, M. Courbariaux, D. Soudry, et al., "Quantized Neural Networks: Training Neural Networks with Low Precision Weights and Activations," The Journal of Machine Learning Research, 2017, vol. 8, no. 1, pp. 6869-6898.

[17] S. Wu, G. Li, F. Chen, L. Shi, "Training and Inference with Integers in Deep Neural Networks," arXiv preprint arXiv:1802.04680, 2018.

[18] H. Zhao, J. Shi, X. Qi, X. Wang, J. Jia, "Pyramid Scene Parsing Network," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2881-2890.

[19] E. Romera, J. M. Alvarez, L. M. Bergasa, R. Arroyo, "ENet: Efficient Residual Factorized ConvNet for Real-Time Semantic Segmentation," IEEE Transactions on Intelligent Transportation Systems, 2017, vol. 19, no. 1, pp. 263-272.

[20] X. Zhang, X. Zhou, M. Lin, J. Sun, "ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6848-6856.

[21] M. Sandler, A. Howard, M. Zhu, et al., "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4510-4520.

[22] W. Wang, Y. Fu, Z. Pan, X. Li, Y. Zhuang, "Real-Time Driving Scene Semantic Segmentation," IEEE Access, 2020, vol. 8, pp. 36776-36788.

[23] A. Paszke, A. Chaurasia, S. Kim, E. Culurciello, "ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation," arXiv preprint arXiv:1606.02147, 2016.

[24] X. Li, J. Ouyang, X. Zhou, "Supervised Topic Models for Multi-Label Classification," Neurocomputing, 2015, vol. 149, pp. 811-819.

[25] X. Li, Y. Wang, A. Zhang, C. Li, J. Chi, J. Ouyang, "Filtering Out the Noise in Short Text Topic Modeling," Information Sciences, 2018, vol. 456, pp. 83-96.

[26] A. Vaswani, N. Shazeer, N. Parmar, et al., "Attention Is All You Need," in Advances in Neural Information Processing Systems, 2017, pp. 5998-6008.

[27] H. Zhang, K. Dana, J. Shi, et al., "Context Encoding for Semantic Segmentation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7151-7160.

[28] S. Woo, J. Park, J. Y. Lee, I. S. Kweon, "CBAM: Convolutional Block Attention Module," in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 3-19.

[29] J. Fu, J. Liu, H. Tian, et al., "Dual Attention Network for Scene Segmentation," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 3146-3154.

[30] H. Jie, S. Li, S. Gang, S. Albanie, "Squeeze-and-Excitation Networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7132-7141.

[31] S. Li, Z. Lin, Q. Huang, "Relay Backpropagation for Effective Learning of Deep Convolutional Neural Networks," European Conference on Computer Vision, Springer, Cham, 2016, pp. 467-482.

[32] T. Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, "Focal Loss for Dense Object Detection," in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2980-2988.

[33] B. Li, Y. Liu, X. Wang, "Gradient Harmonized Single-Stage Detector," in Proceedings of the AAAI Conference on Artificial Intelligence, 2019, vol. 33, no. 01, pp. 8577-8584.

[34] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, B. Schiele, "The Cityscapes Dataset for Semantic Urban Scene Understanding," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 3213-3223.

[35] S. Wu, S. Zhong, Y. Liu, "Deep Residual Learning for Image Steganalysis," Multimedia Tools and Applications, 2018, vol. 77, no. 9, pp. 10437-10453.

[36] T. Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, S. Belongie, "Feature Pyramid Networks for Object Detection," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2117-2125.

[37] G. Huang, Z. Liu, V. D. M. Laurens, K. Q. Weinberger, "Densely Connected Convolutional Networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4700-4708.

[38] A. Bochkovskiy, C. Y. Wang, H. Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," arXiv preprint arXiv:2004.10934, 2020.

[39] G. Lin, F. Liu, A. Milan, C. Shen, I. Reid, "RefineNet: Multi-Path Refinement Networks for Dense Prediction," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019, vol. 42, no. 5, pp. 1228-1242.

[40] S. Mehta, M. Rastegari, A. Caspi, L. Shapiro, H. Hajishirzi, "ESPNet: Efficient Spatial Pyramid of Dilated Convolutions for Semantic Segmentation," in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 552-568.