Front. Struct. Civ. Eng. 2023, 17(5): 732–744
https://doi.org/10.1007/s11709-023-0965-y

RESEARCH ARTICLE

Fast detection algorithm for cracks on tunnel linings based on deep semantic segmentation

Zhong ZHOU[a,b], **Yidi ZHENG**[a], **Junjie ZHANG**[a], **Hao YANG**[b,c*]

[a] *School of Civil Engineering, Central South University, Changsha 410000, China*
[b] *National Engineering Research Center of Highway Maintenance Technology, Changsha University of Science & Technology, Changsha 410000, China*
[c] *School of Traffic and Transportation Engineering, Changsha University of Science & Technology, Changsha 410000, China*
[*] *Corresponding author. E-mail: hyang_cityu@163.com*

ABSTRACT An algorithm based on deep semantic segmentation called LC-DeepLab is proposed for detecting the trends and geometries of cracks on tunnel linings at the pixel level. The proposed method addresses the low accuracy of tunnel crack segmentation and the slow detection speed of conventional models in complex backgrounds. The novel algorithm is based on the DeepLabv3+ network framework. A lighter backbone network was used for feature extraction. Next, an efficient shallow feature fusion module that extracts crack features across pixels is designed to improve the edges of crack segmentation. Finally, an efficient attention module that significantly improves the anti-interference ability of the model in complex backgrounds is validated. Four classic semantic segmentation algorithms (fully convolutional network, pyramid scene parsing network, U-Net, and DeepLabv3+) are selected for comparative analysis to verify the effectiveness of the proposed algorithm. The experimental results show that LC-DeepLab can accurately segment and highlight cracks from tunnel linings in complex backgrounds, and the accuracy (mean intersection over union) is 78.26%. The LC-DeepLab can achieve a real-time segmentation of 416 × 416 × 3 defect images with 46.98 f/s and 21.85 Mb parameters.

KEYWORDS tunnel engineering, crack segmentation, fast detection, DeepLabv3+, feature fusion, attention mechanism

1 Introduction

With the increase in infrastructure stock, structural defects and the associated maintenance needs are becoming increasingly common in China [1–3]. In tunnel construction, lining cracks have adverse effects on the performance and service life of the lining structure [4–6], which may further lead to other defects, such as leakage and concrete spalling. Therefore, the timely detection of defects on surfaces is critical for maintaining tunnel safety.

Traditional detection methods for structural defects typically rely on onsite manual inspection, which significantly wastes time and requires extensive effort [7]. Many nondestructive methods, such as acoustic emission [8], visual imaging [9], and ultrasonic tomography [10], are used in tunnel inspection to improve detection efficiency. However, these methods require manual adjustment of the parameters, and the detection effect is unsatisfactory [11]. With the development of computer technology, image processing techniques (IPTs) and deep learning algorithms have been gradually applied for detecting structural surface defects [12,13]. Although IPTs can rapidly detect structural defects, they require manually designed features, and the detection effect is severely limited by the complexity of the background [14].

Deep learning algorithms use convolutional neural networks (CNNs) to extract defect features with improved generalization and anti-interference abilities [15,16]. Regarding the difference in effects, defect detection methods based on deep learning are mainly

Article history: Received Sep 7, 2022; Accepted Dec 22, 2022

divided into object detection and semantic segmentation. Object detection algorithms highlight defects using rectangular frames and belong to the object-level detection category. To overcome the shortcomings of IPTs, Zhang et al. [17] applied deep learning to the automatic detection of pavement cracks and found that the recognition effect of this method was better than those of the support vector machine and boosting methods. Cha et al. [18] proposed a deep architecture of CNNs without calculating the defect features and accurately detected cracks using multiple rectangular frames. With improvements in inspection equipment, digital single-lens reflex cameras, unmanned aerial vehicles, and depth cameras have also been used for defect detection [19,20]. Zhou et al. [21] improved the YOLO series algorithm [22,23], which uses a lightweight backbone network to extract target features and achieves the real-time detection of various tunnel lining defects. Compared with IPTs, widely used object detection algorithms such as faster regions with CNN (R-CNN) [24], the single-shot detector [25], and YOLO can precisely locate and classify target positions. However, these algorithms cannot quantitatively describe geometric parameters, such as area, length, and orientation, which are not beneficial for accurately evaluating defects.

The semantic segmentation algorithm directly separates the defect pixels from the background, and the defects are visually displayed in the predicted images. To achieve pixel-level detection, Long et al. [26] designed a fully convolutional network (FCN) that converts all fully connected layers into convolutions in CNNs and achieves semantic segmentation through pixel-by-pixel classification. Since then, many scholars have investigated semantic segmentation algorithms [27,28]. Liu et al. [29] used the U-Net model to detect concrete defects under various lighting and cluttered backgrounds. The results showed that U-Net [30] was more efficient than FCN. However, U-Net produces redundant recognition in complex backgrounds. Ji et al. [31] used the DeepLabv3+ model [32] to evaluate five important indicators: fracture length, average width, maximum width, area, and proportion. The method effectively achieved accurate fracture segmentation and quantification. Zhou et al. [33] improved the DeepLabv3+ model and proposed a water leakage segmentation algorithm that could detect the morphological features and spatial distribution of defects in real time. Ali and Cha [34] designed a generative model for defective images based on the attention mechanism and used these images to train the proposed segmentation model.

Previous studies have shown that deep-learning-based semantic segmentation algorithms can detect tunnel cracks intelligently. However, when these algorithms detect tunnel lining cracks in complex environments, the following problems remain.

(1) The model parameters are very large. Traditional deep learning algorithms use many parameters to improve accuracy, which decreases the model detection speed, that is, the frames per second (*FPS*). Achieving rapid and intelligent detection of tunnel cracks can be difficult when the algorithm is embedded in mobile detection equipment.

(2) The edge of the crack is discontinuous. The proportion of tunnel cracks in actual lining images is low. Moreover, the edge information of the cracks is lost after multiple downsamplings, causing existing semantic segmentation algorithms to be unable to achieve high recognition accuracy.

(3) The model has a weak anti-interference ability. Various defects coexist on the surface of the tunnel lining, and the complex background and diverse illuminations interfere with crack identification. The detection accuracy of currently used models does not support practical applications.

A high-precision and lightweight crack segmentation model called LC-DeepLab, based on the DeepLabv3+ structure, was proposed in this study to solve these problems. The main objectives of this study are as follows.

(1) Use a lightweight backbone network. A lightweight neural network, MobileNetV3 [35], was adopted as the backbone network of the model. Depthwise separable convolution was applied to reduce the size of the parameters and the H-swish activation function and the improved squeeze-and-excitation networks (SENet) [36] were adopted to maintain accuracy.

(2) Design a shallow feature fusion module. We extracted half of the shallow design feature fusion module and used dilated convolution to extract crack position information across pixels, resulting in improved accurate crack edges.

(3) Apply the attention module. The method was combined with ECANet [37] after feature fusion to improve the anti-interference ability of the model. ECANet receives information through adaptive one-dimensional convolution cross-channel interaction and detects tunnel cracks in complex backgrounds and illumination.

2 Methodology

2.1 Overview of LC-DeepLab

Many CNN-based segmentation models apply an encoder–decoder structure from an FCN. Based on this approach, some methods have been applied to improve the detection accuracy or speed of models, such as an attention [38] and a spatial pyramid pooling (SPP) modules [39]. Based on the DeepLabv3+ framework, a fast detection algorithm called LC-DeepLab, which is suitable for complex tunnel lining crack detection, was proposed in this study. The overall structure of the proposed LC-DeepLab is depicted in Fig. 1. The model consisted of three main parts: the modified MobileNetV3 for extracting backbone features, the shallow feature fusion and attention modules, and an encoder–decoder structure from DeepLabv3+. Specific strategies of the proposed model are described in the following section.

2.2 MobileNetV3 and improvement methods

The MobileNetV3 structure uses an inverse residual structure and depth-wise separable convolution. The attention module (SENet) and the activation function (H-swish) were used to improve accuracy. The MobileNetV3 structure consisted of a standard convolutional layer, 15 bottleneck layers, and three pointwise convolutional layers. In some bottlenecks, 5 × 5 convolutions were used instead of 3 × 3, and two layers of pointwise convolutions were used at the end of the network instead of batch normalization. The structure of this bottleneck is illustrated in Fig. 2. These changes ensure a high accuracy while minimizing the number of calculations and parameters.

The proposed LC-DeepLab uses the lightweight network, MobileNetV3, to replace Xception [40] in DeepLabv3+ for feature extraction. However, the MobileNetV3 structure contains five downsamplings, whereas the original Xception uses only four downsamplings. Excessive downsampling may result in information loss owing to the characteristics of tiny cracks, and thus, MobileNetV3 is improved. Dilated convolution was applied to the MobileNetV3 structure to reduce information loss caused by downsampling. The last two downsampling layers are contained in the 7th and 13th bottleneck layers in MobileNetV3. From the seventh bottleneck, the stride of convolutions is changed in all bottlenecks to 1, and dilated convolutions with different dilation rates are applied to maintain the receptive field (Table 1).

Atrous convolution gains an increased receptive field by adding holes to the improved MobileNetV3. For the same amount of computation, atrous convolution can effectively prevent the reduction in image resolution caused by downsampling [41]. The atrous convolution structure is shown in Fig. 3. If the receptive field of a standard convolution is 3 × 3, the receptive field of an atrous convolution with a dilation rate of 2 is 5 × 5.

2.3 Shallow feature fusion module

In the training of many CNN-based segmentation models, deep features are extracted via numerous deep convolution operations, which gradually reduce the image resolution. The detection accuracy decreases significantly

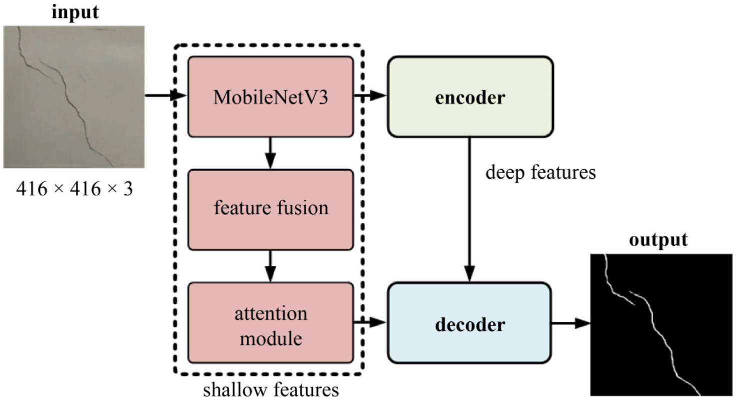

Fig. 1 Overall structure of proposed LC-DeepLab.

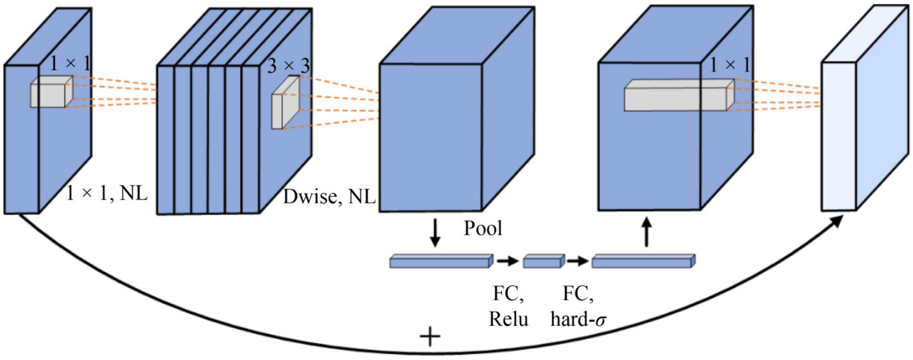

Fig. 2 Bottlenecks of MobileNet v3. Note: NL is a nonlinear activation function; Dwise is a depth-wise separable convolution; Pool is a pooling layer; FC is a fully connected layer.

for defects with a low percentage of pixels in the original image, such as cracks. Pixel classification depends on both high- and low-dimensional features [42]. Thus, the prediction effect is improved by fusing the features of different dimensions [43]. Hence, a shallow feature fusion module is proposed to utilize the location information of the shallow features (Fig. 4).

Table 1 Improvement method for MobileNet v3 network structure

bottleneck	kernel size	stride		dilation rate
		original	improved	
7	3 × 3	2	1	2
8	3 × 3	1	1	2
9	3 × 3	1	1	2
10	3 × 3	1	1	2
11	3 × 3	1	1	2
12	3 × 3	1	1	2
13	5 × 5	2	1	4
14	5 × 5	1	1	4
15	5 × 5	1	1	4

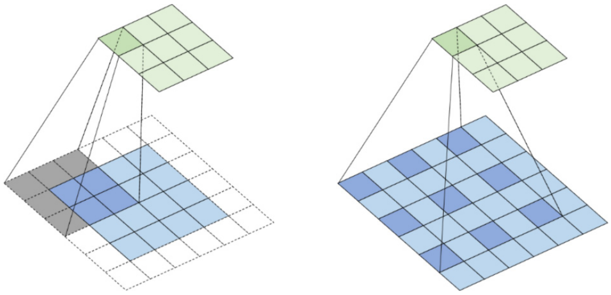

Fig. 3 Standard convolution and atrous convolution.

The lower part is the feature fusion module used to extract shallow features. Here, the feature layers generated by Block 2 and Block 3 in the backbone network MobileNetV3 of DeepLabv3+ are combined. The feature map sizes of Block 2 and Block 3 were half and one-quarter of those of the original image, respectively. First, the feature map in Block 2 was extracted across pixel points using a 3 × 3 dilated convolution at a dilation rate of two. The size was then reduced by a 3 × 3 standard convolution with a stride of two. Finally, this feature map was combined with the feature map in Block 3 to obtain the final shallow features. This final feature layer has richer semantic and spatial information of Block 2 and Block 3, which helps to enhance the segmentation details of the cracks.

2.4 Attention module of ECANet

In a deep CNN, the attention module can effectively improve the adaptability of the CNN. SENet is a representative method that filters the channel weights through feature squeezing and excitation. SENet obtains all channel information through dimensionality reduction and balances model performance with complexity.

However, Wang et al. [37] posit that it is inefficient for SENet to determine all channel relationships and propose a local cross-channel interaction attention mechanism called the ECANet. ECANet designs a method for adaptively selecting the size of one-dimensional convolution kernels to prevent degradation of the model performance caused by dimensionality reduction. Equation (1) is used to calculate the convolution kernel size k (cross-channel interaction coverage), which is related to the number of input channels.

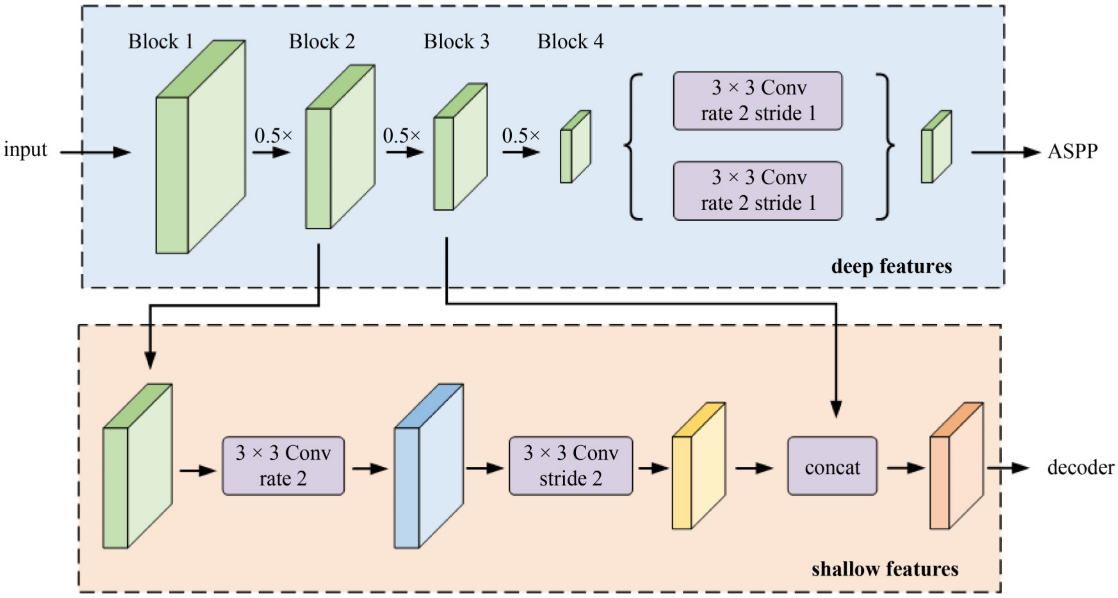

Fig. 4 Shallow feature fusion module.

$$k = |t|_{\text{odd}} = \left|\frac{\log_2 C}{\gamma} + \frac{b}{\gamma}\right|_{\text{odd}}, \tag{1}$$

where C is the current number of input channels, γ and b are constants with values of 2 and 1, respectively, and ‘odd’ is the nearest integer. The ECANet structure is depicted in Fig. 5.

The ECANet is applied to the proposed feature fusion module in LC-DeepLab, which reduces the discontinuity in spatial information caused by dilated convolution. In theory, the attention mechanism focuses on the noticed target pixel and enhances its weight, which can be imposed after all feature fusion parts. The corresponding ablation experiments are discussed in Subsubsection 4.3.2 to demonstrate the effectiveness of the proposed structure.

2.5 Encoder–decoder structure

The proposed LC-DeepLab framework adopts the encoder–decoder structure in DeepLabv3+ (Fig. 6). The encoder network consists of an improved MobileNetV3 and an atrous space pyramid pool (ASPP). Its main function is to compress the feature layer and extract target semantic information. The decoder network consists of multiple convolutional and upsampling layers that restore image size and perform pixel classification. The LC-DeepLab algorithm process for semantic segmentation is as follows.

In the encoder network, the image is input to the improved MobileNetV3 for feature extraction, and two preliminary effective feature layers are obtained. The shallow feature layer results from two downsamplings passed into the shallow feature fusion module. When the image is compressed four times, a deep feature layer is obtained and passed to the ASPP. The ASPP module includes a 1 × 1 convolutional layer, three parallel dilated convolutional layers with different dilation rates, and a pooling layer. After these steps, semantic information of

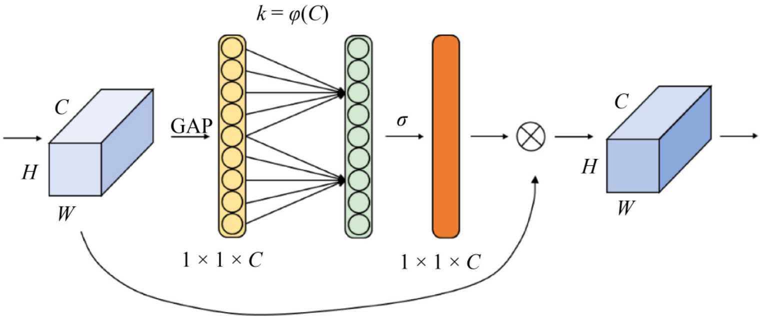

Fig. 5 Network structure of ECANet. Note: H, W, and C are the length, width, and number of channels of the input feature layer, respectively; GAP is the channel-level global average pooling without dimensionality reduction; k is the convolution kernel size; σ is the sigmoid activation function.

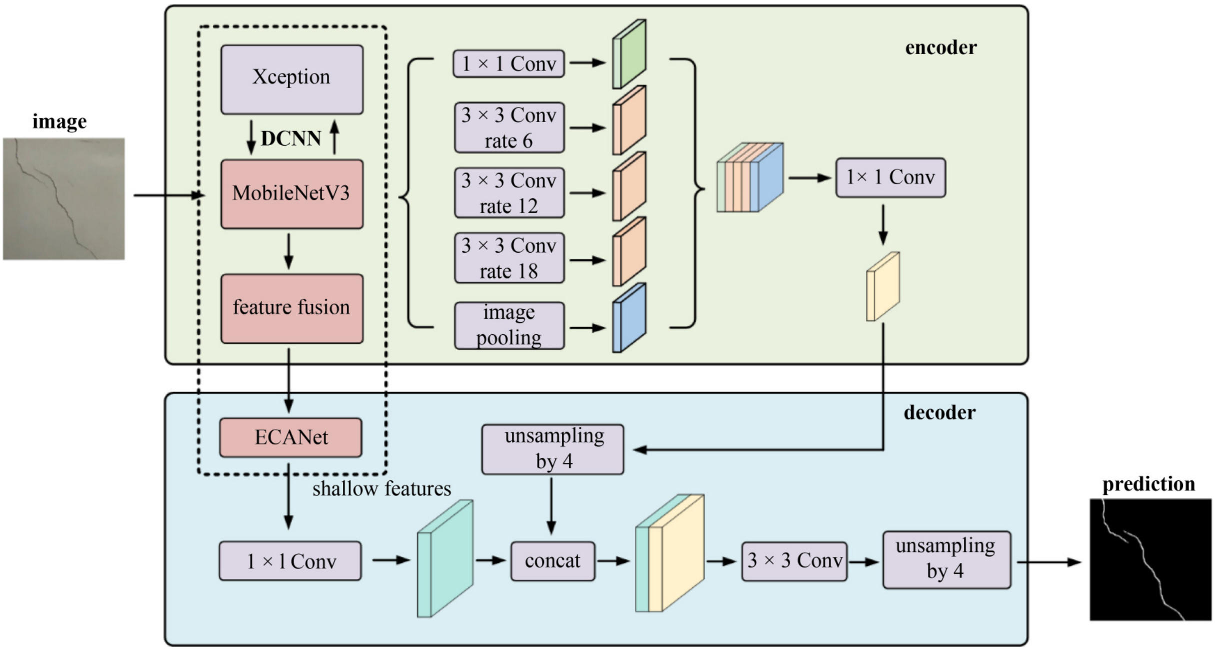

Fig. 6 Encoder–decoder structure of LC-DeepLab and improvement methods.

different scales in the deep feature layer is extracted. The processed deep feature layer is passed to the decoding part after a 1 × 1 convolutional dimension reduction.

In the decoder network, the original shallow feature layer is reduced by a 1 × 1 convolution, and the feature layer after the ASPP is upsampled. The two feature layers are stacked to form the final effective feature layer that condenses the semantic information of the entire image. The final feature layer restores the original size via upsampling and obtains semantic segmentation results after two depth-wise separable convolutions.

3 Model test

3.1 Tunnel lining crack data set

During almost 10 years of tunnel monitoring, the authors captured many photographs of tunnel lining defects. For the data set, 1336 crack images were selected, which contained various colors of light (such as blue, yellow, and black) and complex backgrounds (such as marks, potholes, and noise). The data set of the tunnel lining cracks is shown in Fig. 7.

When the number of photographs is insufficient, the model may undergo overfitting, affecting its generalization ability. Thus, the data set was augmented by rotating, cropping, adding noise, and adjusting brightness (Fig. 8). Finally, the data set containing 2093 complex and diverse crack images was divided into training, validation, and test sets in the ratio of 8:1:1. These sets comprised 1673 training images, 210 validation images, and 210 test images (Table 2). The image size was standardized to 416 × 416 × 3. This division ensured that the original and expanded images were in the same training set.

The tunnel lining crack detection adopted supervised learning. The model learned the mapping relationship between the original and label images in the training set, adjusted the parameters through backpropagation in the validation set, used the mapping relationship to segment the images in the test set, and obtained the model accuracy index. The original images were labeled at the pixel level using the labeling software, LabelMe, to obtain the label images. The label images were maintained to correspond to the original images (Fig. 9).

3.2 Training environment settings

The experimental algorithm was constructed using Python3.8, and Pytorch1.10.0 was used as the deep learning framework. The operating system used for the experiment was Windows 10, the processor (central processing unit) was AMD Ryzen 7 5800H, and the graphics processing unit was NVIDIA GeForce RTX 3060.

Because of the large number of network parameters, the frozen training method was used to improve training efficiency. The model was trained for 200 epochs, and the backbone network, MobileNetV3, was frozen in the first 50 epochs. The network was then fine-tuned. During the

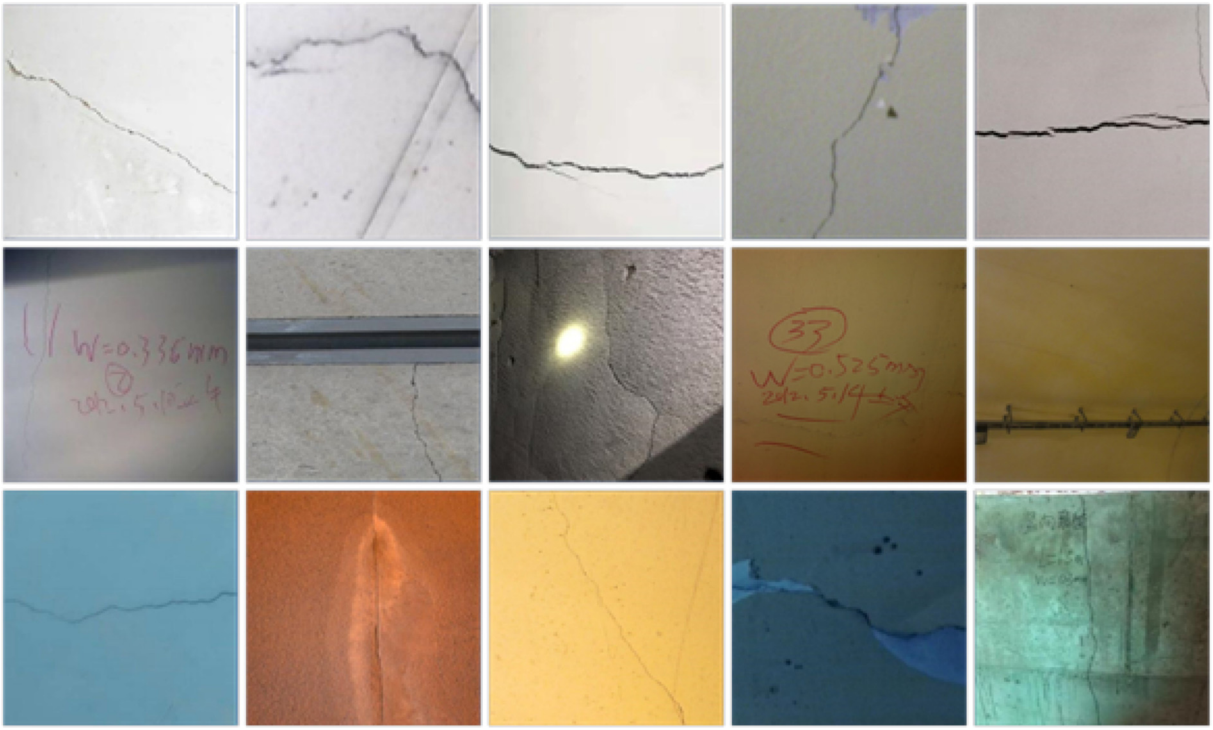

Fig. 7 Data set of tunnel lining cracks.

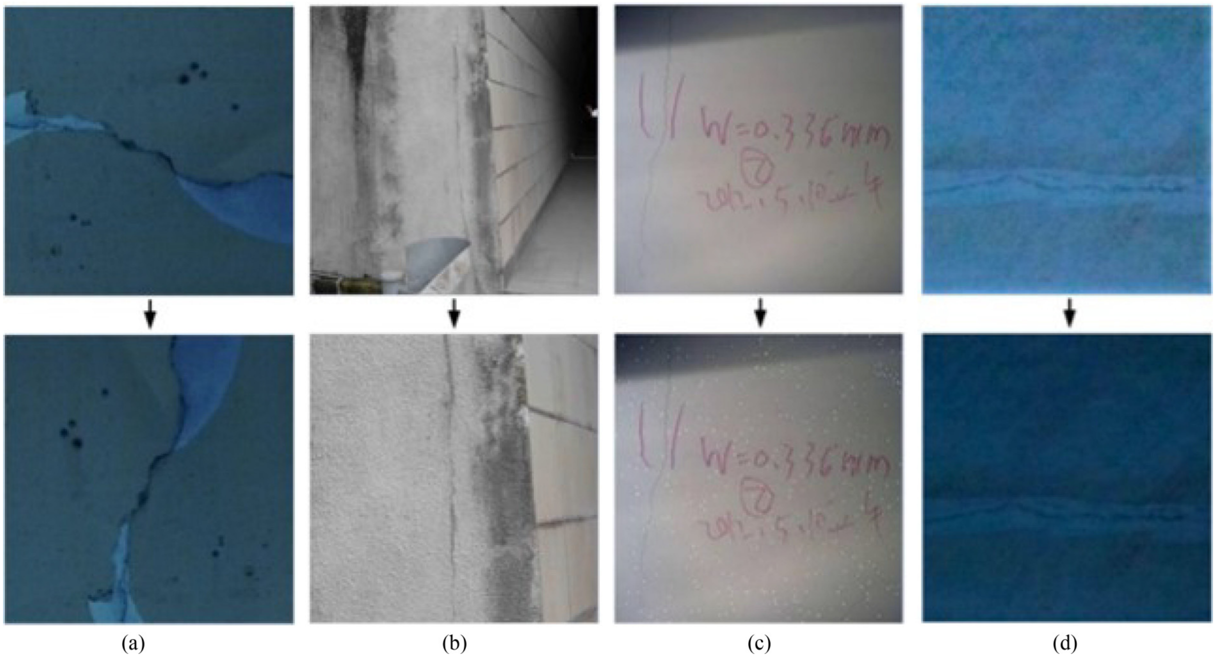

Fig. 8 Expanded data set: (a) rotation; (b) cropping; (c) noise; (d) brightness.

Table 2 LC-DeepLab data set

model	training	validation	test	image size
LC-DeepLab	1673	210	210	416 × 416 × 3

subsequent 150 epochs, the backbone network was unfrozen, and the entire model was trained. Hyperparameters such as the batch size and learning rate were optimized to improve the training efficiency [44]. The optimization results are presented in Table 3. The model used Adam as the optimizer and StepLR as the learning-rate adjustment technique. Before model training, the images were uniformly resized to 416 × 416 pixels via size normalization.

3.3 Evaluation metrics

3.3.1 Training evaluation metrics

The loss function was the basis for determining the convergence trend of the model during training. The *Dice loss* [45] was used as the loss function in this study, and the model continued to converge when the *Dice loss* showed a downward trend. The model used the training

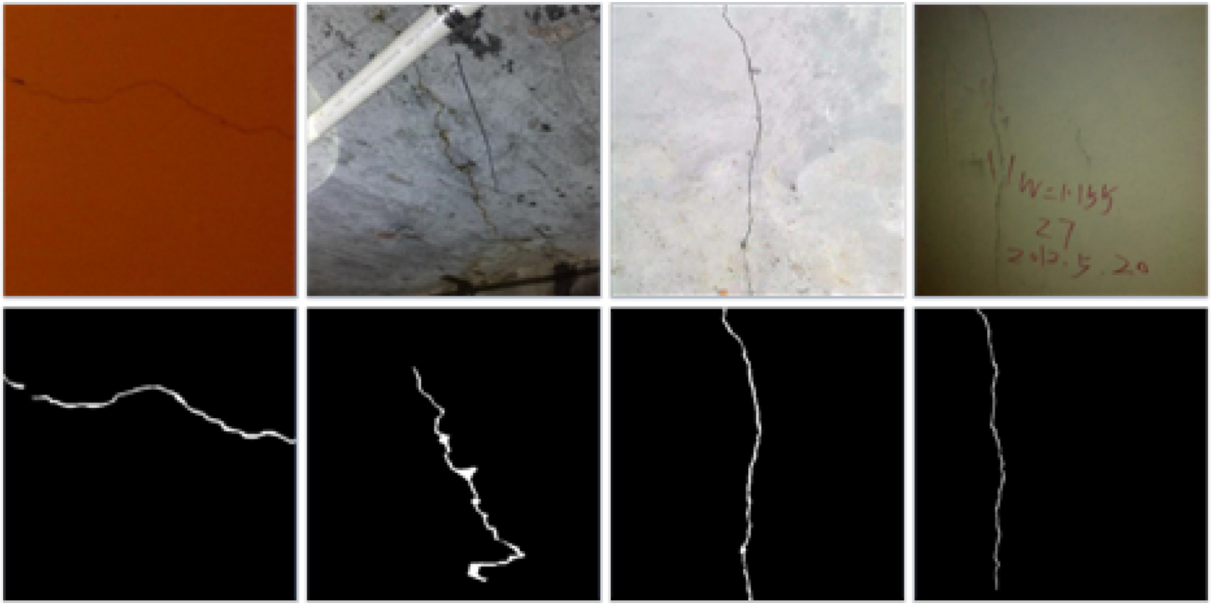

Fig. 9 Image annotation method.

Table 3 Network hyperparameters

hyperparameter	value	
	freeze training	unfrozen training
epoch	50	150
batch size	16	4
learning rate	5×10^{4}	5×10^{5}

set to train the parameters and determine whether the model converged using the validation set. In addition, the validation set adjusted the network parameters through backpropagation to ensure that the *Dice loss* continued decreasing until the model converged fully. Equation (2) was used to compute the *Dice loss*.

$$Dice\ loss = 1 - \frac{2\sum_{i}^{N} p_i q_i + \varepsilon}{\sum_{i}^{N} p_i^2 + \sum_{i}^{N} q_i^2 + \varepsilon}, \tag{2}$$

where p_i is the predicted binary probability, q_i is the true binary probability, ε is a minimum, and i is the input value.

3.3.2 Predictive evaluation metrics

The accuracy evaluation indicators used in this study were the mean pixel accuracy (*mPA*) and the mean intersection over union (*mIoU*). *mPA* is a simple evaluation metric, and *mIoU* is a standard metric for evaluating semantic segmentation [46]. Therefore, *mIoU* was used as the main metric to evaluate the accuracy of the model. Equations (3) and (4) were used to calculate *mPA* and *mIoU*, respectively.

$$mPA = \frac{1}{u+1} \sum_{m=0}^{u} \frac{X_{mm}}{T_m}, \tag{3}$$

where u is the total number of categories of the predicted target, m is the predicted target category, X_{mm} is the total number of pixels successfully predicted in category m, and T_m is the total number of pixels in category m.

$$mIoU = \frac{1}{u+1} \sum_{m=0}^{u} \frac{p_{mm}}{\sum_{n=0}^{u} p_{mn} + \sum_{n=0}^{u} p_{nm} - p_{mm}}, \tag{4}$$

where m and n are different target categories, and p_{mn} is the probability of predicting category m as category n.

FPS was used as the speed evaluation index to analyze the effects of the lightweight improvement of the model on the prediction speed. Equation (5) was used to calculate the *FPS*.

$$FPS = \frac{Num}{Time}, \tag{5}$$

where *Num* is the number of predicted pictures, and *Time* is the time corresponding to the prediction.

4 Test results and discussion

4.1 Loss curves of LC-DeepLab

The training and validation losses were calculated after each training epoch of the LC-DeepLab model. The loss curves are shown in Fig. 10.

The loss value sharply decreased at the beginning of the frozen training, indicating an appropriate learning rate and rapid model convergence. The loss curve gradually flattened as the number of training generations increased. The backbone network remained unfrozen after the 50th epoch. The model loss first increased slightly, decreased gradually, and stabilized, indicating that the network converged. Finally, the training loss of the LC-DeepLab model was stable at approximately 0.13, and the validation loss was stable at approximately 0.16, indicating that the model did not undergo overfitting during the training process.

4.2 Comparison and analysis of different models

Four models (FCN, PSPNet [47], U-Net, and DeepLabv3+) were compared with LC-DeepLab to assess the model performance. The four semantic segmentation models are briefly introduced as follows.

(1) FCN: The FCN supports image inputs of any size and restores feature maps to their original size by upsampling and deconvolution. The segmentation problem is transformed into a pixel classification problem, and end-to-end segmentation can be achieved.

(2) PSPNet: This model designs a unique pyramid pooling module that divides the feature layer into grids of different sizes. In addition, it performs average pooling to aggregate the context information of different regions and

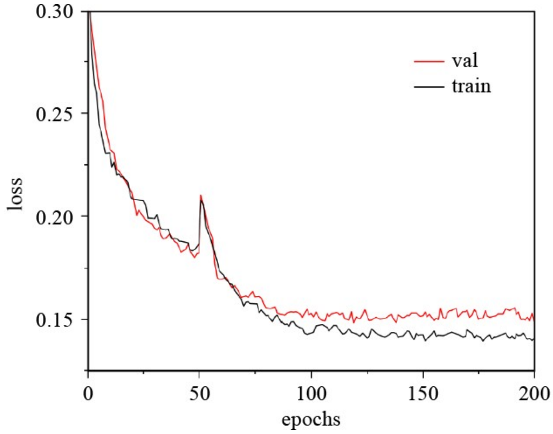

Fig. 10 Loss curve of LC-DeepLab model.

improves the ability to obtain full-text information.

(3) U-Net: The U-Net designs the encoder–decoder structure and splices the feature map of the encoder to the decoding feature map of the corresponding stage. This enables the decoder to learn the relevant information lost during the pooling of the encoder.

(4) DeepLabv3+: ASPP is designed for feature extraction and fusion at different magnifications, and the encoding–decoding structure reduces the loss of semantic information.

The other four classical semantic segmentation models were trained on the same data set, and the same test set was used to evaluate the various performance metrics. The results are presented in Table 4.

The evaluation experiment showed that the *mIoU* obtained using LC-DeepLab on this data set was 78.26%, which were 6.56%, 12.77%, 4.82%, and 7.05% higher than those of the FCN, PSPNet, U-Net, and DeepLabv3+, respectively. The *mPA* of LC-DeepLab was 85.80%, second only to U-Net (86.27%), indicating that LC-DeepLab effectively improved the segmentation accuracy of tunnel-lining cracks. After replacing the network backbone, the LC-DeepLab parameters were only 21.85 Mb, which is significantly lower than those of the other four segmentation algorithms. The FPS reached 46.98 f/s, and each tunnel defect image took only 21.3 ms on average. Therefore, the detection speed of LC-DeepLab is faster than those of other models, and tunnel-lining cracks can be detected in real time.

Figure 11 shows the predicted images of the cracks using different models. Manually detected handwriting was added to the detection images as interference, and the complexity of the images increased from Figs. 11(a) to 11(e). The experimental results showed that LC-DeepLab had the best segmentation effect. The predicted cracks were continuous and less disturbed by the environment, which was more evident when the images were enlarged (Figs. 11(e) and 11(f)). Although DeepLabv3+ and U-Net could predict continuous crack images, they were affected by severe background interference, and a significant part of the handwriting in the background was segmented into cracks (Figs. 11(e) and 11(f)). The FCN and PSPNet lost some crack pixels, which led to a discontinuity in the recognition results (Figs. 11(a) and 11(d)). However, FCN and PSPNet exhibited a stronger anti-interference ability than DeepLabv3+ and U-Net (Figs. 11(c) and 11(e)) and could still accurately distinguish cracks and artificial handwriting for interference.

4.3 Ablation experiments

4.3.1 Effect of different improvement strategies

Based on DeepLabv3+, LC-DeepLab replaces the backbone network with MobileNetV3 and applies the shallow feature fusion module and ECANet. Ablation experiments were conducted using the same data set and hyperparameters to verify the effects of these improvement strategies on the model prediction speed accuracy (Table 5).

The accuracy and speed evaluation results showed that after replacing the backbone network, the *mIoU* increased to 74.80%, the *mPA* increased to 85.99%, and the parameters decreased to 21.70 Mb. After applying the feature fusion and ECA modules, *mIoU* increased to 78.26%. The *mPA* remained at a high level, and the model parameters remained unchanged, indicating that the three improvement strategies of the LC-DeepLab model effectively improved model segmentation accuracy and speed.

Prediction images of DeepLabv3+ and LC-DeepLab under different background interferences (such as marks, depressions, and shadows) were selected to visually demonstrate the model improvement effect (Fig. 12). The first row shows the original image of the tunnel lining. Figures 12(a) and 12(b) show simple manual markings, Figs. 12(c) and 12(d) show varying lighting, and Figs. 12(e) and 12(f) show segments and depressions.

Partially manual annotations were identified as cracks (Figs. 12(a)–12(c)) when using DeepLabv3+. As the cracks narrowed or the lighting worsened, the cracks predicted by DeepLabv3+ became discontinuous (Fig. 12(d)), and some images were completely undetectable (Fig. 12(c)). In addition, low-light parts such as shadows of segments and dents in concrete were misjudged as cracks (Figs. 12(e) and 12(f)). After replacing the backbone network, the model could distinguish cracks from other disturbances more accurately (Figs. 12(c)–12(e)). The accuracy of the predicted crack orientation and edges improved after applying the feature fusion and ECA modules (Figs. 12(c), 12(e), and 12(f)).

4.3.2 Comparison of locations of ECA module application

Different control groups were established to identify the optimal position of the ECA module application. Considering that the attention module improves the adaptive ability of convolutional networks by suppressing irrelevant channel weights, we propose the application of ECANet after three feature fusion modules. These three positions are after the shallow feature fusion, ASPP structure, and deep and shallow feature fusion. The effect

Table 4 Evaluation metrics of model performance

method	amount of parameters (Mb)	*FPS* (f/s)	*mPA* (%)	*mIoU* (%)	ascension of *mIoU* (%)
LC-DeepLab	21.85	46.98	85.80	78.26	–
FCN	269.74	17.48	79.45	71.70	6.56
PSPNet	178.51	23.18	76.38	65.49	12.77
U-Net	94.97	20.04	86.27	73.44	4.82
DeepLabv3+	209.70	12.00	79.06	71.21	7.05

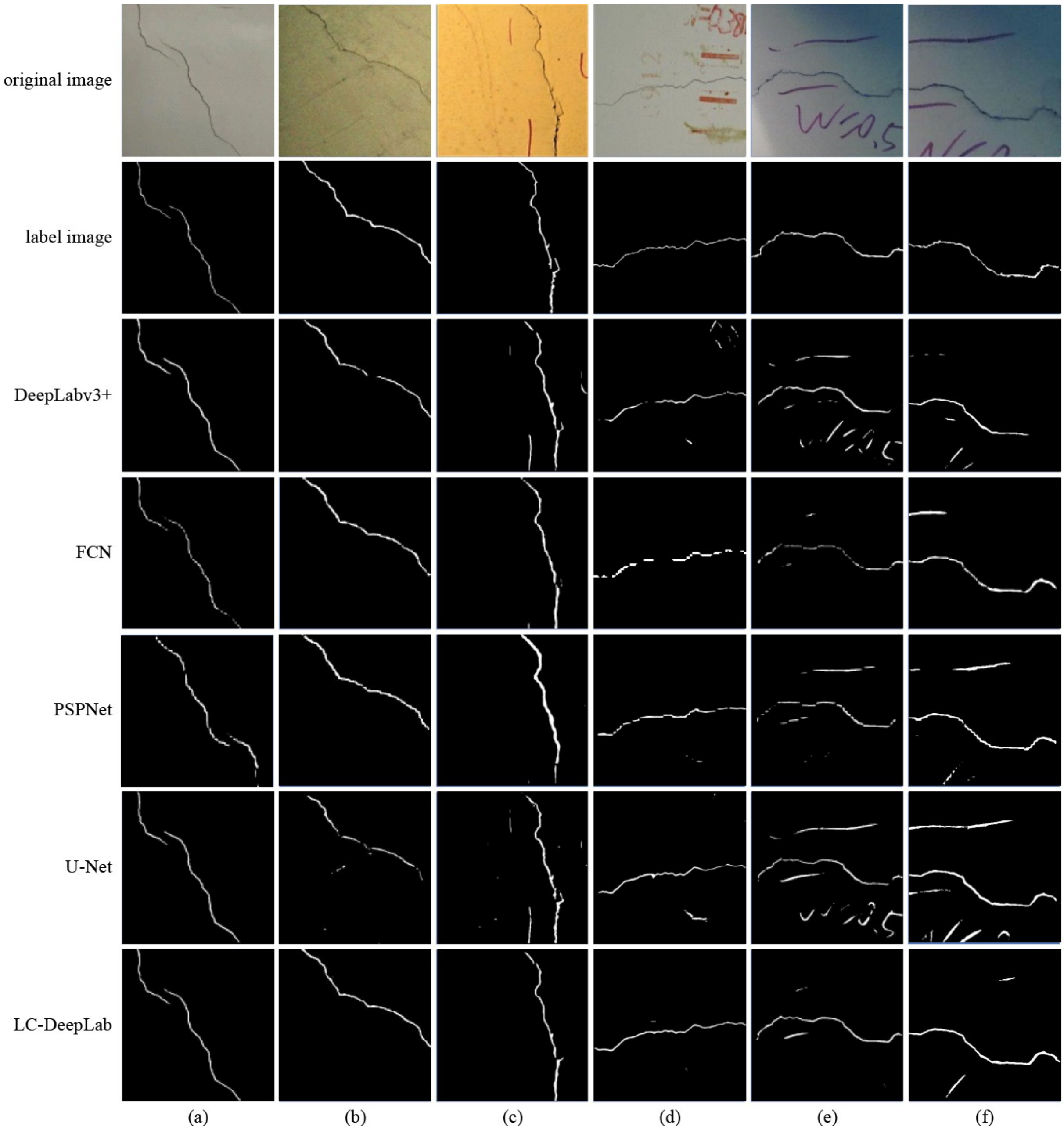

Fig. 11 Test images of different models. (a) Test 1; (b) Test 2; (c) Test 3; (d) Test 4; (e) Test 5; (f) Test 6.

of applying ECANet at different positions on the model accuracy by selecting the model (Group 1) after replacing the backbone network structure and performing feature fusion as the benchmark is presented in Table 6.

Table 5 Results of ablation experiments

MobileNet v3	feature fusion	ECANet	parameter (Mb)	*mPA* (%)	*mIoU* (%)	ascension of *mIoU* (%)
			209.70	79.06	71.21	–
√			21.70	85.99	74.80	3.59
√	√		21.85	84.25	76.71	5.50
√	√	√	21.85	85.80	78.26	7.05

Note: √ represents “used.”

Applying ECANet to different locations is not entirely beneficial for crack detection. For example, applying the attention module to the corresponding positions of the 5th–7th groups reduced the accuracy of the model, possibly because the attention mechanism at this location damaged the channel weights of the original network. In addition, after applying the ECA module three times in the 8th group, its detection effect was not as good as applying a single ECA module in the 2nd–4th groups. Hence, ECANet was applied only after shallow feature fusion to simplify the network structure. The *mIoU* of the model increased to 78.26%, and the *mPA* increased to 85.80%.

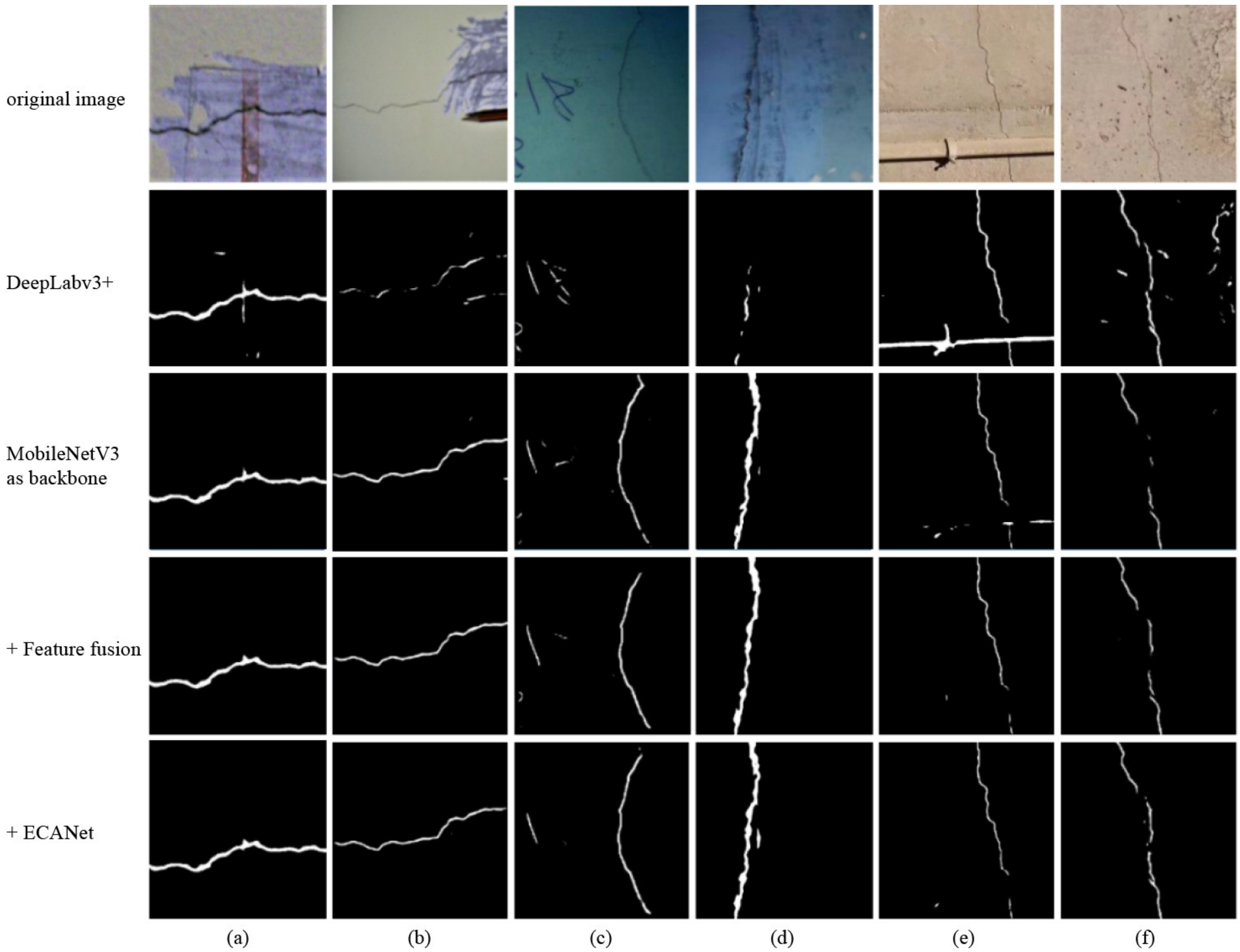

Fig. 12 Comparison of predicted images between DeepLabv3+ and LC-DeepLab. (a) Test 1; (b) Test 2; (c) Test 3; (d) Test 4; (e) Test 5; (f) Test 6.

Table 6 Effect of applying ECANet at different positions on model accuracy

group	A	B	C	*mPA* (%)	*mIoU* (%)	ascension of *mIoU* (%)
1				84.25	76.71	–
2	√			85.80	78.26	1.55
3		√		84.55	77.89	1.18
4			√	85.39	77.93	1.22
5	√	√		82.70	76.08	−0.63
6		√	√	77.06	71.33	−5.38
7	√		√	82.23	74.27	−2.44
8	√	√	√	84.84	77.25	0.54

Note: A is the efficient channel attention (ECA) after the shallow feature fusion, B is the ECA after the ASPP, C is the ECA after deep and shallow feature fusion, and √ represents "used".

5 Discussion

Compared with the traditional semantic segmentation algorithm, the optimization strategy adopted by LC-DeepLab effectively improves the accuracy and speed of tunnel-lining crack identification. The novel algorithm exhibits an excellent anti-interference ability and can accurately segment lining cracks in complex backgrounds under poor lighting conditions. Its detection speed (*FPS*) reaches 46.98 f/s, which supports the dynamic detection of tunnel defects.

A crack detection management platform based on the proposed algorithm can be developed. The model can be built on crack-detection mobile devices, such as robot cars and drones. The artificial intelligence camera automatically captures and detects various defects in the tunnel lining when the equipment moves. The recorded data can be uploaded to the software management platform through the network to realize intelligent detection and maintenance of tunnel projects.

6 Conclusions

This study proposes a fast detection model called LC-DeepLab for cracks on tunnel linings based on the DeepLabv3+ architecture. The model uses a lightweight network, MobileNetV3, instead of Xception for feature extraction. Moreover, a shallow feature fusion module and ECANet are designed to enhance the extraction of shallow features and improve the anti-interference ability of the proposed model. A complex tunnel surface-lining data set was constructed for the experiments to validate the performance of the proposed model. The results showed that the *mIoU* of the LC-DeepLab was 78.26%, and the speed (FPS) reached 46.98 f/s when the input was 416 × 416 × 3. Compared with the four classical semantic segmentation models, the proposed LC-DeepLab exhibits an excellent anti-interference ability because it can detect the edges of cracks more accurately than other models in the complex backgrounds of tunnels.

Acknowledgements This study was supported by the National Natural Science Foundation of China (Grant Nos. 50908234, 52208421), the Open Fund of the National Engineering Research Center of Highway Maintenance Technology, Changsha University of Science & Technology (No. kfj220101), the Natural Science Foundation of Hunan Province (No. 2020JJ4743), and the Research Innovation Project for Postgraduate of Central South University (No. 1053320213484).

References

1. Zhang J, Dai L, Zheng J, Wu H. Reflective crack propagation and control in asphalt pavement widening. Journal of Testing and Evaluation, 2016, 44(2): 838−846
2. Zhou Z, Ding H, Miao L, Gong C. Predictive model for the surface settlement caused by the excavation of twin tunnels. Tunnelling and Underground Space Technology, 2021, 114: 104014
3. Zeng L, Xiao L Y, Zhang J H, Gao Q F. Effect of the characteristics of surface cracks on the transient saturated zones in colluvial soil slopes during rainfall. Bulletin of Engineering Geology and the Environment, 2020, 79(2): 699−709
4. Chiaia B, Marasco G, Aiello S. Deep convolutional neural network for multi-level non-invasive tunnel lining assessment. Frontiers of Structural and Civil Engineering, 2022, 16(2): 214−223
5. Zhang N, Zhu X, Ren Y. Analysis and study on crack characteristics of highway tunnel lining. Civil Engineering Journal, 2019, 5(5): 1119−1123
6. Savino P, Tondolo F. Automated classification of civil structure defects based on convolutional neural network. Frontiers of Structural and Civil Engineering, 2021, 15(2): 305−317
7. Arena A, Delle Piane C, Sarout J. A new computational approach to cracks quantification from 2D image analysis: Application to micro-cracks description in rocks. Computers & Geosciences, 2014, 66: 106−120
8. Falls S D, Young R P. Acoustic emission and ultrasonic-velocity methods used to characterise the excavation disturbance associated with deep tunnels in hard rock. Tectonophysics, 1998, 289(1–3): 1–15
9. Lee C H, Chiu Y C, Wang T T, Huang T H. Application and validation of simple image-mosaic technology for interpreting cracks on tunnel lining. Tunnelling and Underground Space Technology, 2013, 34: 61−72
10. Schabowicz K. Ultrasonic tomography—The latest nondestructive technique for testing concrete members—Description, test methodology, application example. Archives of Civil and Mechanical Engineering, 2014, 14(2): 295−303
11. Dang L M, Wang H, Li Y, Park Y, Oh C, Nguyen T N, Moon H. Automatic tunnel lining crack evaluation and measurement using deep learning. Tunnelling and Underground Space Technology, 2022, 124: 104472
12. Kamaliardakani M, Sun L, Ardakani M K. Sealed-crack detection algorithm using heuristic thresholding approach. Journal of Computing in Civil Engineering, 2016, 30(1): 04014110
13. Wang G, Peter W T, Yuan M. Automatic internal crack detection from a sequence of infrared images with a triple-threshold Canny edge detector. Measurement Science & Technology, 2018, 29(2): 025403
14. Dorafshan S, Thomas R J, Maguire M. Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete. Construction & Building Materials, 2018, 186: 1031−1045
15. Huang H, Li Q, Zhang D. Deep learning based image recognition for crack and leakage defects of metro shield tunnel. Tunnelling and Underground Space Technology, 2018, 77: 166−176
16. Wu X, Li J, Wang L. Efficient identification of water conveyance tunnels siltation based on ensemble deep learning. Frontiers of Structural and Civil Engineering, 2022, 16(5): 564−575
17. Zhang L, Yang F, Zhang Y D, Zhu Y J. Road crack detection using deep convolutional neural network. In: Proceedings of 2016 IEEE International Conference on Image Processing (ICIP). Phoenix, AZ: IEEE, 2016, 3708−3712
18. Cha Y J, Choi W, Büyüköztürk O. Deep learning-based crack damage detection using convolutional neural networks. Computer-Aided Civil and Infrastructure Engineering, 2017, 32(5): 361−378
19. Kang D, Cha Y J. Autonomous UAVs for structural health monitoring using deep learning and an ultrasonic beacon system with geo-tagging. Computer-Aided Civil and Infrastructure Engineering, 2018, 33(10): 885−902
20. Beckman G H, Polyzois D, Cha Y J. Deep learning-based automatic volumetric damage quantification using depth camera. Automation in Construction, 2019, 99: 114−124
21. Zhou Z, Zhang J, Gong C. Automatic detection method of tunnel lining multi-defects via an enhanced You Only Look Once network. Computer-Aided Civil and Infrastructure Engineering, 2022, 37(6): 762−780
22. Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, NV: IEEE, 2016, 779−788
23. Bochkovskiy A, Wang C Y, Liao H Y M. Yolov4: Optimal speed and accuracy of object detection. 2020, arXiv: 2004.10934
24. Ren S, He K, Girshick R, Sun J. Faster R-CNN: Towards real-time object detection with region proposal networks. Advances in Neural Information Processing Systems. New York, NY: Curran Associates, 2015, 28
25. Liu W, Anguelov D, Erhan D, Szegedy. SSD: Single shot multibox detector. In: European Conference on Computer Vision.

Amsterdam: Springer, 2016, 21−37

26. Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Boston, MA: IEEE, 2015, 3431−3440
27. Choi W, Cha Y J. SDDNet: Real-time crack segmentation. IEEE Transactions on Industrial Electronics, 2019, 67(9): 8016−8025
28. Kang D, Benipal S S, Gopal D L, Cha Y J. Hybrid pixel-level concrete crack segmentation and quantification across complex backgrounds using deep learning. Automation in Construction, 2020, 118: 103291
29. Liu Z, Cao Y, Wang Y, Wang W. Computer vision-based concrete crack detection using U-Net fully convolutional networks. Automation in Construction, 2019, 104: 129−139
30. Ronneberger O, Fischer P, Brox T. U-Net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Munich: Springer, 2015, 234−241
31. Ji A, Xue X, Wang Y, Luo X, Xue W. An integrated approach to automatic pixel-level crack detection and quantification of asphalt pavement. Automation in Construction, 2020, 114: 103176
32. Chen L C, Zhu Y, Papandreou G, Schroff F, Adam H. Encoder−decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European Conference on Computer Vision (ECCV). Munich, Germany: Springer, 2018, 801−818
33. Zhou, Z, Zhang, J, Gong, C, Ding H. Automatic identification of tunnel leakage based on deep semantic segmentation. Chinese Journal of Rock Mechanics and Engineering, 2022, 41(10): 2082−2093 (in Chinese)
34. Ali R, Cha Y J. Attention-based generative adversarial network with internal damage segmentation using thermography. Automation in Construction, 2022, 141: 104412
35. Howard A, Sandler M, Chu G, Chen L C, Chen B, Tan M, Wang W, Zhu Y, Pang R, Vasudevan V, Le Q V. Searching for MobileNetV3. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. Seoul: IEEE, 2019, 1314−1324
36. Hu J, Shen L, Sun G. Squeeze-and-excitation networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Salt Lake City, UT: IEEE, 2018, 7132−7141
37. Wang Q L, Wu B G, Zhu P F, Li P, Zuo W, Hu Q. ECA-Net: efficient channel attention for deep convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Seattle, WA: IEEE, 2020
38. Kang D H, Cha Y J. Efficient attention-based deep encoder and decoder for automatic crack segmentation. Structural Health Monitoring, 2022, 21(5): 2190−2205
39. He K, Zhang X, Ren S, Sun J. Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(9): 1904−1916
40. Chollet F. Xception: Deep learning with depthwise separable convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, HI: IEEE, 2017, 1251−1258
41. Chen L C, Papandreou G, Schroff F, Adam H. Rethinking atrous convolution for semantic image segmentation. arXiv:1706.05587
42. Farabet C, Couprie C, Najman L, LeCun Y. Learning hierarchical features for scene labeling. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 35(8): 1915−1929
43. Cha Y J, Choi W, Suh G, Mahmoudkhani S, Büyüköztürk O. Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types. Computer-Aided Civil and Infrastructure Engineering, 2018, 33(9): 731−747
44. Liashchynskyi P, Liashchynskyi P. Grid search, random search, genetic algorithm: A big comparison for NAS. arXiv preprint arXiv:1912.06059
45. Milletari F, Navab N, Ahmadi S A. V-Net: Fully convolutional neural networks for volumetric medical image segmentation. In: Proceedings of 2016 Fourth International Conference on 3D Vision (3DV). Stanford, CA: IEEE, 2016, 565−571
46. Ren Y, Huang J, Hong Z, Lu W, Yin J, Zou L, Shen X. Image-based concrete crack detection in tunnels using deep fully convolutional networks. Construction & Building Materials, 2020, 234: 117367
47. Zhao H, Shi J, Qi X, Wang X, Jia J. Pyramid scene parsing network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, HI: IEEE, 2017, 2881−2890