Lawn Image Semantic Segmentation Method Based on the Improved DeepLabV3+

Caixiang Zhang[1,a*], Mingxuan Zhang[1,b], Qihang Guan[1,c] , Shuping Xu[1,d]

School of Computer Science and Engineering Xi'an Technological University Xi'an, China

[a]848512725@qq.com, [b]3572213105@qq.com, [c]1109835535@qq.com, [d]563937848@qq.com

Abstract: With the in-depth advancement of urbanization in China, the urban green area has gradually increased. The demand for its maintenance and management is also constantly rising. Lawn mowing, as a daily task with high repetition and high labor intensity, urgently needs an intelligent lawn mowing device to replace manual labor. Semantic segmentation technology, as the key technical basis of the perception system of intelligent lawn mowing robots, can accurately identify and distinguish different targets in the scene, and accurately define the boundaries of the operation area at the same time. However, traditional image semantic segmentation methods have problems such as a large number of model parameters, slow reasoning speed, and limited segmentation accuracy, which are difficult to meet the requirements of real-time operations. In response to the above problems, based on the DeepLabV3+ deep learning framework, this study proposes an improved segmentation model. By proposing the serial hollow space pyramid pooling and feature fusion module, the performance of the model is enhanced. And in view of the current situation where there is a lack of public lawn scene segmentation datasets, a dedicated segmentation dataset is constructed for the lawn scene.

Keywords: Machine Vision; Image Semantic Segmentation; DeepLabV3+; Multi-scale Feature Integration; Attention Mechanism

1. Introduction

In recent years, with the rapid development of artificial intelligence (AI) and deep learning technologies, computer vision—an important field—has witnessed profound advancements in its related technologies. Image classification, object detection, and image semantic segmentation are currently three hot research directions in computer vision [1].

As a crucial research direction in the field of computer vision [2], image semantic segmentation is an important technology for understanding information in practical application scenarios [3]. It holds broad application prospects in engineering fields such as intelligent assisted driving [4], industrial robots [5], and medical imaging [6]. At present, image semantic segmentation technologies are mainly divided into two development directions [7-8].The first direction consists of traditional semantic segmentation methods. These methods primarily rely on shallow feature information of images and perform segmentation by extracting features such as color, texture, and geometry from images. For example: Yang Yun et al. initialized image object categories using the bisection principle and block similarity criteria, and proposed a fuzzy threshold remote sensing image segmentation algorithm based on local spatial features [9];Pang Mingming et al. integrated fuzzy mathematics theory into the Canny edge detection algorithm to enhance the ability to recognize local information, thereby improving the algorithm's performance in identifying object

https://doi.org/10.58244/joerai.263889

edges [10];Hu Gaozhen et al. established local feature information using the similarity of local image regions, extracted object edge information via the Canny operator, and fused these features through a Markov model to solve the problem of edge blurring in image segmentation [11];Yang Meng et al. proposed a fuzzy divergence multi-threshold image segmentation algorithm based on the standard deviation method, extending the single-threshold membership function to a multi-threshold form [12];Sun Yang et al. comprehensively utilized the local regional features and object edge features of underwater images, and proposed an edge feature extraction method suitable for underwater scenarios [13].The second direction involves deep learning-based image semantic segmentation algorithms. For example: Long et al. improved the traditional convolutional neural network (CNN) by replacing its fully connected layers with convolutional layers, proposed the Fully Convolutional Networks (FCN), and used deconvolution for image up sampling to obtain final results [14-15];Wei Guo et al. made full use of the low-level and high-level semantic information of images through short connections, and improved the segmentation performance of the network in specific tasks via transfer learning [16];Wang Haiou et al. proposed an improved U-Net medical image segmentation network, which used a filter model to filter external noise and added a normalization layer to the network structure to enhance parameter sensitivity [17];Chen et al. proposed the DeepLabV1 image semantic segmentation model. To address the feature loss issue of FCNs during feature extraction, they replaced traditional convolutions with dilated convolutions to expand the receptive field and improve image segmentation performance [18];Wang et al. proposed the SegNet semantic segmentation network, which used an encoder-decoder structure to optimize the network architecture, reduce network parameters, and fully leverage shallow semantic information to enhance network performance [19];Chen et al. proposed the DeepLabV3+ image semantic segmentation network and constructed a spatial pyramid module. This module connects three dilated convolutions with different dilation rates and one pooling layer in parallel to extract semantic information of different dimensions from the input image [20-21].

In existing research on image semantic segmentation, the segmentation effect of traditional methods is more susceptible to interference from external noise such as illumination and grayscale values, and thus can no longer meet the application requirements of complex practical scenarios. Compared with traditional methods, deep learning-based image semantic segmentation networks use convolutional neural networks (CNNs) to better acquire image feature information and improve the performance of segmentation networks. However, during the feature extraction stage, problems such as loss of detailed feature information, reduced pixel correlation, mis segmentation at image edges, and discontinuous segmentation boundaries still exist.

2. Improved Deeplabv3+ Network

2.1 Improved Network Structure

DeepLabV3+ is a classic semantic segmentation network proposed by Google. In the encoder stage, the backbone feature extraction network performs feature extraction and divides the extracted features into low-level features and high-level features. The low-level features are directly input to the decoder, while the high-level features serve as input to the Atrous Spatial Pyramid Pooling (ASPP) module, which conducts multi-scale feature fusion on the input features. In the decoder stage of the network, the high-level features output by the ASPP module undergo bilinear interpolation upsampling, then are fused with the low-level features. After further upsampling, a prediction image with the same size as the input image is obtained.

The original DeepLabV3+ network uses the Xception network as its backbone feature extraction network. However, Xception has a large number of parameters, a complex structure, and slow

prediction speed, making it difficult to meet the requirements of real-time recognition. Additionally, the ASPP module has a drawback: increasing the dilation rate of dilated convolutions leads to a decline in network performance.

To address the above issues, this study makes the following improvements to the original network: Replace the backbone feature extraction network with the lightweight MobileNetV2 network to reduce the number of network parameters and improve the network's prediction speed; Optimize the original ASPP module and propose a Serial Atrous Spatial Pyramid Pooling (S-ASPP) module. Specifically, the dilated convolutions with dilation rates of 6, 12, and 18 (which worked in parallel in the original module) are replaced with three serially connected dilated convolutions each with a dilation rate of 6, accompanied by forward short connections; Add a Dual Channel Attention Mechanism (DCAM) module after the low-level features output by the backbone feature extraction network and the high-level features output by the S-ASPP module. The structure of the improved DeepLabV3+ network is shown in Figure 1.

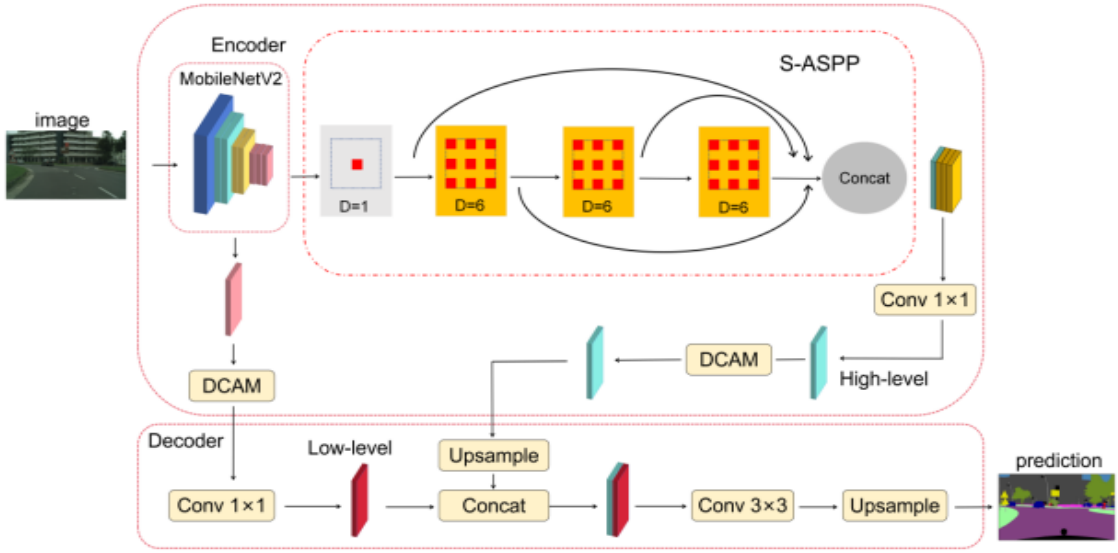

Figure 1. Improvement of the Overall Structure of the DeepLabV3+ Network

2.2 Replacing the Backbone Feature Extraction Network

The original DeepLabV3+ network adopts the Xception network as its backbone feature extraction network. However, Xception has a large number of parameters and a complex structure, which leads to long network prediction and training times and makes it unable to meet the application requirements in complex real-world scenarios. Therefore, the lightweight convolutional neural network MobileNetV2 [22] is used as the backbone feature extraction network. The network structure parameters are shown in Table 1, where: t represents the channel expansion factor; c represents the number of output channels; n represents the number of repetitions; s represents the convolution kernel stride.

An important improvement of the MobileNetV2 network is the introduction of the inverted residual structure, which consists of three parts: channel expansion, feature extraction, and channel compression. Its structure is exactly opposite to the residual structure proposed by the ResNet network [23], and it uses 3×3 depthwise separable convolution kernels to replace traditional 3×3 convolution kernels. This can greatly simplify the network structure, reduce the computational complexity of the network, and improve the robustness and expressive ability of the network. The inverted residual structure is shown in Figure 2.

Table 1 MobileNetV2 Network Structure

Input	Operator	t	c	n	s
$224^2 \times 3$	Conv2d 3×3	-	32	1	2
$112^2 \times 32$	Inverted Residual	1	16	1	1
$112^2 \times 16$	Inverted Residual	6	24	2	2
$56^2 \times 24$	Inverted Residual	6	32	3	2
$28^2 \times 32$	Inverted Residual	6	64	4	2
$14^2 \times 64$	Inverted Residual	6	96	3	1
$14^2 \times 96$	Inverted Residual	6	160	3	2
$7^2 \times 160$	Inverted Residual	6	320	1	1
$7^2 \times 320$	1×1	-	1280	1	1
$7^2 \times 1280$	Avgpool	-	-	1	-
1×1×1280	Conv2d 1×1	-	K	-	-

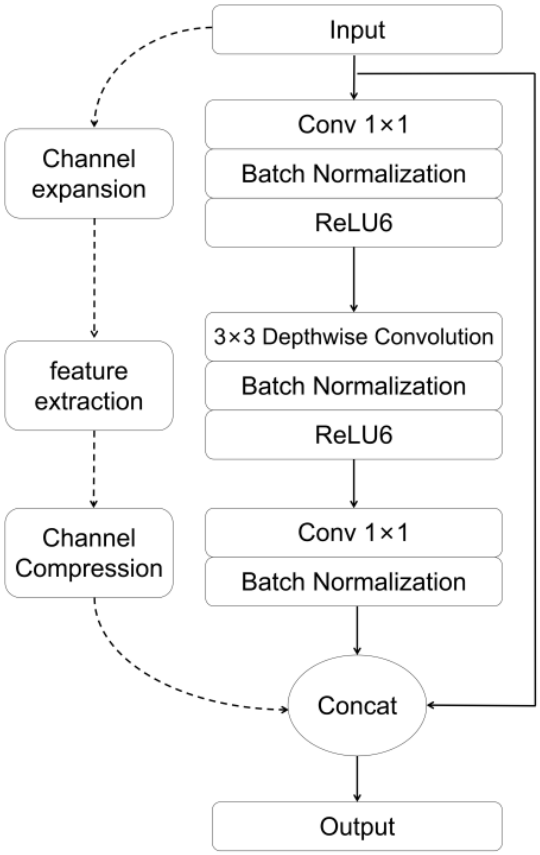

Figure 2. Inverted Residual Structure

2.3 Improved Atrous Spatial Pyramid Pooling

To acquire multi-scale feature information of input images and enhance network segmentation performance, the DeepLabV3+ network incorporates the Atrous Spatial Pyramid Pooling (ASPP) module for multi-scale feature fusion of input feature maps. This structure comprises three parallel 3×3 atrous convolutions with dilation rates of 6, 12, and 18 respectively. However, experimental results indicate that the value of the dilation rate of the atrous convolutions in this module affects network performance. Specifically, increasing the dilation rate of atrous convolutions can improve network performance—by expanding the receptive field, the network can capture more feature

information. Nevertheless, a larger dilation rate simultaneously reduces pixel correlation, which in turn degrades network performance.

To address this issue, this study proposes a Serial Atrous Spatial Pyramid Pooling (S-ASPP) module based on the original ASPP module. In this improved module, the three parallel atrous convolutions (with dilation rates of 6, 12, and 18) in the original ASPP are replaced by three serially connected atrous convolutions each with a dilation rate of 6, supplemented by forward shortcut connections. The structure of the S-ASPP module is illustrated in Figure 3.

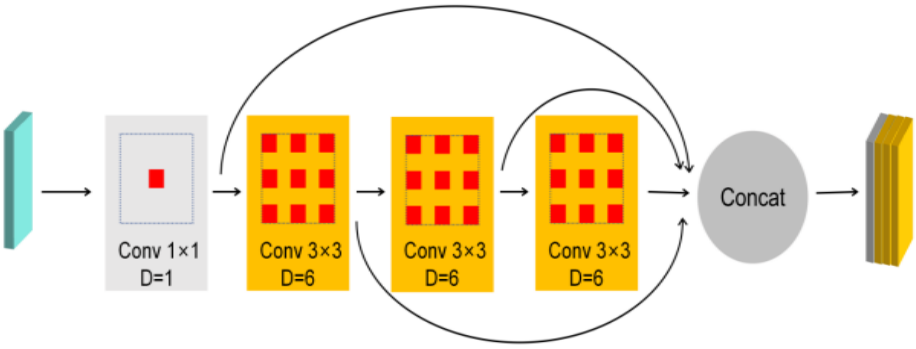

Figure 3. Structure of the S-ASPP Module

Below is the derivation of the equivalence between the serial and parallel connections of atrous convolutions. Assume that the weight of a 3×3 convolution kernel is 1/9, and the input image is denoted as Pi,j, where i and j represent the coordinates of any pixel in the input image. After this pixel undergoes a 3×3 atrous convolution with a dilation rate of 6, the resulting formula is obtained as follows:

$$9P_6^{'}=[P(i,j)+P(i+6,j)+P(i-6,j)+P(i,j-6)+P(i,j+6)$$

$$+P(i+6,j-6)+P(i+6,j+6)+P(i-6,j-6)+P(i-6,j+6)]$$

Similarly, suppose that the pixel values within the range of 6 pixels around any pixel (i,j) in the input image are equal. When the pixel (i, j) sequentially passes through two 3×3 atrous convolutions each with a dilation rate of 6, the formula is derived as:

$$9P66''(i,j)=P(i,j)+P(i+12,j)+P(i-12,j)+P(i,j-12)+P(i,j+12)+$$
$$P(i+12,j-12)+P(i+12,j+12)+P(i-12,j-12)+P(i-12,j+12)$$

In a similar manner, when a 3×3 atrous convolution with a dilation rate of 12 acts on any pixel (i,j) in the input image, the resulting formula is:

$$9P12'(i,j)=P(i,j)+P(i+12,j)+P(i-12,j)+P(i,j-12)+P(i,j+12)+P(i+12,j-12)$$
$$+P(i+12,j+12)+P(i-12,j-12)+P(i-12,j+12)$$

By comparing the calculation results of P66″(i,j) and P12′(i,j), the following formula is obtained:

$$P66''(i,j)=P12'(i,j)$$

The above formula can prove that the result obtained by an input image sequentially passing through two 3×3 atrous convolutions (both with a dilation rate of 6) is approximately equal to the result obtained by the same input image passing through a single 3×3 atrous convolution with a dilation rate of 12, as shown in Figure 4(a). Similarly, it can be concluded that the result of an input image sequentially undergoing three 3×3 atrous convolutions (each with a dilation rate of 6) is approximately the same as the result of the image passing through one 3×3 atrous convolution with a dilation rate of 18, as illustrated in Figure 4(b).

Based on the above proof results, it can be inferred that the effect produced by the parallel connection of three 3×3 atrous convolutions (with dilation rates of 6, 12, and 18 respectively) in the

ASPP module is equivalent to the effect produced by the serial connection of three 3×3 atrous convolutions (each with a dilation rate of 6 and supplemented with forward shortcut connections) in the S-ASPP module, as shown in Figure 4(c).

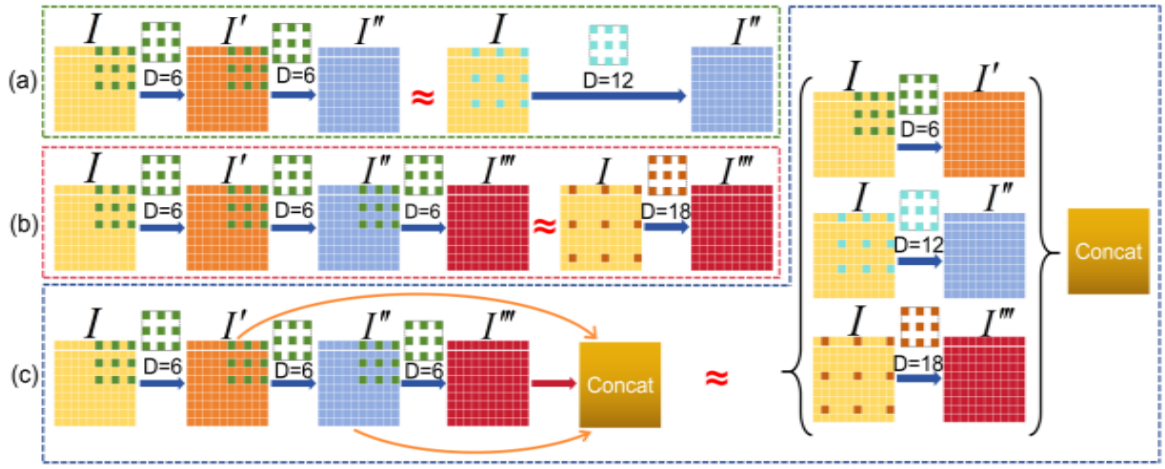

Figure 4. Schematic Diagram of the Relationship Between Multiple Atrous Convolutions Under Parallel and Serial Connections

The receptive field refers to the size of the corresponding convolution region in the previous layer for a specific pixel in the output generated by a convolution operation. Without changing the size of the input image, a larger receptive field can be obtained by increasing the dilation rate of the atrous convolution. This enables the extraction of more detailed global features and enhances network performance. The calculation formula for the receptive field is as follows:

$$R = D \times K - D + 1 \quad (1)$$

In the above formula: R denotes the size of the receptive field ; D represents the dilation rate of the convolution kernel; K stands for the size of the convolution kernel.

In the original ASPP module, all atrous convolutions operate in parallel. Therefore, the receptive field generated by this module is determined by the atrous convolution with the largest dilation rate. Taking the ASPP module with dilation rates of (6, 12, 18) as an example, the maximum receptive field is calculated as:

$$R_{max} = max\{RK = 3, D = 6, RK = 3, D = 12, RK = 3, D = 18\} = RK = 3, D = 18 = 37 \quad (2)$$

In practical operations, a larger receptive field can be achieved by serially connecting multiple atrous convolutions. The method for calculating the receptive field generated by multiple serially connected atrous convolutions is:

$$R_{max} = \sum_{n=1}^{N} R_n - (N - 1) \quad (3)$$

For the S-ASPP module, which consists of three serially connected atrous convolutions (each with a dilation rate of 6), the size of the receptive field it generates is:

$$R_{max}=RK=3,D=6+RK=3,D=6+RK=3,D=6\text{-}2=37 \quad (4)$$

From the calculation results of the above formulas, it can be concluded that the S-ASPP module achieves the same receptive field size as the original ASPP module by serially connecting three atrous convolutions with a dilation rate of 6. Additionally, since the atrous convolutions in the S-ASPP module are connected in series (where the output of the previous layer serves as the input of the next layer), the module can better capture the contextual information of input features. This

addresses the issue of network performance degradation caused by increasing the dilation rate in the original ASPP module.

2.4 Dual-Channel Attention Mechanism.

1) Spatial Attention: The Spatial Attention Mechanism (SAM) module generates a weight vector that has the same size as the input image and a channel count of 1. By multiplying this weight vector with the input image, the importance of each pixel in the input image is determined, allowing the module to focus on the positions of more critical pixels. The structure of the SAM module is illustrated in Figure 5.

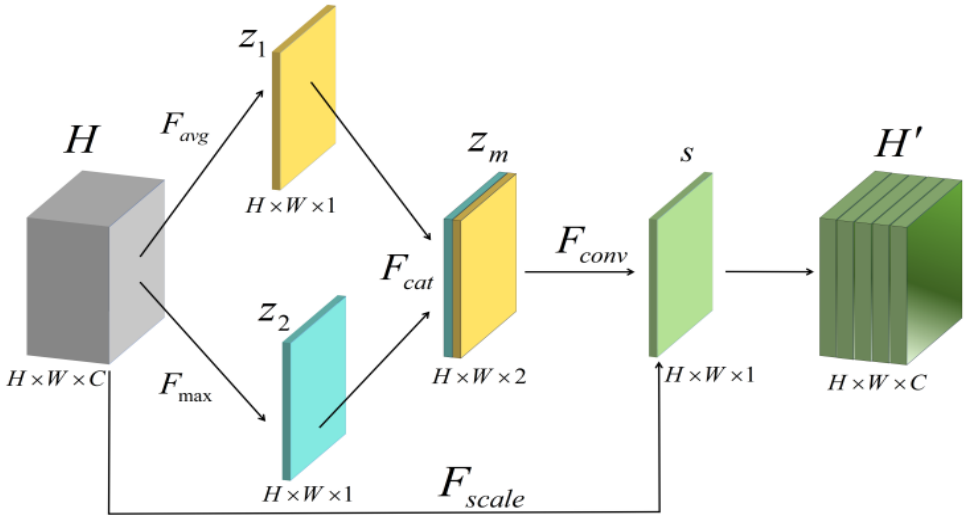

Figure 5. Network Structure of the SAM Attention Module

The specific calculation process consists of two steps: fusion and weighting：

a) Perform average pooling operation Favg and maximum pooling operation Fmax on the feature map. The two obtained 2D vectors $z1$ and $z2$ are stacked along the channel dimension through the Fcat operation to get vector zm, and then a 2D weight vector s is obtained through the convolution operation Fconv. Here, σ denotes the Sigmoid activation function, and $f7\times7$ represents a 7×7 convolution. The calculation formula is as follows:

b) The spatial 2D weight s is multiplied by the input feature map H through the weighting operation Fscale, resulting in the weighted output feature map H'. The calculation formula is as follows:

$$\mathbf{H}' = s \cdot \mathbf{H} \tag{5}$$

2) Channel Attention: The Channel Attention Mechanism (CAM) module utilizes a one-dimensional weight vector to identify feature channels that make greater contributions in the feature map, while filtering out those with minimal feature contributions. Its structure is illustrated in Figure 6.

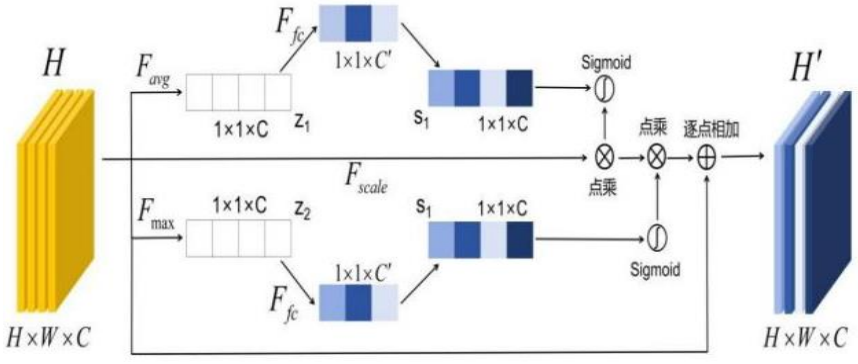

Figure 6. Network Structure of the CAM Attention Module

The specific calculation process mainly includes three steps: pooling, fully connected, and weighting:

a) First, perform the Favg operation and Fmax operation on the input feature map to obtain two one-dimensional vectors, denoted as z_1 and z_2 respectively. Let v_c represent the pixel points in a specific feature channel of the feature map. The calculation formulas are as follows:

$$z_1 = F_{avg}(v_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} v_c(i, j) \tag{6}$$

$$z_2 = F_{max}(v_c) = \max\left(\sum_{i=1}^{H} \sum_{j=1}^{W} v_c(i, j) \right) \tag{7}$$

b) The obtained one-dimensional vectors z_1 and z_2 are compressed into one-dimensional vectors with C' channels through the fully connected operation F_{fc}, and then expanded into one-dimensional weight vectors s_1 and s_2 with C channels. Here, C' = C/r, where r is the compression ratio, σ denotes the Sigmoid function, δ denotes the ReLU function, and f_1 and f_2 denote the fully connected layers. The calculation formulas are as follows:

$$s_1 = F_{fc}(z_1) = \sigma(f_2 \delta(f_1 z_1)) \tag{8}$$

$$s_2 = F_{fc}(z_2) = \sigma(f_2 \delta(f_1 z_2)) \tag{9}$$

c) Finally, through the weighting operation Fscale, the input feature map is multiplied by the two one-dimensional weight vectors s_1 and s_2 respectively, and then added point-wise to the input feature map to obtain the weighted output feature map H'. The calculation formula is as follows:

$$\mathbf{H}' = \mathbf{H} + (s_2 \cdot (s_1)) \tag{10}$$

3. Experiments and results

3.1 Lawn Dataset

Given the current lack of publicly available segmentation datasets for lawn scenarios, this study collected images both offline and online to construct a dedicated lawn scene segmentation dataset, which provides an important foundation for subsequent algorithm research and performance evaluation. To facilitate the model in loading the lawn dataset, Python scripts were used to rename the images in batches, and the dataset was named VOC_grass. Eventually, 646 lawn scene images were obtained. Examples of original images from the dataset are shown in Figure 7.

Figure 7. Examples of Original Images from the Lawn Dataset

3.2 Experimental Configuration and Training Parameters

This experiment is implemented based on the PyTorch network framework, and the specific configurations used in subsequent comparative experiments are shown in Table 2.

Table 2 Experimental Configuration Table

Name	Parameter
Operating system	Windows11
CPU	Intel(R) Core(TM) i9-12900H
GPU	NVIDIA GeForce RTX 3060 Laptop
RAM	16GB
Development language	Python 3.10
Development environment	PyCharm 2023
Web framework	PyTorch 1.7.1
CUDA	Cuda 11.0

In the subsequent comparative experiments, the network training process is divided into a freezing phase and an unfreezing phase. The parameter settings are consistent across all comparative experiments, and the specific training parameters are shown in Table 3.

Table 3 Training Parameters Table

Training details	Parameter setting	
Training Optimizer	Adam	
Optimizer Momentum	0.9	
Learning Rate Strategy	Cos	
Maximum Learning Rate	0.0005	
Minimum Learning Rate	0.000005	
Training Process	Freeze	UnFreeze
Epoch	1~100	101~200
Batchsize	8	4

This experiment adopts the Cosine learning rate strategy. In each iteration, the learning rate decreases based on the specific number of iterations, as well as the benchmarks of the maximum learning rate and the minimum learning rate. The specific calculation formula is as follows:

$$\eta_{new} = \eta_{\min} + \frac{1}{2}(\eta_{\max} + \eta_{\min})\left(1 + \cos\left(\frac{T_{cur}}{T_{\max}}\pi\right)\right) \tag{11}$$

In the above formula, η_{new} represents the current learning rate, η_{max} represents the maximum learning rate, η_{min} represents the minimum learning rate, Tc_{ur} represents the current number of iterations, and T_{max} represents the maximum number of iterations.

In the subsequent experiments, two metrics—Mean Pixel Accuracy (MPA) and Mean Intersection over Union (MIoU)—are used to verify the network performance. The specific calculation formulas are as follows:

$$MPA = \frac{1}{N+1}\sum_{i=0}^{N}\frac{P_{ii}}{\sum_{j=0}^{N}P_{ij}} \tag{12}$$

$$MIoU = \frac{1}{N+1}\sum_{i=0}^{N}\frac{P_{ii}}{\sum_{j=0}^{N}P_{ij} + \sum_{j=0}^{N}P_{ji} - P_{ii}} \quad (13)$$

In the above two formulas, P_{ii} denotes the number of pixels that are actually of class i and predicted as class i; P_{ji} denotes the number of pixels that are actually of class j and predicted as class i; and P_{ij} denotes the number of pixels that are actually of class i and predicted as class j.

3.3 Experimental Results and Analysis

To verify the effectiveness of the improvements proposed in this study, three sets of experiments were designed during the experimental phase. First, the dilation rates of the dilated convolutions in the ASPP module were adjusted, and the ASPP module was replaced with the S-ASPP module for equivalence, so as to verify the effectiveness of the S-ASPP module. Second, for the dual-channel attention mechanism module proposed in this study, experiments were conducted to verify the effectiveness of this improved module for the network. Finally, under the condition that all training parameters were the same, comparative experiments were designed for different semantic segmentation algorithms.

1) Comparative Experiments on the Improved S-ASPP: To verify the effectiveness of the S-ASPP module proposed in this study, it was compared with the ASPP module in DeepLabV3+. Experiments were carried out on the Lawn Dataset to investigate the impact of dilation rates on the performance of dilated convolutions. In response to the problem that the network performance degrades as the dilation rate in the ASPP module increases, multiple groups of experiments with different dilation rates were set up. Table 3.3 lists three scenarios: dilation rate less than 24, dilation rate equal to 24, and dilation rate greater than 24. All experimental parameters were set identically, and MobileNetV2 was used as the backbone network in all cases.

Experimental analysis shows that the S-ASPP module exhibits significant advantages in maintaining the stability of network performance. As shown in Table 4, when the dilation rate exceeds 24, the performance of the traditional ASPP module degrades significantly, while the S-ASPP module still maintains stable performance. Further comparative analysis reveals that under the condition of the same receptive field, the performance of the S-ASPP module is significantly better than that of the ASPP module. In particular, when the dilation rate is greater than 24, compared with the ASPP module, the S-ASPP module achieves an increase of 2.26% in MIoU (Mean Intersection over Union) and 3.36% in MPA (Mean Pixel Accuracy). The experimental results indicate that by decomposing the parallel dilated convolutions with large dilation rates into a series structure of convolutions with small dilation rates, the S-ASPP module effectively alleviates the grid effect caused by the increase in dilation rate while maintaining the same receptive field, thus ensuring the stability of network performance. This fully proves the robustness and effectiveness of the S-ASPP module in semantic segmentation tasks.

Table 4 Performance Comparison Test of ASPP and S-ASPP Under Different Void Fractions

Method	Void Fraction	Receptive Field	MIoU(%)	MPA(%)
ASPP	(6,12,18)	37	78.54	85.03
S-ASPP	(6,6,6)	37	79.85	87.33
ASPP	(18,24,30)	61	76.40	85.28
S-ASPP	(6,6,6,6,6)	61	76.84	86.65
ASPP	(24,30,36)	73	73.50	83.35
S-ASPP	(6,6,6,6,6,6)	73	75.76	86.71

It can be seen from Table 3.5 that after the original network increases the number of shallow feature layers and uses the Feature Fusion module (FF) for feature extraction, its performance is better than that of the original network. By comparing ②, ③, and ④, it can be observed that when three shallow feature layers are selected and fused using the FF module, the network achieves the best segmentation performance, with Mean Intersection over Union (MIoU) and Mean Pixel Accuracy (MPA) reaching 79.70% and 88.57% respectively. Comparing ② and ③, it can be found that after introducing the improved module, the MIoU and MPA of the network increase by 1.16% and 3.54% respectively compared with the original network. This effectively verifies the effectiveness of the Feature Fusion module (FF). Its loss curve is shown in Figure 8.

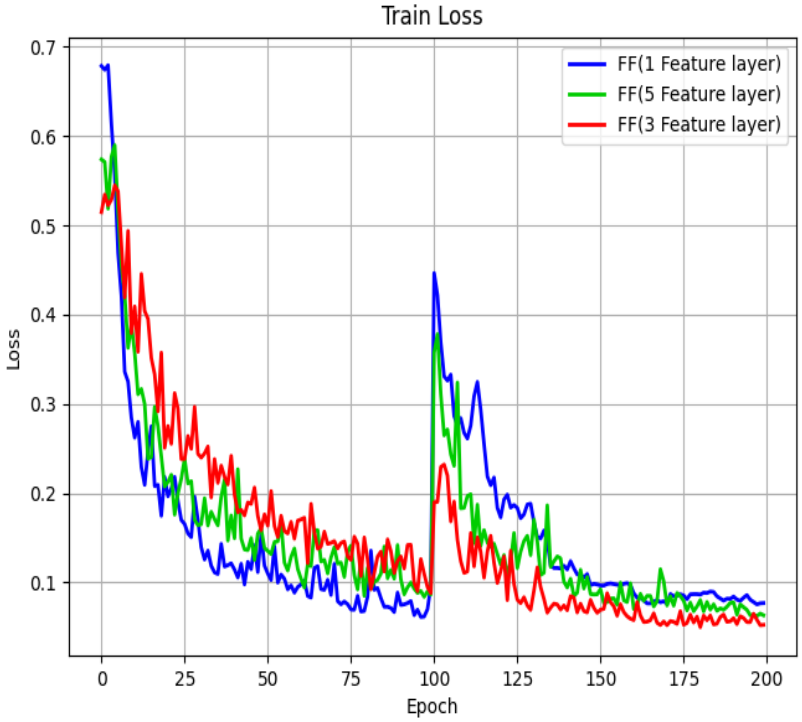

Figure 8. Loss Variation on Lawn Data

As shown in the comparative analysis of the training loss curves in Figure 8, all models tend to converge after completing 150 training epochs. When the Feature Fusion (FF) module is introduced and the number of shallow feature layers is set to 3, the network achieves better segmentation performance with a smaller loss value.

Comparative Experiment of Different Segmentation Network Models To verify the generalization performance of the model proposed in this paper, a horizontal comparison experiment was conducted with mainstream semantic segmentation models in recent years on the lawn dataset. All experiments were set with the same training parameters and number of training iterations.

Figure 9 shows the training loss variation curves of different models on the lawn dataset. It can be observed that after 180 epochs, the loss functions of all models exhibit a convergence trend, and the algorithm proposed in this paper demonstrates better convergence performance compared with the comparison methods. To further verify the effectiveness of the proposed algorithm, experiments were performed on all models using the test set, and the specific experimental results are shown in Table 6.

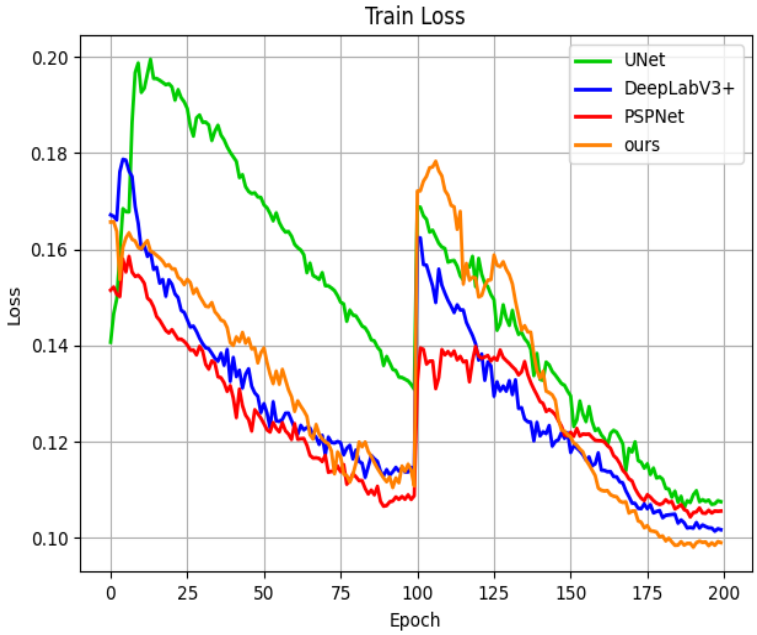

Figure 9. Loss Variation of Each Network on the Lawn Dataset

Table 5 Experimental Results of Each Semantic Segmentation Model on the Lawn Dataset

Network	Backbone network	MIoU(%)	MPA(%)	Params/MB
U-Net	Resnet50	73.65	83.01	167
PSPNet	Resnet50	77.17	86.23	178
DeepLabV3+	MobileNetV2	78.54	85.03	22.4
Ours	MobileNetV2	80.83	88.68	24.7

It can be seen from Table 6 that the MIoU, MPA, and parameter quantity of the algorithm proposed in this chapter on the lawn dataset are 80.83%, 88.68%, and 24.7MB respectively. Compared with U-Net, PSPNet, and DeepLabV3+, the MIoU is increased by 7.18%, 3.66%, and 2.29% respectively, and the MPA is increased by 5.67%, 2.45%, and 3.65% respectively. To intuitively verify the performance advantages of the algorithm proposed in this paper in the image semantic segmentation task, Figure 10 shows a visual comparison of the segmentation effects of various segmentation models on the lawn dataset.

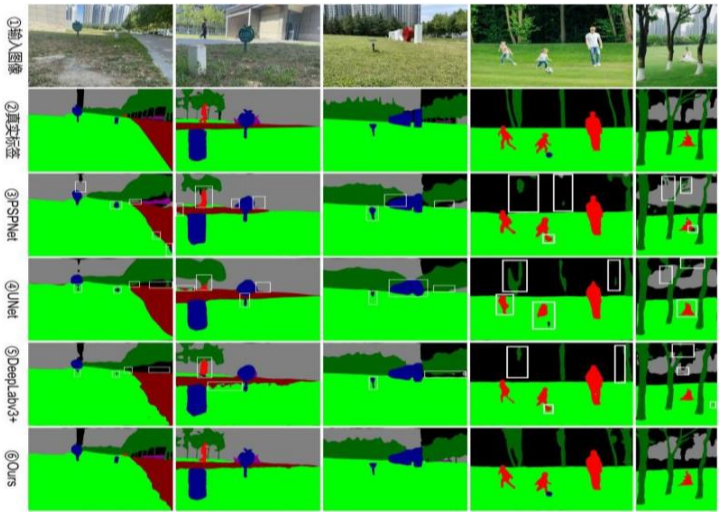

Figure 10. Segmentation Effects of Various Networks on the Lawn Dataset

The visualization results of the lawn dataset are shown in Figure 10. By comparing ② and ③, it can be observed that PSPNet exhibits missed segmentation for targets such as distant pedestrians and tree trunks in the images, and also has issues of unclear segmentation boundaries and missegmentation for targets like buildings and obstacles. Comparing ② and ④ reveals that the U-Net model has relatively poor overall performance in segmentation tasks: when segmenting targets such as trees, pedestrians, and obstacles, the lack of segmentation boundaries is quite severe, and in the second image, the segmentation effect for pedestrians and trees is unsatisfactory. By comparing ③, ④, and ⑤, it can be seen that although DeepLabV3+ also has the problem of unclear segmentation boundaries, it has fewer missegmentation cases compared with PSPNet and U-Net. Finally, by comparing ③, ④, ⑤, and ⑥, it is evident that the improved algorithm proposed in this paper outperforms the other three algorithms. It better addresses the issues of unclear image segmentation boundaries and missegmentation, achieving a more favorable segmentation effect.

4.Conclusion

To improve network performance, an improved semantic segmentation network based on DeepLabV3+ is proposed. The specific improvements are as follows: The lightweight convolutional neural network MobileNetV2 is used to replace the original backbone feature extraction network Xception. On the premise of no loss in network performance, this replacement reduces the number of network parameters and training time, simplifies the network structure, and improves the network prediction speed.

The S-ASPP module, which serially connects several dilated convolutions with small dilation rates, is adopted to replace the ASPP module in the original network. This module enables the acquisition of multi-scale features, allowing the network to maintain stable performance while obtaining a larger receptive field, and addresses the issue of network performance degradation in the original ASPP module caused by increasing the dilation rate of dilated convolutions. Finally, a dual-channel attention mechanism (DCAM) is introduced to focus on regions that contribute more in the image, acquire richer image feature information, and enhance the network’s ability to recognize the boundary contours of objects. Ablation experiments on each module and comparative experiments on the improved network were conducted. The experimental results show that compared with the original algorithm, the proposed improved model not only significantly reduces the number of parameters but also achieves a substantial improvement in network performance. It effectively mitigates problems such as missegmentation at image edges and discontinuous segmentation boundaries.

Acknowledgments

The authors wish to thank the cooperators. This research is partially funded by the National-level College Students' Innovation and Entrepreneurship Fund Project（202510702002X）.

References

[1] Yan Y, Deng C, Li L, et al. A review of image semantic segmentation methods under the background of deep learning[J]. Journal of Image and Graphics, 2023, 28(11): 3342-3362.

[2] Cai Y, Huang X G, Zhang Z A, et al. Real-time semantic segmentation algorithm based on feature fusion[J]. Laser & Optoelectronics Progress, 2020, 57(2): 021011.

[3] Wang L F, Yan C M. Review of semantic segmentation for road scenes[J]. Laser & Optoelectronics Progress, 2021, 58(12): 1200002.

[4] Jiang H, Wang R, Shan S, et al. Adaptive metric learning for zero-shot recognition." IEEE Signal Processing Letters 26.9 (2019): 1270-1274.

[5] Garcia-Garcia, Alberto, et al. "A survey on deep learning techniques for image and video semantic segmentation." Applied Soft Computing 70 (2018): 41-65.

[6] Feng Hu, Wei Liu, Junyu Lu, et al.Urban Function as a New Per spective for Adaptive Street Quality Assessment[J].Sustainability, MDPI, Open Access Journal,2020,12(4).

[7] Zhang Risheng, Yuan Mingting, Ding Junhang, et al. Extraction of Enteromorpha prolifera images based on image threshold segmentation[J]. Techniques of Automation and Applications, 2020, 39(2):83-86.

[8] Zhang W,Wang X,You W,et al.RESLS:Region and Edge Syner getic Level Set framework for image segmentation[J].IEEE Trans actions on Image Processing,2020:57-71

[9] Yang Y, Li Y, Zhao Q H. Fuzzy threshold segmentation of optical remote sensing images with variable classes based on local spatial information[J]. Acta Automatica Sinica, 2022, 48(2):582-593.

[10] Pang M M, An J C. Image segmentation fusing fuzzy LBP and Canny edge[J]. Computer Engineering and Design, 2019, 40(12):3533-3537.

[11] Hu G Z, Xu S J, Meng Y B, et al. Image Segmentation Method Based on Edge-Constrained Local Region MRF[J]. Computer Engineering, 2021, 47(6):253-261,270.

[12] Yang M, Lei B, Shi L N, et al. Fuzzy Divergence Multi-threshold Image Segmentation Based on Standard Deviation Method[J]. Computer Applications and Software, 2020, 37(05):219-225.

[13] Sun Y, Chen Z, Wang H B, et al. Level Set Underwater Image Segmentation Fusing Regional and Edge Features[J]. Journal of Image and Graphics, 2020, 25(04):824-835.Wei Z,Haodi Z,Yujin Y,et al.A semantic segmentation algorithm using FCN with combination of BSLIC[J].Applied Sciences, 2018,8(4):500.

[14] Jichao S,Xiuzhi L,Songmin J,et al.Semantic Segmentation Based on Deep ConvolutionNeural Network[J].Journal of Physics Conference Series,2018,1069:012169.

[15] Wei G, Guo H, An J B, et al. SAR Sea Surface Oil Spill Image Segmentation Method Based on Fully Convolutional Neural Network[J]. Journal of Computer Applications, 2019, 39(S1):182-186.

[16] Wang H O, Liu H, Guo Q, et al. Design of Superpixel U-Net Network for Medical Image Segmentation[J]. Journal of Computer-Aided Design & Computer Graphics, 2019, 31(06):1007-1017.

[17] Chen L C, Papandreou G, Kokkinos I, et al. Semantic image segmentationwith deep convolutional nets and fully connected crfs[J].arxiv preprint arxiv:1412.7062, 2014.

[18] Wang W, Fu Y, Dong F, et al. Semantic segmentation of remote sensing ship image via a convolutional neural networks model[J]. IET Image Processing, 2019, 13(6): 1016-1022.

[19] Chen L C, Papandreou G, Schroff F, et al. Rethinking atrous convolution for semantic image segmentation[J]. arxiv preprint arxiv:1706.05587, 2017.

[20] Fang H, Lafarge F. Pyramid scene parsing network in 3D: Improving semantic segmentation of point clouds with multi-scale contextual information[J]. Isprs journal of photogrammetry and remote sensing, 2019, 154: 246-258

[21] Sandler M, Howard A, Zhu M, et al. Mobilenetv2: Inverted residuals and linear bottlenecks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 4510-4520.

[22] Zhang K, Sun M, Han T X, et al. Residual networks of residual networks: Multilevel residual networks[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2017, 28(6): 1303-1314.

[23] Woo S, Park J, Lee J Y, et al. Cbam: Convolutional block attention module[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 3-19.

[24] Hu J, Shen L, Sun G. Squeeze-and-excitation networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 7132-7141.