

## Research Article

# A Crack Identification Method for Concrete Structures Using Improved U-Net Convolutional Neural Networks

Wenting Qiao,<sup>1,2,3</sup> Hongwei Zhang,<sup>2,3</sup> Fei Zhu,<sup>4</sup> and Qiande Wu <sup>5</sup>

<sup>1</sup>School of Highway, Chang'an University, Xi'an, Shanxi 710064, China

<sup>2</sup>Inner Mongolia Transport Construction Engineering Quality Supervision Bureau, Hohhot, Inner Mongolia 010051, China

<sup>3</sup>Key Laboratory of Transport Industry of Management,

Control and Cycle Repair Technology for Traffic Network Facilities in Ecological Security Barrier Area, Hohhot 010051, China

<sup>4</sup>Jiangsu Fasten Material Analysis & Inspecting Co., Ltd., Jiangyin, Jiangsu 214400, China

<sup>5</sup>School of Civil Engineering, Southeast University, Nanjing, Jiangsu 210000, China

Correspondence should be addressed to Qiande Wu; [wqd15109845609@163.com](mailto:wqd15109845609@163.com)

Received 22 December 2020; Revised 24 March 2021; Accepted 8 April 2021; Published 22 April 2021

Academic Editor: Paolo Lonetti

Copyright © 2021 Wenting Qiao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The traditional method for detecting cracks in concrete bridges has the disadvantages of low accuracy and weak robustness. Combined with the crack digital image data obtained from bending test of reinforced concrete beams, a crack identification method for concrete structures based on improved U-net convolutional neural networks is proposed to improve the accuracy of crack identification in this article. Firstly, a bending test of concrete beams is conducted to collect crack images. Secondly, datasets of crack images are obtained using the data augmentation technology. Selected cracks are marked. Thirdly, based on the U-net neural networks, an improved inception module and an Atrous Spatial Pyramid Pooling module are added in the improved U-net model. Finally, the widths of cracks are identified using the concrete crack binary images obtained from the improved U-net model. The average precision of the test set of the proposed model is 11.7% higher than that of the U-net neural network segmentation model. The average relative error of the crack width of the proposed model is 13.2%, which is 18.6% less than that measured by using the ACTIS system. The results indicate that the proposed method is accurate, robust, and suitable for crack identification in concrete structures.

## 1. Introduction

Cracks are one of the most serious defects in concrete structures, because when they are developed, they tend to reduce the effective loading area. It not only affects the appearance of structures, but also causes corrosion of the internal steel bars and accelerates the aging of structures, thereby affecting the bearing capacity and safety of structures. In addition, cracks originated at the surface, and it is difficult to detect them visually if the crack width is small. Therefore, detecting cracks quickly and accurately is an extremely important means for inspection and safety assessment in concrete bridges. The most widely applied techniques for detecting cracks in concrete are scanning electron microscopy and optical fluorescent microscopy.

With the advancement of science and technology, digital image processing technology, as a powerful tool for crack detection, has been widely used in concrete bridges.

Detection of surface cracks in building structures is using the digital image processing technique with image thresholding. Tong et al. [1] used the gray difference between the crack area and the background and the gray threshold segmentation method to extract the crack. However, the segmentation effect of this method is poor when it is applied to the image of bridge crack with complex background. Hoang [2] proposed a surface crack detection in building structures using image processing technique with an improved Otsu method for image thresholding. He established an intelligent model for automatic crack recognition and analyses. The model using the improved Otsu method could

effectively eliminate noisy pixels and noncrack pixels in crack images. However, the detection effect of the model was not good when the background pixel value and the noise pixel value were lower than the set threshold. Based on histogram estimation and shape analysis, Xu et al. [3] used the multiscale segmentation method to perform threshold segmentation for subblock images with different scales. The location of cracks could be determined according to the linear characteristics of cracks. It could obtain more crack information by using the multiscale segmentation method, but it would introduce more noise.

Because of the limitation of using a certain algorithm alone for crack detection, various fusion algorithms have been generated. Talab et al. [4] proposed the Otsu method and multiple filtering in image processing techniques to detect cracks in concrete structures. The method consists of three steps: ① the image was changed to gray, and Sobel filter was used to detect the crack image; ② a suitable threshold in a binary image was used, all pixels with two categorizations, background and foreground, were classified, and the region area was got after the filter area was used, and the area was changed if it is less than the specific number to get back; ③ major cracks were detected using Otsu method after using Sobel's filtering to elimination of residual noise. Wu et al. [5] proposed a novel and efficient image processing method for extracting cracks from blurred and discontinuous pavement images. First, a series of pretreatments were performed to enhance the difference between the crack and the background. Then, the multiresolution method was used to leave more image features. Finally, max-mean fusion was performed on the obtained images, and threshold determination was made for the connected fracture region to realize fine fracture fusion. Sun et al. [6] proposed a weighted neighborhood pixel segmentation method to automatically detect cracks. The results showed that the developed automated detection and segmentation method is accurate, fast, and robust. Zhang et al. [7] proposed an automatic pavement crack detection method, which used the geodetic shadow removal algorithm to eliminate the shadow, and then established the crack probability map, extracted the crack seed, and deduced the minimum spanning tree to obtain the shape of the crack.

In recent years, neural network technology has been widely used in the processing of crack images. Zhang et al. [8, 9] used convolutional neural network (CNN) to predict whether a single pixel in a crack image belongs to a crack. The proposed algorithm could reflect the details of cracks. However, it needs manually set feature extractor for pre-processing. The size of images has a great influence to the setting of networks. Cha et al. [10] proposed a defect detection algorithm based on Fast Region Convolutional Neural Networks (Fast R-CNN) and compared it with the traditional edge detection algorithms of Canny and Sobel. The results indicated that the algorithm could detect more types of defects. However, the algorithm took a lot of time to process images and could not get a complete crack image. Chen et al. [11] used the convolutional neural network and NB-CNN network fused with naive Bayesian data for fracture detection. The advantage of this algorithm is that it

can detect tiny cracks, but this method can only detect the location of cracks and cannot extract cracks. Li et al. [12] designed a CNN with dual-partition output based on improved Google-net convolutional neural network. This method could extract crack feature information of images, but it could not locate the extracted crack information to the original image location. Yang et al. [13] used the semantic segmentation method based on fully convolutional network (FCN) to detect images of cracks. The detection method could extract relatively complete crack images. Its training time was short. However, its image information loss was large, and the spatial level information location of crack pixels was not accurate enough. Zhu et al. [14] proposed a crack identification algorithm for U-net convolutional neural network, using U-net networks as the front end to extract the crack, and then using threshold method and Dijkstra connection to extract the crack accurately. However, this method was still difficult to solve the problem that feature resolution degradation caused by continuous pooling. An Atrous Spatial Pyramid Pooling (ASPP) module was proposed by Chen et al. [15, 16], which probed convolutional features at multiple scales, with image-level features encoding global context and further boosting performance. It used Atrous convolution in cascade or in parallel to capture multiscale context by adopting multiple Atrous rates, which could increase the details of image features and enhance the effect of dense prediction.

In order to improve the accuracy of crack detection of reinforced concrete structures, a method for identifying cracks of concrete beams based on improved U-net convolutional networks is proposed in this article. Firstly, a bending test of concrete beams is conducted to collect crack images. Secondly, datasets of crack images are obtained by using data augmentation technology. Selected cracks are marked. Thirdly, based on the U-net neural networks, an improved inception module and an Atrous Spatial Pyramid Pooling (ASPP) module are added in the improved U-net model to reduce the data loss in the process of pooling and improve the accuracy of multifeature fusion, which is the innovation of this paper. The new Loss Function Dice Loss is used in the model to improve the sensitivity of the network to pixels of cracks. Finally, the widths of cracks are identified using MATLAB image processing technology.

## 2. Bending Test of Reinforced Concrete Beams

In this article, a bending test of concrete beams is conducted to collect crack images. According to the minimum reinforcement ratio, 6 concrete beams with dimensions of  $200 \times 400 \times 1500$  mm and concrete grade of C30 are designed and constructed. After curing for a certain period, the specimens were loaded, and cracks appeared on the surface. The actual widths of 20 cracks selected are measured by using the crack comprehensive tester HC-F800 (Figure 1).

The widths of cracks were also detected by the ACTIS system of Japan KURABO Co., Ltd. Table 1 gives the parameters of the ACTIS system. The widths measured by the ACTIS system are compared with those measured by the HC-F800 tester, as shown in Table 2. Table 2 indicates that



FIGURE 1: The crack comprehensive tester HC-F800.

TABLE 1: Measurement parameters of the ACTIS system.

Item	Parameter
Working distance (precision)	50 m (0.05 mm)
The measured area	3 m × 4 m
Working angle	360°
Amount of data stored	64 G
Speed of detection	260 m <sup>2</sup> /h
Suitable object	Plane, surface, cylinder
Output format	CAD, Exelt, TXT
Image resolution	7360 × 4912PIX

TABLE 2: Comparison of the measured crack widths.

Crack number	Measured results	ACTIS measurements	Relative error
1	0.74	0.35	52.7
2	1.10	0.40	63.6
3	0.81	0.41	49.4
4	0.35	0.28	20.0
5	0.33	0.24	27.3
6	0.27	0.24	11.1
7	0.37	0.31	16.2
8	0.43	0.35	18.6
9	0.36	0.27	25.0
10	0.43	0.24	44.2
11	0.36	0.25	30.6
12	0.48	0.30	37.5
13	0.37	0.19	48.6
14	0.42	0.34	19.0
15	0.35	0.23	34.3
16	0.58	0.34	41.4
17	0.37	0.31	16.2
18	0.56	0.39	30.4
19	0.57	0.46	19.3
20	0.45	0.31	31.1

the average relative error of the crack width measured by the ACTIS system is 31.8%, indicating that the measured accuracy of the ACTIS system is low.

The 20 crack areas are marked. Then, a digital camera is used to shoot at different shooting distances (3 m, 5 m, 10 m, 15 m, and 20 m) and different shooting angles ( $-30^\circ$ ,  $0^\circ$ , and  $30^\circ$ ). The total number of digital images is 300.

### 3. Bending Test of Reinforced Concrete Beams

Flowchart of the crack detection is shown in Figure 2. Using 300 images to train a deep learning neural network might be overfitting, so it is necessary to divide the original dataset into a training dataset and a test dataset, augment the training dataset, and verify the model on the test dataset. Data augmentation technology includes horizontal mirroring (equation (1)) and vertical mirroring (equation (2)).

$$\text{new\_image}(x, y) = \text{old\_image}(\text{width} - x - 1, y), \quad (1)$$

$$\text{new\_image}(x, y) = \text{old\_image}(x, \text{height} - y - 1). \quad (2)$$

An example of an image is shown in Figure 3. After data augmentation, the LabelMe tool is used to mark the data augmentation. The name of the marked crack is denoted as “ck,” as shown in Figure 4.

### 4. A Crack Identification Method Using Improved U-Net Convolutional Neural Networks

*4.1. U-Net Convolutional Neural Networks and ASPP.* U-net convolutional neural networks are improved fully convolutional neural networks. They make full use of the abstract features obtained by the deep network and the image information contained in the shallow networks. They adopt the method of copy and superposition for feature fusion. Therefore, they could realize automatic segmentation of images effectively and accurately. U-net convolutional neural networks are mainly used in medical image segmentation, which indicates that, in the case of a small amount of deep learning data, the semantic segmentation accuracy of U-net is relatively high, so that it can reduce the workload of concrete detection and improve the efficiency of concrete crack detection. Schematic diagram of U-net convolutional neural networks is shown in Figure 5.

ASPP probes convolutional features at multiple scales, with image-level features encoding global context and further boosting performance. It uses Atrous convolution in cascade or in parallel to capture multiscale context by adopting multiple Atrous rates. Then, it uses  $1 \times 1$  convolution to linearly fuse each channel. Schematic diagram of ASPP is shown in Figure 6.

*4.2. An Improved Inception Module.* In the actual shooting process, the actual size of pixels of actual cracks in bridges was different, and it was more suitable to use different scale convolution kernels to extract features. The original convolution network is replaced with an Improved Inception module [17]. The receptive fields of convolution kernels of different scales are different, and the expression of the receptive fields is

$$F(i) = (F(i + 1) - 1) \times S + K. \quad (3)$$

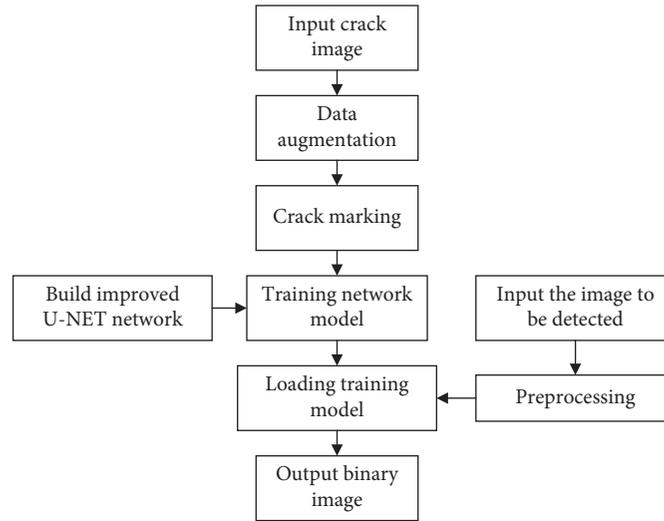


FIGURE 2: Flowchart of the crack detection.

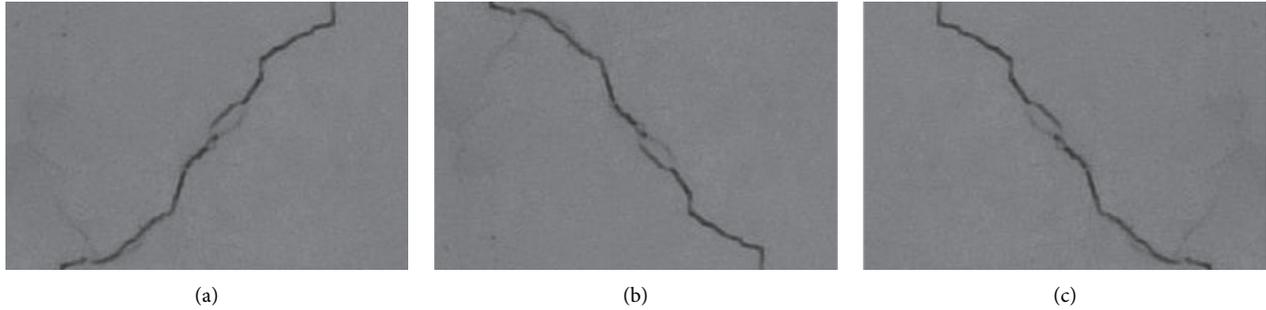


FIGURE 3: Data augmentation. (a) Original image. (b) Image processed by horizontal mirroring. (c) Image processed by vertical mirroring.



FIGURE 4: Examples for marking cracks.

Here,  $F(i)$  is the receptive field of the  $i$ -th layer;  $S$  is the step distance of the  $i$ -th layer;  $K$  is the scale of the convolution kernel.

According to formula (3), it can be calculated that 2 cascaded  $3 \times 3$  convolution kernels and 1 cascaded  $5 \times 5$  convolution kernel have the same receptive field. Therefore, the  $5 \times 5$  convolution kernel in the classic Inception module is replaced with 2 cascaded  $3 \times 3$  convolution kernels.

In the inception module, after  $1 \times 1$  convolution, No. 1 feature map is got.  $3 \times 3$  convolution is used to get No. 2 feature map. After two consecutive  $3 \times 3$  convolutions, the

No. 3 feature map is got. Then, after  $3 \times 3$  max-pooling and  $1 \times 1$  convolution, the No. 4 feature map is got. The four feature maps are spliced to realize image feature extraction of multiple convolution kernels. Its process is shown in Figure 7. This method reduces the parameters without changing the receptive field effect and improves the efficiency of model training.

**4.3. Improved U-Net Neural Networks.** Schematic diagram of improved U-net neural networks is shown in Figure 8. In the dataset, the original image data size is  $256 \times 256$ . During the convolution process, the resolution of the feature map will be reduced. In order to make the boundary information better preserved in the network layer before the convolution operation, the edge of the feature map is mirrored (overlay-tile strategy). During the mirror operation, half of the receptive field is added to each boundary as the mirror edge. The size of the mirror edge used in the convolution process of the model is defined as one pixel, which can ensure that the specification of the feature map remains unchanged during the convolution process, and the information of the last layer of the feature map is retained to the greatest extent.

In the coding part, IM is the inception module. The size of the input crack image is  $256 \times 256 \times 3$ , which is processed

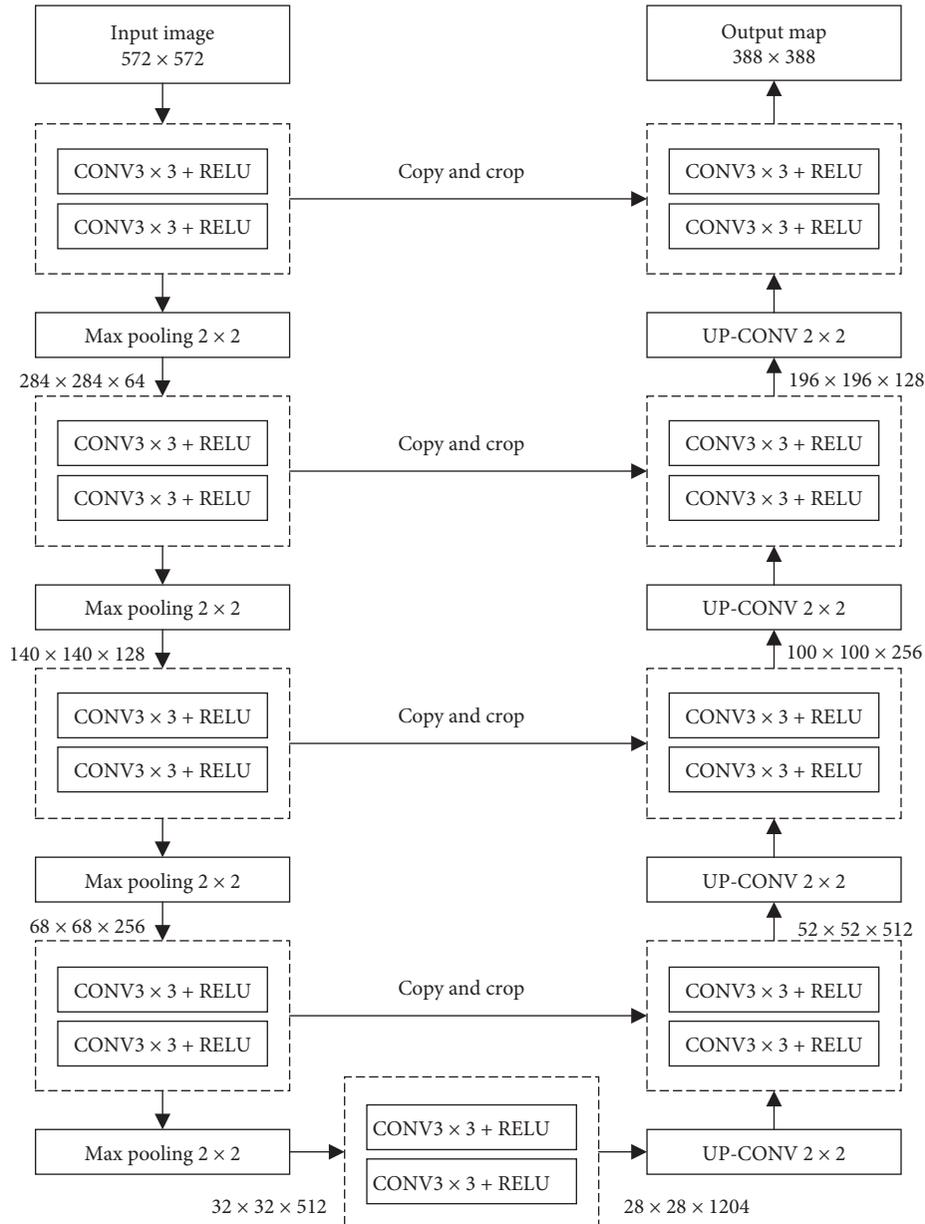


FIGURE 5: Schematic diagram of U-net convolutional neural networks.

by the first inception module. Inside the module, it undergoes 4 branch convolution and pooling operations and then spliced and reduced by  $1 \times 1$  convolution. Finally, the eigenmatrix is normalized after dimension reduction [18]. After the max-pooling process, a feature map with the size of  $128 \times 128 \times 64$  is got. Input the feature map again into the IM module for processing to get an eigenmatrix with the size of  $128 \times 128 \times 128$ . By analogy, after 5 inception modules, an eigenmatrix with a size of  $16 \times 16 \times 1024$  is generated. Finally, the feature map obtained in the downsampling stage is input to the ASPP module for processing.

In the upsampling stage, a deconvolution operation is used to restore the feature map to matrix specifications of the corresponding network layer. The specific process is as follows: first of all, the feature map with the size of

$16 \times 16 \times 1024$  obtained by the ASPP module is upsampled to obtain a feature map with the size of  $32 \times 32 \times 512$ . Then, the fusion layer is used for fusion; that is, the feature map obtained by convolution and the feature map of the same size in the downsampling are spliced in channel dimensions to obtain a feature map. Finally, it is reduced by  $1 \times 1$  convolution. A binary image of  $256 \times 256 \times 1$  is obtained. The training process of an image is completed in the improved U-net neural networks.

**4.4. Loss Functions.** Using the conventional loss cross-entropy function has the problem that the network structure has a good segmentation effect on background pixels, but it is not sensitive to pixels of cracks, resulting in the failure in correctly

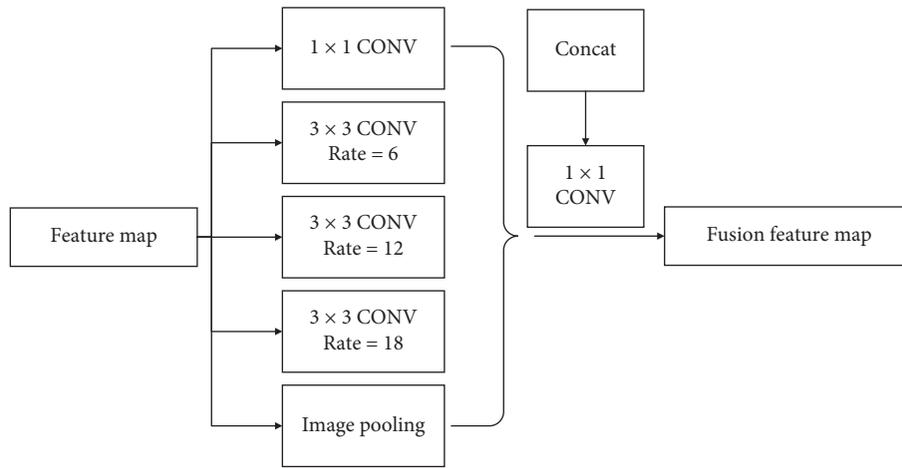


FIGURE 6: Schematic diagram of ASPP.

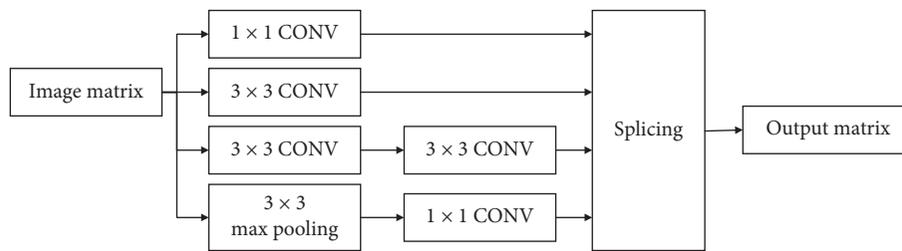


FIGURE 7: An improved Inception module.

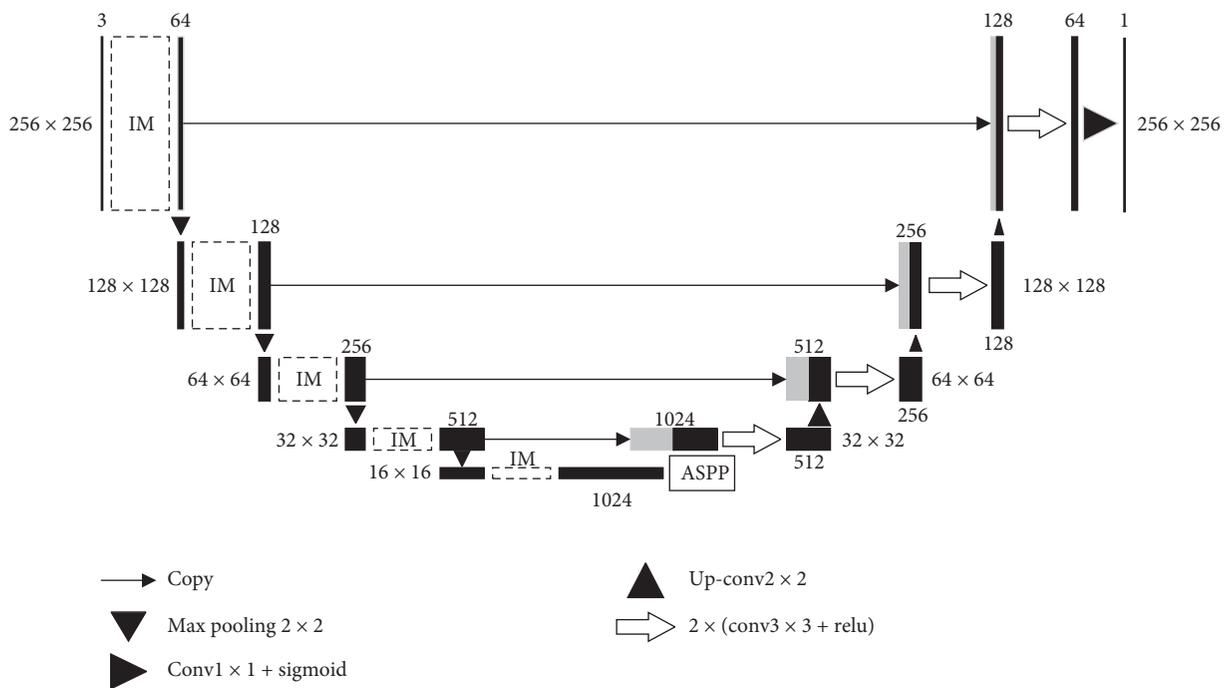


FIGURE 8: Schematic diagram of improved U-net neural networks.

identifying cracks during network prediction. Therefore, the loss function [19] GDL (Generalized Dice loss) is adopted in this article. The calculation formulas are as follows:

$$\text{GDL} = 1 - 2 \frac{\sum_{l=1}^2 w_l \sum_m y_{lm} y_{lm}'}{\sum_{l=1}^2 w_l \sum_m y_{lm} + y_{lm}'} \quad (4)$$

$$w_l = \frac{1}{\left(\sum_{m=1}^M y_{lm}\right)^2}. \quad (5)$$

Among them,  $y_{lm}$  is the actual value of category  $l$  at point  $m$ ;  $y_{lm}'$  is the predicted value of the point;  $w_l$  is the weight.

Each classification sample has a Dice loss function, and the weight of each classification is inversely proportional to the proportion of pixels, thus reducing the influence of the Dice loss function and the area size. Since Dice loss function uses the similarity between the predicted value and the true value as an indicator, it can be used as a loss function for sample imbalance classification problems, so Dice loss function is used as a loss function to improve U-net.

## 5. Results and Discussion

*5.1. Preparation for Experiments.* Training, verifying, and testing are performed on the 64-bit window 10 platform equipped with i7700u CPU, 8G memory, and GTX1060 GPU. The network model is established based on the PyTorch deep learning framework and trained using the GPU. The crack images are collected in the bending test of concrete beams. 1000 crack datasets are got by using the data augment technology. In order to test the feasibility of this algorithm, the dataset is divided into training dataset and test dataset, and 50 images are selected for testing.

*5.2. Parameter Settings and Model Training.* Appropriate training parameters are the key to ensure the accuracy and speed of training. After constant debugging, the training parameter of the improved U-Net algorithm is shown in Table 3.

Batch size is defined as the number of samples required for a training of the model, and its size affects the optimization degree and speed of the model. The value of the batch size is determined by the size of the data set. A reasonable batch size can not only improve the utilization rate of memory, make the GPU run near full load, and improve the speed of training, but also make the direction of gradient descent in the training process more accurate and the network converge faster. In the training process, the value of batch size is set as 1 first. At this time, the gradient of the network generally changes greatly, and it is difficult for the network to converge. As the value increases, the network gradient becomes accurate. When the increase reaches a certain value, the accuracy of the network no longer improves, and the batch size at this point is the optimal value. After adjusting the batch size to the optimal value, epoch needs to be added in order to improve the accuracy of the network.

TABLE 3: Parameter settings.

Item	Batch size	Epoch	Learning rate	Weight decay	Momentum
Value	4	200	0.0001	$2e-5$	0.9

The process of a complete data set running through a neural network once and returning once is called an epoch. In an epoch, the training samples underwent a forward- and backpropagation. In the training process, the neural network could not converge if the number of epochs is too low. If there are too many epochs, it will not only cause computational waste, but also lead to overfitting phenomenon of neural network. In Figure 9, (a) is the result of function overfitting due to the excessive number of epochs; (b) is the optimal solution of the function; (c) shows the result that the number of epochs is too low, and the function is not fitted. The number of epochs is related to the degree of diversity of the data set.

Weight decay is a factor placed before the regular term in the loss function that moderates the effect of model complexity on the loss function. If weight decay is large, the complex model loss function is larger. The use of proper weight decay prevents network overfitting problems. The calculation formula is as follows:

$$C = C_0 + \frac{\lambda}{2n} \sum_w w^2. \quad (6)$$

$C$  is the loss function after regularization,  $C_0$  is the loss function without regularization (the Generalized Dice loss is used in our algorithm), and  $(\lambda/2n)\sum_w w^2$  is a regular item, namely, the sum of the squares of the weighting matrix divided by  $2n$ . The above formula is deduced and calculated as follows:

$$\frac{\partial C}{\partial w} = \frac{\partial C_0}{\partial w} + \frac{\lambda}{n} w, \quad (7)$$

$$w \longrightarrow w - \eta \left( \frac{\partial C_0}{\partial w} + \frac{\lambda}{n} w \right) = \left( 1 - \frac{\eta \lambda}{n} \right) w - \eta \frac{\partial C_0}{\partial w}. \quad (8)$$

The value of  $(1 - (\eta \lambda / n))$  is less than 1 because  $\lambda, \eta, n$  are all positive. Its effect is to reduce the weight  $w$  and control  $w$  within a certain range, so that the complexity of the training network is lower, and the data fitting is better.

Momentum is a method of acceleration in a gradient descent. In this algorithm, the stochastic gradient descent method (SGD) is adopted. The calculation formulas of the SGD method with momentum are as follows:

$$v = \beta * v - \alpha * dx, \quad (9)$$

$$x \longrightarrow x + v. \quad (10)$$

$\beta$  is the coefficient of momentum, and  $v$  is the value of momentum. If the direction of momentum in the last training process is the same as the negative gradient direction, then the gradient descent speed will increase.

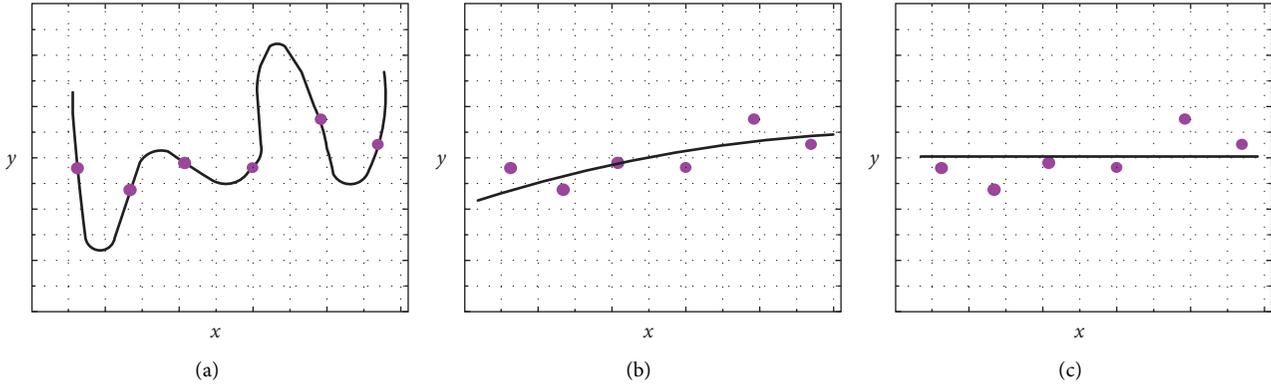


FIGURE 9: Curves of training loss over training epoch.

Learning rate ( $lr$ ) determines whether most optimization algorithms can converge and control the speed of gradient descent of neural networks. The calculation formula is as follows:

$$w = w - \alpha \frac{\partial}{\partial w} \text{loss}(w). \quad (11)$$

$\alpha$  is the learning rate in gradient descent. If the value of learning rate is set too large, and the gradient drops too fast, then the training loss shock will be caused, and the optimal solution may be skipped in the descent process. If the value of learning rate is set too small, the training process will be greatly increased. The relationship between different learning rates and loss values is shown in Figure 10.

Different learning rates lead to different degrees of loss convergence. Therefore, choosing an appropriate learning rate is the key to effectively reduce loss. Leslie proposed a method to select the appropriate learning rate of neural networks in 2015 [20]. The algorithm first selects a learning rate to train the network, and then the learning rate is updated once after 10 epochs. Then, the learning rate and training loss of each epoch were recorded, and the relationship curve between the learning rate and loss was plotted, as shown in Figure 11.

In the functional relationship between learning rate and training loss, a point where the loss changes fastest is selected, and the abscissa of this point is the optimal value of learning rate. The function shown in Figure 10 should be derivative to get the curve of loss change rate over learning rate and then smooth the curve, as shown in Figure 12. The maximum value of the change rate of training loss is 0.0001, so the learning rate in the optimization algorithm is 0.0001.

After setting the parameters, follow the following steps to run the training code:

- (1) Data processing. In order to increase the robustness of the algorithm and enlarge the data set, it is necessary to enhance the data. All tagged images under data: path is first read, and a function called `tr.scalenRotate` is set up for data enhancement including image flipping (range:  $15^\circ$  to  $15^\circ$ ) and image scaling (range: 0.75 to 1.5). Then, the processed data images are adjusted to a specific resolution using a

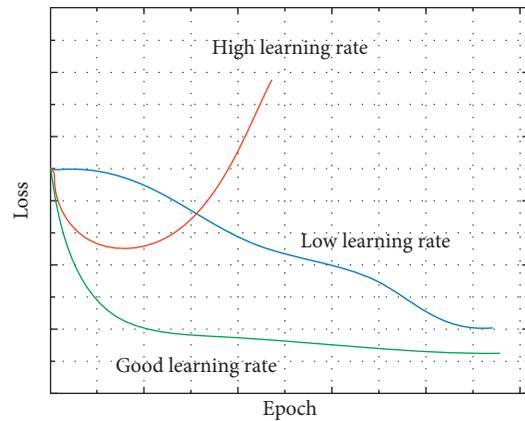


FIGURE 10: Curves of training loss over learning rate.

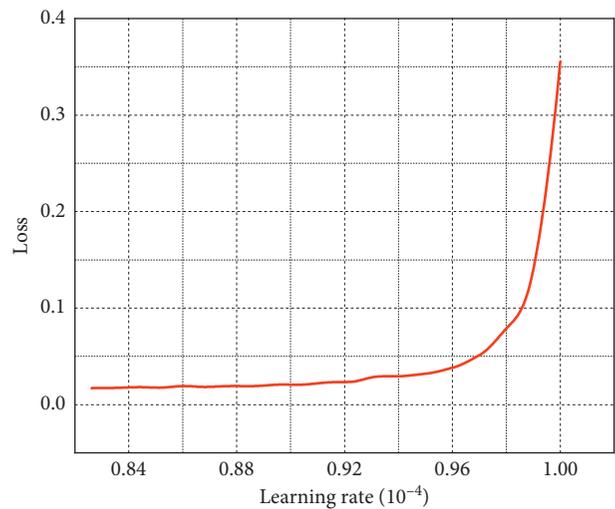


FIGURE 11: Curves of training loss over learning rate in new algorithm.

function named `tr.FixedResize`. Finally, these tensor images are normalized using standard deviation and mean value through a function named `tr.Normalize`, so as to import the constructed model for training. Then, the processed data is counted with the

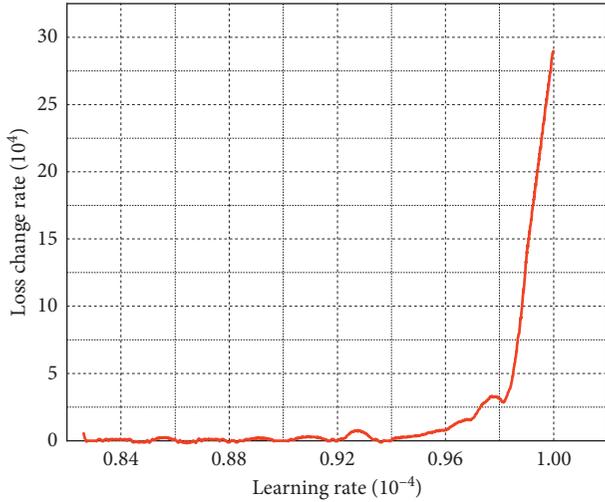


FIGURE 12: Curves of loss change rate over learning rate.

VOCsegmentation command. Finally, the training data is imported with the DataLoader command.

- (2) Model selection and import. Two files named unet\_aspp.py and unet\_model.py are created to write the model code and select the GPU for training.
- (3) Model training. A file named Log.txt will be output including train loss, IoU, and acc.
- (4) The image of the test set is segmented by the trained model. The improved U-net neural network model is used to segment the images of the test set.

**5.3. Evaluation Indicators.** In order to determine the performance of the proposed model, recall, precision, and Dice similarity coefficient (DSC) are used as the evaluation indicators for the segmentation models. The precision represents the probability that the ground truth of the sample is also cracked in all samples, which are predicted to be cracked. The recall indicates the probability of sample being predicted as cracked in all samples labeled as cracked. The functions of recall and precision are shown in equations (12) and (13).

$$f_{\text{Recall}} = \frac{i_{TP}}{i_{TP} + i_{FN}}, \quad (12)$$

$$f_{\text{precision}} = \frac{i_{TP}}{i_{TP} + i_{FP}}. \quad (13)$$

Here,  $i_{TP}$  is the pixel where a crack is predicted to be a crack;  $i_{TN}$  is the pixel where a noncrack is predicted to be a noncrack;  $i_{FN}$  is the pixel where a crack is predicted to be a noncrack; and  $i_{FP}$  is the pixel where a noncrack is predicted to be a crack. The smaller the recall rate is, the higher the probability that the crack will be missed is. The higher the precision is, the lower the probability that a noncrack will be falsely detected as a crack is.

In addition, the most commonly used indicator for evaluating semantic segmentation is mean\_IoU (Mean

Intersection over-Union) [21], which is used to calculate the ratio of noncrack and crack set:

$$\text{mean\_IoU} = \frac{1}{1+k} \sum_{i=0}^k \frac{P_{ii}}{\sum_{j=0}^k P_{ij} + \sum_{j=0}^k P_{ji} - P_{ii}}. \quad (14)$$

In the formula,  $k+1$  is the number of categories in the image;  $P_{ii}$  is the number of pixels correctly classified in category  $i$ ;  $P_{ij}$  is the number of pixels in category  $i$  that are incorrectly classified into category  $j$ ; and  $P_{ji}$  is the number of pixels in category  $j$  that are incorrectly classified into category  $i$ .

**5.4. Model Training Results.** In model training, the loss function value, precision, and recall over training epoch are shown in Figures 13~15. Figure 9 shows curves of training loss over training epoch. Figure 13(a) is the curve obtained using the improved U-net neural networks. Figure 13(b) is the curve obtained using the U-net neural networks. Figure 13(c) is the curve obtained using Segnet. Figure 13 shows the curves of precision over training epoch. Figure 14(a) is the curve obtained using the improved U-net neural networks. Figure 14(b) is the curve obtained using the U-net neural networks. Figure 14(c) is the curve obtained using Segnet. Figure 14 shows the curves of recall over training epoch. Figure 15(a) is the curve obtained using the improved U-net neural networks. Figure 15(b) is the curve obtained using the U-net neural networks. Figure 15(c) is the curve obtained using Segnet.

The loss function value obtained by using the improved U-net neural networks needs about 90 iterations of training to reach stability. The loss function value obtained by using the U-net neural networks needs about 100 iterations. The loss function value obtained by using Segnet needs more than 200 iterations. It can be seen that the loss function using the improved U-net model converges fastest, and the number of epochs required for training is the least.

The maximum training precision obtained using the three models of the improved U-net neural networks, the U-net neural networks, and Segnet is 0.96627, 0.85456, and 0.61531, respectively. Among these three models, the accuracy of the improved U-net model is basically equal to that of the U-net model, while the Segnet model has the lowest training precision and the worst detection effect. The maximum training recall value obtained using the above three models is 0.95112, 0.95231, and 0.89544, respectively. The recall obtained using the improved U-net neural networks and U-net neural networks is higher than that obtained using Segnet, so the proposed method using the improved U-net neural network has the lowest false detection rate.

To verify the effectiveness of the identification method, 6 images are selected for experiments. Figure 16 gives original images. The above six images are taken as examples to compare the three indicators of precision, recall, and mean\_IoU of the three models, as shown in Tables 4~6. It can be seen from Tables 4~7 that the precision, recall, and mean\_IoU of the proposed model in this article are all better

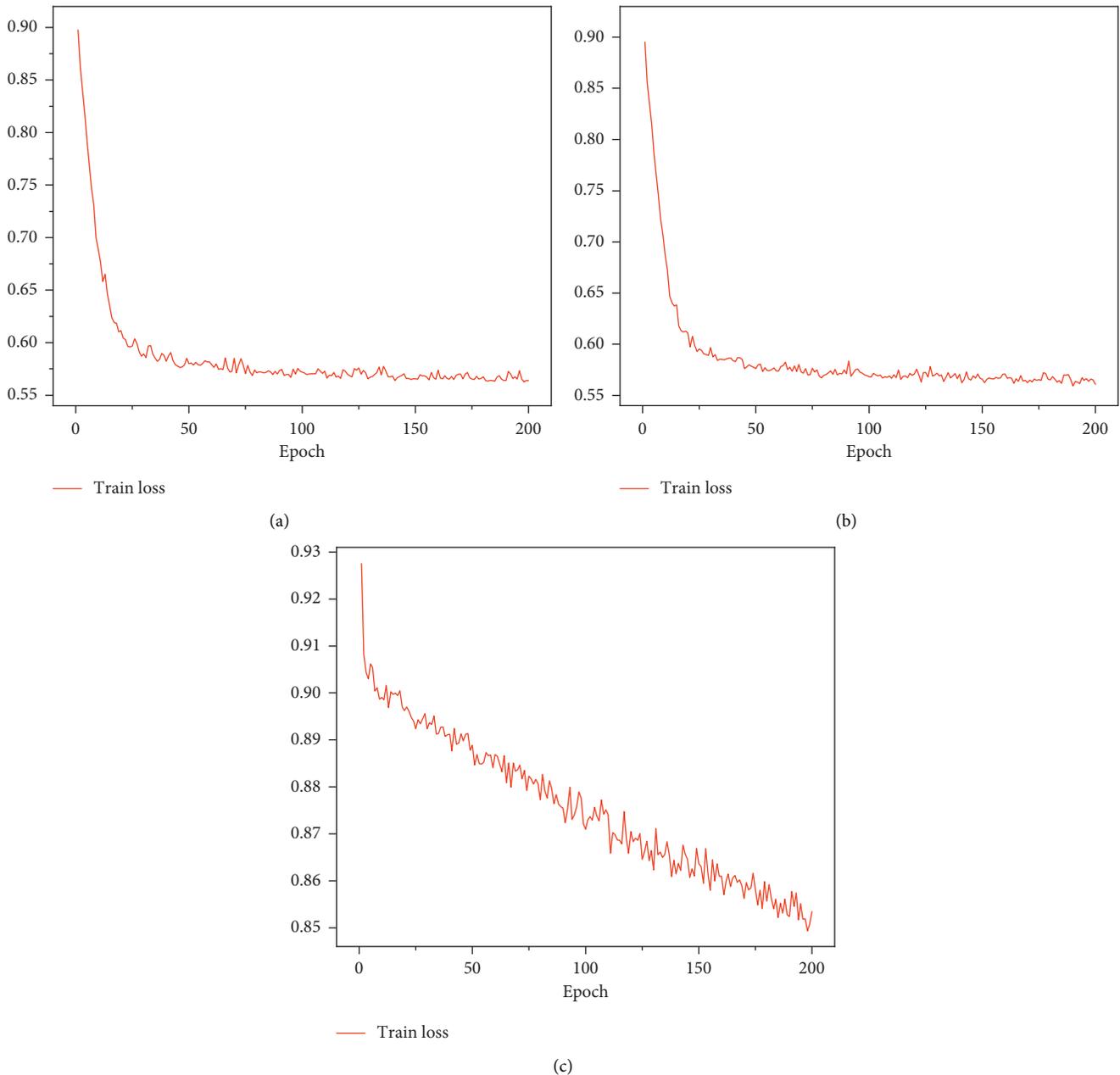


FIGURE 13: Curves of training loss over training epoch.

than those of the other two models when processing the six images in Figure 16.

50 images are used as the test set. The precision, recall, and mean\_IoU values of the 50 images are calculated, respectively. Then, the average value is calculated, as shown in Table 8. As can be seen from Table 8, the average precision of the test set of the proposed model is improved by 11.7% compared with that of the U-net model, the average recall is increased by 0.3%, and the mean\_IoU value is increased by 2.9%, which indicates that the proposed model in this article has high precision and good robustness.

The detection and segmentation results of concrete surface cracks by the three models are shown in Figure 17. The main differences in the image processing results of each

identification method are shown in the red box. From the comparison of evaluation indicators and the image effect of detection, the Segnet image segmentation results are relatively rough, and there are cases of false detection in the crack region. In contrast, the segmentation effect of the U-net model and improved U-net model has been improved. Segnet does not make full use of the features extracted by the network coding part. It gradually restores the feature map to the original image size through simple upsampling and pooling index operations but ignores the connection between pixel positioning and classification, so the segmentation results are more coarse-grained than the other two models. U-net fuses low-level features with high-level features through jump connections to achieve finer

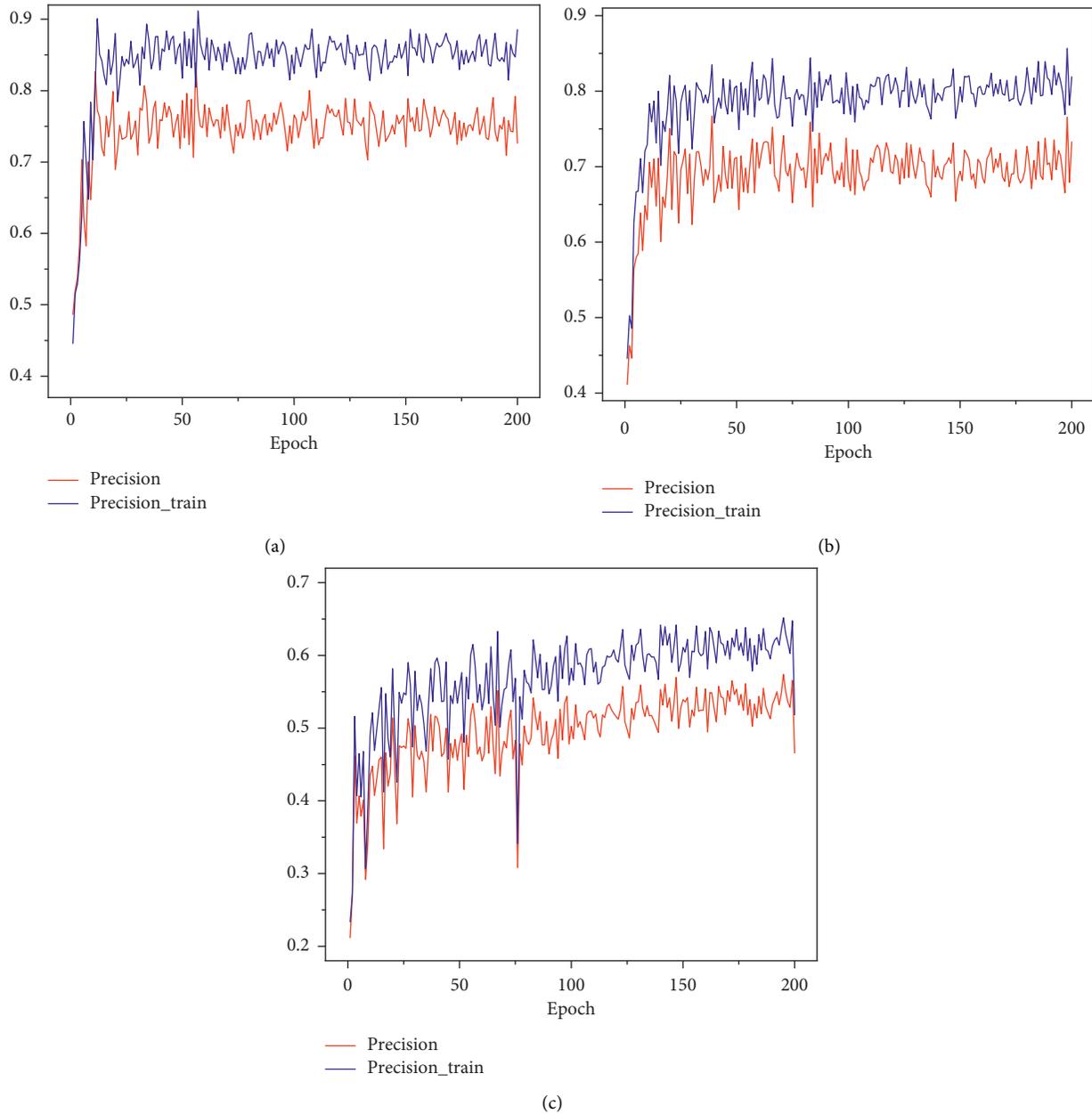


FIGURE 14: Curves of precision over training epoch.

segmentation results. Compared with the U-net model, the segmentation results of the improved U-net model in this article are most similar to the labeled images, with lower leak detection rate and false detection rate, higher segmentation precision, and shorter segmentation time for single images. This is because the improved U-net model not only has a jump-connected structure, but also uses the Inception module and ASPP module to process the feature map before upsampling, which expands the receptive field of the

convolutional layer, so that the image retains more information during the convolutional process and has a powerful ability to identify the crack region.

*5.5. Analysis for Identification Crack Widths.* According to the binary image of concrete cracks obtained by the improved U-net model, MATLAB image processing technology is used to identify the crack width. The specific calculation steps are as follows:

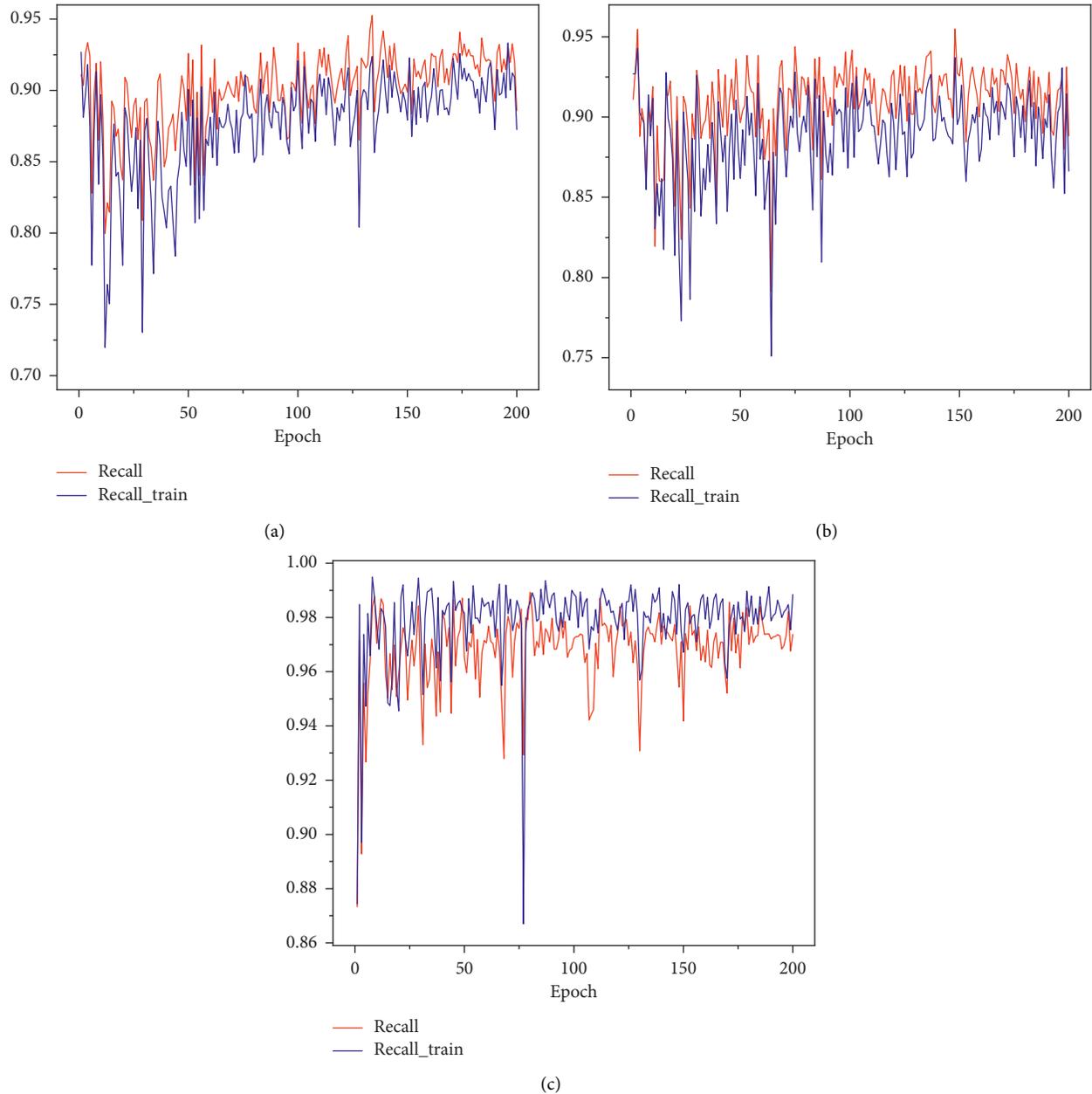


FIGURE 15: Curves of recall over training epoch.

- (1) `Bwskel ()` is used to get its skeleton matrix, as is shown in Figure 17(a).
  - (2) `Bwdist ()` is used to calculate the minimum Euclidean distance from the point in the crack to the crack boundary.
  - (3) Two image matrices are multiplied to obtain an eigen matrix. The value in the eigen matrix will be the radius along the crack skeleton.
  - (4) The eigenvalue is multiplied by 2 to get the width (diameter) of each crack, as is shown in Figure 18(b).
  - (5) The average crack width (presented by pixel values) of the crack segment is calculated.
  - (6) The actual distance represented by each pixel is taken as the scale for conversion to obtain the crack width.
- 20 crack image regions are selected. HC-F800 crack comprehensive tester is used to measure the actual crack width. The results using the proposed crack identification method are compared with the result by the HC-F800 crack comprehensive tester. Comparison results of crack widths are listed in Table 9. The relative error of the proposed model is shown in Figure 19. The average relative error using the proposed method is 13.2%. The average relative error measured by the ACTIS system is 31.8%. It can be concluded that the proposed method is accurate and robust.

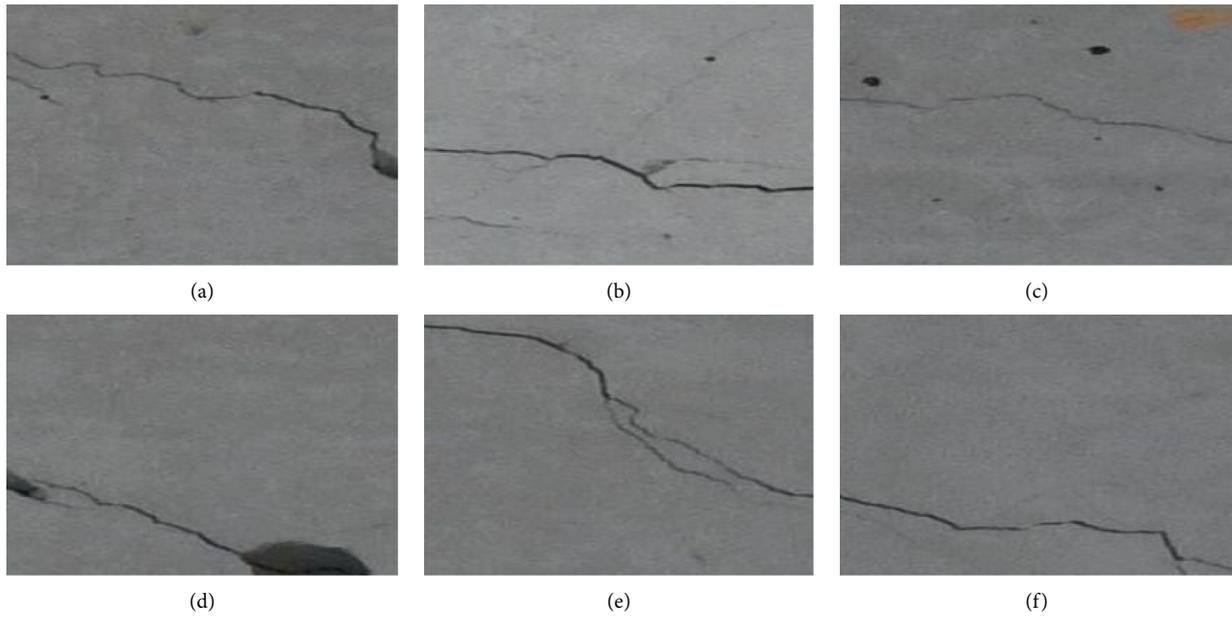


FIGURE 16: Original images of the test set.

TABLE 4: Comparison of precision of the three model test sets.

Image	U-net	Segnet	Improved U-net
Figure 12(a)	0.692	0.528	0.865
Figure 12(b)	0.696	0.465	0.832
Figure 12(c)	0.712	0.564	0.842
Figure 12(d)	0.735	0.535	0.808
Figure 12(e)	0.742	0.549	0.831
Figure 12(f)	0.724	0.573	0.823

TABLE 5: Comparison of recall of the three model test sets.

Image	U-net	Segnet	Improved U-net
Figure 12(a)	0.879	0.884	0.920
Figure 12(b)	0.883	0.933	0.935
Figure 12(c)	0.931	0.967	0.982
Figure 12(d)	0.924	0.932	0.973
Figure 12(e)	0.968	0.875	0.969
Figure 12(f)	0.889	0.884	0.956

TABLE 6: Comparison of mean\_IoU of the three model test sets.

Image	U-net	Segnet	Improved U-net
Figure 12(a)	0.759	0.749	0.819
Figure 12(b)	0.810	0.711	0.815
Figure 12(c)	0.816	0.766	0.832
Figure 12(d)	0.808	0.751	0.823
Figure 12(e)	0.817	0.758	0.827
Figure 12(f)	0.799	0.771	0.806

TABLE 7: Comparison of the maximum values of training evaluation indicators.

Item	U-net	Segnet	Improved U-net
Precision	0.855	0.615	0.966
Recall	0.952	0.895	0.951

TABLE 8: Comparison of the averages of the evaluation indicators.

Item	U-net	Segnet	Improved U-net
Average precision of the test set	0.727	0.536	0.844
Average recall of the test set	0.912	0.876	0.915
Average mean_IoU value of the test set	0.801	0.751	0.830

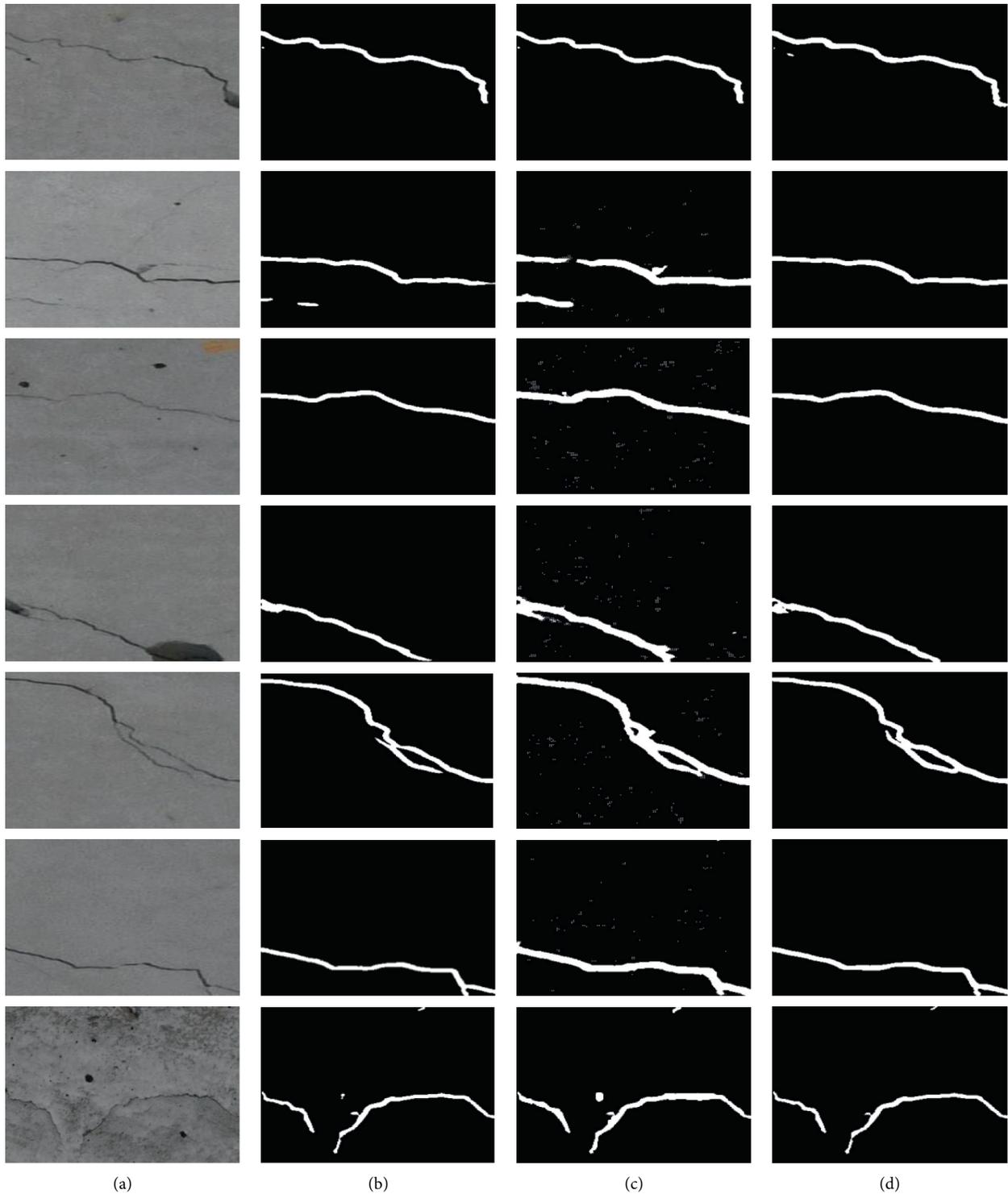


FIGURE 17: Comparison of segmentation results of the three models. (a) Original images. (b) Image processed with U-net. (c) Image processed with Segnet. (d) Image processed with Improved U-net.

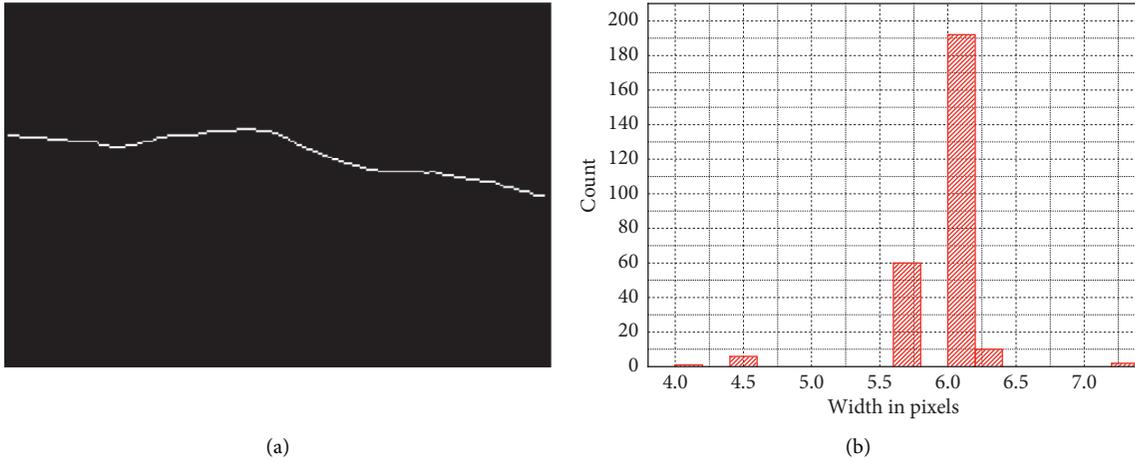


FIGURE 18: Measurement of crack widths. (a) Fracture skeleton diagram. (b) Histogram of crack width distribution.

TABLE 9: Comparison of crack widths.

Crack number	Plotting scale	Measured result	The number of pixels	Test results	Error value	Relative error (%)
1	0.0887	0.40	4.6	0.412	0.014	8.3
2	0.0887	0.38	4.1	0.367	0.008	8.5
3	0.0930	0.48	5.8	0.541	0.061	11.2
4	0.0930	0.37	4.6	0.472	0.059	14.1
5	0.0899	0.43	5.5	0.501	0.071	14.5
6	0.0899	0.63	4.6	0.413	0.053	13.0

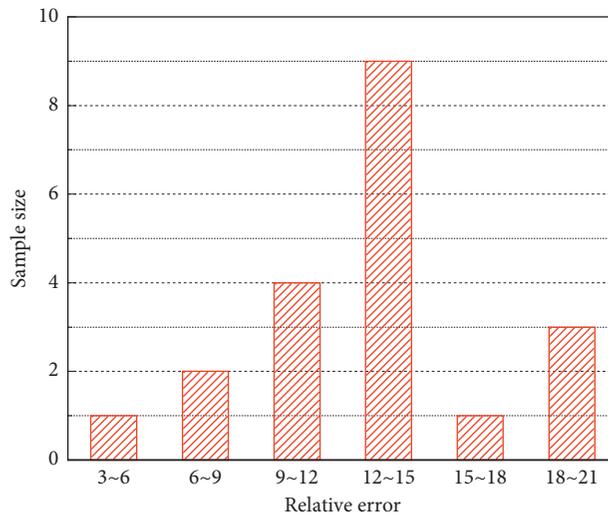


FIGURE 19: Diagram of relative errors of the proposed mod.

### 6. Conclusions

A crack identification method using improved U-net neural networks is presented for the digital image of concrete cracks with complex backgrounds. An improved inception module and an ASPP module based on the U-net neural networks are added in the improved U-net neural network model. A new loss function GDL is used to improve the sensitivity of networks to crack pixels. According to the concrete crack

binary images obtained from the improved U-net model, the widths of cracks are identified using MATLAB image processing technology. The average precision of the test set obtained using the proposed method is 11.7% higher than that of the U-net model. The average relative error of the crack width is 13.2%, which is 18.6% less than that obtained by the ACTIS system. The results indicate that the proposed method in this article has high precision and good robustness, which is a foundation and convenience method for

crack identification in concrete structures. In the following research, the Android software based on the algorithm will be developed to realize the extensive application in practical engineering.

### Data Availability

All the data used to support the findings of this study are available upon request from the corresponding author.

### Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.”

### Acknowledgments

The works was supported by the Transportation Department of Inner Mongolia Autonomous Region (Project no. NJ-2018-27).

### References

- [1] X. Tong, J. Guo, and Y. Ling, “A new image-based method for concrete bridge bottom crack detection,” in *Proceedings of the 2011 International Conference on Image Analysis and Signal Processing IEEE*, pp. 568–571, Wuhan, China, October 2011.
- [2] N. D. Hoang, “Detection of surface crack in building structures using image processing technique with an improved Otsu method for image thresholding,” *Advances in Civil Engineering*, vol. 2018, Article ID 3924120, , 2018.
- [3] Z. Xu, X. Zhao, and H. Song, “Asphalt pavement crack recognition algorithm based on histogram estimation and shape analysis,” *Chinese Journal of Scientific Instrument*, vol. 10, pp. 2260–2265, 2010.
- [4] A. M. A. Talab, Z. Huang, F. Xi, and L. HaiMing, “Detection crack in image using otsu method and multiple filtering in image processing techniques,” *Optik*, vol. 127, no. 3, pp. 1030–1033, 2016.
- [5] G. Wu, X. Sun, and L. Zhou, “Research on crack detection algorithm of asphalt pavement,” in *Proceedings of the IEEE International Conference on Information & Automation*, Lijiang, China, August 2015.
- [6] L. Sun, M. Kamaliardakani, and Y. Zhang, “Weighted neighborhood pixels’ segmentation method for automated detection of cracks on pavement surface images,” *Journal of Computing in Civil Engineering*, vol. 30, no. 2, Article ID 04015021, 2013.
- [7] Q. Zou, Y. Cao, Q. Li, Q. Mao, and S. Wang, “CrackTree: automatic crack detection from pavement images,” *Pattern Recognition Letters*, vol. 33, no. 3, pp. 227–238, 2012.
- [8] L. Zhang, F. Yang, and D. Zhang, “Road crack detection-using deep convolutional neural network,” in *Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP)*, pp. 3708–3712, Phoenix, AZ, USA, September 2016.
- [9] A. Zhang, K. C. P. Wang, B. Li et al., “Automated pixel-level pavement crack detection on 3D asphalt surfaces using a deep-learning network,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, no. 10, pp. 805–819, 2017.
- [10] Y.-J. Cha, W. Choi, and O. Büyüköztürk, “Deep learning-based crack damage detection using convolutional neural networks,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, no. 5, pp. 361–378, 2017.
- [11] F. C. Chen and M. R. Jahanshahi, “NB-CNN: deep learning-based crack detection using convolutional neural network and Naïve Bayes data fusion,” *IEEE Transactions on Industrial Electronics*, vol. 65, no. 5, pp. 4392–4400, 2017.
- [12] S. Li and X. Zhao, “Convolutional neural networks-based crack detection for real concrete surface,” *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems*, vol. 10598, Article ID 105983V, 2018.
- [13] X. Yang, H. Li, Y. Yu, X. Luo, T. Huang, and X. Yang, “Automatic pixel-level crack detection and measurement using fully convolutional network,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 12, pp. 1090–1109, 2018.
- [14] S. Y. Zhu, J. Du, and Y. Li, “Method for bridge crack detection based on the U-Net convolutional networks,” *Journal of Xidian University*, vol. 46, no. 4, pp. 35–42, 2019.
- [15] L. C. Chen, G. Papandreou, F. Schroff, and H. Adam, “Rethinking atrous convolution for semantic image segmentation,” 2017, <https://arxiv.org/abs/1706.05587>.
- [16] L. C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 801–818, Munich, Germany, September 2018.
- [17] G. Zheng, G. Han, and N. Q. Soomro, “An inception module CNN classifiers fusion method on pulmonary nodule diagnosis by signs,” *Tsinghua Science and Technology*, vol. 25, no. 3, pp. 368–383, 2019.
- [18] S. Santurkar, D. Tsipras, A. Ilyas, and A. Madry, “How does batch normalization help optimization?” *Advances in Neural Information Processing Systems*, vol. 31, pp. 2483–2493, 2018.
- [19] W. R. Crum, O. Camara, and D. L. Hill, “Generalized overlap measures for evaluation and validation in medical image analysis,” *IEEE Transactions on Medical Imaging*, vol. 25, no. 11, pp. 1451–1461, 2016.
- [20] L. N. Smith, “Cyclical learning rates for training neural networks,” in *Proceedings of the 2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 464–472, Santa Rosa, CA, USA, March 2017.
- [21] M. A. Rahman and Y. Wang, “Optimizing intersection-over-union in deep neural networks for image segmentation,” in *Proceedings of the International Symposium on Visual Computing*, pp. 234–244, Las Vegas, NV, USA, December 2016.