IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Traitement du Signal
Vol. 39, No. 2, April, 2022, pp. 407-417
Journal homepage: http://iieta.org/journals/ts

MAF-DeepLab: A Multiscale Attention Fusion Network for Semantic Segmentation

Ning Chen[1,2*], Yupeng Chen[1,2], Qinfeng Wang[1,2], Shaopeng Wu[1,2], Hongyi Zhang[3]

[1] College of Marine Equipment and Mechanical Engineering, Jimei University, Xiamen 361021, China
[2] Marine Platform Support System, Fujian University Engineering Research Center, Xiamen 361021, China
[3] School of Opto-Electronics and Communication Engineering, Xiamen University of Technology, Xiamen 361024, China

Corresponding Author Email: cn1972@jmu.edu.cn

https://doi.org/10.18280/ts.390202

Received: 3 December 2021
Accepted: 8 March 2022

Keywords:
atrous spatial pyramid pooling, attention mechanism, deepLab V3+, multiscale features fusion, semantic segmentation

ABSTRACT

The existing semantic segmentation networks mostly focus on extracting and expressing deep image features. But none of them could adequately aggregate contextual information, or utilize features on different scales or layers. To improve prediction results, this paper proposes a multiscale attention fusion network for semantic segmentation (MAF-DeepLab), which highlights important features, and aggregates multi-scale features well. Firstly, the high-level semantic features and low-level texture features were captured by a lightweight feature extraction network. Secondly, cascaded spatial pyramidal pooling (CSPP) were employed to fuse feature extraction branches with different receptive fields, enhancing the correlation between multi-scale features. Finally, a bottom-up attention fusion module was adopted to guide the cascading aggregation of high-level and low-level features, producing detailed saliency maps. MAF-DeepLab achieved an excellent effect of semantic segmentation on two benchmark datasets: CamVid (74.8%) and Cityscapes (83.4%).

1. INTRODUCTION

Semantic segmentation, a challenging task in computer vision and pattern recognition [1] aims to recognize and classify every pixel of a given image from end to end [2]. It is widely utilized in such fields as autonomous driving [3, 4], augmented reality [5], and medical image processing [6, 7]. Traditionally, semantic segmentation splits each image into multiple regions [8], based on artificial features like texture, color, and shape, and then separates the target from the background. However, the segmentation results are often impractical, for the artificial features cannot describe high-level semantic information.

Recently, deep learning has been applied to the field of semantic segmentation. For example, the fully convolutional network (FCN) [9] transforms image-level classification networks into pixel-level semantic segmentation networks, by creative replacing fully connected layers of deep neural networks with fully convolutional layers. But the FCN cannot utilize the rich contextual information in the images, or utilize the features on different scales and layers. What is worse, some pixels are segmented incorrectly or overlooked, and the multi-scale objects often underperform. To address the two issues, many researchers have explored semantic segmentation based on the FCN.

To extract contextual features, DeepLabV1 [10] introduces the FCN-based atrous convolution to increase the receptive field without adding to the computing effort, and to reduce the resolution loss caused by down-sampling. Through spatial pyramid pooling (SPP), pyramid scene parsing (PSP) [11] network extracts features in parallel by pooling branches with four different receptive fields, and thus effectively aggregates global contextual information. Inspired by the SPP, DeepLabV2 [12] combines the atrous convolution with pyramid pooling into the atrous spatial pyramid pooling (ASPP) module, which considerably improves the accuracy of boundary segmentation. With the increase of dilation rate, atrous convolution would degrade gradually. To solve the problem, DeepLabV3 [13] relies on atrous convolution branches with different dilation rates to extract features in parallel, aggregates the features into multi-scale features, and adds the global average pooling (GAP) branch. However, the extracted features are weakly correlated, due to the sparse sampling of the atrous convolution. Hence, the network performs poorly in feature representation.

To fuse multi-scale features, SegNet [14] adds a decoder to the FCN to recover image size and spatial information based on the stored max-pooling index, during the up-sampling operation. The additional decoder makes it more accurate to localize image boundaries. To recover image boundaries, U-Net [15] adopts a U-shaped encoding-decoding structure to fuse deep and shallow features, using skip connections. DeepLabV3+ [16] adds a decoder module to refine the boundary information for DeepLabV3, and fuses high-level and low-level features to enhance the network ability of boundary segmentation. With the aid of a multiresolution fusion module, RefineNet [17] fuses feature maps of different resolutions, and refines the boundary information by a decoder module. For an image, the feature amount varies from region to region. The simple aggregation between high-level and low-level features may lead to incorrect segmentation.

Google's DeepLab [10-13, 16] family of models has greatly advanced the development of semantic segmentation, and realized an unprecedented accuracy of semantic segmentation on several benchmark datasets. This paper comes up with MAF-DeepLab, a semantic segmentation network based on DeepLabV3+ [16] for urban road scenes. The proposed network efficiently acquires 3D attention weights, and guides

the bottom-up fusion of high-level and low-level features through a feature fusion module. In this way, it strikes an efficient balance between computing complexity and segmentation accuracy. The contributions of this work can be summarized as follows.

(1) By replacing the backbone network, the CSPDarkNet was adopted as the feature extraction network [18], aiming to better characterize image details, and focus on multiscale targets.
(2) The cascaded spatial pyramidal pooling (CSPP) module was proposed to reuse the features, and to widen the receptive field, using the dense connection of each depth-separable convolutional layer. This enhances the correlation and utilization of multi-scale features.
(3) A multiscale attention fusion module was designed, which uses 3D attention weights [19] to guide the fusion of high-level and low-level features, and to gradually recover the details and spatial information of high-level semantic features.
(4) The proposed network, which requires only end-to-end training and not post-processing, outperforms advanced networks on CamVid [20] and Cityscapes [21] datasets in semantic segmentation.

2. LITERATURE REVIEW

This section reviews the related literature, highlighting the innovations of our approach.

2.1 Atrous spatial pyramid pooling

The ASPP proposed by DeepLabV3+ [16] consists of four parallel dilation convolutions with different dilation rates, and a global average pooling. The sparse sampling of atrous convolutions weakens the intercorrelation between the extracted features, making the features less expressive. Through dense concatenation, DenseASPP [22] combines the outputs of each dilated convolution to obtain a larger receiver field and denser sampled points. The rincipal component analysis network (PCANet) [23] filters and fuses appropriate contextual information efficiently, by learning long-term dependencies and embedding location information into features. The cascaded hierarchical atrous spatial pyramid pooling (CHASPP) [24] provides a novel layered structure, consisting of multiple convolutional layers, increases the sampling density to solve the degraded representation of detailed local features, which arises from ASPP sparse sampling.

Unlike previous methods, this paper uses multiple depth-separable convolutional branches to extract parallel multiscale features. The output of each branch undergoes channel concatenation, before being imported as the input of the next layer. In addition, global average pooling branches and short-circuit connections are added from input to output to alleviate the poor feature correlation due to sparse sampling.

2.2 Encoder-decoder

Most semantic segmentation networks adopt an encoder-decoder architecture. The encoder gradually augments the receptive field to obtain high-level semantic features, while the decoder slowly recovers the lost semantic information. Through pyramids, the context encoder network (CENet) [25] aggregates multiscale contextual information from deep convolution to shallow convolution through pyramids to obtain a more powerful representation of semantic features. Using a T-shaped decoder structure, Oršić and Šegvić [26] fused semantically rich high-level features with low-level features of textures through lateral connections, which greatly improves the recovery of image details. In DeepLabv3+ [16], the encoder occupies a significant memory overhead, which suppresses the segmentation efficiency. Meanwhile, the decoder fuses low-level texture features through up-sampling only once, without fully utilizing the intermediate layer features. Moreover, feature maps of different levels and scales vary in the abstraction level of semantic information, and in the weight of influence on feature learning. If the maps are directly fused with high-level features, the resulting noise would hinder subsequent feature extraction.

Drawing on the above approaches, this paper redesigns the encoder-decoder architecture of DeepLabV3+ [16], and constructs bottom-up fusion paths to aggregate deep and shallow features, making the segmentation more accurate.

2.3 Attention mechanism

The human vision tends to focus on the salient features of an image. Similarly, the attention mechanism helps the network compute the dependencies between image pixels through matrix operations. The dual attention network (DANet) [27] feeds the extracted features into the spatial attention module and the channel attention module in parallel to learn the spatial relationships and interdependencies between any two positions of the feature map. The cascaded convolutional neural network (CCNet) [28] has a cross-attention module to capture global contextual relationships, using long-range dependencies between pixels. Peng and Ma [29] proposed a dual-attention decoder, which consists of a channel attention mechanism and a spatial attention mechanism, that learns the spatial relationships and interdependencies between any two positions of the feature map. The self-attention feature fusion network (SA-FFNet) [30] introduces attention modules to compress the feature maps from vertical and horizontal directions, aiming to obtain better spatial feature maps with richer information contained in each pixel.

The existing attention modules differ along a single dimension of channel or space. This paper introduces a 3D attention mechanism that directly learns 3D weight coefficients of the influence on the target for different regions in the feature map. In this way, the learning attention weights that vary across channels and space become more flexible.

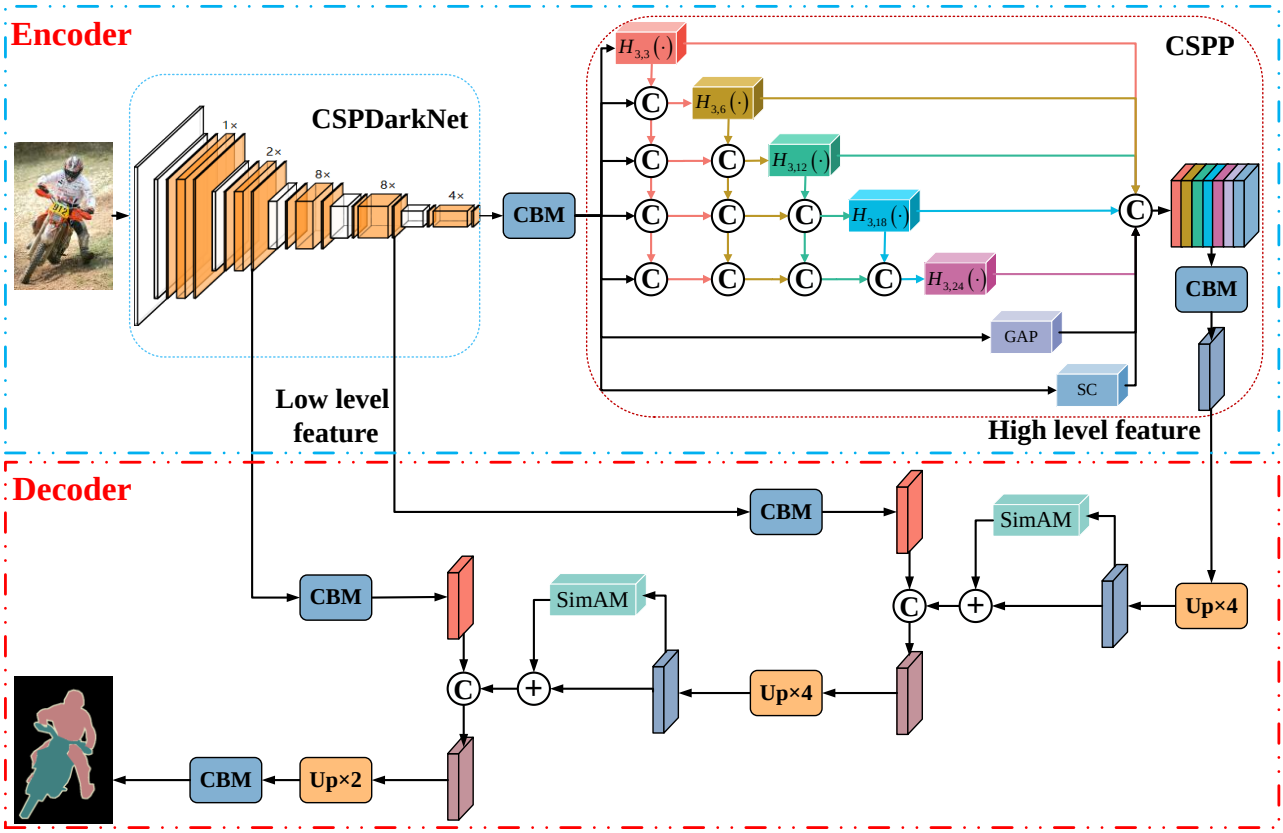
3. METHODOLOGY

This section firstly elaborates on the overall architecture of MAF-DeepLab, and then details the three parts of the model: the encoder backbone for extracting multiscale features, the cascade pyramid pooling module for improving the correlation of long-range features, and the decoder module for multi-scale feature fusion.

3.1 Overall architecture

This paper analyzes the memory usage of each module during DeepLabV3+ runtime and finds that the backbone network occupies the major memory overhead of the model,

which reduces the segmentation efficiency of the model. In addition, the features extracted by ASPP using large expansion rate convolution lacked correlation, resulting in the loss of local pixel information, which was particularly detrimental to the segmentation of small targets. More importantly, the decoder module only fuses the low-level texture features by up-sampling once, and the features in the middle layer are not fully utilized, which makes it difficult to recover the local and spatial information of the image and leads to poor object segmentation. For these reasons, this paper redesigns the encoder-decoder architecture of DeepLabV3+, producing the MAF-DeepLab framework (Figure 1).



Note: CBM is a nonlinear transformation function consisting of convolution, batch normalization, and Mish activation function; $H_{k,d}(\cdot)$ is the depth-wise separable atrous convolution (DSAConv) with the convolution kernel and dilation rate; Up, +, ×, and C indicate up-sampling, element-wise addition, element-wise multiplication, and channel-wise concatenation, respectively.

Figure 1. Architecture of MAF-DeepLab

In the new framework, the encoder takes a lightweight CSPDarkNet [18] as the backbone network, which continuously abstracts high-level semantic features after continuous down-sampling. Next, the CSPP module acquires correlated multiscale features by densely connecting branches with different receptive fields. After that, the attention fusion module in the decoder aggregates the multiscale features bottom-up with the low-level texture features obtained from the backbone network. Finally, the high-level features are reduced to the size of the input image through nonlinear interpolation, yielding the pixel-level semantic labels. Based on these improvements, the network can better describe image details, focus on multi-scale targets, and enhance the correlation of multi-scale features.

3.2 Backbone network replacement

The memory overhead is the main constraint of the segmentation efficiency of DeepLabV3+ [16]. This paper pretrains the CSPDarkNet [18] in the ImageNet to build a backbone network, and to eliminate the fully connected and pooling layers in the network (Figure 2). The purpose is to reduce model complexity, and enhance the extraction capability of multi-scale features, from the perspective of network structure design.

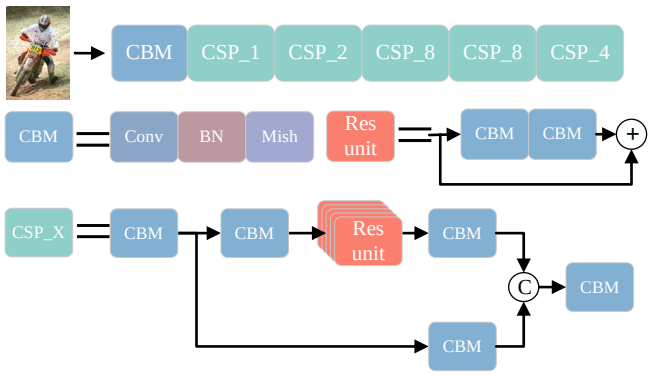


Figure 2. Structure of CSPDarkNet

More specifically, the smooth Mish [31] activation function prevents saturating gradients, and spurs the propagation of deep information; Res unit is the residual unit, composed of CBM and residual connection, responsible for preventing vanishing gradients or exploding gradients; CSP_X, internally composed of CBM and Res unit structure, realizes down-sampling through convolution with a step size of 2.

The network depth is mainly expanded by superimposing the CSP_X structure, which divides the feature mapping of the incoming base layer into two parts. The main path extracts

deep features by convolution, and the bypass copies the feature mapping map of the base layer and then merges them by cross-stage hierarchy. The number of network parameters is not only reduced by feature reuse, but also the gradient disappearance problem is effectively alleviated. Finally, the backbone network is down-sampled five times (steps 2, 4, 8, 16, and 32) to obtain the output of multiscale feature maps.

3.3 Cascaded spatial pyramidal pooling

The CSPP is inspired by DenseASPP [22]. The CSPP layers are densely connected to each other with the same resolution of each feature map, such as to ensure tandem connection in the channel dimension.

The cascade method is illustrated in Figure 3, where the branches of CSPP are densely connected and the convolutions with different expansion rates depend on each other, not only constituting a dense feature pyramid, but also acquiring a larger receptive field. The obtained feature maps are concatenated in the channel dimension to obtain multi-scale features and contextual information.

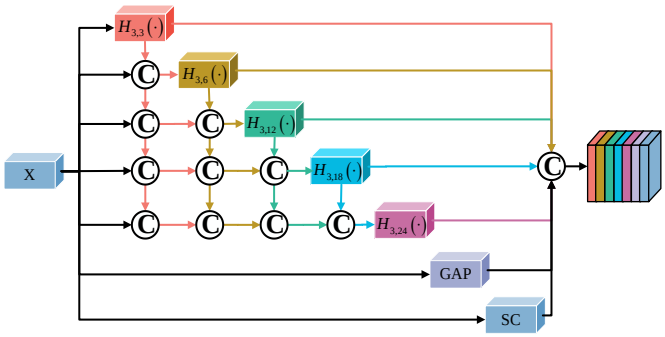


Figure 3. The structure of the CSPP

The first l DSAConv cascade representation outputs a feature map:

$$y_l = H_{k,d}([y_{l-1}, y_{l-2}, \ldots, y_0]) \tag{1}$$

The dimensionality cl of the layer l output feature map can be calculated by:

$$c_l = c_{in} + n \times (l\text{-}1) \tag{2}$$

where, $H_{k,d}(\cdot)$ is DSAConv; [···] is the cascade of all layers; c_{in} is the dimensionality of the input feature map; l is the number of layers in the bottleneck layer.

GAP and SC are short for global average pooling, and skip connections, respectively. For the feature maps obtained by GAP, bilinear interpolation and 1×1 convolution are performed to ensure that they have the same resolution as the internal feature maps of the CSPP; 1×1 convolution is implemented to ensure that the skip connect (SC) from the input to output has the same resolution as the internal feature maps of the CSPP. Next, all feature maps are concatenated in the channel dimension. Then, the output y_{cspp} of the CSPP can be expressed by:

$$y_{cspp} = [y_l, y_{l-1}, y_{l-2}, \ldots, y_0, gap(x), x] \tag{3}$$

The channel dimensionality c_{out} of the output feature map can be calculated by:

$$c_{out} = c_l + n + n = c_{in} + n \times (l+1) \tag{4}$$

where, gap(x) is global average pooling; x is the input feature map.

3.4 Multi-scale attention fusion

Contrary to the existing channel or spatial attention modules, Sun Yat-sen University developed a non-participatory attention mechanism called SimAM [19], based on neuroscience theory. Information-rich nodes usually differ in firing pattern, and inhibit surrounding nodes. Before assigning relatively high importance to the nodes with null inhibition effect, the linear differentiability between nodes can be measured by:

$$e_t^* = \frac{4(\hat{\sigma}^2 + \lambda)}{(\text{t-}\hat{\mu}^2) + 2\hat{\sigma}^2 + 2\lambda} \tag{5}$$

Formula (5) shows that, the sharper the difference between node t and a surrounding node, the greater the importance of that node. $1/e*$ t is the mean and variance of the channel excluding the target node. The hyperparameter λ is set to 0.0001.

Drawing on the SimAM [19] attention mechanism, this paper designs a simple attention up-sampling module (SAU) as shown in Figure 4. For the input low-level features, the channel dimensionality of the input feature map was set to 48 through 1×1 convolution, to reduce the proportion of low-level features and prevent the weakening of multiscale features. Thereafter, the high-level features are up-sampled to the same resolution as the low-level features through transpose convolution. The resulting 3D attention weights of the high-level feature distribution help to enhance the complementary information, and suppress redundant and interference information through dynamic weighting. Then, the adjusted high-level features are added element by element with the channel-compressed feature maps, producing fused feature maps. Finally, a feature map with channel dimensionality of 256 is obtained by 3×3 convolutional fusion, so that the number of input channels is equal to that of output channels.

Assuming that the pre-processed high-level features and low-level features are $X_h \in R^{H \times W \times Ch}$ and $X_l \in R^{H \times W \times Cl}$.

$$X_h' = MX_h + X_h \tag{6}$$

$$X_f = Cat[X_h', X_l] \tag{7}$$

where, M is the distribution weight of the high-level features obtained by SimAm processing, $X\hbar$ is the high-level features after SimAM processing, and X_f is the fused features after channel concatenation.

The low-level features with stride 2 and stride 8 are first extracted from the backbone network. Then, the high-level features with stride 32 are input to the CSPP module for extracting multi-scale high-level features. After that, SAU up-sampling is performed to aggregate high-level and low-level features step by step, and to recover the spatial and detailed information of the high-level features. Finally, the fused feature map, which is generated through bottom-up fusion of

multiple branches, is recovered to the size of the input image, via bilinear interpolation, producing the predicted semantic segmentation map. In this way, the semantic segmentation becomes more accurate, while the computing efficiency is maximized.

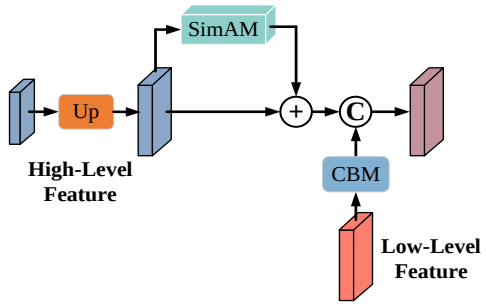


Figure 4. The up-sampling structure of SAU

4. EXPERIMENTS

This section firstly introduces the datasets, procedure, and metrics of the experiments. Next, the effectiveness of the proposed components was examined through ablation. Then, our method was compared with existing techniques, and the segmentation performance of each model was quantified to demonstrate the effectiveness of our method. In the end, the results of our method on two datasets were visualized, and analyzed qualitatively.

4.1 Datasets

Two benchmark datasets of semantic segmentation are adopted: CamVid [20] and Cityscapes [21]. The details of the datasets are reported in Table 1. For network training and testing, each dataset was split into a training set, a validation set, and a test set by the ratio given in Table 1.

Table 1. Details of the datasets

Dataset	Resolution	Classes	Train	Val	Test
CamVid	960×720	12	490	140	71
Cityscapes	2048×1024	19	2975	500	1525

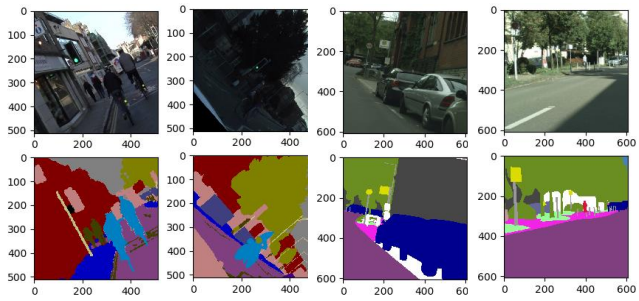


Figure 5. Pre-processing of two data sets

To enhance the effectiveness of model training, the input images and labels into the network are preprocessed simultaneously to ensure that the images and labels are corresponding to each other. First, the input images and labels are scaled randomly in the range of [0.5, 2], then rotated randomly in the range of [60, 90] degrees, and finally the output images and labels are of the required size for the network using the center crop. It is worth noting that if the crop size is larger than the size of the image and label, the blank area is filled with 0 value. The effect of the pre-processed images and labels is shown in Figure 5.

4.2 Procedure

All experiments were conducted on NVIDIA workstations within the hardware and software environment in Table 2. The CSPDarkNet, pre-trained on ImageNet, was taken as the backbone for semantic segmentation. The final fully connected and classification layers were removed from the network. Parallel training with mixed precision was implemented to speed up the training, and batch normalization was employed to enhance network generalizability.

Table 2. Hardware and software environment

Project	Details
CPU	Intel® Core™ i9-10940X @ 3.50GHz
GPU	NVIDIA GeForce RTX 3080×2
RAM	8.00 GB×2
System	Windows 10
CUDA	11.1.1
Python	3.9.0
PyTorch	1.8.1

Note: CPU, GPU, and RAM are short for central processing unit, graphics processing unit, and random-access memory, respectively.

Considering the large sample differences in each dataset, the class imbalance was solved by counting the proportion of each class in each dataset, and assigning a weight to each class by the cross-entropy loss function.

$$W(p,q) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{n}\lambda_i p(x_{ij})\log(\frac{e^{y_i}}{\sum_{i=1}^{n}e^{y_i}}) \tag{8}$$

where, λ_i denotes the weight of category j, y_i denotes the output of category j, $p(x_{ii})$ denotes the true value, n denotes the number of categories, and m denotes the small sample size.

In the early stages of training, the distance to the target is long, requiring a larger learning rate, but too large a learning rate tends to lead to instability in the training process. To avoid this problem, a warm-up phase was performed at the beginning of the training, and when the training was more stable, the learning rate was adjusted back to the initial value, and then the learning rate gradually decayed to 0. Therefore, in this paper, the Ranger optimizer [32] is used as the optimization function of the network and the learning rate is dynamically adjusted by the 'ploy' strategy.

$$LR = \begin{cases} lr \times \dfrac{epoch}{w\_epoch} & 0 < epoch < w\_epoch \\ lr \times (1 - \dfrac{epoch}{\mathrm{m}\_epoch})^{power} & w\_epoch < epoch < \mathrm{m}\_epoch \end{cases} \tag{9}$$

During the training, the power was set to 0.9, the w_epoch was set to 10, and the batch size to 4 images per GPU. For the CamVid dataset, the images and labels were randomly cropped to 512×512, lr was set to 0.02, and the m_epoch was set to 150. For the Cityscapes dataset, the images and labels were randomly cropped to 608×608, lr was set to 0.002, and the m_epoch was set to 200. Note that the models were pretrained

in ImageNet to improve the model convergence speed and segmentation effect through transfer learning.

4.3 Metrics

The performance of our method was evaluated comprehensively with the following quantitative metrics: pixel accuracy (PA), mean intersection-over-union (mIoU), number of network parameters (Params), number of floating point operations (FLOPs), and inference speed (FPS). These metrics are commonly used to evaluate the effect of semantic segmentation.

The mIoU is the average of the intersection-over-union between the predicted value and true value of each class. Params and FLOPs are positively correlated with model complexity, and hardware requirements. FPS is the inference speed of the model. The greater the FPS, the more efficient the model inference. These metrics can be respectively calculated by:

$$mIoU = \frac{1}{k+1}\sum_{i=0}^{k}\frac{p_{ii}}{\sum_{j=0}^{k} p_{ij} + \sum_{j=0}^{k} p_{ji} - p_{ii}} \tag{10}$$

$$Params = \left(k_h \times k_w \times C_{in} + 1\right) \times C_{out} \tag{11}$$

$$FLOPs = H' \times W' \times Params \tag{12}$$

$$FPS = n/(t_{end} - t_{start}) \tag{13}$$

where, k+1 is the number of classes; P_{ij} is the number of class j pixels predicted as class i pixels; P_{ii} is the number of correctly predicted pixels; P_{ji} is the number of class i pixels predicted as class j pixels; $H \times W \times C_{in}$ is the input feature map; $k_h \times k_w$ is the convolution kernel; $H \times W \times C_{out}$ is the output feature map; t_{strat} and t_{end} are the start and end signs of inference, respectively; n is the number of test images.

4.4 Ablation analysis

The ablation experiments focus on the effects of three modules, namely, the backbone network, the CSPP, and the MAF, on the proposed network, as well as the mechanism of each module. The segmentation accuracy and detection efficiency of each model were quantified by Params, FLOPs, mIoU, and FPS.

4.4.1 Ablation experiment on backbone network

Several popular deep neural networks were tested. The fully connected layer and classification layer were removed from each network. The feature maps were obtained through 8 steps of down-sampling, and taken as low-level features to input into the decoder of DeepLabV3+. The size of the input image was set to 512×512, and the batch size to 8.

As shown in Table 3, ResNet and Xception as the backbone network led to relatively high mIoUs. But the two networks involve many more parameters than CSPDarkNet. MobileNet and DenseNet require relatively few parameters. However, their segmentation accuracy was not as good as that of CSPDarkNet. To sum up, using CSPDarkNet as the feature extraction network strikes a balance between real-time performance and segmentation accuracy, fully utilizes the system resources for multiscale feature fusion, and achieves the best overall performance.

Table 3. Network performance with different backbones

Backbone	Params (M)	FLOPs (G)
Xception [33]	54.71	82.76
ResNet [34]	59.36	50.05
CSPDarkNet [18]	40.06	33.33
DensNet [35]	16.21	22.25
MobileNet [36]	5.82	40.21

4.4.2 Ablation experiment on CSPP module

Taking ASPP as the baseline, the ablation experiment on the CSPP compares the difference between the ASPP with dilation convolution being replaced by depth-separable convolution, and that with the CSPP of each dilation convolution branch. As shown in Table 4, from group 1 to group 2, Params and mIoU were reduced by 15.58% and 0.20%, respectively. This means replacing atrous convolution with depthwise separable atrous convolution can maintain a high detection accuracy with a lightweight model. From group 2 to group 3, the Params and mIoU were improved by were 1.94% and 0.65%, respectively. Thus, the hybrid receptive field fusion can improve the model segmentation performance, at a small cost of computing overhead and video memory usage.

Table 4. Comparison between different fusion mechanisms with ASPP as the baseline

Baseline	Params (M)	FLOPs (G)	mIoU (%)
ASPP	40.06	33.33	71.34
CSPP (DSAConv)	33.82	31.72	71.14
CSPP (DSAConv+MRFF)	34.49	31.90	71.79

Note: CSPDarkNet network is taken as the backbone network of DeepLabV3+; DSAConv is deep separable convolution; MRFF is cascaded receptive field fusion.

A comprehensive analysis of Table 4 shows that the improved CSPP outperformed the original ASPP by 0.45%, and reduced the number of parameters by 16.15% and the number of operations by 4.5%. This is because the correlation of features between distant convolutions is improved by the dense connections between each expanded convolutional branch and improved the representation of multi-scale features.

Furthermore, two control experiments were carried out on the CamVid dataset with the CSPP as the baseline, trying to disclose how the combination between the number of branches of the expanding convolution and the dilation rate influences the segmentation performance. The experimental results are shown in Table 5. Control groups 2 and 3 were obtained by fixing the number of number of branches of the expanding convolution, and changing the dilation rate. It can be seen that the params and FLOPs of the network remained constant. The segmentation performance could be improved, if the receptive field is enhanced by increasing the dilation rate of the expanding convolution within a certain range.

Table 5. Effect of different combinations of expansion rates on model segmentation performance

Dilation rate	Params (M)	FLOPs (G)	mIoU (%)
CSPP (6, 12, 18)	34.49	31.90	71.79
CSPP (4, 8, 12, 16)	35.10	32.05	72.32
CSPP (6, 12, 18, 24)	35.10	32.05	72.83

4.4.3 Ablation experiment on MAF module

The effectiveness of the MAF was verified by importing the low-level features S_x obtained by x stride down-sampling to the decoder. Then, the input multiscale features were aggregated bottom-up by the SAU module. Taking the CSPDarkNet as the backbone network, the authors set the dilation rate combinations of the CSPP to (6, 12, 18, 24), while fixing the input image size to 512×512 and the batch size to 4. The experimental results are shown in Table 6.

Table 6. Effect of different scale feature inputs on model segmentation performance

Multi-scale feature input	Params (M)	FLOPs (G)	mIoU (%)
[S_8]	33.00	32.05	72.83
[S_2, S_8]	35.24	110.34	74.72
[S_2, S_4, S_8]	35.39	112.79	74.77
[S_2, S_8, S_{16}]	34.37	48.41	74.06

In groups 1-3, the mIoU value increased with the resolution of the low-level features S_x in the input network. The reason is that the high-level features are rich in semantic information, but the ability to express detailed information is relatively weak. The fusion between high-level and low-level features diversifies feature mapping, facilitating the segmentation of small target objects and boundaries. The mIoU of group 4 was 0.635 smaller than that of group 3, but the two groups varied significantly in FLOPs. Despite the constant number of low-level feature maps in the input model, the large resolution of these maps leads to a sharp growth in the memory overhead of the model.

4.5 Comparison with the state-of-the-art

The proposed network was tested on the CamVid dataset to quantify the impact of different modules on the computing cost and segmentation performance, using various metrics.

4.5.1 Comparison of computing cost

The computing cost of our network was compared with the state-of-the-art, where the input network has the same batch size and image resolution. The test results are displayed in Table 7.

Compared with the baseline, the modified backbone network reduced the number of parameters and operations of the model by 32.51% and 33.40%, respectively, and its mIoU dropped by 1.57%. This is attributable to the fact that the lightweight CSPDrakNet as the backbone network greatly reduces the number of parameters and computing load. The hardware requirement on the computer is reduced at the cost of some detection accuracy. Note that layers like the activation function are ignored, and the measured values were slightly smaller than the actual ones.

Table 7. Comparison of computing costs

Baseline	Params (M)	FLOPs (G)	mIoU (%)
ResNet	59.36	50.05	72.80
CSPDarkNet	40.06	33.33	71.23
CSPDarkNet+CSPP	35.10	32.05	72.83
CSPDarkNet+CSPP+MAF	35.57	112.97	74.82

The data on groups 2 and 3 show that adding the CSPP module to the network reduced the model parameters and memory consumption by 12.38% and 0.03%, respectively, while increasing the mIoU by 1.60%. The reason is that the branches of the CSPP module support feature multiplexing, and widen the receptive field through dense connection. This strengthens the information exchange between branches, exerting a positive effect on the model accuracy.

The data on groups 3 and 4 show that the mIoU was improved by 2.21%, revealing the considerable impact of multi-scale attention fusion on model performance. The aggregation between high-level and low-level features increases the computing overhead and memory consumption simultaneously, dragging down the detection efficiency by 0.75 fps.

In summary, the results in Table 7 suggest that all three proposed modules can improve the overall performance, especially through the optimization of the backbone network optimization and the fusion between modules.

4.5.2 Comparison of inference time

Figure 6 compares the mIoU and inference speed of the four networks on the CamVid test set. The inference was performed on a NVIDIA GeForce RTX 3080 with a batch size of 1 and an input image resolution of 512×512. It can be observed that, compared with the baseline, CSPDarkNet as the backbone network sacrificed some detection accuracy for the fastest inference speed. The addition of the CSPP brought a comparable segmentation performance with the baseline, while significantly speeding up the inference. The introduction of the MAF substantially improved the detection accuracy, despite a slight reduction of the inference speed.

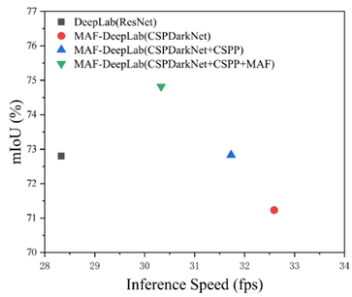


Figure 6. Comparison of inference mIoU and inference speed on the CamVid test set

4.5.3 Comparison of splitting performance

Figure 7 shows the confusion matrix of method on the CamVid test set. The confusion matrix reflects the relationship between the predicted labels and the true labels. If there are many elements on the diagonal, then the segmentation is highly effective.

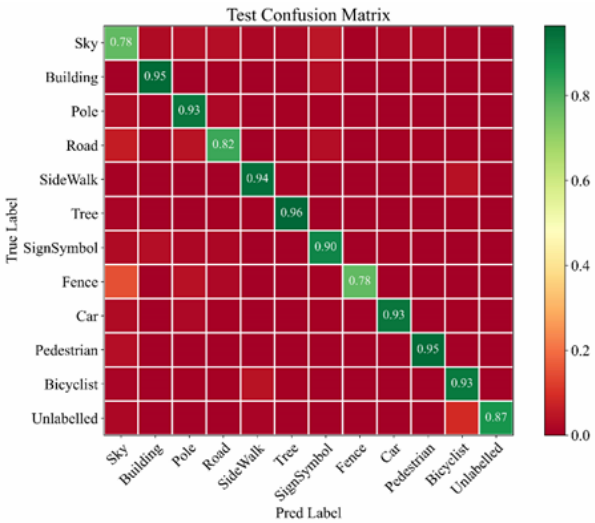
As can be seen from Figure 7, almost all the classes concentrated on the diagonal, especially building and pole. This fully manifests the effectiveness of our network. It can also be learned that the model segmented several classes rather poorly, namely, fence, road, and bicyclist. Our network has difficulty in distinguishing between these similar classes.

Our network was further contrasted with the state-of-the-a on the Cityscapes dataset. It can be clearly inferred from Table 8 that MAF-DeepLab achieved the best segmentation performance with an mIoU of 74.8%, a 3.4% improvement compared to DeepLabV3+, and realized the best performance in 13 out of the 19 classes included in the dataset.

Table 8. Performance of our network and the state-of-the-art on the Cityscape test set

Classes	SegNet [14]	EDANet [37]	RefineNet [17]	PSPNet [11]	DeepLabV3+ [16]	Ours
Road	96.4	98.6	98.2	98.3	**98.7**	98.3
Sidewalk	73.2	86.1	83.3	86.9	87.0	**88.4**
Building	84.0	93.5	91.3	93.5	**93.9**	93.3
Wall	28.5	56.2	47.8	58.4	**61.5**	59.1
Fence	29.0	63.3	50.4	63.7	63.9	**64.5**
Pole	35.7	69.7	56.1	67.7	72.4	**73.5**
T-Light	39.8	77.3	66.9	76.1	78.2	**79.6**
T-Sign	45.2	81.3	71.3	80.5	82.2	**83.5**
Vegetation	87.0	93.9	92.3	93.6	93.0	**94.9**
Terrain	63.8	72.9	70.3	72.2	**73.0**	72.8
Sky	91.8	95.7	94.8	95.3	92.0	**96.3**
Person	62.8	87.3	80.9	86.8	87.0	**90.5**
Rider	42.8	72.9	63.3	71.9	73.3	**74.5**
Car	89.3	96.1	94.5	**96.2**	92.4	96.1
Truck	38.1	76.8	64.6	77.7	78.0	**79.2**
Bus	43.1	89.5	76.1	**91.5**	90.9	91.2
Train	44.2	86.5	64.3	83.6	88.9	**91.7**
Motorcycle	35.8	72.2	62.0	70.8	73.8	**75.6**
Bicycle	51.9	78.2	70.0	77.5	78.9	**82.1**
mIoU	57.0	81.5	73.6	81.2	82.1	**83.4**

Note: The mIoU and mean mIoU of each class are presented; the best performance in each class is in bold.



Note: The mIoU and mean mIoU of each class are presented; the best performance in each class is in bold.

Figure 7. Confusion matrix on the CamVid test set

4.6 Visual analysis

Our network was tested on two benchmark datasets of semantic segmentation, namely, CamVid and Cityscapes. To qualitatively analyze model performance, the relatively complex scenarios were selected to visualize the segmentation results.

4.6.1 Results on the CamVid dataset

Figure 8 visualizes the results of our network on the CamVid dataset. It can be seen that the class boundaries were clearly predicted, and the spatial details were processed more securely than those in DeepLabV3+. This is consistent with the results in Figure 6. However, the segmentation effect needs to be further improved on confusing classes like fence, sidewalk, and road.

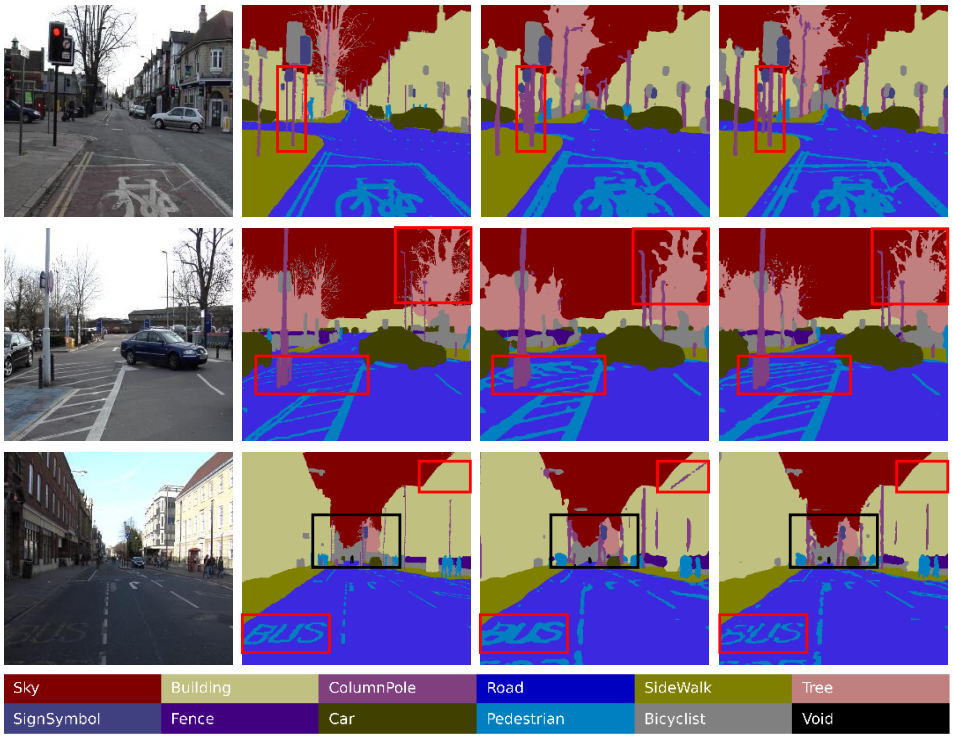
4.6.2 Results on the Cityscapes dataset

Figure 9 visualizes the results of the original and improved DeepLabV3+ on the Cityscapes validation set. It can be learned that our network correctly segmented most objects, except some minor details. The segmentation results of our network were the closest to the true labels. In particular, the confusing classes of fences, traffic signs, and signals were missing or ambiguous in DeepLabV3+. However, our network failed to classify similar-looking objects like sidewalks and roads, leaving a room for improvement.

5. CONCLUSIONS

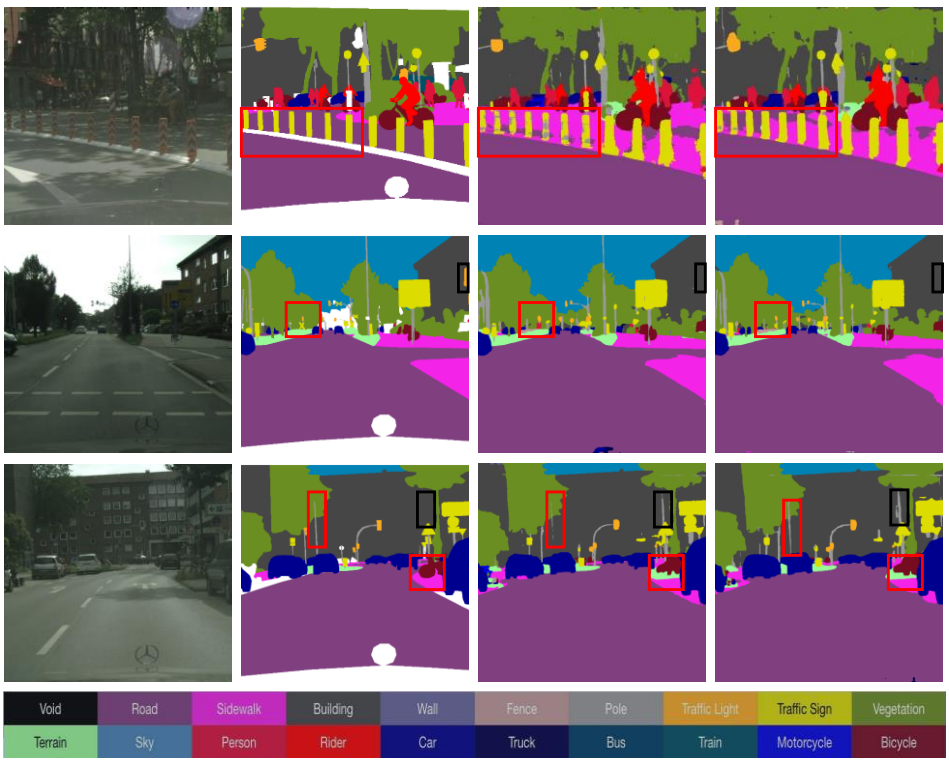
This paper proposes a novel encoder-decoder network called MAF-DeepLab, which can effectively aggregate contextual information, and fuse features at different scales. Unlike the latest encoder-decoder networks, our network adopts a lightweight backbone to enhance its ability to capture multiscale features, and improve its operating efficiency. The global average pooling was combined with densely connected depthwise separable atrous convolution into the CSPP, which substantially improve the handling of multiscale features, and strengthen the interconnectedness of acquired features. Besides, the attention fusion module was introduced to improve the focus on the salient features of the feature map, and facilitate the fusion between deep and shallow features. Then, experiments were carried out on public datasets like CamVid and Cityscapes. Through quantitative and qualitative analyses, it was found that the proposed MAF-DeepLab can effectively prevent incorrect and incomplete segmentation, and achieve an excellent segmentation effect of multi-scale objects.

Despite the lightweight design of the backbone network, the efficiency of our network is partly sacrificed, as low-level and high-level features are fused repeatedly to enhance the segmentation ability of multi-scale objects along with boundary details. In addition, the proposed network does not perform satisfactorily on confusing classes of objects. The future work will try to improve the representation of confusing class features based on lightweight network structures.



Note: From left to right are the original image, the true label, the label predicted by DeepLabV3+, and the label predicted by MAF-DeepLab. The label of each class is given a specific color. The red boxes highlight the best performance.

Figure 8. Visualization results of on the CamVid test set



Note: From left to right are the original image, the true label, the label predicted by DeepLabV3+, and the label predicted by MAF-DeepLab. The label of each class is given a specific color. The red boxes highlight the best performance.

Figure 9. Visualization results of on the Cityscapes validation set

ACKNOWLEDGMENT

This work was supported by Fujian Provincial Natural Science Foundation (Grant No.: 2021J01851) and National Natural Science Foundation Incubation Program, Jimei University (Grant No.: ZP2020045).

REFERENCES

[1] Minaee, S., Boykov, Y.Y., Porikli, F., Plaza, A.J., Kehtarnavaz, N., Terzopoulos, D. (2021). Image segmentation using deep learning: A survey. IEEE Transactions on Pattern Analysis and Machine

Intelligence. https://doi.org/10.48550/arXiv.2001.05566
[2] Hao, S., Zhou, Y., Guo, Y. (2020). A brief survey on semantic segmentation with deep learning. Neurocomputing, 406: 302-321. https://doi.org/10.1016/j.neucom.2019.11.118
[3] Li, X., Ma, H., Yi, S., Chen, Y., Ma, H. (2021). Single annotated pixel based weakly supervised semantic segmentation under driving scenes. Pattern Recognition, 116: 107979. https://doi.org/10.1016/j.patcog.2021.107979
[4] Sun, J., Li, Y. (2021). Multi-feature fusion network for road scene semantic segmentation. Computers & Electrical Engineering, 92(12): 107155. https://doi.org/10.1016/j.compeleceng.2021.107155
[5] Zhou, Q., Wang, Y., Fan, Y., Wu, X., Zhang, S., Kang, B., Latecki, L.J. (2020). AGLNet: Towards real-time semantic segmentation of self-driving images via attention-guided lightweight network. Applied Soft Computing, 96: 106682. https://doi.org/10.1016/j.compeleceng.2021.107155
[6] Elaraby, A., Elansary, I. (2021). A framework for multi-threshold image segmentation of low contrast medical images. Traitement du Signal, 38(2): 309-314. https://doi.org/10.18280/ts.380207
[7] Wang, X., Li, Z., Huang, Y., Jiao, Y. (2022). Multimodal medical image segmentation using multi-scale context-aware network. Neurocomputing, 486: 135-146. https://doi.org/10.1016/j.neucom.2021.11.017
[8] Yu, H., Yang, Z., Tan, L., Wang, Y., Sun, W., Sun, M., Tang, Y. (2018). Methods and datasets on semantic segmentation: A review. Neurocomputing, 304: 82-103. https://doi.org/10.1016/j.neucom.2018.03.037
[9] 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, pp. 3431-3440. https://doi.org/10.1109/CVPR.2015.7298965
[10] Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L. (2014). Semantic image segmentation with deep convolutional nets and fully connected CRFs. arXiv preprint arXiv:1412.7062. https://doi.org/10.48550/arXiv.1412.7062
[11] 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, pp. 2881-2890. https://doi.org/10.1109/CVPR.2017.660
[12] Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L. (2017). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Transactions on Pattern Analysis and Machine Intelligence, 40(4): 834-848. https://doi.org/10.1109/TPAMI.2017.2699184
[13] Chen, L.C., Papandreou, G., Schroff, F., Adam, H. (2017). Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587. https://doi.org/10.48550/arXiv.1706.05587
[14] Badrinarayanan, V., Kendall, A., Cipolla, R. (2017). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(12): 2481-2495. https://doi.org/10.1109/TPAMI.2016.2644615
[15] 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. https://doi.org/10.48550/arXiv.1505.04597
[16] 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. 801-818. https://doi.org/10.48550/arXiv.1802.02611
[17] Lin, G., Milan, A., Shen, C., Reid, I. (2017). Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1925-1934. https://doi.org/10.1109/CVPR.2017.549
[18] Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M. (2020). Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934. https://doi.org/10.48550/arXiv.2004.10934
[19] Yang, L., Zhang, R.Y., Li, L., Xie, X. (2021). Simam: A simple, parameter-free attention module for convolutional neural networks. In International Conference on Machine Learning, pp. 11863-11874.
[20] Brostow, G.J., Fauqueur, J., Cipolla, R. (2009). Semantic object classes in video: A high-definition ground truth database. Pattern Recognition Letters, 30(2): 88-97. https://doi.org/10.1016/j.patrec.2008.04.005
[21] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3213-3223. https://doi.org/10.1109/CVPR.2016.350
[22] Yang, M., Yu, K., Zhang, C., Li, Z., Yang, K. (2018). Denseaspp for semantic segmentation in street scenes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3684-3692. https://doi.org/10.1109/CVPR.2018.00388
[23] Sang, H., Zhou, Q., Zhao, Y. (2020). Pcanet: Pyramid convolutional attention network for semantic segmentation. Image and Vision Computing, 103: 103997. https://doi.org/10.1016/j.imavis.2020.103997
[24] Lian, X., Pang, Y., Han, J., Pan, J. (2021). Cascaded hierarchical atrous spatial pyramid pooling module for semantic segmentation. Pattern Recognition, 110: 107622. https://doi.org/10.1016/j.patcog.2020.107622
[25] Zhou, Q., Wu, X., Zhang, S., Kang, B., Ge, Z., Latecki, L.J. (2022). Contextual ensemble network for semantic segmentation. Pattern Recognition, 122: 108290. https://doi.org/10.1016/j.patcog.2021.108290
[26] Oršić, M., Šegvić, S. (2021). Efficient semantic segmentation with pyramidal fusion. Pattern Recognition, 110: 107611. https://doi.org/10.1016/j.patcog.2020.107611
[27] 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/CVF Conference on Computer Vision and Pattern Recognition, pp. 3146-3154. https://doi.org/10.1109/CVPR.2019.00326
[28] 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/CVF International Conference on Computer Vision, pp. 603-

612. https://doi.org/10.1109/ICCV.2019.00069
[29] Peng, C., Ma, J. (2020). Semantic segmentation using stride spatial pyramid pooling and dual attention decoder. Pattern Recognition, 107: 107498. https://doi.org/10.1016/j.patcog.2020.107498
[30] Zhou, Z., Zhou, Y., Wang, D., Mu, J., Zhou, H. (2021). Self-attention feature fusion network for semantic segmentation. Neurocomputing, 453: 50-59. https://doi.org/10.1016/j.neucom.2021.04.106
[31] Misra, D. (2019). Mish: A self regularized non-monotonic activation function. arXiv preprint arXiv:1908.08681.
[32] Zhang, M., Lucas, J., Ba, J., Hinton, G.E. (2019). Lookahead optimizer: K steps forward, 1 step back. Advances in Neural Information Processing Systems, 32.
[33] Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1251-1258. https://doi.org/10.1109/CVPR.2017.195
[34] He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770-778. https://doi.org/10.1109/CVPR.2016.90
[35] Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q. (2017). Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700-4708. https://doi.org/10.1109/CVPR.2017.243
[36] Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861. https://doi.org/10.48550/arXiv.1704.04861
[37] Lo, S.Y., Hang, H.M., Chan, S.W., Lin, J.J. (2019). Efficient dense modules of asymmetric convolution for real-time semantic segmentation. In Proceedings of the ACM Multimedia Asia, pp. 1-6. https://doi.org/10.48550/arXiv.1809.06323