Fast Semantic Segmentation Model PULNet and Lawn Boundary Detection Method

Xiaoxia Li¹,², Jingjing Chen¹, Yuanzheng Ye¹,*, Shunli Wang¹,² and Xueyuan Wang¹,²

¹ School of Information Engineering, Southwest University of Science and Technology, Mianyang 621010, China
² Robot Technology Used for Special Environment Key Laboratory of Sichuan Province, Mianyang 621010, China
*Email: 391606040@qq.com

Abstract. To quickly and accurately identify the lawn area and boundary positions of different scenes, environments, and seasons, we propose a new semantic segmentation model PULNet and lawn boundary detection methods. Firstly, the ResNet50 network is improved to expand its effective receptive field, a Pooling pyramid (P) and an Upsampling dimensionality reduction structure (U) is constructed based on the Dilated_ResNet50 network. Secondly, a fast and accurate PULNet semantic segmentation network is proposed integrating the image Local detail information structure (L). Finally, an Eight-neighbor coding method is designed to accurately locate the border of the lawn. Experiments on the ADE20K dataset obtained the mean Intersection over Union (mIoU) and mean Pixel Accuracy (mPA) 32.86% and 75.65% respectively. The average speed is 82.7 frames per second on a platform with GTX 1080Ti GPU. Compared with the Fully Convolutional Network (FCN) the mIoU and mPA are increased by 3.47% and 4.33% respectively, and the speed is 11 times higher. The proposed method can be used for fast and accurate lawn semantic segmentation and boundary detection.

1. Introduction
With the rapid development of artificial intelligence and big data, data with visual information is growing exponentially. The purpose of studying computer vision is to extract targets with semantic information from massive amounts of video and image data, so that computers can better understand and solve problems in the real world, bringing great convenience to people. Although target detection can identify the location and category of the target in the image, it cannot detect the specific boundary of the target, nor can it accurately detect large-area or irregular targets in the image, such as lawns, lakes, sky, and wall cracks. However, applications such as medical treatment, intelligent robots, and drones [1-3] generally operate in a large-area specific area, requiring a computer to identify the location of the target area and its boundary. This is an area target identification and boundary location problem summarized as a boundary detection problem. For the lawn area and its boundary detection problem, it is necessary to analyze the semantics of the scene in the image to identify the lawn and the non-lawn area, and locate the lawn and locate its boundary position on this basis. Target region
recognition is an image semantic segmentation problem. Image semantic segmentation is mainly divided into two methods: image segmentation methods based on artificially designed features and semantic segmentation methods based on Convolutional Neural Network (CNN). Image segmentation based on artificially designed features mainly includes methods such as threshold method [4-6], clustering [7-9], and texture [10-12]. These methods have real-time speed, but they are extremely prone to problems such as voids, mutual "contamination" and misunderstandings between regions with similar features, resulting in inaccurate boundary positioning. The performance of image segmentation depends on the selection of features, and CNN can automatically learn features, and learn different features at different levels: the low-level convolutional layer can express the detailed information of the image, learn the local area characteristics of the image, and help locate the boundary of each target area in the image, the high-level convolutional layer can express the semantic information of the image and learn deep-level abstract features, which is beneficial to the classification of each target area in the image. With the excellent performance of VGGNet, Google Inception Net, and ResNet [13-15] in the ImageNet visual competition [16], in 2014 Long J et al. [17] first used Fully Convolutional Network (FCN) for semantic segmentation, and proposed the model using a learnable deconvolution structure for upsampling to compensate for the loss of detail caused by multiple standard convolutions and pooling layers, and perform pixel-by-pixel classification, which achieves better segmentation results than methods based on artificially designed features. However, the learnable deconvolution layer structure increases the amount of calculation, and the network lacks local detail information and semantic information, and there is more serious intra-class inconsistency. Badrinarayanan V et al. [18] proposed the SegNet model. Its encoder and decoder are in a symmetrical structure. The VGG-16 without the fully connected layer is designed as an encoder for feature extraction, and an index function is designed in the decoder to record the position of each feature point in the feature map after pooling in the feature map before pooling so that it is convenient to enlarge the feature map. SegNet has a good image segmentation effect, but its effective receptive field [19] is small, and high-level semantic information is insufficient. Chen LC et al. proposed the DeepLabV1 model [20], which uses dilated convolution to expand the effective receptive field of the feature extraction network, improves high-level semantic information, and uses conditional random fields to replace the SoftMax classification layer to improve the accuracy of classification, but the details of the image are missing. To make up for the detailed information of the image and continue to improve the effective receptive field of CNN, Chen LC and others have successively proposed DeepLabV2 and DeepLabV3 models [21-22], using dilated convolution with different expansion rates and the global pooling layer to expand the effective receptive field to compensate the loss of detail information caused by the pooling layer. Zhao H et al. [23] proposed PSPNet, which uses DeepLab's similar feature extraction network and builds a pooled pyramid classification layer, which has a better semantic segmentation effect. Although the above model has a high semantic segmentation accuracy, it has a large amount of calculation, a slow speed, and a more complicated model. For lawn datasets with more sample diversity and fewer categories, overfitting is prone to occur. Zhao H et al. [24] proposed ICNet, which uses multi-scale images as input and uses cascaded feature fusion modules to form a multi-task objective function, which reasonably reduces the complexity of the model and greatly improves the speed of semantic segmentation, but it also reduces the ability to express high-level semantic information.

Therefore, this paper proposes a PULNet semantic segmentation network: to reduce the complexity of the Resnet-50 model and increase the effective receiving range of the network, the Dilated ResNet50 network structure is constructed. To improve the invariance of features to image translation and multi-scale changes, a Pooling pyramid structure (P) is constructed. To accelerate the speed of the semantic segmentation network and enhance the detailed information of the feature map, an Upsampling dimensionality reduction structure (U) is designed. To compensate for the loss of detailed information caused by the Pooling pyramid and Upsampling dimensionality reduction structure, an image Local detail information structure (L) is constructed. For the problem of fast positioning of the boundary of the lawn, this paper proposes an Eight-neighbor coding method that can
record the changing trend of pixel values.

2. Related Work

2.1. Feature Map Extractor

With the development of Deep Convolution Neural Networks (DCNNs), the accuracy of VGGNet, Google InceptionNet, and ResNets in the ImageNet-1000 competition has been significantly improved. Semantic segmentation networks such as FCN, SegNet, and Deeplab remove the fully connected layer of the deep neural network and use the remaining part to extract the feature map of the image, so it is called a feature map extractor [25]. ResNets [15] was proposed by Kaiming He et al. The combination of its batch normalization algorithm and residual block largely solves the problems of overfitting, gradient explosion, and gradient disappearance. Therefore, it is favored by most researchers [20-22]. The commonly used network structures of ResNets are ResNet-50, ResNet-101 and ResNet-152. In figure 1, the residual block is the main structure of ResNets, including the Identity Block (IdB) in figure 1 (a) and the Convolution Block (CB) structure in (b). Figure 2 shows the overall structure of ResNet-50. ResNet-50 consists of Conv1_X to Conv5_X and fully connected classification layers.

![Residual block diagram](image1)

**Figure 1.** Residual block diagram.

![ResNet50 diagram](image2)

**Figure 2.** ResNet50 diagram.

2.2. Effective Receptive Field and Dilated Convolution

As shown in figure 3, in the convolution process, the pixel value in the middle of the image is more commonly used. In other words, the pixels in the middle of the image contribute far more to the gradient than the edge pixels during training. Luo W et al. [19] proposed that the contribution of pixels from the edge to the center to the training gradient roughly conforms to the Gaussian distribution, and the receptive field with higher contribution is called the effective receptive field. They confirmed that expanding the convolution and pooling layers can improve the effective receiving field. Many researchers tend to use dilated convolution to learn more global context information [25,26]. Dilated convolution expands \( d \times 1 \) zero values between two adjacent weights of the filter, and then convolutes according to the standard convolution method, where \( d \) is the expansion rate. Assuming that the size of the standard convolution kernel is \( k \times k \), the size of the convolution kernel for dilated
convolution is shown as equation (1):

\[ n = k + (k - 1) \times (d - 1) \]  

(1)

In semantic segmentation, dilated convolution can effectively improve the semantic expression ability and translation invariance of the output feature map. Unlike pooling, it can retain more image details due to its trainable parameters.

Figure 3. Effective receptive field diagram.

2.3. Method of Using Multi-layer Feature Maps and Pyramid Feature Maps

Some recent methods improve the performance of segmentation and detection by using different layers in DCNNs. Networks such as FCN [17] and SSD [27] summarize the scores of each category on multiple scales to improve accuracy. There are two main types of pyramid feature maps: pyramid convolutional feature maps and pyramid pooling feature maps. FPN [28] uses feature maps of different layers to form a convolutional feature map pyramid, which naturally improves the detection efficiency of small-scale and occluded objects. DeeplabV2 and DeeplabV3 [21-22] use dilated convolutional feature map pyramids of different dilation rates to enhance the ability of feature maps to express image details, improve the effective receptive fields, and solve multi-scale problems to a certain extent. PSPNet [23] uses pyramid pooling features to enhance the performance of image scene understanding without losing too many image details. Therefore, using contextual features such as multi-layer feature maps and pyramid feature maps can improve the recognition and detection performance of objects in complex scenes.

3. PULNet Semantic Segmentation Model

This paper proposes a PULNet semantic segmentation model to improve the speed and accuracy of semantic segmentation in deep learning, it is suitable for lawn segmentation. In figure 4, the green square is the standard convolution output feature map, the red square is the dilated convolution output feature map, the yellow square is the pooling feature map, and the purple square is the feature map for prediction.

The PULNet model first scales the input image to a ratio of 1/2, and then improves ResNet50 to Dilated_ResNet50 network, which reduces the number of parameters and expands the receptive field of the network but compared with deep networks such as ResNet100 and ResNet152, the feature semantic expression ability is slightly insufficient. Therefore, a Pooling pyramid (P) is designed, the pooling layer of the pyramid structure improves the invariance of the effective receptive fields and features to the rotation, translation, and multi-scale changes of the image while losing less detail information. To further accelerate the speed of the semantic segmentation network, reduce the complexity of the model, and enhance the detailed information of the feature map, and Upsampling dimensionality reduction structure (U) is designed, which reduces the dimensionality of the feature map while fusing the underlying feature map with rich detailed information. To further compensate for the loss of detail information caused by the pooling pyramid and the Upsampling dimensionality reduction structure, an image Local detail information network (L) is designed, using the original size
image as input, the feature map extracted by the shallow convolution network is additively fused with the output feature map of the Upsampling dimensionality reduction structure, and then the feature maps are spliced and upsampling after dilated convolution with different dilation rates to obtain the output layer.

![Figure 4. PULNet semantic segmentation network.](image)

### 3.1. Dilated_ResNet50 Feature Extractor

As shown in figure 5, to meet the real-time and accuracy requirements of semantic segmentation, reduce the complexity of the model, and improve the generalization ability, a Dilated_ResNet50 network structure is designed. Firstly, it gives up the final average pooling layer, the feature expansion layer, and the fully connected layer of ResNet50. Secondly, the number of channels of the output feature maps of modules other than Conv1_x is changed to 128, 256, 512, and 1024 to reduce the feature size of the network, and the Conv3_1 output feature map is bilinear interpolated to half of the size of the input feature map, which further improves the speed of semantic feature extraction. Finally, to avoid insufficient semantic expression ability and improve the effective receptive fields of the network, the 3×3 standard convolution in Conv4_x and Conv5_x is changed to 3×3 dilated convolution with a dilation rate of 2.

![Figure 5. Dilated_ResNet50 diagram.](image)

### 3.2. P Structure

The pooling pyramid integrates the feature map context information of different regions to improve the effective receptive field, thereby strengthening the semantic expression ability of the feature map, and weakening the loss of detail information caused by a single pooling layer feature map. To compensate for the loss of detail information and further improve the speed of the semantic segmentation network, a Pooling pyramid (P) is constructed. In figure 6, the input and output feature maps of the merged pyramid are both 1/32 of the input image size. Firstly, the input feature map is pooled with the step size and the window size are both 1, 1/2, 1/3, 1/4 of the input feature map to form a pooling pyramid with four pooled feature maps. Secondly, bilinear interpolation is used to
convert each feature map to 1/32 of the original image size. Finally, the feature maps are additively fused.

**Figure 6.** Pooling pyramid (P).

### 3.3. **U Structure**

Although the P structure can improve the effective receptive field but lacks detailed information. To further enhance the detailed information of the semantic segmentation network, construct an Upsampling and dimensionality reduction structure (U). As shown in figure 7, the convolution process of the Upsampling dimensionality reduction structure. On the right are the specific parameters of the convolution. "Conv1×1, 256, 1, BN, ReLU" represents standard convolution with the filter size is 1×1, the number of channels is 256, and the step size is 1, followed by batch normalization and activation functions. "Dilated_Conv3×3, 128, 2, BN" represents a dilated convolution with a filter size of 3×3, the number of channels is 128, and an expansion rate of 2, followed by batch normalization. The Upsampling is a bilinear interpolation method. The process of the Upsampling dimensionality reduction structure is: First, the pooling feature map with the number of channels of 1024 is reduced to 256 with a filter of 1×1 size, and after double upsampling, the dilated convolution is used with the number of channels is 128, the filter size is 3×3 and the expansion rate is 2. Secondly, the number of channels of the output feature map of the Conv3_1 residual block in the Dilated_ResNet50 basic network is reduced to 128 with a 1×1 filter. Finally, the output feature maps of the above two processes are additively fused, followed by the same upsampling and dilated convolution process. Both feature maps after upsampling are used as the final feature maps for regression and prediction. And the prediction result is set to P_{1/16} and P_{1/8}.
Figure 7. Upsampling dimensionality reduction structure (U).

3.4. L Structure
A local detail information structure is established based on context information to further improve the ability of feature maps to express local details of an image. Figure 8 is an image local detail information network. The input is the size of the original image. The process is as follows: Firstly, twice standard convolution is used, the number of channels is 32, the filter size is 3×3, the step size is 2, and each convolution has a batch normalization (BN) and activation function ReLU. Secondly, use a filter with 64 channels, the same size and step size as before, and then use a 1×1 filter to change the number of channels to 128, which is additively fused with the above output feature map of the Upsampling dimension reduction structure. Finally, the outputs of the dilated convolution with the expansion rate of 4, 7, and 9, are spliced and double upsampling, then the number of channels is changed and upsampling to obtain the output layer. The upsampled feature map is used as the final feature map for regression and prediction. And the prediction result is set to P\textsubscript{1/4}.

Figure 8. Local detail information structure (L).

4. Multi-task Loss Function
To sum up, this paper defines the prediction results of the three feature maps with the size of 1/4, 1/8, and 1/16 of the original image as P\textsubscript{1/4}, P\textsubscript{1/8}, and P\textsubscript{1/16} respectively. The multi-task loss function is established as follows:

1. A 1×1 filter is used to change the number of channels of the three feature maps to the number of training categories n, and change the shape of the feature map to a vector form.

2. The size of the label image (the category and the pixel value are equal) is scaled to the size of the feature map, and the shape of the feature map is changed to a vector form and is masked. The purpose of the mask is to extract the values less than or equal to the number of categories to form a label vector G and record the position index in the feature map of each feature value in G. The result of step (1) is taken out according to the index to form a prediction vector P.

3. The cross-entropy loss L is calculated according to equation (2) for P and G.

\[
C = -\frac{1}{n} \sum_{x} [y(x)\ln a(x) + (1 - y(x))\ln(1 - a(x))] \tag{2}
\]
In equation (2), \( n \) is the total number of samples, \( x \) is the samples, \( y \) is the true value and \( a \) is the predicted value.

Three cross-entropy losses \( L_1 \), \( L_2 \) and \( L_3 \) are obtained by the above three output feature maps. The multi-task loss function is as follows:

\[
L_{\text{total}} = \lambda_1 L_1 + \lambda_2 L_2 + \lambda_3 L_3 + \frac{1}{2} \|w\|^2
\]

(3)

In equation (3), \( \frac{1}{2} \|w\|^2 \) is the regularization of the parameters.

5. Lawn Boundary Positioning Method

Figure 9. Lawn and lawn segmentation mask image.

Figure 9 (a) is the lawn image, (b) is the image segmented and masked by PULNet. The color of the non-turf mask is black (0, 0, 0), the color of the lawn mask is green (4, 250, 7). Because there are obstacles such as trees, chairs, and pebbles on the lawn, the segmentation map obtained has holes. Usually, the pixel-by-pixel traversal method is used to determine the categories on the left and right sides of the boundary line to locate the lawn boundary.
Figure 10. Eight-neighbor coding method boundary point location diagram.

Figure 10 is a schematic diagram of the location of the boundary points of the lawn segmentation binary image (the lawn is 1, the obstacles and other non-turf categories are 0). To quickly detect the boundaries of lawns and obstacles, an Eight-neighbor coding method is proposed. $A_{ij}$ is the center point of the $3 \times 3$ window, and $(i, j)$ are the coordinates in the image.

$$C_{km} = \begin{cases} 1 & \text{if } A_{pq} = 1 \\ 0 & \text{if } A_{pq} = 0 \end{cases}$$

(4)

In equation (4), $C_{km}$ is a code for each window in the Eight-neighbor, so $k, m \in [0, 2]$, $k$ and $m$ are not 1 at the same time, $p \in [i-1, i+1]$, $q \in [j-1, j+1]$, and $A_{pq} \neq A_{ij}$.

$$N_d = \sum_{k=0}^{2} \sum_{m=0}^{2} C_{km}$$

(5)

In equation (5), $d$ is the step length of $3 \times 3$ window traversal.

Eight-neighbor coding traversal method: the image is traversed in a certain step from bottom to top, from left to right with a $3 \times 3$ window composed of eight pixels, the number of lawn pixels in the window is recorded as $N_d$, if $N_d < 4$, then the window is not scanned to the lawn boundary point; if $N_d \geq 4$, the lawn boundary point is not determined, the window moves to the right. If $N_d$ keeps increasing, the window finds the lawn boundary point, $(i, j)$ is the lawn boundary point coordinates, stop traversing right. If the window is scanned from left to right, the starting point is $N_d \geq 7$, and as the window moves to the right, $N_d$ will not change or change very little, it means that the area is all lawn. If $N_d$ continues to decrease, there are obstacles in the area, and Eight-neighbor coding traversal method determines the location of boundary points according to the changing trend of pixels.

6. Experimental Results

| Configuration     | Model              |
|-------------------|--------------------|
| CPU               | I7-7700K           |
| GPU               | NVIDIA GEFORCE GTX1080Ti |
| RAM               | 16G                |
| Operating system  | Ubuntu16.04        |
In Table 1, the deep learning framework used in this experiment is TensorFlow1.8, using CuDNN V6 and Cuda9.1 versions, and the model is trained on a single GPU 1080Ti server.

6.1. Introduction to DataSet

6.1.1. ADE20K Dataset. The ADE20K dataset has more than 20,000 detailed semantic segmentation and scene understanding images, of which the number of images in the training set is 20,000, the number of images in the validation set is 2,000, and a batch of test sets. The dataset has a total of 150 categories, including indoor and outdoor scenes such as people, cars, sky, lawns, and roads. It was the largest and most commonly used semantic segmentation and scene understanding dataset at the time, which was quite challenging. Figure 11 shows the ADE20K verification set and its annotations. This paper uses the dataset in the "ADEChallengeData2016" folder on the ADE20K official download site for training and verification. The feature of this dataset is that the pixel value of the object in the image is the same as the category of the object. For example, if the lawn is the 10th category, the pixel value of the lawn is marked as (10, 10, 10); for the convenience of observation, the third line in figure 11 uses the image mask.

![Figure 11. ADE20K validation set and annotation images.](image)

6.1.2. Self-built Lawn Dataset. The self-built lawn dataset is taken from different scenes such as campus playgrounds, parks, and highways, in different environments such as light, shadow, and rain, lawns in different seasons, and there are obstacles such as pebbles, trails, pedestrians, and flower stands in the lawn. There are a total of 9134 images in this dataset, and they are all accurately labeled, including 6000 in the training set, 1000 in the validation set, and 2134 in the test set. Examples are shown in figure 12.
6.2. Experimental Process and Hyperparameter Setting

6.2.1. Training Process. The training process of the model is as follows:

(1) Using part of the ImageNet 1000 classification dataset to pre-train the Dilated_ResNet50 basic network and save the trained model parameters.

(2) Removing the fully connected classification layer, adding the Pooling pyramid (P), Upsampling dimensionality reduction structure (U), and the image Local detail information structure (L) to construct the PULNet semantic segmentation network, transferring the trained parameters in (1) to PULNet, and using the ADE20K dataset to train the network. Firstly, the segmentation results of $P_{1/4}$, $P_{1/8}$, and $P_{1/16}$ on the ADE20K verification set is used to prove the role of the Pooling pyramid, the Upsampling dimensionality reduction structure, and the image Local detail information network. And the results are compared with the advanced algorithms in recent years.

(3) Making a lawn segmentation dataset, changing the number of categories of $P_{1/4}$, $P_{1/8}$, and $P_{1/16}$ in the PULNet network, transferring the model trained on the ADE20K dataset in (2) to the current model, fine-tuning the model using the self-built dataset, finally testing the effect of lawn segmentation and boundary detection online.

6.2.2. Hyperparameter Setting. The hyperparameters of the model are set as follows:

Firstly, the same parameter settings are used as ResNet in the above (1) in the ADE20K training process, the input image is scaled to 512×512, the number of categories is set to 150, where 0 is the background category. The model is trained by the Momentum optimizer with a momentum of 0.9 and a decay rate of 0.001 with 90,000 steps, the learning rate is initialized to 0.05, and it decays exponentially with the number of training steps, and the parameters of the loss function $\lambda_1$, $\lambda_2$, and $\lambda_3$ are 0.6, 0.4, and 0.16 respectively.

Secondly, the pre-training model is fine-tuned on the ADE20K dataset before training on the self-built lawn dataset. During the training process, the input image is scaled to 640×480 size, and the number of categories is set to 11, where 0-9 are background categories. 10 is the lawn category. The reason for this setting is that the lawn category of the ADE20K dataset is the 10th category, which is conducive to the adjustment of the gradient during fine-tuning. The Momentum optimizer with a momentum of 0.9 and a decay rate of 0.001 is used to train the model 40,000 steps. The learning rate is initialized to 0.001 and decays exponentially with the number of training steps. The parameters of the loss function $\lambda_1$, $\lambda_2$, and $\lambda_3$ are 0.6, 0.4, and 0.1 respectively.

6.3. ADE20K Dataset Experimental Results

Figure 13 shows the test results of the ADE20K verification set. The first line is the GT mask, and the second line is the test result of the CascadeNet method as a comparison benchmark; the third to fifth lines are the prediction results of the three feature maps in the PULNet network, $P_{1/4}$, $P_{1/8}$, $P_{1/16}$. Among them, $P_{1/16}$ is the prediction result of the output feature map of the Pooling pyramid (P), which shows that it has a good semantic expression ability. The output feature map prediction result $P_{1/8}$ of the additive fusion of the low-level Conv3_1 and the Upsampling dimensionality reduction structure (U) has more detailed information than $P_{1/16}$. Based on the above, the prediction result $P_{1/4}$ of the fused
image Local detail information structure (L) not only has stronger semantic expression ability but also has more detail information, and has better results in the segmentation of the lawn.

Figure 13. ADE20K validation set test results.

Table 2. Comparison of ADE20K validation set indicators.
| methods         | mIoU(%) | mPA(%) | fps  |
|-----------------|---------|--------|------|
| FCN-8S[17]     | 29.39   | 71.32  | 7.5  |
| SegNet[18]     | 21.64   | 71.00  | 36.1 |
| ICNet[24]      | 32.30   | 73.60  | 71.0 |
| PULNet\_P/16     | 29.63   | 71.40  | 85.3 |
| PULNet\_P/8     | 32.56   | 74.00  | 84.5 |
| PULNet\_P/4     | **32.86** | **75.65** | **82.7** |

The last three rows in Table 2 are the experimental results of this PULNet model. The accuracy of PULNet\_P/4 has been significantly improved, and its speed is 10 times faster than the FCN model. Combined with figure 13, the P, U, and L structures designed in this paper have good performance, and their accuracy is gradually improved. Besides, as one can see from the average speed that the P, U, and L structures have lower time complexity.

6.4. Experimental Results of Self-Built Lawn Dataset

Figure 14 shows the effect of PULNet on a typical lawn segmentation. The first two original images were taken in summer, with obstacles such as pedestrians, park chairs, boxes, and strong sunlight and shadows. The middle two original images were taken in autumn, with plants, flowerpots, and footpaths. The fifth one was taken in low light on a rainy day, and the last one was taken in winter, in a shaking environment, the image is blurred and the resolution is low, and the lawn type of each photo is different. As one can see that the PULNet has a good segmentation effect in a variety of environments, and has a strong generalization ability.

Figure 15 shows a comparison of the segmentation effects of different algorithms. Both PSPNet
and ICNet have good effects on the segmentation of lawns: PSPNet has a deeper network level and richer semantic information, but lacks a lot of image details, resulting in the expansion of the green lawn area in the segmentation map. The dilated convolution with the expansion rate of a multiple of 2 leads to irregular image edges. ICNet has richer image detail information and insufficient semantic information, which leads to the expansion of the black non-turf area in the segmentation map. Combined with the mask, it can be seen that PULNet is better than the above two methods.

Table 3. Comparison of lawn test set indicators on self-built lawn dataset.

| methods       | mIoU(%) | fps |
|---------------|---------|-----|
| PSPNet\(^{(23)}\) | 86.41   | 3.5 |
| DeeplabV1\(^{(20)}\) | 82.32   | 0.5 |
| ICNet\(^{(24)}\) | 90.17   | 58.5|
| PULNet        | 96.32   | 67.3|

Table 3 shows the comparison of the test indicators on the self-built lawn dataset. The input image size is 848×480. PULNet has a greater advantage in segmentation accuracy and speed.

Figure 16 shows the detection result of the lawn boundary. The first two rows are segmentation mask images, and the last two rows are boundary detections on the original image. The pictures show part of the field test scene. It has good detection accuracy for the borders of different colors, different varieties, and flatness lawns under strong light, shadow, and rainy conditions, and it can better distinguish the pedestrian walks, pebbles, and bushes of similar colors in the lawn. The Eight-neighbor coding traversal method in this paper has high accuracy and speed for the location of the boundary. On a notebook platform with GTX1050 GPU and i5-7300hq CPU, when the input image is 848×480 the
average speed of the entire lawn boundary detection process is 30.3fps.

7. Conclusion
This paper proposes a PULNet semantic segmentation network for the detection of lawn boundaries, which includes Dilated_ResNet50, Pooling pyramid (P), Upsampling dimensionality reduction structure (U), and image Local detail information structure (L), which can quickly analyze the range of the lawn in the image, and then use Eight-neighbor coding traversal method to record the changing trend of pixel values and locates the boundary. PULNet has the merits of a few parameters and high accuracy, which is suitable for hardware environments such as portable computers, high-performance embedded, and mobile platforms.

Acknowledgements
This research was supported by the National Natural Science Foundation of China (No. 61771411), the Sichuan Science and Technology Program (No. 2020YFSY0062 and 2019YJ0449) and Postgraduate Innovation Fund of Southwest University of Science and Technology (18ycx119).

References
[1] Guo Z, Li X, Huang H, et al. 2019 Deep Learning-Based Image Segmentation on Multimodal Medical Imaging IEEE Transactions on Radiation and Plasma Medical Sciences 3(2) 162-9
[2] Zhu X, Zhao H, Liu Y, et al. 2010 Segmentation and classification of range image from an intelligent vehicle in an urban environment 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems 2010 pp 1457-1462
[3] Kalantar B, Mansor S B, Sameen M I, et al. 2017 Drone-based land-cover mapping using a fuzzy unordered rule induction algorithm integrated into object-based image analysis International Journal of Remote Sensing 38(8-10) 1-22
[4] Osuna-Enciso V, Cuevas E, Sossa H 2013 A comparison of nature-inspired algorithms for multi-threshold image segmentation Expert Systems with Applications 40(4) 1213-9
[5] Chen J, Guan B, Wang H, et al. 2018 Image Thresholding Segmentation Based on two dimensional histogram using gray level and local entropy information IEEE Access 6(99) 5269-5275
[6] Matić T, Aleksić I, Željko Hocenski, et al. 2018 Real-time biscuit tile image segmentation method based on edge detection ISa Transactions 2018 246-54
[7] Tan K S, Mat Isa N A, Lim W H 2013 Color image segmentation using adaptive unsupervised clustering approach Applied Soft Computing 13(4) 2017-36
[8] Lei T, Jia X, Zhang Y, et al. 2019 Superpixel-based fast fuzzy c-means clustering for color image segmentation IEEE Transactions on Fuzzy Systems 27(9) 1753-66
[9] Shen J, Dong X, Peng J, X, et al. 2019 Submodular Function Optimization for Motion Clustering and Image Segmentation IEEE Transactions on Neural Networks and Learning Systems 30(9) 2637-49
[10] Singh M, Singh S, Gupta S 2014 An information fusion based method for liver classification using texture analysis of ultrasound images (Elsevier Science Publishers B. V.) 19 91-6
[11] Nabizadeh N, Kubat M 2017 Automatic tumor segmentation in single-spectral MRI using a texture-based and contour-based algorithm Expert Systems with Applications 77 1-10
[12] Akbulut Y, Guo Y, Şengür A, et al. 2018 An effective color texture image segmentation algorithm based on Hermite transform Applied Soft Computing 67 494-504
[13] Simonyan K, Zisserman A 2015 Very deep convolutional networks for large-scale image recognition International Conference on Learning Representations (ICLR) pp 1-14
[14] Szegedy C, Liu W, Jia Y, et al. 2015 Going deeper with convolution Computer Vision and Pattern Recognition (CVPR) (Boston, MA, IEEE) pp 1-9
[15] He K, Zhang X, Ren S, et al. 2016 Deep residual learning for image recognition Computer Vision and Pattern Recognition(CVPR) (Las Vegas, NV, IEEE) pp 770-8
[16] Russakovsky O, Deng J, Su H, et al. ImageNet Large Scale Visual Recognition Challenge[J]. International Journal of Computer Vision, 2015, 115(3): 211-252.
[17] Long J, Shelhamer E, Darrell T. Fully Convolutional Networks for Semantic Segmentation[C]. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, 2015: 3431-3440.
[18] Badrinarayanan V, Kendall A, Cipolla R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Scene Segmentation[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2017, 39(12): 2481-2495.
[19] Luo W, Li Y, Urtasun R, et al. Understanding the effective receptive field in deep convolutional neural networks[J]. Advances in neural information processing systems, 2016, 29: 4898-4906.
[20] Chen LC, Papandreou G, Kokkinos I, et al. Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs[J]. Computer Science, 2014(4): 357-361.
[21] Chen LC, Papandreou G, Kokkinos I, et al. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2016, 40(4): 834-848.
[22] Chen LC, Papandreou G, Schroff F, et al. Rethinking Atrous Convolution for Semantic Image Segmentation[J]. 2017. arXiv:1706.05587v3 [cs.CV]:1-9.
[23] Zhao H, Shi J, Qi X, et al. Pyramid Scene Parsing Network[C]. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017: 6230-6239.
[24] Zhao H, Qi X, Shen X, et al. ICNet for Real-Time Semantic Segmentation on High-Resolution Images[C]. European Conference on Computer Vision(ECCV), Springer, 2018: 1-9.
[25] Gu Y, Zhong Z, Wu S, et al. Enlarging Effective Receptive Field of Convolutional Neural Networks for Better Semantic Segmentation[C].2017 4th IAPR Asian Conference on Pattern Recognition (ACPR). IEEE, Nanjing, 2017: 388-393.
[26] Yu F, Koltun V. Multi-Scale Context Aggregation by Dilated Convolutions[C]. International Conference on Learning Representations (ICLR). 2016. arXiv: 1511.07122: 1-13.
[27] Liu W, Anguelov D, Erhan D, et al. SSD: Single Shot MultiBox Detector[C]. European Conference on Computer Vision(ECCV), Springer, 2016: 21-37.
[28] Lin T Y, Dollár, Piotr, Girshick R, et al. Feature Pyramid Networks for Object Detection[C]. Computer Vision and Pattern Recognition(CVPR). Honolulu: IEEE, 2017: 936-944.