Pothole road detection and identification based on improved DeepLab V3+

Guohong Yang*, Xuhong Xiang#, Wangze Rui#, Suqing Yu#, Lipeng Fei#

College of Rongcheng, Harbin University of Science and Technology University, Weihai, China 264300

* Corresponding Author Email: 15550013755@163.com

#These authors contributed equally.

Abstract. This paper proposes an improved DeepLab V3+ model based on attention mechanism, which can accurately locate road potholes and effectively extract multi-scale semantic information. The improved model uses the combination of Canny edge extraction algorithm and deep learning method to input rough edges as additional semantic information into the model, and introduces channel attention mechanism and separable cavity convolutional neural network to optimize feature extraction and model size. The effectiveness of the model is proved by training the model on the open source data set and testing its performance in multiple dimensions. The trained model is used to deduce the image on the test set, and the binary mask of the road pothole is obtained, and the refined edge is obtained by edge extraction algorithm. For the second problem, the pothole part of the semantic segmentation map is counted by pixel points, and the proportion of pothole area in the whole image area is obtained, and the result is written into the file. Finally, the advantages and disadvantages of the model are evaluated, and the application of the model in other fields is discussed.

Keywords: DeepLab V3+, channel attention mechanism, Canny algorithm, separable cavity convolutional neural network.

1. Introduction

As major cities at home and abroad continue to promote the construction concept of "smart city", automatic driving has also received close attention from all walks of life. When there are potholes on the road surface, if the car can not be avoided in a timely and reasonable manner, it will not only greatly reduce people's travel comfort, but also damage the body and cause property losses. In view of the above influences, how to detect and identify potholes accurately and quickly will gradually become an important research topic in the field of automatic driving. At the same time, the identification and detection of potholes on the road is crucial to maintaining road safety and ensuring the quality of vehicle driving. In the task of pothole road detection, the traditional classification method often fails to achieve good results because of the complexity and variability of pothole image features. The computer vision technology based on deep learning can automatically extract the most important features from the image, such as the contours, textures and forms of potholes, and transform them into representations that are easier to classify, with the help of the powerful feature extraction and characterization capabilities of deep learning. Therefore, this contest intends to realize accurate and rapid automatic recognition of pothole status of unknown road images through the analysis, feature extraction and modeling of marked road images.

Based on the above background, this paper intends to solve the following problems by designing an accurate and efficient road pothole status recognition algorithm:

(1) In order to complete road obstacle avoidance and road repair work, it is necessary to more accurately identify the edge, area and other characteristics of potholes. Build a model to extract the edge of the pothole in the road.

(2) Estimate the area of the pothole according to the identified pothole edge, and calculate the proportion of the pothole area in each image to the entire image area.

2. Research methods

2.1. Analysis of road pothole surface edge extraction

To build the edge extraction of the road pothole surface. Firstly, this study considers that edge extraction algorithms are generally divided into two categories. One is the traditional image processing using edge extraction operators to extract image edges; The other is to use deep learning methods, based on convolutional neural networks, using supervised training, the trained model on the unknown data set to reason. As for the traditional method, the image requirements are very strict, and the accuracy of the algorithm often varies greatly due to different image data. For deep learning methods, paired training data sets are particularly necessary. In addition, as problem 2 requires estimating the area of road potholes based on the extracted image edge, this study will use the image semantic segmentation method, combine the traditional edge extraction algorithm and deep learning method to establish a semantic segmentation model, and train it on the paired training set. Then the trained model is deduced to the test set to get the segmentation map of road potholes. Based on the segmentation map, the edge is extracted and the area is estimated.

2.2. Estimate the area of potholes

The area of the pothole is estimated based on the identified pothole edge, and the proportion of the pothole area in each image to the entire image area is calculated. Therefore, the trained model can be placed on the test set for reasoning, and the proportion of pothole area can be estimated based on the reasoning results of the model.

3. Data acquisition and processing

Because this team adopted the form of image segmentation to solve the problem, that is, supervised training of the model, and then reasoning on the test set based on the trained model. Therefore, additional data collection and processing work is required first. We used the road pothole data set mentioned in literature [1] to train the model accordingly. The data set consists of two parts, namely the original road image data and the corresponding pothole image segmentation mask. Some of the data are shown in Table 1.

Table 1. Display of partial pothole data sets

Data	Sample display
Original image	
Mask	
Original image	
Mask	

3.1. Data Preprocessing

3.1.1 Data Enhancement

Based on the existing sample data set, in order to make the model more generalized on the test set, we use the data enhancement operation to expand the data set. The methods of data expansion can be divided into single sample data expansion and multi-sample data expansion. Single-sample data augmentation involves performing related operations on a single sample, while diversity augmentation generates new samples from multiple samples. In this competition, our team selected

six methods for single-sample enhancement techniques, namely horizontal folding, vertical folding, Gaussian noise, motion blur, random brightness, and random saturation. These six methods are divided into three categories, namely, pattern change, image blur and color change. The enhancement effect of single sample data is shown in Table 2; For the multi-data expansion technology, our team used zooming, splicing and other operations, and the multi-data enhancement effect is shown in Table 3.

Table 2. Examples of single-sample enhancement effects

Master drawing	Pattern change		Image blur		Color change	
	Horizontal folding	Vertical fold	Gaussian ambiguity	Motion blur	Random brightness	Random saturation

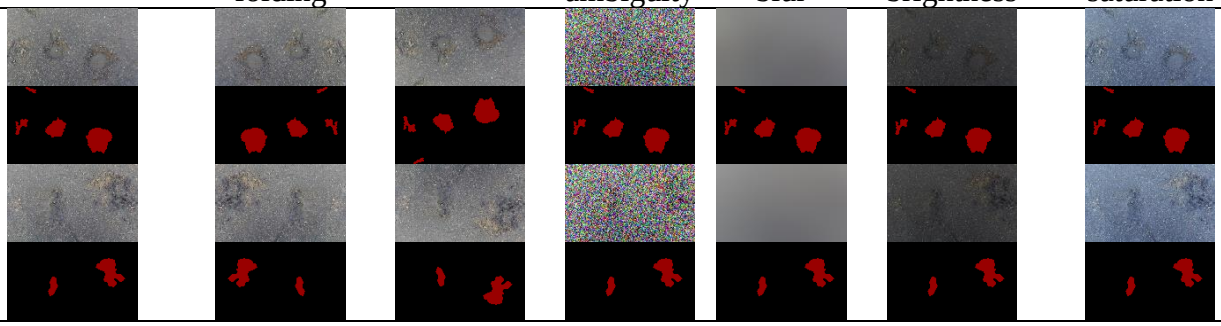
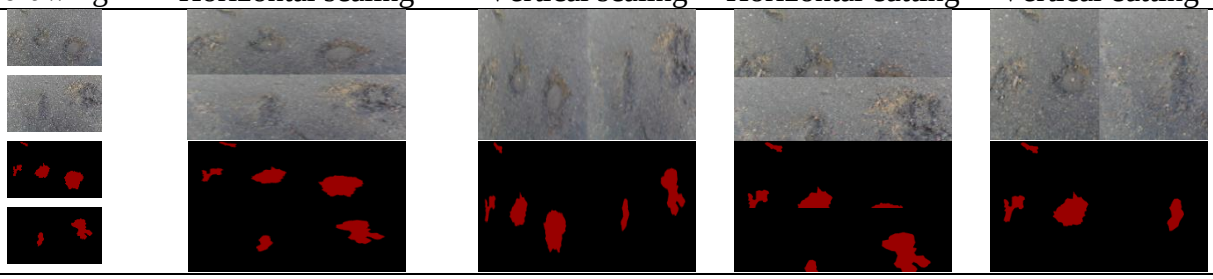


Table 3. Examples of multi-copy enhancements

Master drawing	Zoom		slice	
	Horizontal scaling	Vertical scaling	Horizontal cutting	Vertical cutting

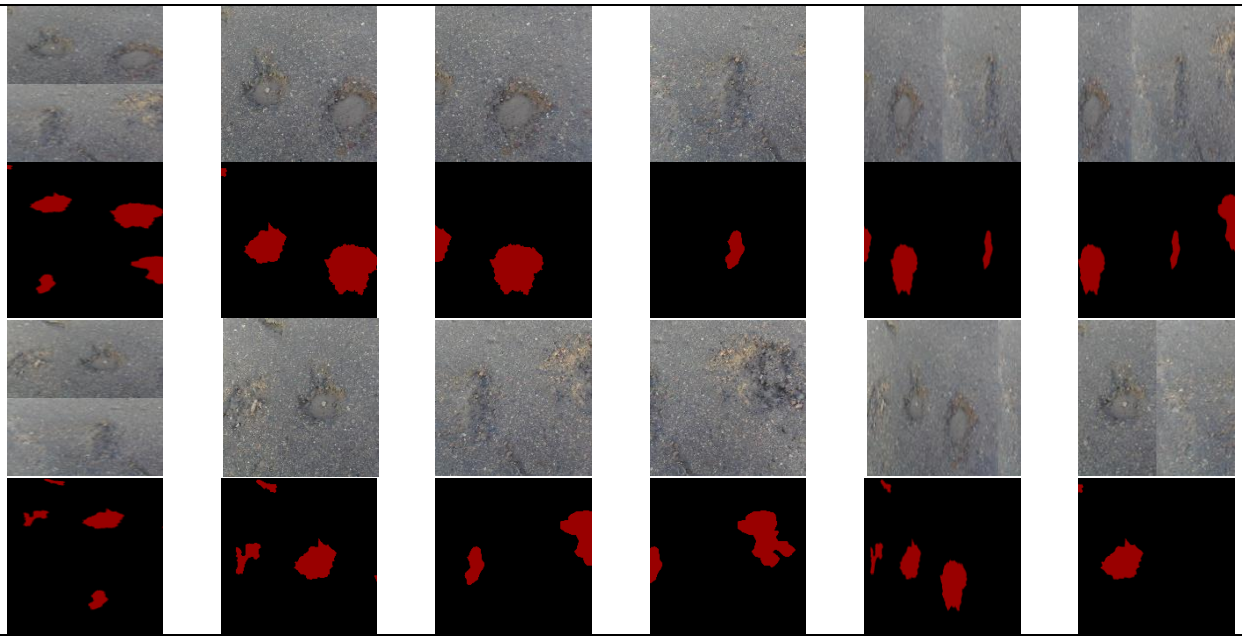


Using two different data enhancement methods, the raw data is eventually scaled up to 1000 image label pairs, and the model is trained, validated, and tested on these 1000 image label pairs.

3.1.2 Data geometric transformation

The geometric transformation of image mainly includes two parts. First, format unification, that is, all the image formats in the data set are uniformly converted to RGB three-channel format. Next comes the size adjustment. Through observation, it is found that the width and height of all images are 1328 pixels and 800 pixels respectively. Therefore, the image size is uniformly converted to 800×800×3 format, which not only preserves the original information of the image to the maximum extent, but also facilitates model training. Therefore, all the enhanced images are cropped, and the cropped data is shown in Table 4.

Table 4. Data tailoring sample presentation



3.2. Production of data sets

3.2.1 Implementation of data set

The 1000 sets of images obtained after image enhancement will be used for model training, validation and testing, which we name RDB (RoadDataBase). In this folder, there are two folders, the "Images" folder and the "Labels" folder, which store 1000 original images and 1000 binary labels of the corresponding images respectively.

3.2.2 Division of data sets

After the data augmentation, our team divided the processed 1000 pairs of data into datasets. The rational division of data sets is beneficial to the training, verification and testing of the model, and can improve the generalization ability of the model to a certain extent and reduce the risk of overfitting. Using random sampling, our team ran a 6:2:2 training set from two folders: Validation set: test set, with no return sampling, forming a list of training sets named "train.txt", validation sets named "val.txt", and test sets named "test.txt".

3.2.3 Feature scaling of data set

In the process of training data, when the image format is read as input, it is necessary to convert the image data into the tensor format of the machine learning input data to facilitate feature extraction. Different units will lead to different calculations, and larger scale features will play a decisive role. In addition, if the range of input data values is too large, it will affect the effectiveness of gradient descent in finding the best advantage. Therefore, in order to eliminate the effects of unit, scale and range differences between features, and to treat the features of each dimension equally, it is also necessary to scale the image pixel value in a specified small area according to a uniform method, without changing any information of the image. This process is called feature scaling, abbreviated FS, and can be achieved using both normalization (normalization of the dispersion of a variable) and normalization (normalization of the standard deviation of a variable) methods. Our team used a standardized approach to scale the input of the training data within the interval [-1, 1].

The standardized mathematical expression is as follows:

$$O_{ij} = \frac{I_{ij} - E(I)}{S(I)} \tag{1}$$

Where, I represents all pixel values of the input image, and I_ij represents single pixel values of the input image; E(I) represents the mean pixels of the input image, and S(I) represents the standard deviation of the input pixels. The standardization method adjusts the pixels to the [-1, 1] interval.

$$E(I) = \frac{1}{mn}\sum_{j=1}^{m}\sum_{i=1}^{n} I_{ij} \tag{2}$$

$$S(I) = \sqrt{\frac{1}{mn}\sum_{j=1}^{m}\sum_{i=1}^{n}(I_{ij} - E(I))^2} \tag{3}$$

The normalized mathematical expression is as follows:

$$O_{ij} = \frac{I_{ij} - m(I)}{M(I) - m(I)} \tag{4}$$

Where, I represents all pixel values of the input image, I_ij represents a single pixel value of the input image, m(I) and m(I) represent the maximum and minimum pixel values of the input image respectively, and O_ij represents the pixel values of row i and column j of the output image. The normalization method adjusts the pixel value to the interval [0,1].

4. Establishment of model

4.1. Overall model architecture

The architecture of the semantic segmentation network proposed in this paper for road pothole detection is shown in the figure 1 below. Before the image enters the deep convolutional neural network (DCNN) of the encoder, the edge detection module of the Canny operator is used to extract the rough edge of the pothole of the road image and obtain the boundary information. Then, the output result of the edge detection module is spliced with the original image in the channel dimension and input into the deep convolutional neural network module (ResNet50). Learn the initial features of the image. The preliminary features are then input into our proposed channel attention module to generate refined features, which are connected with the input features and sent to the Spatial feature Pyramid pooling module (ASPP) to introduce multi-scale information. In the ASPP module, we decompose the 3×3 void convolution into two cascade networks of 3×1 and 1×3 to form a deeply separable convolution form. Finally, the bottom feature and high feature are further integrated to improve the segmentation boundary accuracy in the Decoder part.

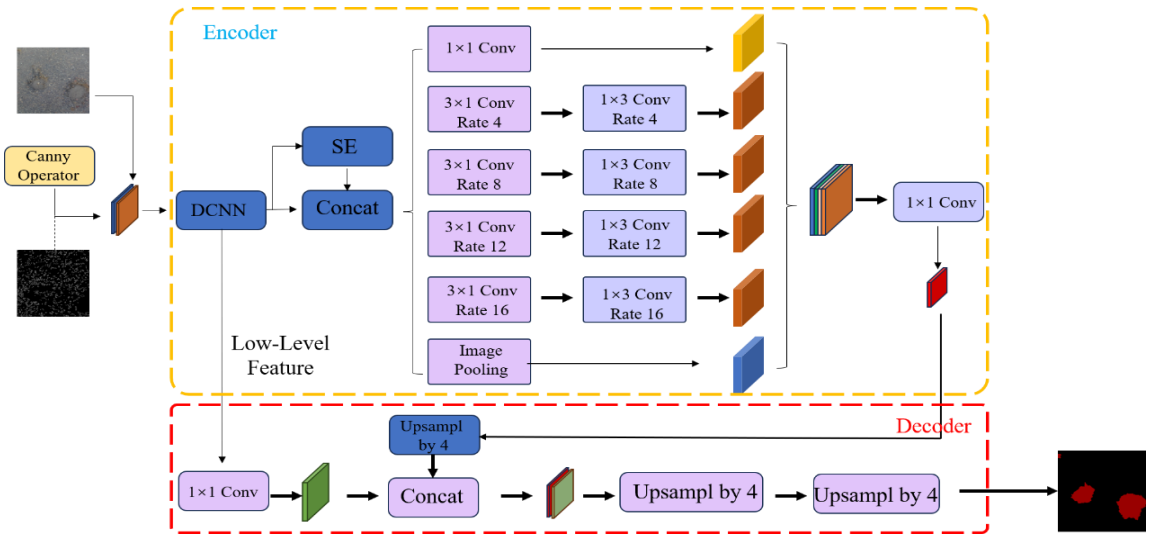


Figure 1. General model frame diagram

4.1.1 Edge detection module

Canny algorithm [1] is a kind of edge detection algorithm with good performance, which has a good effect on identifying the position with the strongest change of gray value in the image. In this

paper, Canny operator is used to extract rough road pothole boundary, which mainly includes the following three steps.

1) Gaussian filter denoising

The Canny operator has a remarkable effect on identifying the locations in the image where the gray value changes strongly, but the noise is also the place where the gray value changes greatly. It is easy to be identified as false edge, which affects the effect of land and water boundary detection. Therefore, it is necessary to filter and de-noise the original image. But at the same time, filtering will inevitably cause the image edge blur, increase the uncertainty of edge extraction, at the same time, if you want to improve the accuracy of edge detection, it will also amplify the noise. Practical engineering experience shows that the Gaussian convolution kernel can balance the contradiction between anti-noise interference and edge detection accuracy. This team uses 5×5 Gaussian convolution kernel for filtering.

2) Calculate the gradient to obtain the candidate edge

The most important feature of the edge is the sharp change of the gray value. Since the image is discrete data, the change in the gray value can be described by the difference between the two pixels. In this paper, the Sobel operator is convolved with the original image to obtain the difference between the x and y axes of the image, and then the gradient and direction of a certain point are further obtained. Since the places where the gray value changes dramatically may or may not be edges, the set of possible edges is obtained in this step.

3) Refine the edges

Non-maximum suppression and double threshold screening were used to refine and screen the edges. The effect of non-maximum suppression is to turn a multi-pixel wide edge into a single-pixel wide edge. Generally, it is the place where the change of gray value is relatively concentrated. The points with the largest change of gray value in the gradient direction are preserved, and the other points are not preserved. After non-maximum suppression, there are still many possible edge points, so it is necessary to set low threshold and high threshold. First, the pixels whose gray value changes less than the low threshold are eliminated, and the pixels whose gray value changes more than the high threshold are retained. Then, if the gray value changes between the two thresholds and contains pixels that have a greater gray value change than the high threshold, they are retained, and if not, they are eliminated to make the edge as closed as possible.

4.1.2 Channel Attention module

The mechanism of attention originated from the study of human vision. Due to a bottleneck in information processing, the human visual system selectively focuses on a portion of all information entering the eye while ignoring other visible information. This is a way for human visual system to make reasonable use of limited visual information processing resources.

There is no strict mathematical definition of attention mechanism, and in traditional information processing, such as feature extraction of local images and the use of sliding window methods, it can be regarded as an attention mechanism. In neural networks, an attention mechanism is an additional neural network that can either strictly select certain parts of the input or assign different weights to different parts of the input in order to filter out important information from a large amount of information.

In convolutional neural networks, the channel attention mechanism is one of the attention mechanisms that is implemented through automatic learning, that is, using another new neural network to obtain the importance level of each channel in the feature graph, and then using this importance level to assign a weight to each feature, enabling the neural network to focus on certain feature channels. Enhance factor mapping channels that are useful for the current task and suppress factor channels that are less useful for the current task. The representative model of channel attention mechanism is SE attention mechanism [2]. The following figure 2 shows the network structure.

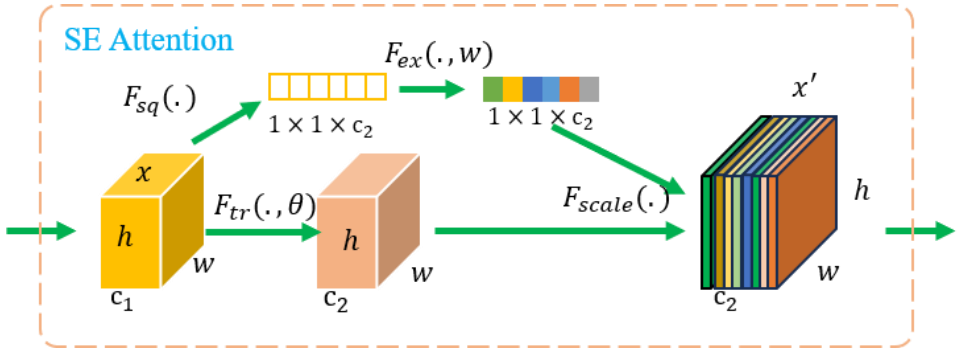


Figure 2. SE attention module

The realization process of SE attention mechanism is mainly divided into the following three steps:

①Squeeze operation

Through global averaging pooling, the two-dimensional feature (H*W) of each channel is compressed into one real number, and the feature map is transformed from h*w*c to 1*1*c

②excitation operation

A weight value is generated for each feature channel, that is, the correlation between the channels is constructed through the two fully connected layers, and the number of output weight values is the same as the number of channels in the input feature map.

③Scale operation

The normalized weights obtained above are weighted to the features of each channel, and the weighted feature map is generated.

In summary, the channel attention mechanism is simple to construct and easy to deploy without introducing new functions or layers. In addition, it also has good characteristics in terms of model and computational complexity. Therefore, the team will also adopt this component as the basic component of the model built by the team.

4.1.3 Improving the ASPP Module

DeepLab series models are a series of deep learning models for image segmentation. The first version of Deeplab v1[3] was proposed in ICLR in 2015. It combines deep convolutional neural network (DCNN) with dense conditional random field (DenseCRFs) model to obtain multi-scale context information. However, continuous pooling and downsampling can reduce resolution. Therefore, in the subsequent improvement, ASPP module is introduced. The core of ASPP module is a series of hollow convolution. As a variant of convolutional neural network, hollow convolution defines the spacing of each value when the convolution kernel processes data by introducing the hyperparameter dilation rate. The biggest feature of void convolution is that it can increase the receptive field without changing the resolution of the image output feature map. At the same time, it can obtain the semantic information of the image at different scales, which can greatly improve the accuracy of semantic segmentation tasks. In the ASPP module of DeepLab V3+ model [4-5], the traditional 3×3 cavity convolutional learning has too much redundant information and too many parameters, resulting in too long training time. So the team made the following improvements to the ASPP module:

The original expansions of the ASPP module were 6, 12, and 18. In this experiment, structures with expansion rates of 4, 8, 12 and 16 are selected to extract more effective information of low-resolution feature maps, so that the ASPP module can display finer results on the subdivision basis.

The layered convolution in ASPP is decomposed into 3×1 and 1×3 convolution while maintaining its expansion rate. These two layers of convolutional networks are cascaded together to form a deeply separable convolutional form. The improved ASPP module structure has 33% fewer convolution parameters than the original, which is faster. It can effectively reduce the amount of network computation and extract important multi-scale semantic information.

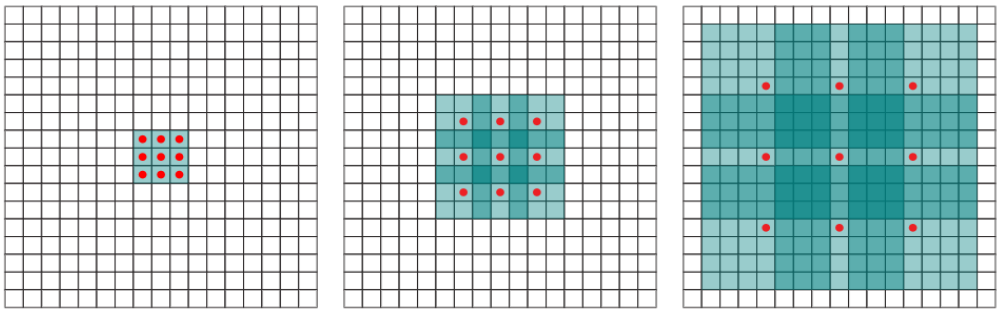


Figure 3. Three different dilated convolutions, the size of the convolution kernel is 3×3, the step size is 1, and the dilation rate is 1,2,4 (from left to right)

4.2. Loss function

The role of the loss function in deep learning is to measure the difference between the predicted value and the true value, and the smaller the difference, the better the model fits. The loss function is essentially a type function used to calculate the difference between the predicted value and the true value. In semantic segmentation problems, Dice Loss function is often used, and the corresponding expression is as follows:

$$L_{Dice} = 1 - \frac{2\sum_{i=1}^{N} y_i y_i'}{\sum_{i=1}^{N} y_i + \sum_{i=1}^{N} y_i'} \tag{5}$$

In this expression, y_i and y_i^' represent the predicted and true value of the i th pixel, respectively, and N represents the total value of the pixel.

In the above paper, our team proposed a new semantic segmentation network for road pothole detection, which improved on the architecture of the original DeepLab V3+ network. The specific form of the network was to add edge extraction pre-processing module and channel attention module for deep feature extraction. The original ASPP module is improved, the traditional 3×3 cavity convolutional module is changed into a separable form, and the original expansion rate is changed to extract more effective information under low resolution. Finally, the network is trained and optimized based on the Dice Loss function.

5. Results and analysis

5.1. Experimental setup and evaluation index

5.1.1 Experimental environment and model hyperparameters

The training and experimental results of the model are closely related to the configuration of software and hardware environment and the selection of hyperparameters. For the algorithm proposed by our team, the software and hardware configuration information of the algorithm is shown in Table 5, and the preliminary parameter configuration information of the experiment is shown in Table 6. In addition, our team uses dependent libraries such as Open CV, Numpy, and Scipy.

Table 5. Experimental environment

	configuration information	Version information
hardware	GPU	NVIDIA GeForce RTX3080Ti
	Video memory	24G
	CPU	Intel(R) Xeon(R) Silver 4214R CPU @ 2.40G
	Internal memory	96G
software	Operating system	Ubuntu 20.04.3 LTS
	Programming language	Python 3.8.10
	Model frame	Pytorch 1.7.1
	CUDA version	CUDA 11.4

Table 6. Model hyperparameters

Parameter	parameter configuration
Training rounds	200
Lot size	8
Thread count	4
Loss function	Dice Loss
optimizer	Adam
Learning rate	1e-3
Betas	(0.9, 0.999)
Eps	1e-8
Weight Decay	1e-6

5.1.2 Evaluation indicators

In our experiment, we used a 12-fold cross-validation strategy to quantify the performance of each model, that is, each model was evaluated 12 times, and pixel-level accuracy (Pre), recall rate (Rec), accuracy rate (Acc), F-score (Fsc), and union intersection (IoU) were calculated. We calculate 12 sets of averages for each indicator, expressed as mPre, mRec, mAcu, mFsc, and mIoU. In addition, the size of the model is measured in MB. The formulas and explanations for each evaluation indicator are described below.

(1) ACC, P, R, F1, IoU

$$ACC = \frac{(TP+TN)}{(TP+TN+FP+FN)} \tag{6}$$

$$Pre = \frac{TP}{TP+FP} \tag{7}$$

$$Rec = \frac{TP}{TP+FN} \tag{8}$$

$$Fsc = 2\frac{P\times R}{P+R} \tag{9}$$

$$IoU = \frac{TP}{TP+TP+FN} \tag{10}$$

In the formula, TP represents the number of samples correctly classified as positive; TN represents the number of samples correctly classified as negative; FP represents the number of samples misclassified as positive; FN indicates the number of samples for which the error classification is negative.

(2) MB (Mega Byte)

In this study, the model volume was lightly optimized to reduce hardware costs, so MB (megabytes) was introduced to measure the memory size of the model. A MB, also known as a Megabyte, is a storage unit in a computer that is commonly used to describe the size of data.

5.1.3 Introduction to the comparison model

To verify the validity of our model, we compared it with four other models: FCN[6], SegNet[7], U-Net[8], DeepLab V3+.

5.2. Experimental results and analysis

5.2.1 Hyperparameter optimization of the model

During the training of a deep learning model, some hyperparameters of the model, such as learning rate, batch size, number of hidden units, and type of optimizer, will not only affect the final convergence of the model, but also have a significant impact on the accuracy of the model. Therefore, we choose the learning rate and batch size as the hyperparameters to be determined by the model, and set different level values for these two hyperparameters for the subsequent training comparison. The selected factor level values are shown in Table 7.

Table 7. Factors and their level values

	Level 1	Level 2	Level 3
Factor A(learning rate)	0.0001	0.001	0.01
Factor B (Batch size)	4	8	16

After determining the above factors and their corresponding level values, we chose to use grid search to determine the best combination of learning rate and batch size. In the data set preparation stage, we used the files "train.txt" and "val.txt" to carry out repeated experiments on all the above models, and obtained the optimal combination value of each group of factors in each model, as shown in Table 8 below:

Table 8. Optimal combination of hyperparameters for each model

	Learning-rate	Batch-size
FCN	0.0001	8
SegNet	0.001	4
U-Net	0.0001	8
DeepLab V3+	0.0001	4
Improved DeepLab V3+	0.001	4

5.2.2 Evaluation of the model

After hyperparameter optimization, the optimal model is obtained on the training set, and then the five methods are quantitatively tested on the test set "test.txt". The test results of each model are shown in Table 9 below:

Table 9. Test results of each model

Model	mPre(%)	mRec(%)	mAcu(%)	mFsc(%)	mIoU(%)
FCN	56.039	75.114	93.881	62.570	47.627
SegNet	35.626	72.157	87.247	44.275	32.709
U-Net	70.168	64.479	94.874	63.046	48.347
DeepLabV3+	78.873	66.375	97.094	70.937	54.304
Improved DeepLab V3+	**83.900**	**69.849**	**97.740**	**74.165**	**60.034**

In addition, we also performed additional performance comparisons of DeepLab V3+ models before and after the improvements, as shown in Table 10 below

Table 10. Comparison of DeepLab V3+ models before and after improvement

Model	mPre	mRec	mAcu	mFsc	mIoU	Time	MB
DeepLab V3+	78.873	66.375	97.094	70.937	54.304	139.34	467.21
Improved DeepLab V3+	83.900	69.849	97.740	74.165	60.034	**121.97**	**385.14**

Through the comprehensive analysis of Table 9 and Table 10, it is found that the improved model has achieved the best results in all indicators, indicating that the improved model has a more stable performance in road pothole segmentation and can obtain more accurate results. By comparing the image segmentation speed and model size before and after the improvement, it can be seen that although Canny operator and channel attention module are added to the improved module when segmenting the original image of 800×800 size, the simplification of ASPP module by our team has removed redundant weight parameters, resulting in a significant reduction in the final number of parameters. Image processing speed has also improved. To sum up, the improved module has higher segmentation accuracy, and has certain improvements in segmentation rate and network structure compared with the original model.

5.2.3 Verification of the improved model -- ablation experiment

In order to further verify the validity of each improved part of our model, we conducted ablation experiments on the improved model. The ablation experiments were mainly divided into three aspects: first, to verify the effectiveness of the channel attention module; Second, verify the effectiveness of the improved ASPP module; Third, verify the validity of adding edge extraction operators. The three verification methods all use the control variable method to carry out experiments, that is, by comparing the performance of the model before and after the addition of modules. The results of the ablation experiment are shown in Table 11-13.

Table 11. Ablation experiment -- Channel attention module

Setting	Evaluation index				
Attention	mPre(%)	mRec(%)	mAcu(%)	mFsc(%)	mIoU(%)
--	83.472	67.531	95.653	73.092	59.495
✓	**83.900**	**69.849**	**97.740**	**74.165**	**60.034**

Table 12. Ablation experiment -- ASPP module

Setting	Evaluation index				
ASPP	mPre(%)	mRec(%)	mAcu(%)	mFsc(%)	mIoU(%)
--	83.047	68.621	96.486	72.047	58.472
✓	**83.900**	**69.849**	**97.740**	**74.165**	**60.034**

Table 13. Ablation experiment -- edge extraction module

Setting	Evaluation index				
Canny	mPre(%)	mRec(%)	mAcu(%)	mFsc(%)	mIoU(%)
--	83.647	69.561	97.275	73.857	59.036
✓	**83.900**	**69.849**	**97.740**	**74.165**	**60.034**

Through the above three groups of ablation experiments, the team confirmed that the addition or improvement of each module contributes to the improvement of model performance. It also reflects the effectiveness and stability of our team's algorithm.

6. Conclusion

In this paper, the experimental environment and model parameters are introduced first. Then, starting from the requirements of comparison[9] and practical application scenarios[10], this paper selects several evaluation indicators and some comparison models related to this field. In the course of the experiment, several hyperparameters of the model are optimized to obtain the optimal model. On the basis of the optimal model, the effectiveness of the proposed method is proved by model evaluation. At the same time, the validity and stability of our model are again proved by ablation experiments. Finally, based on the trained model, this paper deduces the test set and obtains the segmentation map of road potholes. Based on the segmentation map, canny algorithm is used to extract the edge and count the number of pixels. Finally, the pothole proportion of each pothole map is calculated and written into the specified file with Python language.

Reference

[1] Feng Y, Han B, Wang X, et al. Self-Supervised Transformers for Unsupervised SAR Complex Interference Detection Using Canny Edge Detector [J]. Remote Sensing, 2024, 16(2).

[2] Hu J, Shen L, Sun G. Squeeze-and-Excitation Networks [C]. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018.

[3] Jianshe S, Yuguang Y, Daxin Z, et al. Automatic segmentation of cardiac magnetic resonance images based on multi-input fusion network [J]. Computer Methods and Programs in Biomedicine, 2021, 209(prepublish): 106323.

[4] Chen L-C, Zhu Y, Papandreou G, et al. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation [M]. Cham: Springer International Publishing, 2018: 833–851.

[5] Chen T, Guo S, Deng X, et al. CVPCNN: Conditionally variational parameterized convolution neural network for HRRP target recognition with imperfect side information [J]. Signal Processing, 2024, 218109391.

[6] Shouhang D, Jianghe X, Shihong D, et al. IMG2HEIGHT: height estimation from single remote sensing image using a deep convolutional encoder-decoder network [J]. International Journal of Remote Sensing, 2023, 44(18): 5686-5712.

[7] Wang G, Hu J, Zhang Y, et al. A modified U-Net convolutional neural network for segmenting periprostatic adipose tissue based on contour feature learning [J]. Heliyon, 2024, 10(3): e25030.

[8] LU Huanyu, ZHANG Yonghong, MA Guangyi et al. Water extraction from remote sensing images based on semi-supervised adversarial learning [J/OL]. Computer Engineering, 1-17 [2023-12-15].

[9] Sang Zijiang, Shao Yeqin, Xu Changyan. Prostate MR Image segmentation based on improved 3D UNet [J]. Journal of Optoelectronics. Laser,2023,34(12):1337-1344.

[10] Wang Jianxin. Research on image segmentation technology based on Driving scene [D]. Guangxi Normal University, 2023.