LM-DeeplabV3+: A Lightweight Image Segmentation Algorithm Based on Multi-scale Feature Interaction

Xinyu Hou , Peng Chen * , Haishuo Gu

Posted Date: 15 January 2024

doi: 10.20944/preprints202401.1052.v1

Keywords: street view images; semantic segmentation; DeeplabV3+; attention mechanism; loss function

Article

LM-DeeplabV3+: A Lightweight Image Segmentation Algorithm Based on Multi-Scale Feature Interaction

Xinyu Hou, Peng Chen * and Haishuo Gu

People's Public Security University of China, Beijing 100038, China; 2022211511@stu.ppsuc.edu.cn
* Correspondence: chenpeng@ppsuc.edu.cn

Abstract: Street view images can help us better understand the city environment and potential characteristics. With the development of computer vision and deep learning, the technology of semantic segmentation algorithms has become more mature. However, DeeplabV3+, which is commonly used in semantic segmentation, has shortcomings such as a large number of parameters, high requirements for computing resources, and easy loss of detailed information. Therefore, this paper proposes LM-DeeplabV3+, which aims to greatly reduce parameters and computations of the model while ensuring the segmentation accuracy. Firstly, the lightweight network MobileNetV2 is selected as the backbone network, and the ECA attention mechanism is introduced after MobileNetV2 extracts shallow features to improve the ability of feature representation; secondly, the ASPP module is improved, and on its basis, the EPSA attention mechanism is introduced to achieve cross-dimensional channel attention and important feature interaction; thirdly, a loss function named CL loss is designed to balance the training offset of multiple categories and better indicate the segmentation quality. This paper conducted experimental verification on the Cityspaces dataset, and the results showed that the mIoU reached 74.9%, which was an improvement of 3.56% compared to DeeplabV3+; the mPA reached 83.01%, which was an improvement of 2.53% compared to DeeplabV3+.

Keywords: street view images; semantic segmentation; DeeplabV3+; attention mechanism; loss function

1. Introduction

The continuous development of deep learning and street view image provides new perspectives for city feature recognition. The element extraction of street view environment can help us better understand the city environment and potential features, and combining street view data and multi-dimensional feature recognition technology provides refined technical support for city design [1]. With the development of deep learning and computer vision, the technology of extracting elements from street scenes based on semantic segmentation algorithms is becoming more mature. Street view images have the advantages of wide coverage, providing street level landscape information, and low data collection costs, providing a large sample data source and new research ideas for city environmental assessment research [2]. Street view images contain a large amount of visual information about city space, including static city architectural environments and dynamic pedestrians or vehicles on the streets, providing new perspectives and dimensions for city spatial environment analysis [3]. Researchers have extracted elements from street view images and conducted some research related to city perception and safety [4-6].

In recent years, computer vision and deep learning have brought many new opportunities to the field of geographic space [7]. So far, many semantic segmentation algorithms have been widely applied in different fields. From the perspective of model structure, semantic segmentation algorithms mainly have two categories: classification based on information fusion and classification based on encoder-decoder [8]. The method based on information fusion improves the utilization of the model by increasing the number of layers in the network [9]. Representative algorithms include the fully convolutional network (FCN) algorithm [10] and a series of improved algorithms [11], such as FCN-32S, FCN-16S, and FCN-8S. The method based on encoding and decoding [12] improves the accuracy of the network by using different backbone network forms and pyramid pooling modules.

 doi:10.20944/preprints202401.1052.v1

Representative algorithms include the Pyramid Scene Parsing Network (PSPNet) [13] and DeepLabV3+ [14].

With the emergence of more and more image semantic segmentation algorithms, the challenges to computer computing resources and hardware memory requirements are gradually increasing. The Deeplab [15] series is a series of semantic segmentation algorithms developed by the Google team based on FCN. From 2014 to 2018, the Deeplab series released four versions, namely v1, v2, v3, and v3+. The most influential change of DeepLabV3+ is the use of a new backbone network Xception [16], and the addition of an upsampling decoder module modeled after U-Net [17]. The original DeepLabV3 is used as an encoder, and the addition of a decoder results in a new model with better boundary segmentation performance. DeepLabV3+ also integrates the Spatial Pyramid Pooling (SPP) [18] and encoder-decoder into one. DeepLabV3+ refers to a very common feature fusion strategy in object detection, which preserves a considerable amount of shallow information in the network. At the same time, it replaces depthwise separable convolutions with dilated depthwise separable convolutions and proposes Atrous Spatial Pyramid Pooling (ASPP), which can achieve faster speed while ensuring effectiveness.

However, the DeeplabV3+ algorithm has high computational complexity and high memory consumption, making it difficult to deploy on platforms with limited computing power. Moreover, DeeplabV3+ cannot fully utilize multi-scale information when extracting image feature information, which can easily lead to the loss of detail information and affect segmentation accuracy. For some application scenarios that require real-time feedback, such as autonomous driving or real-time monitoring, the limitation of computing resources makes it difficult to complete segmentation tasks in a timely manner. Under limited computing resources, semantic segmentation models cannot achieve good segmentation results. However, many existing lightweight segmentation models (such as BiseNetv2 [19]) use detail and semantic branches to balance low-level and high-level semantic information. Although this model effectively reduces the number of network parameters, its segmentation accuracy is not optimistic. In 2017, Google proposed the MobileNet network, which is a representative of lightweight networks aimed at significantly reducing model size and accelerating model computation speed while sacrificing model performance. There are currently three versions of MobileNet, namely MobileNetV1 [20], MobileNetV2 [21], and MobileNetV3 [22]. MobileNetV2 performs well in semantic segmentation tasks. As a lightweight convolutional neural network, it maintains high accuracy while having fewer parameters and lower computational complexity.

In this regard, this paper proposes an improved algorithm based on the mainstream semantic segmentation model DeeplabV3+ to solve the problem of high model resource consumption, while taking into account segmentation accuracy and obtaining key category information. The main contributions of this paper are summarized as follows:

(1) On the basis of using lightweight network MobileNetV2 as the backbone network to alleviate network architecture, ECA attention mechanism is introduced after extracting shallow features in MobileNetV2, which improves feature representation ability without damaging network structure.

(2) Improvements made to the ASPP module: replacing ASPP Pooling with Strip Pooling effectively expands the receptive field range of the network, making it suitable for more complex scenarios; Furthermore, the EPSA attention module has been introduced to effectively establish long-term dependencies between multi-scale channel attention and achieve cross dimensional channel attention interaction of important features.

(3) CE loss may have the problem of training results shifting towards a larger number of categories when faced with class imbalance, and the measurement of CE loss usually cannot indicate segmentation quality well on the validation set. Therefore, we have designed a loss that combines CE loss and Lovasz loss and named it CL Loss.

This paper validates the effectiveness of the method on the Cityspaces dataset and trains new pre training weights that are more suitable for semantic segmentation of city street view images.

2. Materials and Methods

2.1. The overall framework of LM-deeplabV3+

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 January 2024 doi:10.20944/preprints202401.1052.v1

LM-deeplabV3+ is based on the lightweight MobileNetV2 backbone network. Due to the use of pre trained weights in MobilenetV2, in order to improve image segmentation performance without damaging the original network structure, shallow features were extracted and ECA attention mechanism was incorporated into MobilenetV2. Improvements have been made to the ASPP module by using Strip pooling instead of ASPP pooling, which effectively expands the receptive field range of the network and can be applied to more complex scenarios. The EPSA attention module has also been introduced, effectively establishing long-term dependencies between multi-scale channel attention and achieving cross dimensional channel attention important feature interaction. Considering the problem of imbalanced weight allocation among multiple categories and the fact that its metrics on the validation set often cannot effectively indicate the quality of segmentation in the original CE loss, as well as the instability of the training process in Lovasz loss, we designed CL loss, which takes into account the stability and effect of the loss function.

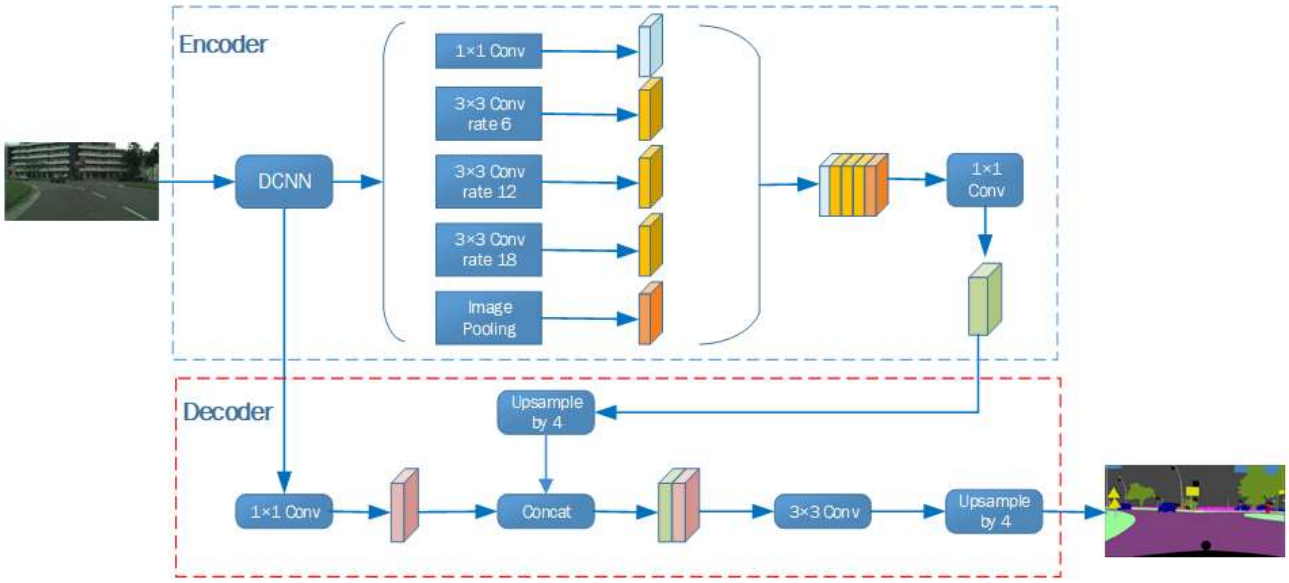


Figure 1. The structure of DeeplaV3+.

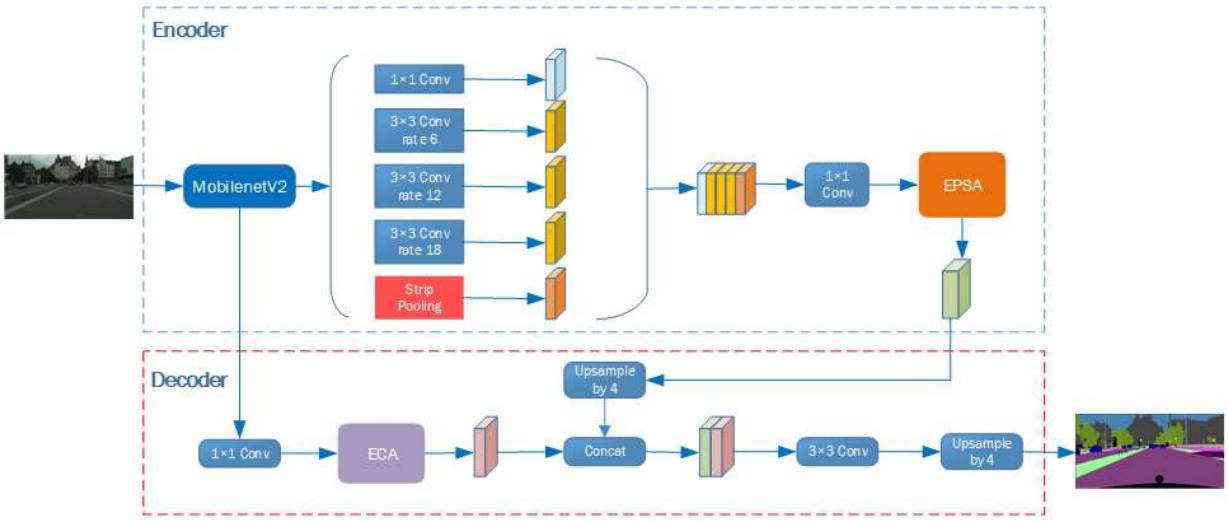


Figure 2. The structure of LM-DeeplaV3+.

2.2. Strip pooling

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 January 2024 doi:10.20944/preprints202401.1052.v1

For semantic segmentation tasks, receptive field size and full-text contextual information are crucial for the final prediction results. Strip pooling [23] can effectively expand the receptive field range of the backbone network, and is a long strip-shaped pooling kernel deployed along the spatial dimension. Parallel connection of Strip pooling in the ASPP module can enhance its ability to perceive remote context.

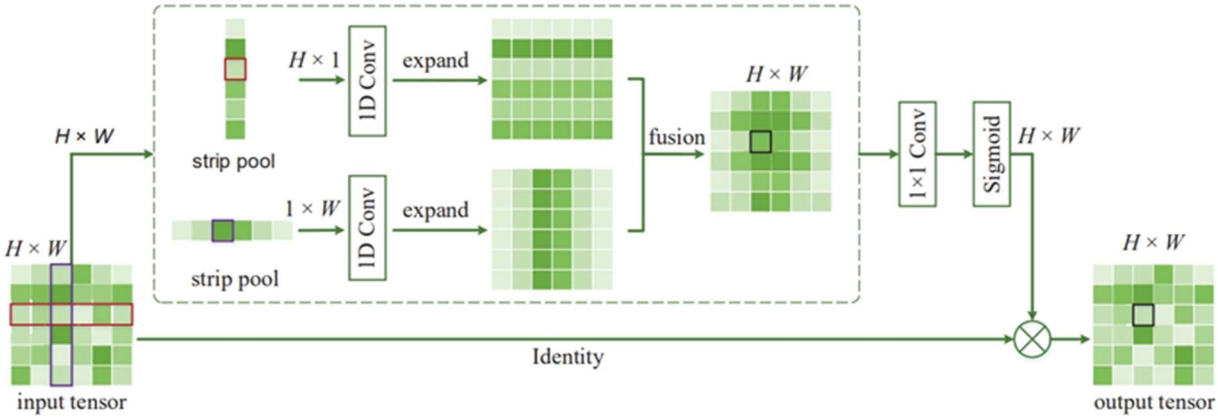


Figure 3. The structure of Strip pooling.

In Strip pooling, the current feature map is divided into several strip regions according to certain rules, and each strip pooling window is pooled along the horizontal or vertical dimension. When two spatial dimensions are pooled, the feature values of columns or rows are weighted average.

For the input image, the calculation formula for row vector output is as follows:

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

The calculation formula for column vector output is as follows:

$$y_i^V = \frac{1}{H} \sum_{0 \le j < H} x_{i,j} \tag{2}$$

In the case of horizontal and vertical banded pooling layers, due to the long and narrow shape of the kernel, it is easy to establish long-term dependencies between discretely distributed regions and encode them in a banded shape. Meanwhile, due to its narrow kernel shape in another dimension, it also focuses on capturing local details. These characteristics make the proposed strip pooling different from traditional spatial pooling that relies on square kernels.

Let $x \in R^{C \times H \times W}$ be an input tensor, where C, H, and W respectively represent the number of channels, height, and width. We feed x into two parallel pathways, each of which contains a horizontal or vertical strip pooling layer. The vertical and horizontal outputs are $y^h \in R^{C \times H}$ and $y^v \in R^{C \times W}$ respectively. After combining the two, the output is as follows, yielding $y \in R^{C \times H \times W}$:

$$y_{c,i,j} = y_{c,i}^h + y_{c,j}^v \tag{3}$$

Then, the output z is computed as:

$$z = Scale(X, \sigma(f(y))) \tag{4}$$

where $Scale()$ represents multiplication, σ Represents the sigmoid function and f represents 1×1 Convolution.

2.3. EPSANet and ECANet

2.3.1. EPSANet

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 January 2024 doi:10.20944/preprints202401.1052.v1

The Efficient Pyramid Split Attention (EPSA) [24] module can handle the spatial information of multi-scale input feature maps and effectively establish long-term dependencies between multi-scale channel attention, achieving cross dimensional channel attention interaction of important features, providing stronger multi-scale feature expression and serving semantic segmentation tasks.

The implementation of EPSA mainly consists of four steps:

(1) Implement the proposed Split and Concat (SPC) module. SPC first divides the input tensor into S groups, and the convolution kernel size K in each group increases sequentially. Considering that when K is relatively large, the computational workload will also be higher. Therefore, each group is further grouped and convolved, with a specific number of groups $G = 2^{\frac{K-1}{2}}$. After undergoing convolutions of different sizes, concatenate them on the channel. The operation of the SPC module is shown in Figure 4.

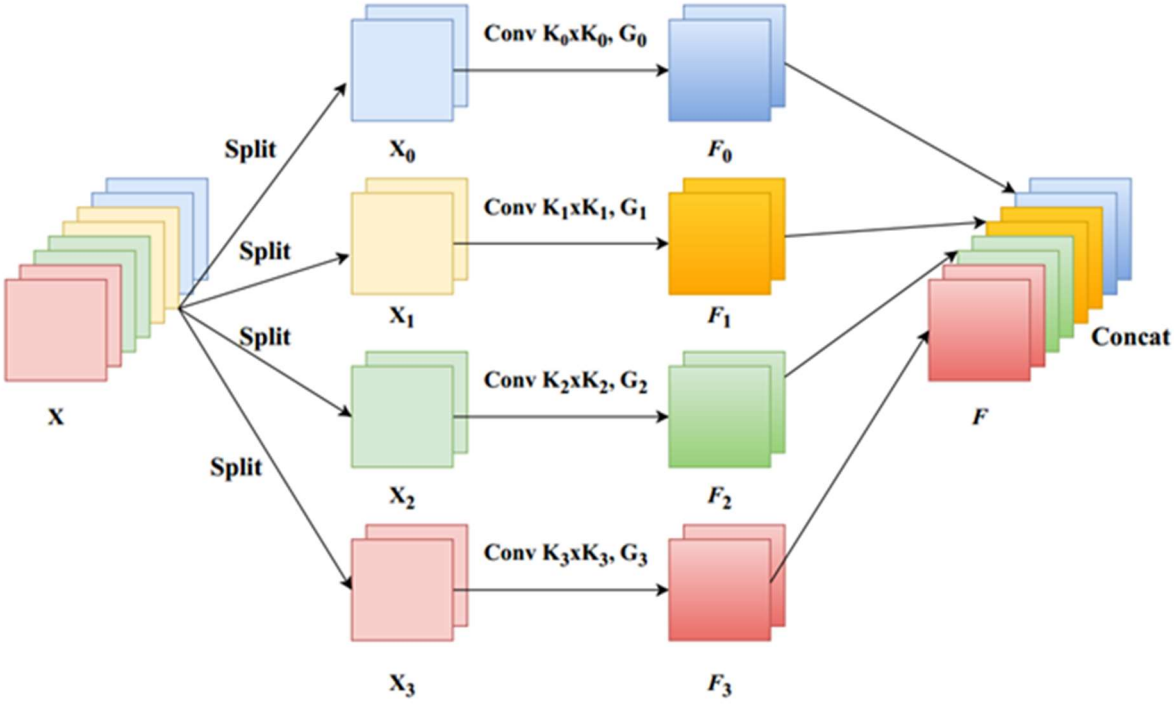


Figure 4. Illustration of SPC module when S=4.

where "Split" represents uniform segmentation in the channel dimension and "Concat" represents connecting features in the channel dimension.

(2) Extracting attention from feature maps of different scales using the SEWeight[25] module to obtain channel attention vectors. The SEWeight is shown in Figure 5.

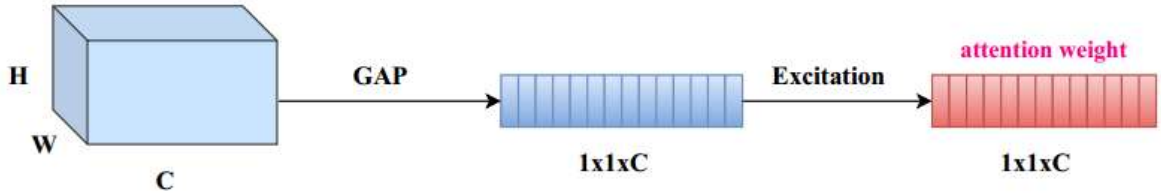


Figure 5. The structure of SEWeight.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 January 2024 doi:10.20944/preprints202401.1052.v1

(3) Use softmax to recalibrate the attention vectors of each channel and obtain recalibrated weights for multi-scale channels.

(4) Perform a product operation on the recalibrated weights and corresponding feature map application elements.

Finally, a refined feature map with richer multi-scale feature information is obtained as the output. The complete EPSA module is shown in Figure 6.

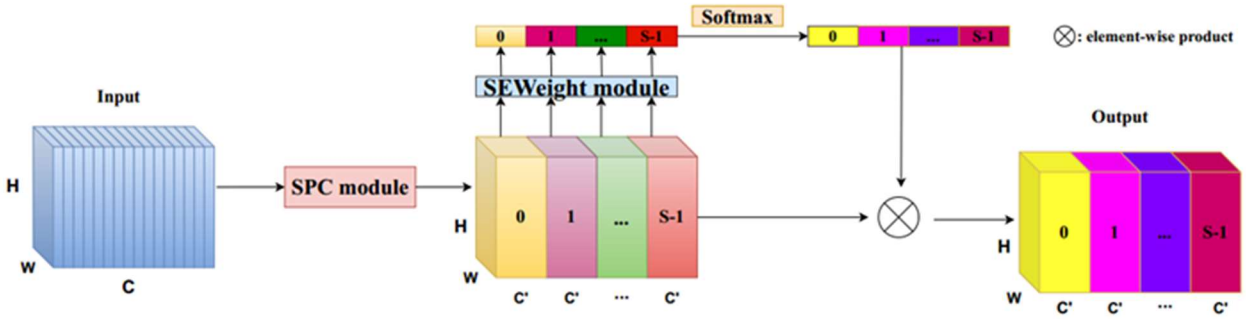


Figure 6. The structure of EPSA module.

The EPSA module can integrate multi-scale spatial information and cross channel attention into the blocks of each separated feature group, achieving better information interaction between local and global channel attention.

2.3.2. ECANet

ECANet [26] is an efficient neural network module that effectively captures channel relationships in images and enhances feature representation by introducing channel attention mechanisms. ECANet mainly consists of three parts. Firstly, using global pooling to transform the spatial matrix into a one-dimensional vector, a feature map of size $1 \times 1 \times C$ is obtained. Secondly, the convolution kernel size k is adaptively determined based on the number of network channels:

$$k = \psi(C) = \left| \frac{\log_2(c)}{\gamma} + \frac{b}{\gamma} \right|_{odd} \tag{5}$$

where $|t|_{odd}$ represents the odd number closest to t, Set a and b to 2 and 1 respectively.

Finally, an adaptive size convolution kernel is used for convolution operations, and the weights of each channel corresponding to the feature map are weighted to obtain the feature map of the input image. We multiply the normalized weights with the original input feature map channel by channel to generate a weighted feature map.

$$wx = \sigma\big(C1D_k(y)\big)x \tag{6}$$

where σ represents the sigmoid function, C1D represents one-dimensional convolution, and y represents the pooling output of x.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 January 2024 doi:10.20944/preprints202401.1052.v1

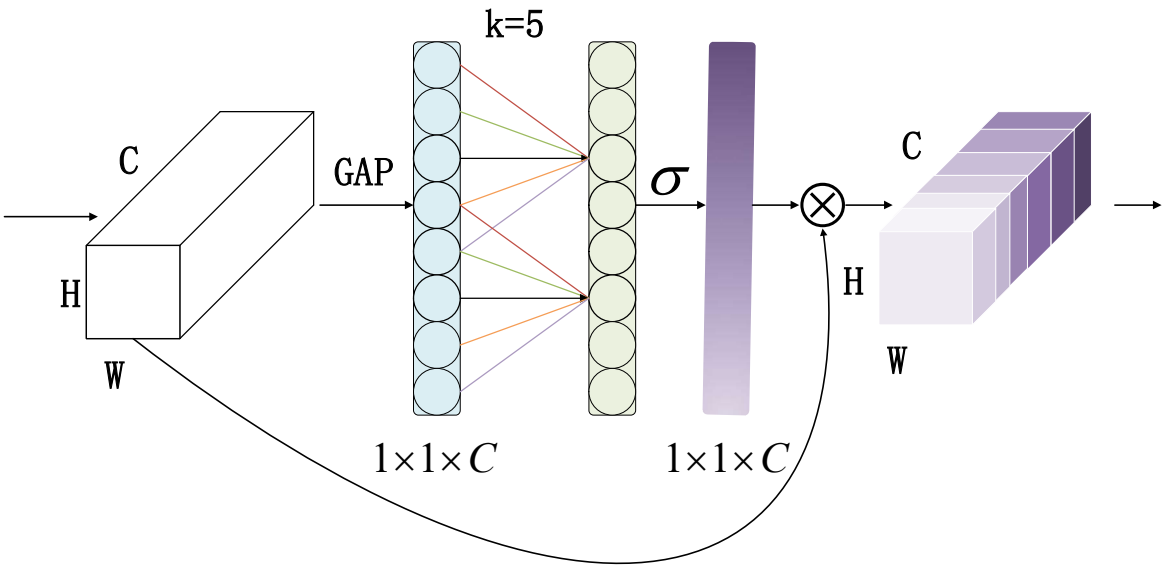


Figure 7. The structure of ECA module.

ECANet avoids dimension reduction and effectively captures cross channel interaction information. It not only enhances spatial features, making the model pay more attention to important information parts, but also enhances feature expression ability, significantly increasing the receptive field.

2.4. CL Loss

The loss function is a method of measuring the difference between the predicted and true values of a model.

2.4.1. CE Loss

The Cross Entropy Loss (CE Loss) function is a commonly used loss function in image segmentation tasks, which reflects the distance between the predicted probability distribution of the model and the true label probability distribution, and performs well in image segmentation tasks. For multi class tasks, the value of CE loss is:

$$CE(p_i, \hat{p}_i) = -\sum_{i=1}^{N} p_i \, log(\hat{p}_i) \tag{7}$$

where N is the number of categories, p_i is the true label of the *i-th* category, and $\hat{p}_i$ is the predicted distribution of the *i-th* category.

2.4.2. Lovasz Loss

Lovasz-softmax loss [27] is a loss function that directly optimizes the semantic segmentation metric mean Intersection over Union (mIoU). For the multi class segmentation problem, Lovasz uses softmax to map the model output to a probability distribution $f \in [0,1]$.

$$\text{loss}(f) = \frac{1}{N \times P} \sum_{n=1}^{N} \sum_{p=1}^{P} \left(\frac{y_p^n \hat{y}_p^n}{y_p^n + \hat{y}_p^n - y_p^n \hat{y}_p^n} \right) \tag{8}$$

where N is the total number of categories, P is the total number of pixels, y_p^n is the true label of the category n of pixel p, and $\hat{y}_p^n$ is the predicted probability of the category n of pixel p.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 January 2024 doi:10.20944/preprints202401.1052.v1

3. Results

3.1. Dataset and experimental configuration

3.1.1. Dataset

The Cityscapes dataset, also known as the City Landscape dataset, is a large-scale dataset that records street scenes in 50 different cities. This data aggregation focuses on the semantic understanding of city street scenes, with 5000 images of driving scenes in city environments and dense pixel annotations of 19 categories.

3.1.2. Experimental configuration

Our experimental environment is shown in Table 1. We use GTX3090(24G), CUDA version 11.8, Pytorch 2.0.0, and Python version 3.8.

Table 1. Experimental environment.

Name	Configuration
CPU	Intel(R) Xeon(R) Gold 6330 CPU
GPU	RTX3090（24G）
CUDA version	11.8
PyTorch	2.0.0
Python version	3.8

3.2. Evaluating indicator

In image segmentation, the mIOU and the mPA (mean Pixel Accuracy) are commonly used to measure the performance of segmentation models. The larger the values of mIoU and mPA of a model, the higher the image segmentation accuracy of the model.

The mIoU is the ratio of the intersection and union of the predicted results of the model segmentation and the actual results, expressed as:

$$mIoU = \frac{1}{k+1}\sum_{i=0}^{k}\frac{p_{ii}}{\sum_{j=0}^{k}p_{ij}+\sum_{j=0}^{k}p_{ji}-p_{ii}} \tag{9}$$

The mPA is the proportion of the total number of correctly classified pixels in each category to the total number of that category, and is averaged. The specific expression is as follows:

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

where $k+1$ represents the total number of categories, p_{ij} represents the number of pixels whose real category is i but predicted as category j, p_{ji} represents the number of pixels whose real category is j but predicted as category i, p_{ii} represents the number of pixels whose true category is i and correctly predicted as category i.

Params is the total number of parameters that need to be trained in model training, used to measure the size of the model, i.e. the computational space complexity.

Floating point operations (FLOPs) refer to the computational complexity of a model, which can be used to measure the complexity of an algorithm. However, there is controversy over using FLOPs to evaluate models, as the computational speed of the model is also related to memory throughput, etc. Therefore, this paper considers FLOPs as a secondary evaluation indicator.

3.3. Results and analysis

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 January 2024 doi:10.20944/preprints202401.1052.v1

3.3.1. Experimental parameter settings

Crop_ size is the image size, Batch_ size is the number of images processed in a single training session, Val_ batch_ size is the number of word processing images during validation, Output_ stride is the ratio of the input graph to the output graph, Lr is the learning rate, Lr_ policy is the learning rate strategy, Step_ size is the number of times the weight is updated, Iterations is the total number of iterations, and Weight_ decay is weight decay. If Batch_size is set to a larger size (such as 8 or 16), the effect will be better, but it also requires more computing resources. Considering the limited hardware resources in the experimental environment, we set Batch_ size to 4. When Output_ stride is set to 8, the segmentation accuracy slightly improves, but it requires more computational resources, so we set Output_stride to 16.

After training and comparison, we have determined the most suitable parameter settings, as shown in Table 2.

Table 2. Experimental Parameter Settings.

Parameter	Value
Crop_size	768×768
Batch_size	8
Val_batch_size	4
Output_stride	16
Lr	0.1
Lr_policy	poly
Step_size	10000
Iterations	30000
Weight_decay	0.0001

3.3.2. Comparison of different models

In order to better validate the segmentation performance of the proposed model in this article, we selected different models for comparative analysis. Table 3 shows the experimental results of different models on the Cityscapes dataset. When using the same backbone network, both mIoU and mPA of the DeeplabV3+ model is improved compared to the DeeplabV3 model. We found that the LM-DeeplabV3+ model showed significant improvement compared to the original DeeplabV3+ model with Xception as the backbone network. The mIoU and mPA increased by 3.69% and 2.53% respectively, while Params was one-fifth of the original model and FLOPs were one-quarter of the original model. Compared to LM-DeeplabV3+ with Resnet101 as the backbone network, the mIoU and mPA of LM-DeeplabV3+ decreased by 1.79% and 0.34% respectively, but Params was one eighth of the former and FLOPs was one-seventh of the former. Compared to the Deeplabv3+ model with MobileNetV2 as the backbone network, although the Params and FLOPs of LM-DeeplabV3+ slightly increased, its mIoU and mPA increased by 2.35% and 2.01% respectively.

Table 3. Comparison of Cityscapes validation set performance on different models.

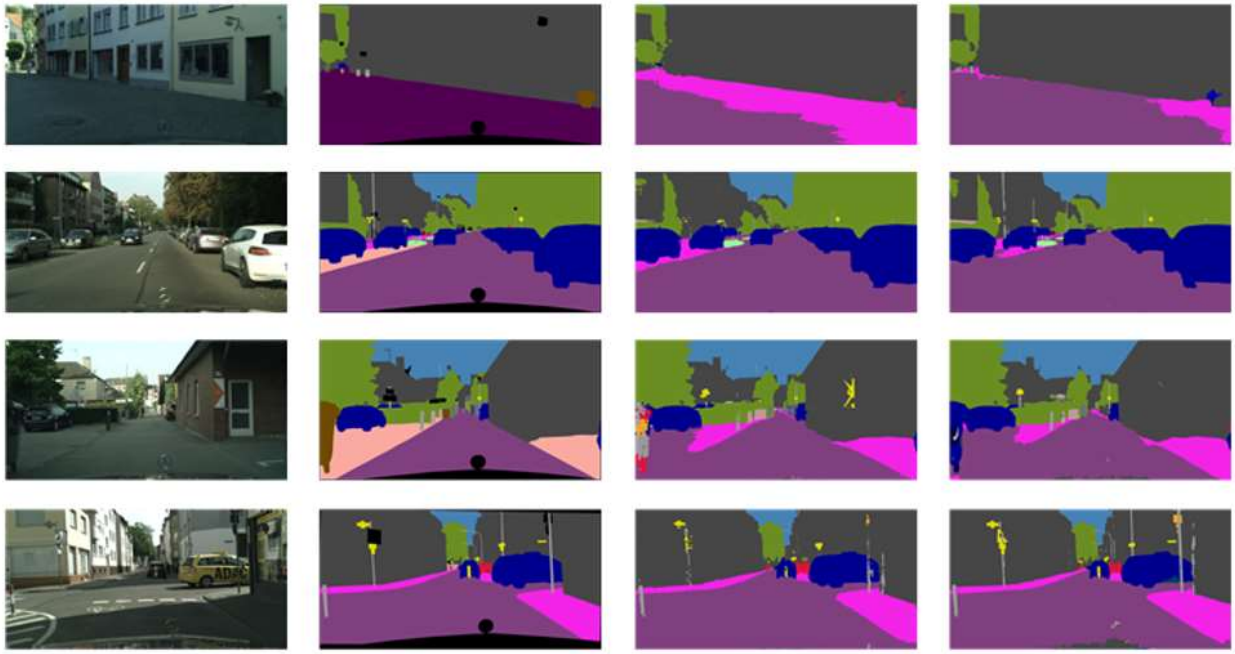
Model	Backbone	mIoU/%	mPA/%	Params/M	FLOPs/G
Deeplabv3	Xception	65.20	76.10	36.93	302.85
Deeplabv3	Resnet50	74.58	82.37	39.64	368.87
Deeplabv3	Resnet101	75.61	84.02	58.64	544.06
Deeplabv3	Mobilenetv2	71.43	79.89	5.11	48.46
Deeplabv3+	Xception	71.21	80.48	37.05	324.65
Deeplabv3+	Resnet50	75.92	83.35	39.76	389.88
Deeplabv3+	Resnet101	76.69	83.35	58.75	565.07

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 January 2024 doi:10.20944/preprints202401.1052.v1

Deeplabv3+	Mobilenetv2	72.55	81.00	5.22	69.07
LM-Deeplabv3+		**74.90**	**83.01**	**7.20**	**83.56**

[1] The size of the image is (3,768,768).

From Figure 8, we can see that LM-DeeplabV3+ has a significant improvement in segmentation performance, which is closer to the label compared to DeeplabV3+. Compared to DeeplabV3+, LM-DeeplabV3+ has more accurate pixel recognition and fewer incorrect pixel recognition.



(**a**) (**b**) (**c**) (**d**)

Figure 8. Comparison of Cityscapes dataset segmentation results.: (a) The original images of street view images. (b) The label images of street view images. (c) The segmentation effect of DeeplabV3+. (d) The segmentation effect of LM-DeeplabV3+.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 January 2024 doi:10.20944/preprints202401.1052.v1

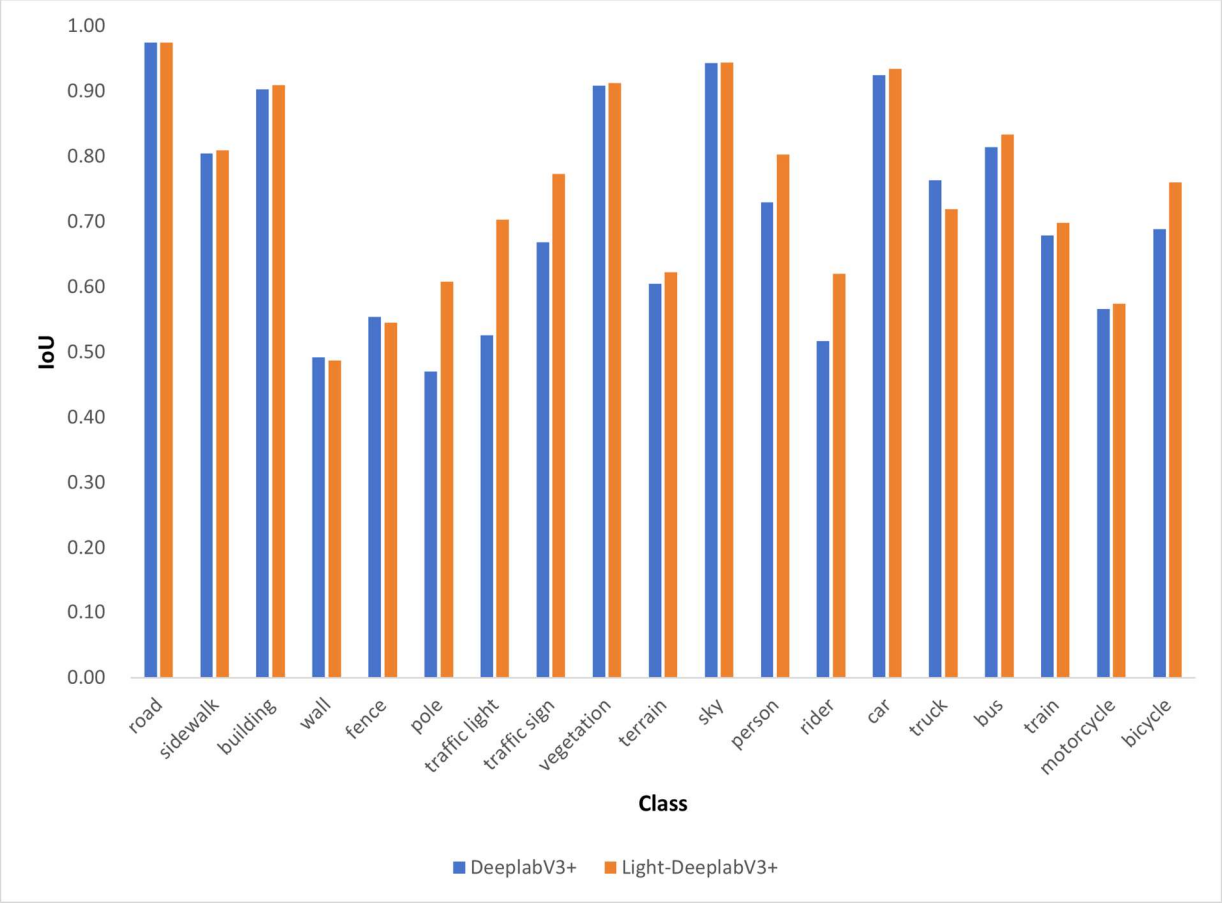


Figure 9. Comparison of IoU for Street View Elements between DeeplabV3+and LM-DeeplabV3+.

From Table 4 and Figure 8, we can find that our proposed model has improved the segmentation performance of most street view elements, with significant improvements in IoU for elements such as Pole, Traffic Light, Traffic Sign, Terrain, Person, Rider, and Bicycle, which has a significant positive impact on segmentation performance. Among them, the IoU growth of Traffic light and Pole was the highest, reaching 17.73% and 13.74% respectively. At the same time, only the IoU of Truck showed a relatively significant decrease, but still at a high level, with almost no negative impact on the overall segmentation effect.

Table 4. The IoU of each street view element on the Cityscapes validation set.

Elements.	DeeplabV3+	LM-DeeplabV3+	Improvement
Road	0.9747	0.9743	-0.0004
Sidewalk	0.8048	0.8092	0.0043
Building	0.9025	0.9094	0.0068
Wall	0.4918	0.4876	-0.0042
Fence	0.5541	0.5454	-0.0088
Pole	0.4704	0.6078	0.1374
Traffic Light	0.5259	0.7032	0.1773
Traffic Sign	0.6683	0.7732	0.1049
Vegetation	0.9085	0.9122	0.0038
Terrain	0.6044	0.6229	0.0184
Sky	0.9427	0.9441	0.0013
Person	0.7297	0.8028	0.0731

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 January 2024 doi:10.20944/preprints202401.1052.v1

Rider	0.5170	0.6204	0.1033
Car	0.9247	0.9340	0.0093
Truck	0.7632	0.7189	-0.0443
Bus	0.8139	0.8338	0.0198
Train	0.6790	0.6983	0.0193
Motorcycle	0.5658	0.5739	0.0081
Bicycle	0.6888	0.7599	0.0712

3.3.3. Ablation experiment

To ensure the final improvement effect of the model, this paper uses the control variable method to conduct experiments on each improved module.

This paper conducted a series of ablation experiments. From Table 5, it can be observed that replacing ASPP Pooling with Strip Pooling resulted in a 0.73% increase in mIoU and a 0.56% increase in mPA. From Table 6, it can be found that only after MobileNetV2 extracts shallow features and adds ECA attention mechanism, mIoU increases by 0.55% and mPA increases by 0.3%; Only after incorporating the EPSA attention mechanism in the ASPP module, mIoU increased by 1.31% and mPA increased by 1.08%. It can be found from Table 7 that on the improved light deeplbav3+, the mIoU and mPA using lovasz loss are increased by 2.22% and 1.71% respectively compared with those using CE loss; Using CL loss increased mIoU by 2.35% and mPA by 2.01% compared with using CE loss. From Table 8, it can be observed that the improvements in each module have played a positive role, effectively enhancing the accuracy of the algorithm.

Table 5. Comparison of different pooling layers on DeeplabV3+.

Backbone	Pooling	mIoU/%	mPA/%
MobilenetV2	ASPP Pooling	72.55	81.00
MobilenetV2	Strip Pooling	73.28	81.56

Table 6. Comparison of different attention mechanisms on DeeplabV3+.

Backbone	Attention	mIoU/%	mPA/%
MobilenetV2		72.25	81.00
MobilenetV2	ECA	72.80	81.30
MobilenetV2	EPSA	73.56	82.08

Table 7. Comparison of different loss functions on LM-DeeplabV3+.

Backbone	loss function	mIoU/%	mPA/%
MobilenetV2	CE loss	72.55	81.00
MobilenetV2	Focal loss	73.55	81.64
MobilenetV2	Lovasz loss	74.77	82.71
MobilenetV2	CL Loss	**74.90**	**83.01**

Table 8. Different combinations of ablation experiments.

Group	Strip Pooling	EPSA	ECA	CL Loss	mIoU/%	mPA/%
1	×	×	×	×	72.55	81.00
2	√	×	√	×	73.49	81.67
3	√	√	×	×	73.60	82.02

 doi:10.20944/preprints202401.1052.v1

4	√	√	√	×	73.83	81.95
5	**√**	**√**	**√**	**√**	**74.90**	**83.01**

[2] when CL Loss is ×, the default CE loss function is used.

4. Conclusions

This paper proposes an improved LM-DeeplabV3+model to solve the problems of large parameter counts, high computational resource requirements, insufficient attention to detail information, and insufficient accuracy of lightweight models in existing semantic segmentation models. Firstly, the backbone network is replaced with the lightweight network MobilenetV2, and the ECA attention mechanism is introduced after the shallow features of MobilenetV2 to improve segmentation performance without damaging the network structure. Replacing the ASPP Pool of the original ASPP module with a Strip Pool can be applied to more complex image scenes; Furthermore, the EPSA attention module has been introduced to effectively establish long-term dependencies between multi-scale channel attention and achieve cross dimensional channel attention interaction of important features. Then we designed a loss function suitable for multi class tasks. Although CE loss performs well in segmentation tasks, there may be a problem of training results shifting towards a larger number of categories when faced with imbalanced categories, and the measurement of CE loss on the validation set usually cannot indicate the quality of segmentation well. Therefore, we designed a loss function that combines CE loss and Lovasz loss. After using CE to find the direction of gradient descent, the Lovasz loss is used to directly optimize the semantic segmentation metric mIoU. Compared to using only CE loss or Lovasz loss, this can achieve better segmentation results and we named it CL Loss.

This paper was validated on the Cityspaces dataset and found that:

(1) LM-DeeplabV3+ effectively improves accuracy while significantly reducing parameter and computational complexity: Tthe mIoU reaches 74.9%, which is 3.56% higher than DeeplabV3+; The mPA reached 83.01%, which is 2.53% higher than the basic network.

(2) LM-DeeplabV3+ significantly improves the IoU of elements such as Pole, Traffic Light, Traffic Sign, Terrain, Person, Rider, Bicycle, etc. Among them, the improvement in Traffic light and Pole was the largest, reaching 17.73% and 13.74% respectively. Overall, it has a significant positive impact on segmentation performance. At the same time, only Truck's IoU showed a relatively significant decrease, but still at a high level, with almost no negative impact on the overall segmentation effect.

(3) The improved LM-DeeplabV3+ has been proven to be the optimal result through ablation experiments, and new pre training weights that are more suitable for semantic segmentation of city street scenes have been trained.

In the future, we will further optimize the backbone network to achieve better segmentation results while reducing the number of model parameters and computational complexity.

Author Contributions: Conceptualization, X.H.; methodology, X.H.; validation, H.G.; formal analysis, X. H.; data curation, X.H. and H.G.; writing—original draft preparation, X.H.; writing—review and editing, P.C. and H.G.; visualization, X.H.; supervision, P.C.; project administration, P.C.; funding acquisition, P.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Research and Innovation Project of Graduate Students Supported by Top-notch Innovative Talents Training Funds of the People's Public Security University of China (2022yjsky012).

Data Availability Statement: The Cityscapes dataset is publicly available https://www.cityscapes-dataset.com/dataset-overview/ .

Acknowledgments: We thank anonymous reviewers for their comments on improving this manuscript.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Ye Yu, Zhang Zhaoxi, Zhang Xiaohu, et al. Human-scale Quality on Streets: A Large-scale and Efficient Analytical Approach Based on Street View Images and New Urban Analytical Tools[J]. Urban Planning International. 2019,34(01):18-27. https://doi.org/10.22217/upi.2018.490
2. Liying ZHANG, Tao PEI, Yijin CHEN, Ci SONG, Xiaoqian LIU. A Review of Urban Environmental Assessment based on Street View Images[J].Journal of Geo-information Science, 2019, 21(1): 46-58. https://doi.org/10.12082/dqxxkx.2019.180311.
3. Voordeckers D, Meysman F J R, Billen P, et al. The impact of street canyon morphology and traffic volume on NO2 values in the street canyons of Antwerp[J]. Building and Environment. 2021, 197: 107825. https://doi.org/10.1016/j.buildenv.2021.107825.
4. Zhang F, Zhang D, Liu Y, et al. Representing place locales using scene elements[J]. Computers, Environment and Urban Systems. 2018, 71: 153-164. https://doi.org/10.1016/j.compenvurbsys.2018.05.005.
5. Fan Zhang, Bolei Zhou, Liu Liu, Yu Liu, Helene H. Fung, Hui Lin, Carlo Ratti. Measuring human perceptions of a large-scale urban region using machine learning[J]. Landscape and Urban Planning[J]. 2018, Volume 180, Pages 148-160. https://doi.org/10.1016/j.landurbplan.2018.08.020.
6. Liu L, Xie H F, Yue H. A comparative study of different street view image micro-environment extraction methods for explaining street property crimes[J]. Journal of Geo-information Science. 2023, 25(7):1432-1447. https://doi.org/10.12082/dqxxkx.2023.220917.
7. GAO Song. A Review of Recent Researches and Reflections on Geospatial Artificial Intelligence[J]. Geomatics and Information Science of Wuhan University, 2020, 45(12): 1865-1874. https://doi.org/10.13203/j.whugis20200597.
8. Minaee, S., Boykov, Y., Porikli, F., et al., 2021. Image segmentation using deep learning: a survey[J]. IEEE Trans. Pattern Anal. Mach. Intell. 44 (7), 3523–3542. https://doi.org/10.1109/TPAMI.2021.3059968.
9. Minaee, Shervin, Wang, Yao. An ADMM approach to masked signal decomposition using subspace representation[J]. IEEE Trans. Image Process. 2017, 28, 3192–3204. https://doi.org/10.1109/TIP.2019.2894966.
10. E. Shelhamer, J. Long and T. Darrell. Fully Convolutional Networks for Semantic Segmentation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1 April 2017, vol. 39, no. 4, pp. 640-651. https://doi.org/10.1109/TPAMI.2016.2572683.
11. Biao, W., Yali, G., Qingchuan, Z.. Research on Image Semantic Segmentation Algorithm Based on Fully Convolutional HED-CRF[C]. 2018 Chinese Automation Congress (CAC). IEEE. 2018, pp. 3055–3058. https://doi.org/10.1109/CAC.2018.8623459.
12. Liu, A., Yang, Y., Sun, Q., Xu, Q.. A deep fully convolution neural network for semantic segmentation based on adaptive feature fusion. 2018 5th International Conference on Information Science and Control Engineering (ICISCE). 2018, pp. 16–20. https://doi.org/10.1109/ICISCE.2018.00013.
13. H. Zhao, J. Shi, X. Qi, X. Wang and J. Jia. Pyramid Scene Parsing Network[J]. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 6230-6239. https://doi.org/10.1109/CVPR.2017.660.
14. CHEN L C, ZHU Y, PAPANDREOU G, et al. Encoder-decoder with atrous separable convolution for semantic image segmentation[C]. Proceedings of the European Conference On Computer Vision(ECCV).2018, 801-818. https://doi.org/10.48550/arXiv.1802.02611.
15. L. -C. Chen, G. Papandreou, I. Kokkinos, K. Murphy and A. L. Yuille. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1 April 2018, vol.40, no.4, pp.834-848. https://doi.org/10.1109/TPAMI.2017.2699184.
16. F. Chollet. Xception: Deep Learning with Depthwise Separable Convolutions[J]. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 1800-1807. https://doi.org/10.1109/CVPR.2017.195.
17. RONNEBERGER O, FISCHER P, B R. U-net: Convolutional networks for biomedical image segmentation[M]// Navab N, Hornegger J, Wells W M, et al. Medical image computing and computer assisted intervention-MICCAI 2015. Lecture notes in computer science. Cham: Springer, 2015, 9351: 234-241. https://doi.org/10.1007/978-3-319-24574-4_28.
18. K. He, X. Zhang, S. Ren and J. Sun, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904-1916, 1 Sept. 2015, https://doi.org/10.1109/TPAMI.2015.2389824.
19. Yu, C., Gao, C., Wang, J. et al. BiSeNet V2: Bilateral Network with Guided Aggregation for Real-Time Semantic Segmentation. Int J Comput Vis 129, 3051–3068 (2021). https://doi.org/10.1007/s11263-021-01515-2.
20. Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, Hartwig Adam. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. CoRR abs. 2017, 1704.04861. https://doi.org/10.48550/arXiv.1704.04861.

21. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L. -C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 4510-4520. https://doi.org/10.1109/CVPR.2018.00474.
22. A. Howard et al., "Searching for MobileNetV3," 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea (South), 2019, pp. 1314-1324. https://doi.org/10.1109/ICCV.2019.00140.
23. Q. Hou, L. Zhang, M. -M. Cheng and J. Feng, "Strip Pooling: Rethinking Spatial Pooling for Scene Parsing," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 2020, pp. 4002-4011. https://doi.org/10.1109/CVPR42600.2020.00406.
24. Zhang, H., Zu, K., Lu, J., Zou, Y., Meng, D. (2023). EPSANet: An Efficient Pyramid Squeeze Attention Block on Convolutional Neural Network. In: Wang, L., Gall, J., Chin, TJ., Sato, I., Chellappa, R. (eds) Computer Vision–ACCV 2022. ACCV 2022. Lecture Notes in Computer Science, vol 13843. Springer, Cham. https://doi.org/10.1007/978-3-031-26313-2_33.
25. J. Hu, L. Shen and G. Sun. Squeeze-and-Excitation Networks. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 7132-7141. https://doi.org/10.1109/CVPR.2018.00745.
26. Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo and Q. Hu, "ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 2020, pp. 11531-11539. https://doi.org/10.1109/CVPR42600.2020.01155.
27. M. Berman, A. R. Triki and M. B. Blaschko. The Lovasz-Softmax Loss: A Tractable Surrogate for the Optimization of the Intersection-Over-Union Measure in Neural Networks. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City, UT, USA. 2018, pp. 4413-4421. https://doi.org/10.1109/CVPR.2018.00464.