Journal of Combinatorial Mathematics
and Combinatorial Computing
www.combinatorialpress.com/jcmcc

A Strip Pooling Attention Network for Urban Scene Images Segmentation

Yaojie Zhang[1,✉], Yanling Li[1]

[1] *Department of Computer Science, Changzhi University, Changzhi, Shanxi, 046011, China*

ABSTRACT

The field of urban scene image segmentation is a crucial task in the field of computer vision. Aiming at the problems of large parameter count and insufficient image segmentation accuracy of the traditional DeepLabV3+ model, an improved lightweight DeepLabV3+ model is designed. The overall performance of the model is improved by replacing the Xception backbone network with MobileNetV2, introducing the band pooling module and the densely connected null pyramid module in ASPP, and using the GD-FAM multi-feature fusion module in the fusion stage. Using Cityscapes as the dataset, the model experiment results show that compared with the traditional Deeplabv3+ model, this paper's method increases the target category IoUs of urban scenes such as pedestrians, cyclists, and columns by 3.1%, 4.41%, and 6.74%, respectively. Therefore, the segmentation effect of the model in this paper is significantly better than the segmentation effect of other models. The mIoU of the MobileNetV2 backbone network is 4.91% higher than the baseline model. The loss function change curve of the model shows that it tends to converge after 100 iterations. In summary, the overall segmentation performance of the improved model is significantly improved.

Keywords: image segmentation, urban scenes, improved DeepLabV3+ model, band pooling, GD-FAM

1. Introduction

The eye is an important channel of communication between humans and the world, and in today's highly advanced digital media technology, visually dependent media such as images and videos have penetrated into all aspects of life. Humans can easily understand the key information in an image, but this is a complex task for computers, which are required to recognize both the categories of objects in an image and the regions occupied by different objects from each other [1]. For this reason, computer vision tasks have been continuously categorized and refined, resulting in a series of branches, including image classification, target detection and semantic segmentation [2-3]. Among them, image classification and target detection are the foundation of computer vision tasks, image classification

✉ Corresponding author.
E-mail address: zhang_yaojie@163.com (Y. Zhang).
Received 12 January 2024; Revised 05 March 2024; Accepted 25 December 2024; Published Online 16 April 2025.
DOI: 10.61091/jcmcc127b-389

task is to assign corresponding category labels from a set of classification labels to the input image, and target detection task is not only to assign classification labels to the image, but also to locate the position of a specific target in the image [4-7]. The semantic segmentation task assigns category labels to each pixel point at the pixel level in order to achieve segmentation detection for a specific target category and fine labeling of its boundaries [8-9].

Urban scenes have always been a major challenge for computer vision tasks because they are more complex environments, which makes it exceptionally difficult for models to accurately detect and recognize specific targets [10-12]. Meanwhile, due to the complexity of the types of objects in urban scenes, the size and dimensions of the same target will constantly change, which requires more adaptive processing capability of the model [13-15]. Therefore, obtaining an image segmentation algorithm that can guarantee superior performance and detection speed in urban scenes is of great value for its development and application, and will bring more innovations and breakthroughs in the field of computer vision [16-17]. The existing algorithms for image segmentation are not yet fully mature, and their backbone network has limited ability to extract features, and high computational complexity and high demand for hardware resources, which makes it difficult to achieve ideal detection results in urban scenes [18-20]. The rapid development of image segmentation in recent years has benefited from the re-emergence of deep learning, however, many existing network architectures and segmentation algorithms mostly pile up a large number of convolutional layers or design models with complex structures, and some even sacrifice speed for higher accuracy [21-24]. Based on this, an image segmentation algorithm based on banded pooled attention network is proposed, which can extract more delicate features of the image, thus realizing the accurate segmentation of objects in the image, and is more suitable for tasks with high processing accuracy [25-27].

The traditional DeepLabV3+ model uses Xception as the backbone network, which has problems such as high computational cost and large amount of parameter data. In order to solve these problems, oriented to complex urban scene image segmentation, this paper proposes an improved lightweight DeepLabV3+ model. The backbone network is replaced with MobileNetV2 to realize the model lightweight. The band pooling module and dense connected empty pyramid module (DenseASPP) are introduced on the basis of the original ASPP, which results in a larger sensory field. Then the CBAM attention mechanism, i.e., channel and spatial attention mechanism, is fused to better handle both spatial and channel information. In the different resolution feature fusion stage, a gated dual-stream alignment module is added to improve the integration performance. Training and testing of the improved DeepLabV3+ model is completed using the Cityscapes dataset to validate the effectiveness of the algorithm.

2. Theory and methodology

Semantic segmentation of urban street scenes is an important research direction within the field of computer vision, which aims to classify urban street scene images into different semantic regions, such as pedestrians, buildings, vehicles, roads, etc., essentially classifying each pixel in the image with an accurate semantic category to categorize the pixels and understand the image [28].

2.1. Pooling operation

Pooling operation is a very common processing in convolutional neural networks, which aims to mimic the human visual system and realize the dimensionality reduction of the data, which can effectively suppress the noise, reduce the parameters of the network, and prevent the phenomenon of overfitting. Maximum pooling and average pooling are most commonly used in most deep learning networks. The application of stochastic pooling in the target detection task shows that its effect is close to that of Dropout and can prevent overfitting. Combinatorial Pooling utilizes both Maximum Pooling and

Average Pooling to obtain richer features within the feature extraction layer. Pyramid Pooling is essentially a combination of spatial pooling at multiple scales, which provides better access to contextual information and obtains more effective features in multi-scale space.Pyramid Pooling is introduced into the saliency detection task, and the proposed MPPNet uses pyramid channels of different sizes to aggregate and merge the extracted semantic features more effectively. The application of global second-order pooling (GSoP) to different visual tasks shows that the introduction of GSoP in the network helps in the learning of higher-order representational capabilities, and the second-order network based on GSoP has better representation learning performance compared to the first-order network based on global average pooling. The application of strip pooling (SP) in the field of semantic segmentation enriches the sensory field by using the strip window as the pooling window, so that pooling is no longer limited to the square window, and the effective parsing of complex scenes is successfully achieved [29].

2.2. Strip pooling

To be able to effectively understand complex scenes in cities, there are usually two approaches to modeling the relationship between pixel points in an image.

1) Introducing self-attention or nonlocal mechanisms in network models, but the large computational effort associated with the computation of the relationship matrix between each spatial location weakens the real-time performance of the algorithm.

2) Null convolutional sense fields allow the model to learn contextual information during training. All of these methods perform feature extraction on the input feature map within a square window. When banded objects with large scales (wrenches, pliers) are present, the square sliding window introduces interfering information from irrelevant regions, limiting the flexibility of the network in capturing anisotropic contextual information.

2.2.1. Strip pooling module. The band pooling kernel can effectively establish dependencies between windows and encode feature maps within windows. Meanwhile, fusion with feature maps of other dimensions can model the detailed information of feature maps.

First, the input $x \in R^{W\times H\times C}$ is then averaged with a water semipooling kernel and a vertical pooling kernel, and the outputs are $y_{c,i}^{h} \in R^{C\times H}, y_{c,j}^{v} \in R^{C\times W}$, respectively; Then, a 1D convolution with a convolution kernel of 3 is used to encode the features within the 1D convolution window, which is expanded to generate $y_{c,i,j}^{H} \in R^{w\times H\times C}, y_{c,i,j}^{W} \in R^{w\times H\times C}$. Fusing them for feature fusion, there:

$$y_{c,i,j} = y_{c,i,j}^{H} + y_{c,i,j}^{W} \tag{1}$$

$$y_{c,i,j} = y_{c,i,j}^{H} + y_{c,i,j}^{W} \tag{2}$$

After passing through the 1×1 convolution with sigmoid activation layer, it is multiplied with the input original feature map, and the feature values of each position on the resulting output feature map y are correlated with multiple positions on the input, so as to achieve the purpose of modeling the context. The sense field of band pooling is long and narrow, avoiding the establishment of too many unnecessary connections between long-range pixel points, which also effectively suppresses the introduction of irrelevant information.

2.2.2. Strip Pooling Prediction Subnetwork. The model introduces banded pooling, which prevents information interference from irrelevant regions while extracting long-distance dependencies, and is able to take into account local detail information. The band pooling module is added to the 3-branch prediction network of key point, scale, and center point bias to form a band pooling detection network. Whether the banded pooling module is shared by the 3 branches. In terms of performance, the shared

scheme maintains the computation rate of the original algorithm to a great extent, with little accuracy improvement. Whereas the scheme without parameter sharing leads to a substantial increase in accuracy, but there is a decrease in the computation rate.

2.3. IoU losses

2.3.1. Original network training loss. The original network is trained for the prediction of centroid heat maps, and the learning process is supervised using a focal loss computed pixel-by-pixel:

$$L_k = -\frac{1}{N}\sum_{xyc}\begin{cases}(1-\hat{Y}_{xyc})^{\alpha}\log\hat{Y}_{xyc} & Y_{xyc}=1\\ (1-Y_{xyc})^{\beta}\hat{Y}_{xyc} & \log(1-\hat{Y}_{xyc}), Y_{xyc}\neq 1\end{cases} \tag{3}$$

Where: α and β are hyperparameters of focal loss. As for scale prediction and bias prediction, the learning process is supervised by L1 loss:

$$L_{off} = \frac{1}{N}\left|\hat{O}_p - (\frac{p}{R} - \overline{p})\right| \tag{4}$$

$$L_{size} = \frac{1}{N}\sum_{k=1}^{N}\left|\hat{S}_{pk} - S_k\right| \tag{5}$$

Where: $\hat{O}_p$ is the value predicted by the bias branching model. $\frac{p}{R}$ is the position coordinate divided by the subsampling rate. $\tilde{p}$ is the position of the feature map target centroid $\hat{S}_{pk}$ is the height and width predicted by the scale branch. S_k is the height and width of the real frame, which is $S_k = (x_{\max}^{(k)} - x_{\min}^{(k)}, y_{\max}^{(k)} - y_{\min}^{(k)})$, for any class of objects. The overall supervised learning loss, on the other hand, is a weighted sum of the individual losses:

$$L_{\det} = L_k + \lambda_{size}L_{size} + \lambda_{off}L_{off} \tag{6}$$

Where: $\lambda_{size}, \lambda_{off}$ is the weight of the corresponding loss, which is usually taken as $\lambda_{size} = 0.1, \lambda_{off} = 1.$

2.3.2. IOU losses. The regression of bounding box size during training is divided into two branches, height and width. The two branches are independent of each other during the learning process, calculating their respective losses and summing with the corresponding length truth and width truth, respectively.

However, in reality, the length and width features are not independent, and are often closely related to the object category, containing a wealth of features. In addition, the judgment of whether an object is detected and accurate or not is often carried out by comparing the intersection ratio between the detection frame and the real frame with the threshold, which is greater than that is considered as a successful detection, and vice versa is a failure. For this evaluation criterion, height and width independent learning is obviously not suitable.

For this reason, the loss function L1 for supervised scale branch learning is changed to the IoU loss which is more suitable for scale learning:

$$IoU(A,B) = \frac{A\cap B}{A\cup B} = \frac{A\cap B}{|A|+|B|-A\cap B} \tag{7}$$

Where: A is the area S of the region surrounded by the real frame, $S = W\times H$; B is the area of the region surrounded by the prediction frame.

Then the expression for the IoU loss is:

$$L_{loss} = 1 - IoU(A,B) \tag{8}$$

3. Improved lightweight DeepLabV3+ model design

The DeepLabV3+ model relies on more complex backbone networks (e.g., Xception), which results in high computational costs and a large number of parameters, limiting its potential for application on resource-constrained devices. In addition, although its adoption of the empty space pyramid pooling (ASPP) technique effectively enlarges the receptive field and improves the model's adaptability to scale variations, its ability to capture detailed features is still limited, especially when dealing with elongated or complex and variable objects. To address these issues, this study proposes several key improvements for DeepLabV3+ [30].

The improved lightweight DeepLabV3+ model backbone uses the more lightweight MobilenetV2, which replaces the original Xception and greatly reduces the number of network parameters, making it better for mobile applications. The original ASPP part is replaced with a densely connected null pyramid module that adds a bar pooling module to obtain a larger sensory field and denser sampling points, while the bar pooling makes the detailed features more accurate. Subsequently, the CBAM attention mechanism is added to better handle spatial and channel information.

3.1. Adding densely connected ASPP with band pooling

In this paper, we replace the original ASPP with DenseASPP that incorporates band pooling. DenseNet uses more densely connected convolutions for better performance, but the number of channels in the neural network increases rapidly due to the dense connections. Therefore DenseASPP uses multiple 1×1 convolutions in order to limit the number of rapidly growing channels to reduce the number of operational parameters of the network as well as the amount of computation. And through experiments, it is known that the quality of image output is more reasonable when the expansion rate is less than 4, so the two are combined. The cascading of dense convolutions at the same time achieves a more balanced segmentation effect. The spatial pyramid module uses three convolutions with different expansion rates, one ordinary convolution as well as one average pooling layer, and the densely connected ASPP employs layer-hopping connections to share information between the front and the back to obtain a larger sensory field to provide more comprehensive global information. When the convolution kernel size of the null convolution is k and the expansion rate is r, the receptive field of the null convolution is:

$$R=(r^{'}-1)\times(k-1)+k \tag{9}$$

The expansion rates of DenseASPP used in this paper are 1, 2, 3, 4, and 8 times that of 3, respectively. The relationship between information utilization β and sensory wildness is as follows:

$$\beta=\frac{\text{The number of pixels in a feature map that participate in an effective operation}}{\text{Feel the total amount of elements in the field}} \tag{10}$$

Since ordinary pooling leads to increased complexity of the model and in some cases the target objects may have long structures. The use of a large square pooling window does not solve this problem well as it will inevitably incorporate contaminated information from irrelevant domains. Therefore, the strip pool module is introduced to cover more complex streetscape information. Improving the effectiveness of streetscape segmentation. The standard spatial average pooling is given by:

$$y_{io,jo}=\frac{1}{h\times w}\sum_{0\le i<h}\sum_{0\le j<w}x_{i_o\times h+i,j_o\times w+j} \tag{11}$$

The above equation represents the average pooling of a $h\times w$, i.e., for the feature map x of $H\times W$, the mean values in each of its $h\times w$ regions are averaged, and the output feature map obtained is $H/h\times W/w$. However, it may contain a large number of irrelevant regions when dealing with irregular

targets In order to solve the above problem Strippooling is proposed, which will perform pooling operations along the horizontal or vertical direction.

By adopting strip pooling, deep learning models can effectively enhance their ability to capture image global information without significantly increasing the computational burden, especially for those application areas that rely on long-range spatial features, such as high-resolution image processing and large scene understanding. Therefore, band pooling provides a new means for deep neural networks to capture spatial relationships simply and effectively, which helps to improve the overall performance of the model.

The horizontal pooling formula is expressed as shown in equation (12):

$$y_i^h = \frac{1}{W} \sum_{0 \le J < w} x_{i,j} \tag{12}$$

The vertical pooling equation is expressed as shown in equation (13):

$$y_j^h = \frac{1}{W} \sum_{0 \le J < w} x_{i,j} \tag{13}$$

3.2. CBAM Attention Mechanism

CBAM attention mechanism is a combination of Channel and Spatial attention mechanism.Traditional attention such as SENet mostly works on the analysis of channel attention.CBAM not only pays attention to the weights of each channel, but likewise pays attention to the weight of each pixel. This attention mechanism first performs channel attention mechanism and subsequently this result is processed through spatial attention mechanism. The input image is first subjected to Maximum Feature Pooling and Global Average Pooling, these two components are first pooled for height and width. These two data are used as the number of channels for the feature input into two shared fully connected layer parts, the first fully connected layer with fewer neurons and the second fully connected layer with the same number of neurons as the input number of channels fully connected layer. The pooled result gets two features through the fully connected layer and subsequently the two features are added together and the value is fixed through Sigmoid function and this feature map has the same number of channels as the input feature map to complete the channel attention addition.

3.3. GD-FAM multi-feature fusion

DeepLabV3+ network only utilizes the first down-sampled feature maps in the decoder part of the backbone network part, which is not fully utilized in the coding side to improve the capability of feature representation of the network. In this paper, all the downsampled features of 1/8 and 1/16 feature maps are downsampled in MobileNet, resulting in a waste of semantic information. In order to obtain more detailed underlying semantic features, the sampling coefficients for will be the second and third low-level semantic features were merged, respectively. Meanwhile, in order not to affect the output of the DenseASPP module, the second 1/16 downsampling is carried out, and the first three downsampled low-level semantic features are reduced to 64, 32, and 16, respectively. The influence of low-level semantic information on high-level semantic information is avoided.

4. Model realization and analysis of experimental results

4.1. Description and comparison of experimental data

In general, data is one of the most important parts for machine learning including deep learning tasks nowadays. For the task of semantic segmentation of urban scenes, it is even more beneficial from large-scale datasets. Currently, the Cityscapes dataset is one of the large-scale datasets used for semantic segmentation of urban scenes, which was created specifically to address the fact that previous data did not adequately capture the complexity of real-world cities. The Cityscapes dataset consists of a

large number of different stereoscopic video sequences recorded on streets in 50 different cities in and around Germany. The dataset consists of 5,000 high-quality finely labeled images and 20,000 coarsely labeled images, which are used to help implement semantic segmentation methods that utilize large amounts of weakly labeled data. The finely labeled images and roughly labeled images in the dataset are shown. Critically, Cityscapes surpasses previous attempts at dataset size, annotation richness, scene variability and complexity.Cityscapes' data recording and annotation methods were carefully designed to capture the high variability of outdoor street scenes. The camera system and post-processing capabilities of the dataset reflect the current state-of-the-art in the automotive field.

The number of pixels per fine annotation class (y-axis) and their associated categories (x-axis) are shown in Fig. 1.The Cityscapes dataset defines 29 visual classes for annotations, which can be categorized into eight main categories: planar, architectural, natural, vehicular, sky, object, human, and blank label. Based on their frequency, relevance from an application perspective and practical considerations regarding annotation work, as well as promoting compatibility with existing datasets, among other reasons. A smaller number of categories were excluded from the benchmark, using the remaining 19 categories for the evaluation.

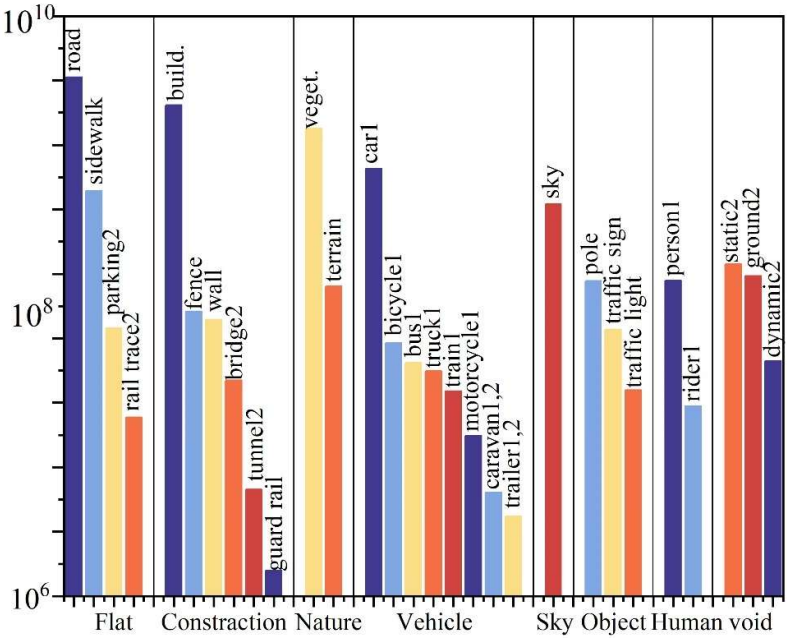

Fig. 1. Each detail annotation class (y axis) and its related categories (x axis)

Cityscapes has up to two orders of magnitude more annotated pixels compared to CamVid, DUS and KITTI.Various dataset pairs are shown in Table 1The majority of all annotated pixels in Cityscapes are coarse annotations, providing many separate but related training samples but lacking information close to the object boundaries.Cityscapes(fine) represents the 5000 finely annotated images in the dataset and cityscapes(coarse) represents the 20,000 roughly annotated additional maps in the dataset and the images in the CamVid dataset were scaled up to 1290 × 740 pixels to maintain the original aspect ratio.

Table 1. Various data sets are compared

Data set name	The number of pixels (109)	Data tag density (%)
DUS	0.16	63.22
KITTI	0.24	89.25
CamVid	0.67	97.84

Cityscapes(fine)	9.58	97.3
Citescapes(coarse)	26.14	68.6

The number of annotations per category in each dataset was compared as shown in Figure 2. It is worth noting that the differences in the datasets stem from inherently different configurations; Cityscapes involve dense downtown traffic, wide roads, and large intersections, while KITTI consists of less busy suburban traffic scenes. As a result, KITTI shows significantly fewer flat structures and humans and more natural scenes. In terms of overall construction, DUS and CamVid are closer to the construction of the Cityscapes dataset. However, the CamVid dataset has more sky information and the DUS dataset has more blank label information. After the above analysis, it is obvious that the Cityscapes dataset exceeds the previous datasets in all aspects, and it is the current dataset that can approximate the real-world urban scene situation. Therefore, this dataset is selected as the training and validation test dataset in the training and testing of this paper.

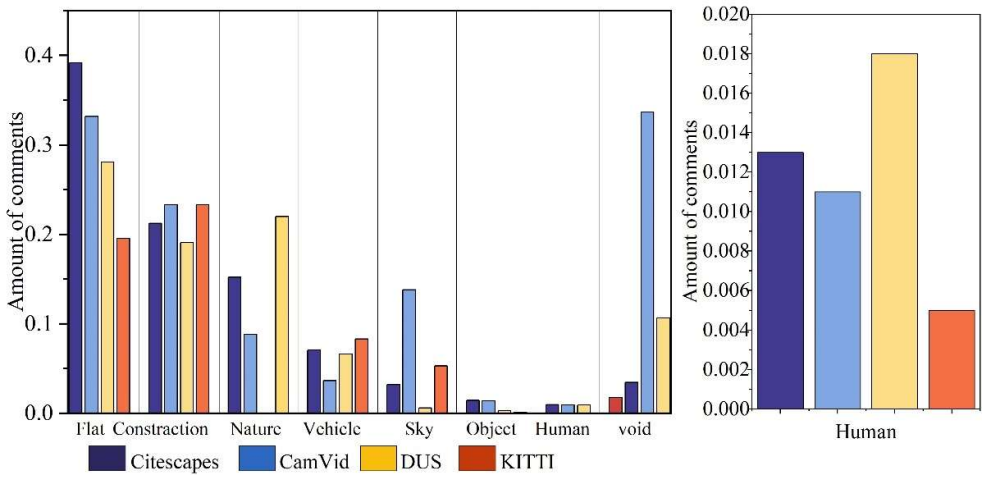

Fig. 2. The proportion of annotated pixels

4.2. Introduction to the experimental environment

The software version as well as the hardware configuration of the network model implementation in this paper are shown in Table 2.

Table 2. Experimental software version and hardware configuration

Name	Version or type
Processor	Intel Xeon E5-1620 v3
Memory	32G
Graphics card	TITAN X(12G)
Hard disk	256SSD+2T
System	Ubuntu 16.04
Language	Python 3.6.7
Compiling environment	PyCharm 2017.3
Frame	PyTorch 0.4.1

4.3. Experimental results and analysis

The comparative experimental results of each algorithm on the Cityscapes dataset are shown in Figure 3. Compared with the recent state-of-the-art segmentation algorithms such as Deeplabv3+, PSANet and MCNet, the mIoU of the improved DeepLabv3+ model is improved by 4.128%, 4.81% and 2.345%, respectively, which indicates that this paper's method is more suitable for thermal infrared image segmentation. The method in this paper not only has a slightly higher overall segmentation performance mIoU than other methods, but also has a slightly higher segmentation performance for targets such as small targets and edges, for example, compared with Deeplabv3+, the method in this paper improves the IoU in the category of small targets such as pedestrians, cyclists, and columns by 3.1%, 4.41%, and 6.74%, respectively. Compared to MCNet, this paper's method improves IoU by 1.54%, 2.18%, 2.58%, 7.8%, 2.6%, 0.88%, 2.14%, and 3.1% in the categories of pedestrians, cyclists, cars, trunks, railings, trees, buses, and columns, respectively. The segmentation results of the above part of the algorithm are visualized, and through observation and analysis, it is found that the algorithms in this chapter have slightly better segmentation effects on each category, for example, the segmentation algorithm in this paper is better than Deeplabv3+ segmentation algorithm in terms of details such as small targets and edges, and has a clearer effect on the edge processing of the target structure. More specifically, this paper's algorithm in the front and rearview mirrors of buses, pedestrians, upper and lower limbs, tree trunks and branches, columns and other details of the segmentation effect is significantly better than Deeplabv3+ and PSPNet algorithms segmentation effect.

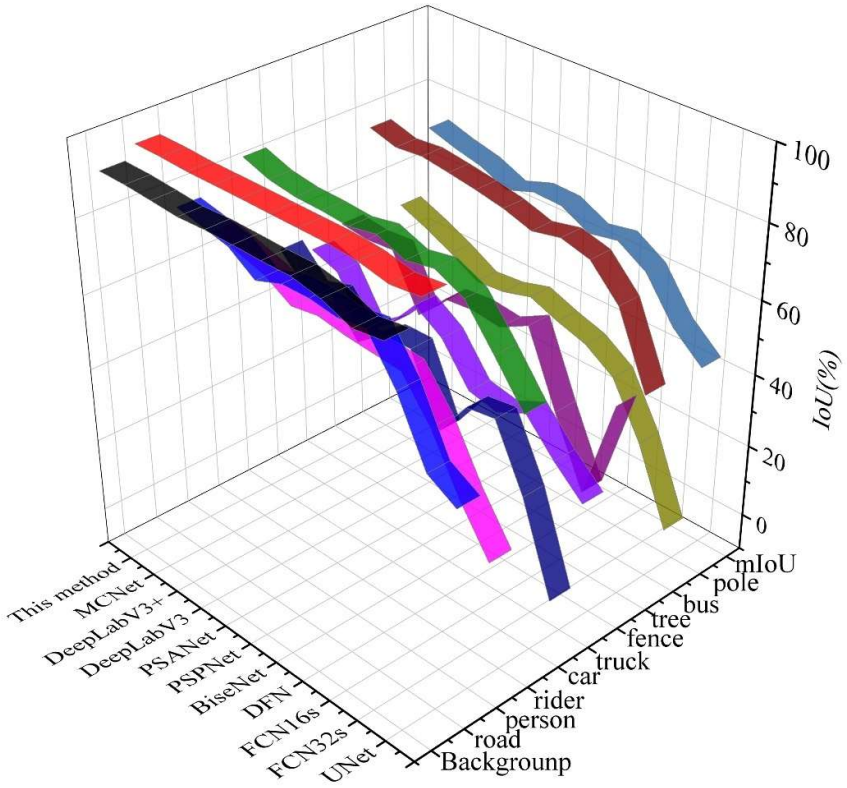

Fig. 3. Comparison of segmentation performance of popular algorithm

The performance improvement of the backbone network for the ablation experiment is shown in Table 3. With Xception as the backbone network, the mIoU of the overall network is improved by 4.9% over the baseline model when the method of this paper is embedded in ASPP. With MobileNetV2 as the backbone network, the mIoU of the overall network is improved by 4.91% over the baseline model. The MobileNetV2-based model outperforms Xception's model by only 1.00%, indicating that when the network reaches a certain depth, the network performance is not proportional to the depth of the network.

Table 3. The performance of the main dry network is improved

Method	Backbone	mIoU(%)	Δ(%)
DenseNet	Xception	65.12	—
DenseNet+DenseASPP	Xception	70.02	—
DenseNet+DenseASPP+ CBAM	Xception	71.14	1.12 ↑
DenseNet+DenseASPP	MobileNetV2	70.03	—
DenseNet+DenseASPP+ CBAM	MobileNetV2	72.14	2.11 ↑

The loss function variation curve of the improved lightweight DeepLabV3+ model network when validating the Citycapes dataset is shown in Fig. 4, and it can be found that the model loss values converge after about 100 iterations.

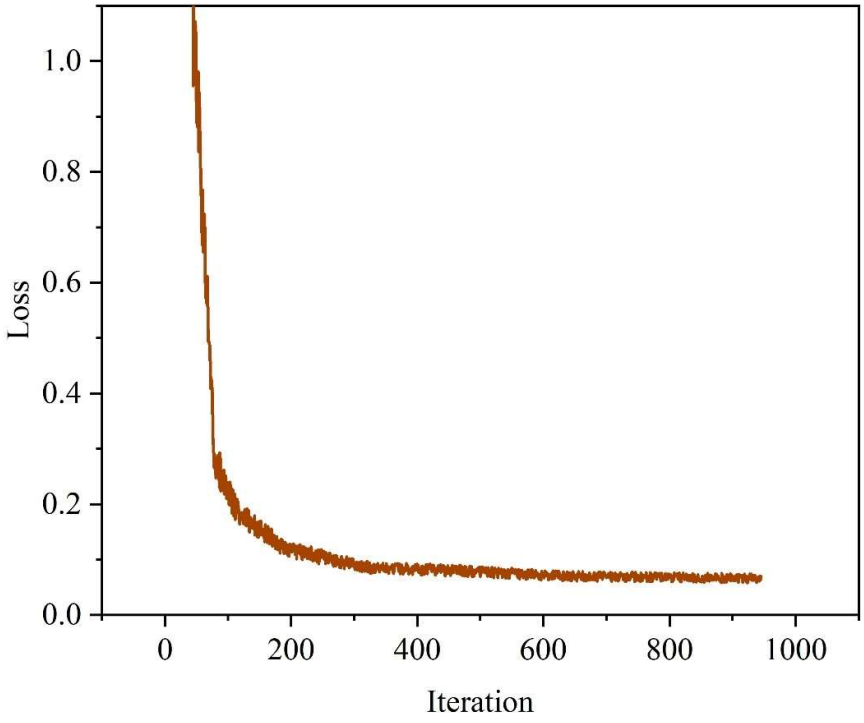

Fig. 4. Loss function diagram

In order to obtain better experimental results, a stronger network (MobileNetV2) was chosen to improve the performance of this model. The best segmentation performance of this model on Citycapes validation set reaches 72.14%. The performance comparison of this paper's method with the most relevant current methods is shown in Table 4. The three methods that are most similar to the method of this paper are selected including: DCN, PSANet, and MCNet.The module performance of this paper is optimal with an improvement of 1.92%, 1.98%, and 3.94%, respectively.

Table 4. Experiment with the most similar method

Method	mIoU(%)	Δ(%)	Imput size	GFLOPs	param(M)
FCN	60.45	—	512×512	278.41	68.54
DCN	70.22	—	512×512	—	—
GCN	70.25	—	512×512	278.41	68.71
PSANet	70.16	—	512×512	279.56	78.22
MCNet	68.20	—	582×725	178.94	35.78
Ours	72.14	3.94 ↑	512×512	255.53	63.78

The ablation experiments for each module designed in this paper are shown in Table 5. The ablation experiments are performed for each module using the baseline model with the backbone network as Xception and MobileNetV2, respectively. When Xception is used as the backbone network, the segmentation performance of the proposed DenseASPP module and CBAM module, for the baseline model, are both slightly improved by about 0.54% and 1.13%, respectively. When the DenseASPP, CBAM and GD-FAM modules are embedded simultaneously, the segmentation performance is improved by about 2.39%. When embedding DenseASPP, CBAM and GD-FAM modules in the baseline model where the backbone network is MobileNetV2, its segmentation performance improves by about 2.92%. It shows that the method in this paper is indeed effective.

Table 5. Ablation of various components

Method	Backbone	mIoU(%)	Δ(%)	GFLOPs
Traditional deeplabv3+	Xception	45.2	—	—
DenseNet	Xception	67.12	—	182.36
DenseNet+DenseASPP	Xception	67.66	0.54 ↑	177.78
DenseNet+DenseASPP+ CBAM	Xception	68.79	1.13 ↑	177.84
DenseNet+DenseASPP+ CBAM +GD-FAM	Xception	70.05	2.39 ↑	177.94
DenseNet	MobileNetV2	68.45	—	254.97
DenseNet+DenseASPP	MobileNetV2	69.94	1.49 ↑	255.42
DenseNet+DenseASPP+ CBAM	MobileNetV2	70.72	2.27 ↑	256.78
DenseNet+DenseASPP+ CBAM +GD-FAM	MobileNetV2	71.37	2.92 ↑	25563

Different loss function pairs were subjected to ablation experiments as shown in Table 6. It is found that if only the edge loss is used to supervise it, its effect is not very obvious, and the segmentation performance is only improved by 0.6%, and although the edge feature enhancement can remove the edge noise to enhance the edge segmentation effect, the number of edge pixels accounts for a smaller number and the edge pixel category cannot be supervised, which leads to a slightly worse segmentation effect. However, considering the edge pixel category information and supervising it, the segmentation performance is improved significantly. However, the percentage of edge pixels is relatively small, so the overall segmentation performance improvement is not very obvious. Finally, the small target features are supervised and it is found that the OHEM loss supervises its training, which is slightly better than the general symplectic entropy loss function.

Table 6. Ablation experiment based on different loss functions

Method	Lseg	Le-bce	Le-ohem	mIoU(%)	Δ(%)
DenseNet	✓	—	—	71.3	—
—	—	✓	—	71.9	0.6 ↑
—	✓	✓	—	72.2	0.9 ↑
—	✓	—	✓	72.3	1 ↑

5. Conclusion

In this paper, the traditional DeepLabV3+ image segmentation model is mainly improved by using innovative methods such as introducing CBAM attention mechanism, replacing the backbone network structure, incorporating densely connected ASPP with band-pooling, and fusing GD-FAM modules. It is examined in Cityscapes dataset, and the results show that the lightweight DeepLabV3+ segmentation model improved in this paper is better than the traditional Deeplabv3+ segmentation model in terms of details such as small targets and edges, such as front and rearview mirrors of buses, upper and lower limbs of pedestrians, branches of tree trunks, and columns, etc., and the edge processing effect is more clear. The loss value of the improved model in this paper tends to converge after 100 iterations. In the ablation experiments carried out in each module, embedding DenseASPP, CBAM and GD-FAM modules, the segmentation performance is better than the baseline model, and its overall network mIoU value is improved by about 2.92%.

References

[1] Ibrahim, M. R., Haworth, J., & Cheng, T. (2020). Understanding cities with machine eyes: A review of deep computer vision in urban analytics. Cities, 96, 102481.

[2] Zhou, W., Liu, J., Lei, J., Yu, L., & Hwang, J. N. (2021). GMNet: Graded-feature multilabel-learning network

for RGB-thermal urban scene semantic segmentation. IEEE Transactions on Image Processing, 30, 7790-7802.

[3] Yang, Z., Yu, H., Feng, M., Sun, W., Lin, X., Sun, M., ... & Mian, A. (2020). Small object augmentation of urban scenes for real-time semantic segmentation. IEEE Transactions on Image Processing, 29, 5175-5190.

[4] Elngar, A. A., Arafa, M., Fathy, A., Moustafa, B., Mahmoud, O., Shaban, M., & Fawzy, N. (2021). Image classification based on CNN: a survey. Journal of Cybersecurity and Information Management, 6(1), 18-50.

[5] He, X., & Peng, Y. (2017). Fine-grained image classification via combining vision and language. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 5994-6002).

[6] Gupta, A. K., Seal, A., Prasad, M., & Khanna, P. (2020). Salient object detection techniques in computer vision—A survey. Entropy, 22(10), 1174.

[7] Pathak, A. R., Pandey, M., & Rautaray, S. (2018). Application of deep learning for object detection. Procedia computer science, 132, 1706-1717.

[8] Wiley, V., & Lucas, T. (2018). Computer vision and image processing: a paper review. International journal of artificial intelligence research, 2(1), 29-36.

[9] Saini, S., & Arora, K. (2014). A study analysis on the different image segmentation techniques. International Journal of Information & Computation Technology, 4(14), 1445-1452.

[10] Dong, G., Yan, Y., Shen, C., & Wang, H. (2020). Real-time high-performance semantic image segmentation of urban street scenes. IEEE Transactions on Intelligent Transportation Systems, 22(6), 3258-3274.

[11] Arulananth, T. S., Kuppusamy, P. G., Ayyasamy, R. K., Alhashmi, S. M., Mahalakshmi, M., Vasanth, K., & Chinnasamy, P. (2024). Semantic segmentation of urban environments: Leveraging U-Net deep learning model for cityscape image analysis. Plos one, 19(4), e0300767.

[12] Huerta, R. E., Yépez, F. D., Lozano-García, D. F., Guerra Cobian, V. H., Ferrino Fierro, A. L., de León Gómez, H., ... & Vargas-Martínez, A. (2021). Mapping urban green spaces at the metropolitan level using very high resolution satellite imagery and deep learning techniques for semantic segmentation. Remote Sensing, 13(11), 2031.

[13] Pan, Z., Xu, J., Guo, Y., Hu, Y., & Wang, G. (2020). Deep learning segmentation and classification for urban village using a worldview satellite image based on U-Net. Remote Sensing, 12(10), 1574.

[14] Zou, Y., Weinacker, H., & Koch, B. (2021). Towards urban scene semantic segmentation with deep learning from LiDAR point clouds: a case study in Baden-Württemberg, Germany. Remote Sensing, 13(16), 3220.

[15] Ibrahim, M. R., Haworth, J., & Cheng, T. (2021). URBAN-i: From urban scenes to mapping slums, transport modes, and pedestrians in cities using deep learning and computer vision. Environment and Planning B: Urban Analytics and City Science, 48(1), 76-93.

[16] Neupane, B., Horanont, T., & Aryal, J. (2021). Deep learning-based semantic segmentation of urban features in satellite images: A review and meta-analysis. Remote Sensing, 13(4), 808.

[17] Ankareddy, R., & Delhibabu, R. (2025). Dense Segmentation Techniques Using Deep Learning for Urban Scene Parsing: A Review. IEEE Access.

[18] Kim, B., & Ye, J. C. (2019). Mumford–Shah loss functional for image segmentation with deep learning. IEEE Transactions on Image Processing, 29, 1856-1866.

[19] Liu, X., Deng, Z., & Yang, Y. (2019). Recent progress in semantic image segmentation. Artificial Intelligence Review, 52, 1089-1106.

[20] Yu, Y., Wang, C., Fu, Q., Kou, R., Huang, F., Yang, B., ... & Gao, M. (2023). Techniques and challenges of image

segmentation: A review. Electronics, 12(5), 1199.

[21] Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N., & Terzopoulos, D. (2021). Image segmentation using deep learning: A survey. IEEE transactions on pattern analysis and machine intelligence, 44(7), 3523-3542.

[22] Ghosh, S., Das, N., Das, I., & Maulik, U. (2019). Understanding deep learning techniques for image segmentation. ACM computing surveys (CSUR), 52(4), 1-35.

[23] Chouhan, S. S., Kaul, A., & Singh, U. P. (2019). Image segmentation using computational intelligence techniques. Archives of Computational Methods in Engineering, 26, 533-596.

[24] Luo, P., Wang, G., Lin, L., & Wang, X. (2017). Deep dual learning for semantic image segmentation. In Proceedings of the IEEE international conference on computer vision (pp. 2718-2726).

[25] Chen, C., & Zhang, H. (2023). Attention Block Based on Binary Pooling. Applied Sciences, 13(18), 10012.

[26] Yuan, Z. (2024). SPCANet: congested crowd counting via strip pooling combined attention network. PeerJ Computer Science, 10, e2273.

[27] Sakthipriya, G., Padmapriya, N., & Venkateswaran, N. (2024). TSDAnet: texture strip dual attention network for intraclass texture classification. Signal, Image and Video Processing, 18(11), 7597-7610.

[28] Chen Qi,Zhang Zhenxin,Chen Siyun,Wen Siyuan,Ma Hao & Xu Zhihua. (2022) .A self-attention based global feature enhancing network for semantic segmentation of large-scale urban street-level point clouds. International Journal of Applied Earth Observation and Geoinformation,113.

[29] Peng Chen,Wenxuan He,Feng Qian,Guangyao Shi & Jingwen Yan. (2025) .A synergistic CNN-transformer network with pooling attention fusion for hyperspectral image classification. Digital Signal Processing,160,105070-105070.

[30] Arpan Mahara,Md Rezaul Karim Khan,Liangdong Deng,Naphtali Rishe,Wenjia Wang & Seyed Masoud Sadjadi. (2025) .Automated Road Extraction from Satellite Imagery Integrating Dense Depthwise Dilated Separable Spatial Pyramid Pooling with DeepLabV3+. Applied Sciences,15(3),1027-1027.