

## Research Article

# Enhanced Precision in Dam Crack Width Measurement: Leveraging Advanced Lightweight Network Identification for Pixel-Level Accuracy

Zihao Wu,<sup>1</sup> Yunchao Tang ,<sup>2</sup> Bo Hong,<sup>3</sup> Bingqiang Liang,<sup>3</sup> and Yuping Liu<sup>3</sup>

<sup>1</sup>College of Engineering, South China Agricultural University, 483 Wushan Road, Guangzhou 510642, China

<sup>2</sup>Guangdong Lingnan Township Green Building Industrialization Engineering Technology Research Center, Zhongkai University of Agriculture and Engineering, Guangzhou 510225, China

<sup>3</sup>Guangdong Zhenzheng Construction Engineering Testing Co., Ltd., Guangzhou, China

Correspondence should be addressed to Yunchao Tang; ryan.twain@zhku.edu.cn

Received 2 May 2023; Revised 2 July 2023; Accepted 23 August 2023; Published 2 September 2023

Academic Editor: B. B. Gupta

Copyright © 2023 Zihao Wu 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.

In dam engineering, the presence of cracks and crack width are important indicators for diagnosing the health of dams. The accurate measurement of cracks facilitates the safe use of dams. The manual detection of such defects is unsatisfactory in terms of cost, safety, accuracy, and the reliability of evaluation. The introduction of deep learning for crack detection can overcome these issues. However, the current deep learning algorithms possess a large volume of model parameters, high hardware requirements, and difficulty toward embedding in mobile devices such as drones. Therefore, we propose a lightweight MobileNetV2\_DeepLabV3 image segmentation network. Furthermore, to prevent interference by noise, light, shadow, and other factors for long-length targets when segmenting, the atrous spatial pyramid pooling (ASPP) module parameters in the DeepLabV3+ network structure were modified, and a multifeature fusion structure was used instead of the parallel structure in ASPP, allowing the network to obtain richer crack features. We collected the images of dam cracks from different environments, established segmentation datasets, and obtained segmentation models through network training. Experiments show that the improved MobileNetV2\_DeepLabV3 algorithm exhibited a higher crack segmentation accuracy than the original MobileNetV2\_DeepLabV3 algorithm; the average intersection rate attained 83.23%; and the crack detail segmentation was highly accurate. Compared with other semantic segmentation networks, its training time was at least doubled, and the total parameters were reduced by more than 2 to 7 times. After extracting cracks through the semantic segmentation, we proposed to use the method of inscribed circle of crack outline to calculate the maximum width of the detected crack image and to convert it into the actual width of the crack. The maximum relative error rate was 11.22%. The results demonstrated the potential of innovative deep learning methods for dam crack detection.

## 1. Introduction

The degree of structural damage of cracks in concrete structures is a fundamental indicator for diagnosing the health of dams. Owing to the long-term exposure to large water pressure and the effects of water scouring, infiltration, and erosion, cracks inevitably occur, making real-time health inspection of dam structures necessary to detect such cracks in time [1, 2].

However, the manual crack detection method, which is relatively simple and has a low detection efficiency, has resulted in tragic incidents such as 236 deaths at the Canyon Lake Dam due to errors in crack detection [3] and the failure of Austin Dam, which resulted in the death of 76 people, due to inadequate management and monitoring of damage and cracks [4]. Therefore, it is necessary to realize efficient and automated dam crack detection.

*1.1. Motivation.* To overcome the problems of the manual detection process, various digital techniques have been studied to achieve crack detection and feature extraction in civil infrastructure [5], such as edge detection (Canny filter and Sobel filter), threshold segmentation, and 3D reconstruction [6–8]. Although these traditional image processing methods can ideally complete crack detection on the measurement surface, these traditional image processing methods are faced with the interference of different textures of the measurement surface, changes in light, and the presence of stains and debris on the measurement surface. These factors cause problems such as continuous detection, false detection, or missed detection.

With the advancements in computer science and artificial intelligence, machine learning has been widely adopted for image recognition, object detection, and classification [9–11], which provides a new prospect for crack detection. These algorithms contain common processing steps; for example, first, the crack image is divided into many small images; a feature extraction algorithm is then used to construct feature vectors for the small images; thereafter, machine learning algorithms are applied to train the feature vectors; and finally, the generated model is used to detect cracks in images [12]. However, the method of constructing feature vectors using machine learning is highly cumbersome, and in addition, the detection effect of cracks in different scenarios is considerably different.

In recent years, with rapid advancements in deep learning technology [13, 14], it has become the mainstream method in the field of crack detection [15, 16]. Kim and Cho [17] used a deep learning method based on convolutional neural network (CNN), which possesses a high detection speed for concrete surface cracks and can realize the automatic detection of cracks. Gopalakrishnan et al. [18] proposed different classifiers based on the VGG16 neural network model for detecting cracks in concrete and asphalt pavement images. Although these deep learning algorithms can automatically detect cracks, they cannot exactly locate the cracks; therefore, they possess limitations toward structural health maintenance and management [19].

Target detection is a deep learning recognition method in computer vision [20], which can realize automatic detection of cracks, in addition to accurately locating the cracks. Redmon [21] proposed a YOLO algorithm that can quickly detect and identify cracks. Teng et al. [22] compared the crack detection performances of YOLOv2 and 11 feature extractors to prove the excellent crack detection of YOLOv2. Subsequently, YOLOv3 and YOLOv4 were proposed. Yang [23] found that the accuracy of the trained YOLOv3 model in detecting cracks was satisfactory, confirming that deep learning can be an efficient and powerful crack detection technology. Yu et al. [24] improved on the basis of YOLOv4, accelerated the detection speed, and realized the real-time detection of bridge cracks for drones or other machinery and equipment. Yao et al. [25] proposed a new pavement crack detection method based on YOLOv5 by combining the attention mechanism. Cha et al. [26] applied a faster R-CNN for automatic crack detection, which could realize real-time detection through a camera, with an average accuracy of

94.7%. Maeda et al. [27] compared the accuracy and processing speed of several target detection algorithms and, finally, decided to use a CNN based on MobileNet and Inception V2 to detect various types of road damages through an onboard camera. In the object detection method, cracks are marked with bounding boxes; however, because the distribution paths and shapes of cracks are irregular, object detection cannot provide high-precision detection information with respect to these aspects.

Pixel segmentation is another deep-learning recognition method in computer vision. Pixel segmentation, as the name suggests, facilitates the identification of target objects at the pixel level. Pixel segmentation is divided into instance segmentation and semantic segmentation. For crack detection, semantic segmentation is a better choice [28–32]. Dung and Anh [33] proposed a deep fully convolutional network (FCN)-based crack detection method for the semantic segmentation of concrete crack images and found that cracks could be reasonably detected and the crack density was accurately assessed. With the emergence of semantic segmentation networks such as U-Net [34], deep learning-based image segmentation has gradually attracted substantial research attention. Zhang et al. [35] proposed a new method for crack detection on 3D asphalt pavement, called CrackNet-R (RNN), which achieved a high accuracy (88.89%) for crack detection. Ju et al. [36] proposed a network structure called CrackU-Net, which achieved pixel-level crack detection through convolution, pooling, transposed convolution, and cascade operations, and the effect was better than the traditional FCN and U-Net network. Cui et al. [37] added an attention mechanism to the U-Net network, and the effect in crack detection was improved. Park et al. [38] used semantic segmentation and migration learning techniques to develop an effective crack detection method that can accurately detect crack locations and shapes by performing pixel-level classification on concrete structure images. Khayatazad et al. [39] used an algorithm based on texture features and morphological operations to identify and locate cracks and deformations in the structure.

Owing to the small pixel ratio of cracks, compared with object detection, pixel segmentation can more accurately identify and locate cracks [40–42] and can provide the distribution path and shape of cracks, which are important information for evaluating the structural health of dams. To obtain quantified information such as crack length, average width, maximum width, area, and ratio, the current study considered the use of semantic segmentation algorithms to detect cracks.

Recently, the DeepLabV3+ semantic segmentation method created by Chen et al. [43] has been widely used for pixel segmentation. DeepLabV3+ adds encoder and decoder modules on the basis of DeepLabV3, which combines the advantages of spatial pyramid pooling and can effectively capture global information. Currently, DeepLabV3+ has been successfully applied in many fields, such as gear pitting measurement [44], lychee detection [45], and the classification of trees, shrubs, and grasses [46]. The success of DeepLabV3+ in these areas prompted us to apply it for crack detection.

Considering the use of drones and other equipment for real-time crack detection, we utilized the lightweight and efficient network MobileNetV2 [47] as the backbone network for DeepLabV3+. MobileNetV2 has demonstrated its effectiveness in real-time detection of orchard kiwifruit [48], masks [49], and cracks through wall-climbing robots [50]. It provides efficient feature extraction, meets real-time requirements, and has excellent migration learning capabilities, making it ideal for crack detection on UAVs and similar devices.

To incorporate more global information, we adjusted the dilation rate parameter in the atrous spatial pyramid pooling (ASPP) module and introduced a  $1 \times 1$  convolution module to increase the convolution density and minimize information loss, thus improving information search efficiency. In place of the parallel structure in ASPP, we employed a multifeature layer fusion structure described in references to capture intricate crack features. The issue of imbalanced positive and negative samples was addressed by utilizing the Dice loss [51–54].

The improved MobileNetV2\_DeepLabV3 network enables the identification of cracks, which can be further utilized to measure crack width in images using the inscribed circle method and imaging principle of the crack profile. This allows for the estimation of the maximum width of real cracks.

*1.2. Contributions.* The contributions of this study are three-fold:

- (1) Utilizing the lightweight and efficient MobileNetV2 network as the backbone of DeepLabV3+, with all standard convolutions replaced by DSC to minimize the number of parameters and enable real-time detection.
- (2) Modifying the dilation rate parameter in the ASPP module, incorporating a  $1 \times 1$  convolution module to enhance convolution density and reduce information loss, employing a multifeature layer fusion structure in place of the parallel structure in ASPP, and using Dice loss to address the crack and background pixel imbalance problem.
- (3) Introducing a method that leverages the inscribed circle of the crack outline to measure crack width in the image.

*1.3. Organization.* In this paper, the implementation process of the dam crack-based width measurement algorithm includes three major parts: crack image segmentation, crack backbone refinement, and crack width measurement, as shown in Figure 1.

The remainder of this paper is organized as follows. Section 2 briefly outlines the structures of DeepLabV3+ and MobileNetV2 and proposes methods for modifying the dilation rate of atrous convolution, adjusting the loss function, and altering the structure of ASPP to enhance network performance. Section 3 introduces a method to measure the image crack width using the inscribed circle of

the crack outline and describes the conversion process from image crack width to real crack width using a depth camera. Moreover, Section 4 conducts extensive experiments and analyzes the results. Lastly, Section 5 concludes this study.

## 2. Method and Principle

Since its introduction, the DeepLabV3+ network has often been used for high-precision image segmentation due to its excellent capacity [55]. While the original DeepLabV3+ provides satisfactory image segmentation accuracy, it struggles to produce continuous crack identification in real-world applications, particularly when dealing with small and irregular cracks on dams. Therefore, this section outlines the improvements made to the DeepLabV3+ network to better capture global information and enhance the accuracy of dam crack detection.

This section first introduces the DeepLabV3+ network structure and the MobileNetV2 network structure, followed by a description of the improved MobileNetV2\_DeepLabV3+ structure.

*2.1. DeepLabV3+ Network Structure.* The DeepLabV3+ algorithm is currently one of the best semantic segmentation algorithms [56] for solving image segmentation problems. The model improves the backbone network based on the DeepLabV3+ network and introduces the encoder and decoder modules. The overall network structure of DeepLabV3+ is shown in Figure 2.

In the encoder, the DeepLabV3 network is used as the encoder module, which is primarily composed of the backbone network (DCNN) and the spatial pyramid (ASPP), for high-level and low-level feature layers. In the encoder, we used dilated convolutions with different dilation rates for feature extraction in the ASPP module, where the dilation rates were  $3 \times 3$  convolutions of 6, 12, and 18, for improving the receptive field of the network. This further enables the network to exhibit different feature feelings and used  $1 \times 1$  convolution to decrease the number of channels as well as reduce the parameters and used image pooling to prevent network overfitting. The feature layer was stacked and fused, and the  $1 \times 1$  convolution was performed to fuse the number of channels to obtain the green feature layer shown in Figure 1.

In the decoder, the low-level feature layer generated by DCNN enters the decoder for  $1 \times 1$  convolution, and the number of channels is adjusted. The level features' layer is subjected to feature fusion through the results obtained from the  $1 \times 1$  convolution, and the fused feature layer is subjected to  $3 \times 3$  convolution for feature extraction. Finally, the input and output images are maintained at the same dimensions through quadruple upsampling, and the prediction result is obtained.

*2.2. MobileNetV2 Backbone Network Structure.* Numerous backbone networks can be employed for DeepLabV3+, as demonstrated in Table 1, which presents the parameters of some common networks. Lightweight backbone networks

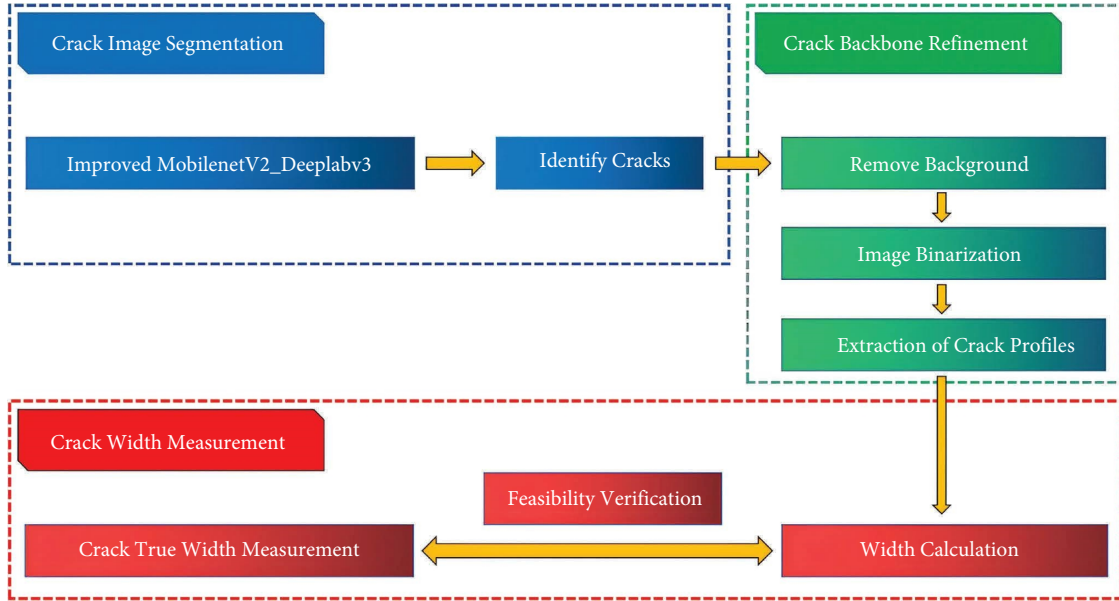


FIGURE 1: Technology road map.

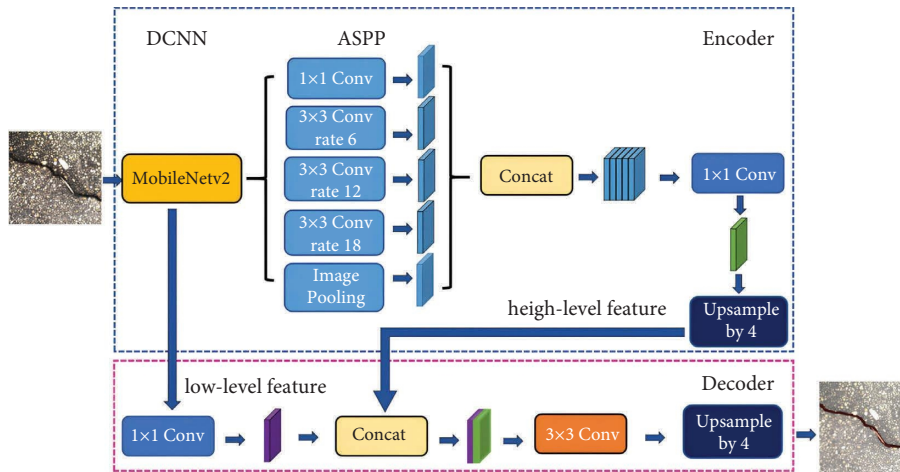


FIGURE 2: Original MobileNetV2\_DeepLabV3+ network structure.

can effectively enhance detection speed and reduce memory usage, making the model more suitable for real-time detection. Consequently, in the encoder module, we opted for the lightweight MobileNetV2 network as the backbone network (DCNN) for feature extraction.

MobileNetV2 [47] is a lightweight network model introduced by the Google team in 2018. The overall network structure of MobileNetV2 is displayed in Table 2. Boasting a simple, streamlined structure with significantly fewer parameters and low latency, MobileNetV2 can extract crack feature information more quickly while maintaining a consistent accuracy.

The standard convolution process is shown in Figure 3. The input was passed through four convolution layers of  $3 \times 3$  convolution kernels, and four feature maps were output, where the number of parameters of the convolution layer was  $4 \times 3 \times 3 \times 3 = 108$ .

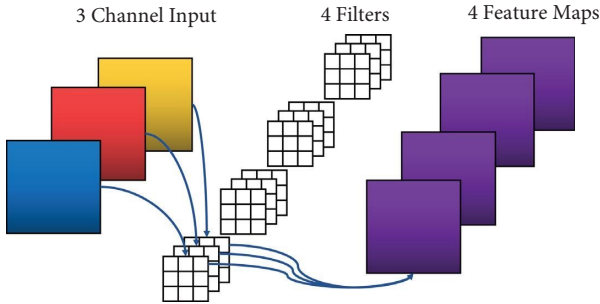
The DSC is a combination of depthwise convolution (DW) and pointwise convolution (PW). The convolution operation is shown in Figure 4. PW performs a  $1 \times 1$  convolution cross-channel combination on the input image, whereas DW is a convolution kernel corresponding to one input channel and performs spatial convolution on each input channel independently. Therefore, as shown in Figure 4, a four-channel image was subjected to DW to generate four feature maps, and finally, the same output dimension as the standard convolution process could be obtained. The overall effect was the same as the standard convolution. However, the number of parameters in the convolution layer was  $1 \times 1 \times 3 \times 4 + 3 \times 3 \times 4 = 48$ , which is only about 44% of the number of parameters compared to the standard convolution. Therefore, the number of parameters of the DSC module was considerably reduced compared with that of the ordinary convolution module, indicating that the

TABLE 1: Parameters of common models.

| Models       | Parameters (M) |
|--------------|----------------|
| VGG16        | 138.36         |
| ResNet50     | 25.56          |
| ResNet101    | 44.55          |
| Inception V3 | 27.16          |
| Xception     | 23.03          |
| MobileNetV2  | 3.5            |

TABLE 2: MobileNetV2 structure.

| Inputs                | Operators               | Expansions | Channels | Numbers | Strides |
|-----------------------|-------------------------|------------|----------|---------|---------|
| $244^2 \times 3$      | Conv2D                  | —          | 32       | 1       | 2       |
| $112^2 \times 32$     | Bottleneck              | 1          | 16       | 1       | 1       |
| $112^2 \times 16$     | Bottleneck              | 6          | 24       | 2       | 2       |
| $56^2 \times 24$      | Bottleneck              | 6          | 32       | 3       | 2       |
| $28^2 \times 32$      | Bottleneck              | 6          | 64       | 4       | 2       |
| $28^2 \times 64$      | Bottleneck              | 6          | 96       | 3       | 1       |
| $14^2 \times 96$      | Bottleneck              | 6          | 160      | 3       | 2       |
| $7^2 \times 160$      | Bottleneck              | 6          | 320      | 1       | 1       |
| $7^2 \times 320$      | Conv2D<br>$1 \times 1$  | —          | 1280     | 1       | 1       |
| $7^2 \times 1280$     | AvgPool<br>$7 \times 7$ | —          | —        | 1       | —       |
| $1 \times 1 \times k$ | Conv2D<br>$1 \times 1$  | —          | $k$      | —       | —       |

FIGURE 3: Ordinary  $3 \times 3$  convolution process.

integration of the MobileNetV2 network and DSC module can achieve a faster speed and a smaller network volume, with low computational load extract crack features.

**2.3. Improved MobileNetV2\_DeepLabV3.** To enhance the network's performance for crack feature extraction, an improved DeepLabV3+ method is proposed. This method replaces the feature layer parallel structure in ASPP with a feature layer fusion structure, adjusts the dilation rate in the ASPP module, and adds a few parameters to strengthen the fracture characteristic information while reducing the interference of environmental factors. These improvements enable a greater accuracy in crack identification without sacrificing the model detection efficiency.

Three modifications were incorporated based on the original DeepLabV3+ network structure (Figure 5).

The convolution interval of the original hole convolution in ASPP was small, which resulted in the loss of some global information. Increasing the dilation rate can reduce the loss of global information (Figure 6). The red dots in Figure 6(a) represent the pixels used. The green regions represent the pixel information obtained by the conventional  $3 \times 3$  convolution. Figure 6(b) uses a dilation rate of 1 hole convolution, which can be observed from the number of red dots. The obtained information increases from 9 in Figure 6(a) to 49 in Figure 6(b). Figure 6(c) shows the dilation rate of 2. The convolution exhibits more pixel information, up to 121, such that the hole convolution is increased. Therefore, the increase in the dilation rate can promote the acquisition of more pixel information from the image and improve the ability of the network to capture global information. The segmentation of targets with long lengths such as cracks will be disturbed by factors such as noise, light, and shadows. Furthermore, it is necessary to improve the ability to capture global information. The dilation rate of the atrous convolution in the original ASPP module was increased from 6, 12, and 18 to 6, 18, and 24, thereby effectively enhancing the ability to obtain pixel information and consequently improving the capacity to capture global information. In addition, a new dilation rate 3 feature layer was added to ensure that features near the cracks receive focused attention. As a result, the entire system can utilize more pixel information to improve recognition performance when faced with complex backgrounds. Furthermore, a  $1 \times 1$  convolution layer was introduced after the atrous convolution to increase the convolution density, which helps to reduce information loss while obtaining more information.

In the original ASPP module of DeepLabV3+, the feature layers operate in parallel, preventing them from sharing any fracture feature information. By fusing the feature layers, fracture feature information can be shared among the layers, and expansion convolutions with different expansion rates can be interdependent, ultimately increasing the range of the receptive field [50]. The structure of the ASPP before and after the improvement is illustrated in Figure 7.

The improvement structure can better capture the global information of the dam crack image by increasing the perceptual field. The following equation shows the receptive field:

$$\text{RF} = (\text{Rate} - 1) \times (k - 1) + k, \quad (1)$$

where RF denotes the receptive field, Rate denotes the dilation rate, and  $k$  denotes the convolution kernel size.

According to equation (1), the maximum perceptual field  $\text{RF}_{\max}$  for a single feature layer is

$$\begin{aligned} \text{RF}_{\max} & (\text{RF}_3^3, \text{RF}_3^6, \text{RF}_3^{18}, \text{RF}_3^{24}) \\ & = \text{RF}_3^{24} \\ & = 49, \end{aligned} \quad (2)$$

where  $\text{RF}_3^6$  indicates the field of perception of the expanded convolution for an RF convolution kernel size of  $3 \times 3$  and a dilation rate of 6.

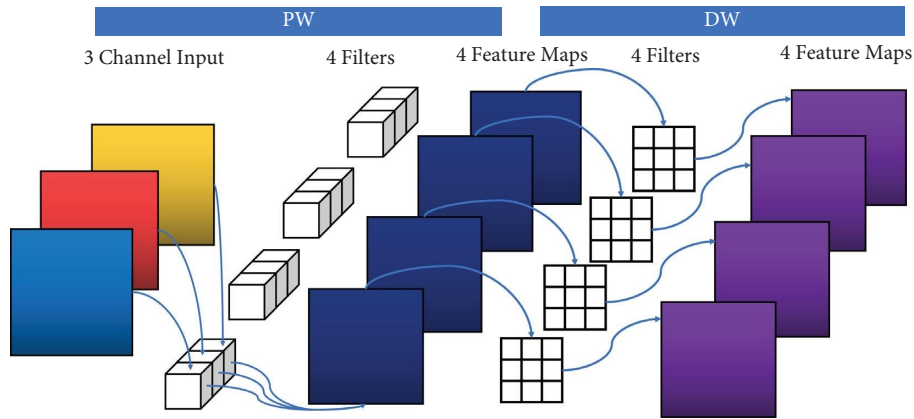


FIGURE 4: Depthwise separable convolution (DSC)  $3 \times 3$  process.

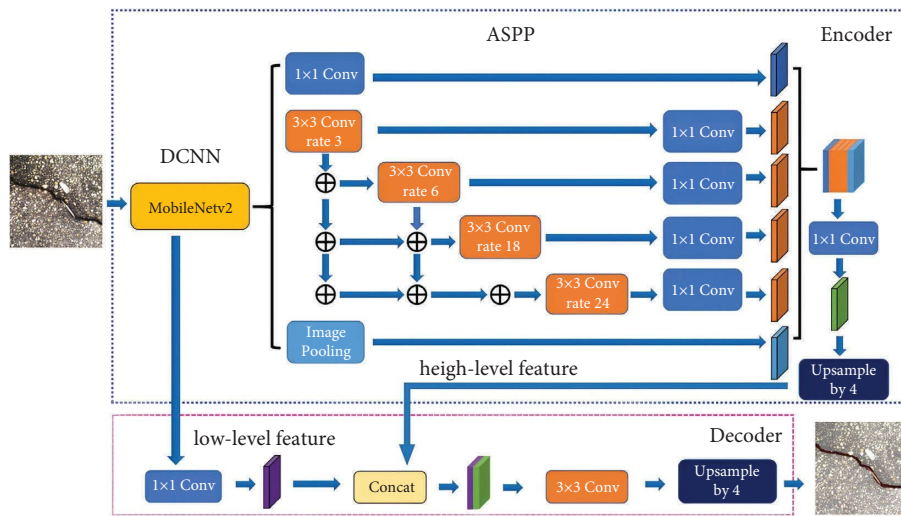


FIGURE 5: Improved MobileNetV2\_DeepLabV3 network structure.

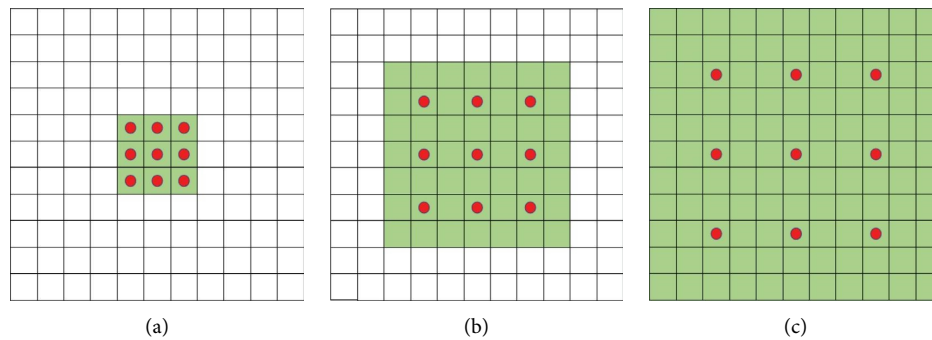


FIGURE 6:  $3 \times 3$  ordinary convolution and hole convolution structure: (a)  $3 \times 3$  convolution, (b)  $3 \times 3$  atrous convolution with a dilation rate = 1, and (c)  $3 \times 3$  atrous convolution with a dilation rate = 2.

By fusing feature layers, a larger perceptual field can be obtained. For the perceptual field obtained by fusing two feature layers,

$$RF = RF_1 + RF_2 - 1, \quad (3)$$

where  $RF_1$  and  $RF_2$  are two different feature layers.

Thus, according to equation (4), the maximum perceptual field obtained after fusing feature layers in this paper is 105.



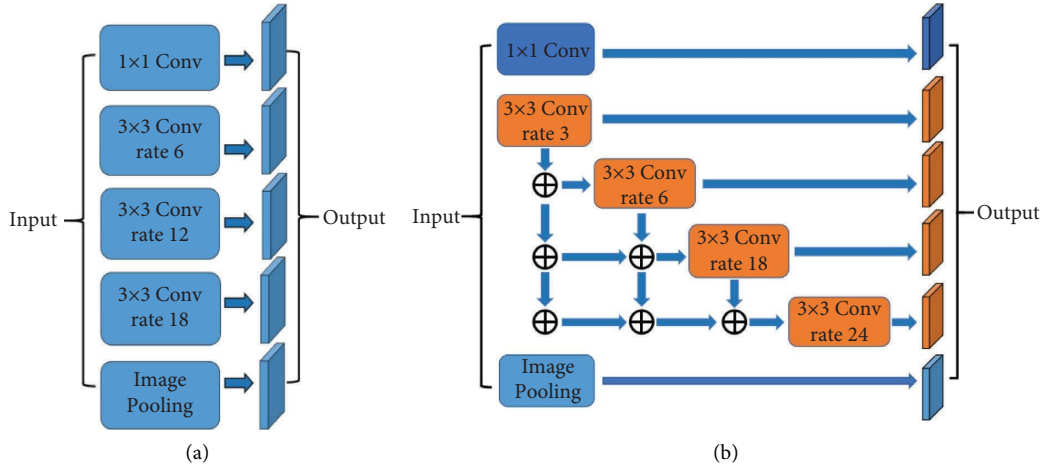


FIGURE 7: ASPP structure: (a) before improvement and (b) after improvement.

$$\begin{aligned} RF_{\max} &= RF_3^3 + RF_3^6 + RF_3^{18} + RF_3^{24} - 3 \\ &= 105. \end{aligned} \quad (4)$$

By fusing feature layers in the ASPP module, the model can obtain a larger receptive field, which improves the model's use of crack information and ultimately the overall crack detection accuracy.

In crack images, most cracks occupy a pixel area much smaller than the background area. This leads to an imbalance of positive and negative samples during algorithm training, resulting in weight bias that affects the segmentation performance of the algorithm. To solve this issue, the Dice loss (equation (5)) is used instead of the cross-entropy loss function:

$$\text{Dice loss} = 1 - \frac{2\sum_{i=1}^N y_i \hat{y}_i}{\sum_{i=1}^N y_i + \sum_{i=1}^N \hat{y}_i}, \quad (5)$$

where  $y_i$  and  $\hat{y}_i$  denote the label value and the predicted value of pixel  $i$ , respectively, and  $N$  is the total number of pixel points, which is equal to the number of pixels in a single image multiplied by the batch size.

### 3. Crack Width Calculation

**3.1. Extraction of Pixels in Cracks.** For cracks with different widths in a dam, the inscribed circle method of the crack contour was used to calculate the crack width of the image. First, the crack image identified by the improved MobileNetV2\_DeepLabV3 was removed, and only the crack was retained. As we only require obtaining the pixel points within the crack outline, we first enclosed the crack with the smallest circumscribed rectangle (Figure 8(d)). Thereafter, all pixels except those contained in the minimum circumscribed rectangle were removed from the image to reduce the amount of computation. The ray method in geometry was used to determine whether the pixels of the circumscribed rectangle were within the crack contour, and the pixels within the crack contour were extracted.

**3.2. Identification of the Maximum Width of the Crack.** The minimum side length of the minimum circumscribed rectangle of the crack outline was set to  $s$ . The pixel points  $x_i$  ( $i = 0, 1, 2, \dots, n$ ) in the crack outline were considered as dots. Initially, the minimum radius  $sr$  of the circle was set to 0. The maximum radius  $mr$  was set as  $s/2$ . The dichotomy method was subsequently used to find the maximum radius. If the circle drawn with this length as the radius contains no intersection with the crack outline, then half of the sum of the maximum radius and the minimum radius is used as the subsequent minimum radius. When the drawn circle and the point outside the crack outline were not selected, the radius calculated by the dichotomy method is selected as the subsequent maximum radius. Until the circle and the crack outline have exactly two intersection points, the circle was based on these parameters. The pixel point was the center of the circle, which was relative to the maximum inscribed circle of the crack outline. The whole process is shown in Figure 9.

Finally, the maximum inscribed circle radii of all pixels in the crack outline were compared (Figure 10) to obtain the maximum width of the crack outline, and the circle was drawn.

**3.3. Estimating the Actual Maximum Width of the Crack.** The D435i depth camera utilizes a method of acquiring depth information for a point on an image by emitting infrared structured light and capturing the shape change of the light that is reflected back. It achieves this by utilizing a depth sensor to analyze the shape change of the structured light. By calculating the distance between the camera and the point on the surface of the object, it generates a depth image that provides a quantitative measurement of the distance to various points on the object's surface. Therefore, we used the Intel RealSense D435i depth camera to capture the crack image and to obtain the distance from the camera to the measurement surface. Thereafter, the crack width was calculated through the crack image. The principle of true width is shown in Figure 11, where points P1 and P2 are the

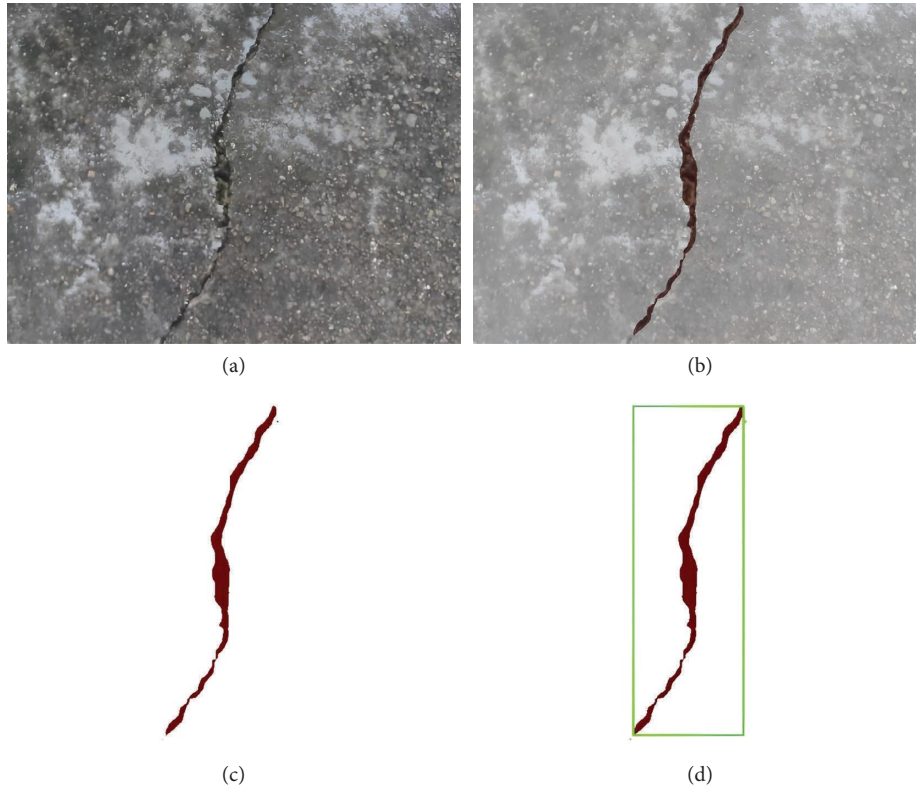


FIGURE 8: Extraction process of the minimum circumscribed rectangle of the crack: (a) original image, (b) identified cracks, (c) removing background, and (d) the smallest circumscribed rectangle surrounding the crack.

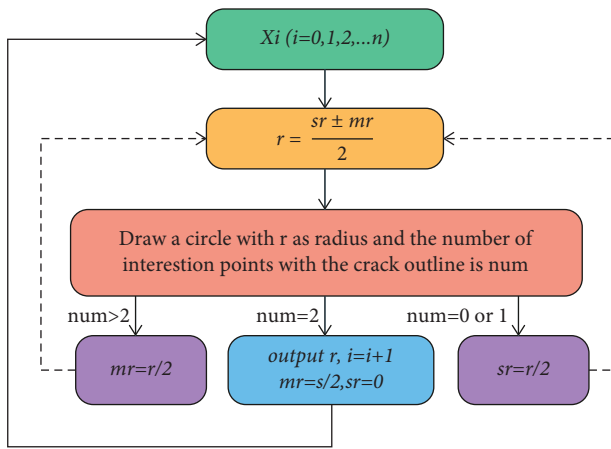


FIGURE 9: General process of using the bisection method to find the inscribed circle with the pixel as the center in the crack outline.

intersection points of the inscribed circle with the maximum width as the diameter of the crack on the image and the crack contour. Thus, P1 and P2 are the largest crack contours on the world coordinate system obtained by the imaging principle. In Figure 11,  $u_x$  is the relative offset of the origin of the imaging plane coordinate system from the  $v$ -axis along the  $u$ -axis of the pixel plane coordinate system;  $v_y$  is the relative offset of the origin of the imaging plane coordinate system from the  $u$ -axis along the  $v$ -axis of the pixel plane coordinate system; and  $u$  is the value along the  $u$ -axis on the



FIGURE 10: Extracted inscribed circle with the maximum width as the diameter.

pixel plane coordinate system and  $v$  is the value along the  $v$ -axis of the pixel point on the pixel plane coordinate system.

The depth ( $d$ ) from the measurement surface to the camera was measured by the principle of depth camera structured light. When the depth of the depth camera and the measurement surface is known, the depth camera is single-targeted to obtain the internal parameter matrix,  $\begin{pmatrix} f_x & 0 & u_x \\ 0 & f_y & v_y \\ 0 & 0 & 1 \end{pmatrix}$ . The scale factors  $f_x$  and  $f_y$  are extracted, and the values of the coordinate points ( $x_1$  and  $y_1$ ) and ( $x_2$  and  $y_2$ ) of the real fracture are calculated using equations (6)



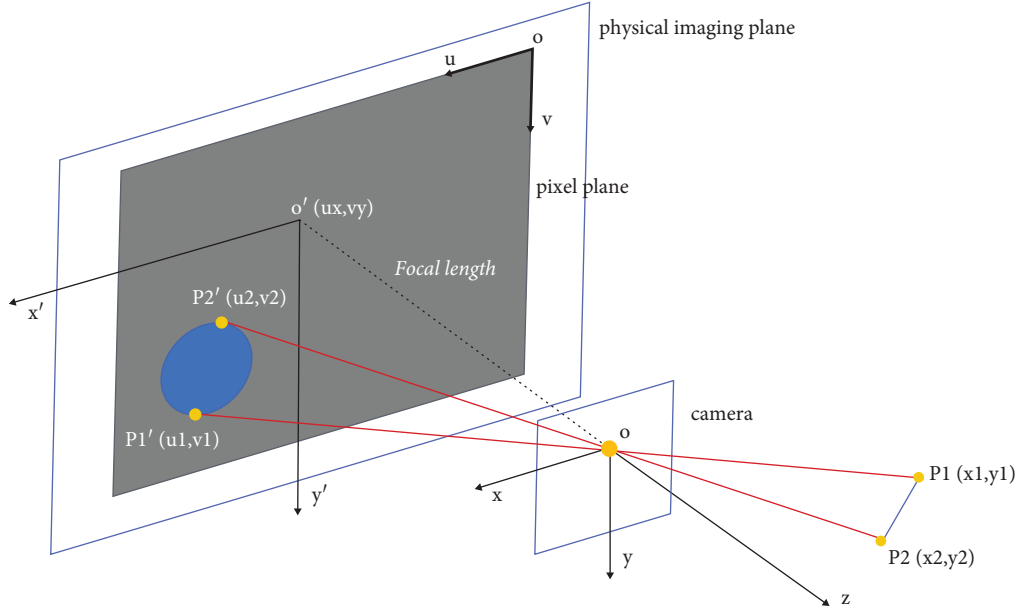


FIGURE 11: Diagrammatic representation of camera imaging.

and (7), by using the principles of imaging and similar triangle:

$$x_i = \frac{u_i - u_x}{f_x} \times d \quad (i = 1, 2), \quad (6)$$

$$y_i = \frac{v_i - v_y}{f_y} \times d \quad (i = 1, 2). \quad (7)$$

Finally, the maximum width of the real crack is calculated by the following equation:

$$\text{width} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}. \quad (8)$$

#### 4. Fracture Segmentation Results and Analysis

To verify the effectiveness of our proposed method, we conducted experiments on existing semantic segmentation neural networks, by comparing the improved MobileNetV2\_DeepLabV3 with MobileNetV2\_DeepLabV3, ENet, BiSeNetV2, FCN, U-Net, and PSPNet.

**4.1. Evaluation Indicators.** In this study, four evaluation criteria were used to assess the performance of the network for semantic segmentation: mRecall rate (equation (9)), mPrecision rate (equation (10)), intersection ratio IoU (equation (11)) for cracks, and average intersection ratio mIoU (equation (12)). TP represents the positive samples predicted by the network model as positive classes, TN represents the negative samples predicted by the network model as negative classes, FP represents the negative samples predicted by the network model as positive classes, and FN represents the positive samples predicted by the network model as negative classes.

The mRecall rate indicates the average of the quantity of data accurately predicted in the positive class data in the test set (the prediction result is a positive class). The mPrecision rate indicates the proportion of truly positive samples in the predicted outcome among the samples predicted to be positive. The average intersection and union ratio MIoU represents the ratio of the intersection and union of the prediction results of the model and the true value of each category. The IoU represents the ratio of the intersection and union of the prediction results of the model and the true value of one category.

$$m\text{Recall} = \frac{1}{n+1} \sum_{i=0}^n \frac{TP}{FN + TP}, \quad (9)$$

$$m\text{Precision} = \frac{1}{n+1} \sum_{i=0}^n \frac{TP}{FP + TP}, \quad (10)$$

$$\text{IoU} = \frac{TP}{TP + FP + FN}, \quad (11)$$

$$m\text{IoU} = \frac{1}{n+1} \sum_{i=0}^n \frac{TP}{TP + FP + FN}. \quad (12)$$

**4.2. Experimental Conditions and Model Environment.** The experimental environment in this study was configured with the Ubuntu 20.0 operating system, Intel Core i5-10400F CPU processor, NVIDIA GeForce RTX 2060 graphics card, and open-source PyTorch deep learning framework. The raw data for the experiments were images of real dam cracks collected through digital equipment. Table 3 shows the use of experimental data, where “size” denotes the image size in pixels and “number” denotes the number of images used.

TABLE 3: The use of images.

| Type            | Size (pixel) | Number |
|-----------------|--------------|--------|
| Training data   | 512 × 512    | 1560   |
| Test data       | 512 × 512    | 760    |
| Validation data | 512 × 512    | 1404   |



FIGURE 12: Equipment used for crack identification and measurement: (a) depth camera and (b) electronic Vernier caliper.

The Intel RealSense D435i depth camera was used to measure the distance from the measuring surface to the camera, and the 0–200 mm electronic digital caliper was used to measure the width of the real crack (Figure 12).

The LabelMe tool was used to annotate the collected dataset to obtain the corresponding annotation map. The annotation of the crack data is shown in Figure 13.

The TensorBoard tool attached to the PyTorch was used to observe the change of the loss value during training (Figure 14). It can be observed that when the epoch reached 100, the loss value tended to be stable. Therefore, the selected epoch was 150.

**4.3. Ablation Studies.** Based on the theoretical analysis in Section 3, ablation studies were designed to verify the effectiveness of rate modification,  $1 \times 1$  convolution, feature layer fusion, and Dice loss function. The following networks were tested: (1) DeepLabV3+ network without any improvement, denoted as DL; (2) DeepLabV3+ with only rate modification, denoted as R + Dp; (3) DeepLabV3+ network with only  $1 \times 1$  convolution, denoted as C + DL; (4) DeepLabV3+ with a feature layer fusion structure, denoted as F + DL; and (5) DeepLabV3+ with the Dice loss function, denoted as D + DL. The results of the ablation experiments are shown in Table 4 to verify the effectiveness of the algorithm.

From Figure 15, it can be seen that the original DeepLabV3+ network has a good ability to extract crack feature pixels but can be further improved. Among them, rate modification, feature layer fusion structure, and the addition of  $1 \times 1$  convolution do not improve the original loss function because of the extreme imbalance between positive and negative samples. However, the original DeepLabV3+ network improved significantly after adding Dice loss, and the DeepLabV3+ network with a feature layer fusion structure and the addition of  $1 \times 1$  convolution showed even

more improvement. This indicates that these changes make the network pay more attention to the characteristics of cracks, solve the problem of imbalance between positive and negative samples, and improve the network's ability to identify cracks.

**4.4. Crack Segmentation Experiment Results.** To compare the accuracy and training speed of the algorithm, the segmentation effect of the dam crack image was tested. The original MobileNetV2\_DeepLabV3, U-Net, PSPNet, Xception-DeepLabV3+, and improved MobileNetV2\_DeepLabV3 networks were tested with the same dataset. MIOU and crack IoU were used as the evaluation metrics for network segmentation accuracy. In Table 5, the experimental results are shown, where the total parameters of each network are represented as total params and the time used by each network to train under the same conditions is represented as train time, and FPS stands for frames per second.

Table 5 shows that the total parameters of U-Net, PSPNet, and FCN, which are common semantic segmentation networks, are 2.3 times, 5.9 times, and 3.2 times higher than MobileNetV2\_DeepLabV3+. Moreover, the training time of U-Net and FCN is more than twice that of MobileNetV2\_DeepLabV3+. However, the FPS of MobileNetV2\_DeepLabV3+ is higher than that of U-Net and FCN. Therefore, using MobileNetV2 as the backbone of DeepLabV3+ achieves a lightweight effect.

Although BiSeNetV2 and ENet are lightweight networks with a small number of parameters and training time, they have only 24.34% IoU and 54.17% IoU for dam cracks, making them unsuitable for crack identification. On the other hand, the improved MobileNetV2\_DeepLabV3+ showed the best segmentation results, with an overall MIOU value increase of 2.11% and a significant improvement in the dam crack region with an IoU value increase of 4.67%. It only increased the number of parameters by 29.79 MB and



FIGURE 13: Labeling of the crack data with LabelMe.

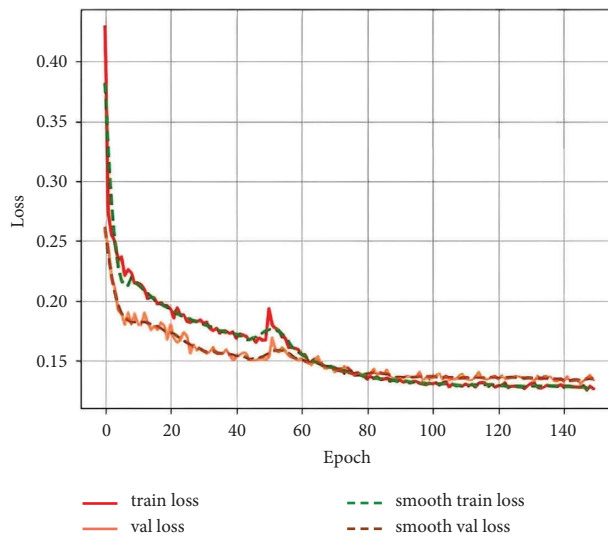


FIGURE 14: Improved MobileNetV2\_DeepLabV3 training and validation loss curves.

TABLE 4: Ablation studies.

| Model          | MIoU (%)     | IoU (%)      | mRecall (%) | mPrecision (%) |
|----------------|--------------|--------------|-------------|----------------|
| DL             | 81.12        | 62.29        | 86.96       | 88.76          |
| R + DL         | 81.14        | 62.83        | 87.34       | 89.63          |
| C + DL         | 81.04        | 62.64        | 87.05       | 89.80          |
| F + DL         | 80.59        | 61.69        | 87.71       | 89.35          |
| L + DL         | 82.19        | 64.89        | 87.98       | 90.57          |
| R + C + DL     | 81.26        | 63.07        | 87.87       | 89.24          |
| R + L + DL     | 82.72        | 65.96        | 89.96       | 89.51          |
| C + L + DL     | 82.78        | 66.07        | 89.13       | 90.20          |
| R + C + L + DL | 82.85        | 66.21        | 89.85       | 89.57          |
| F + C + DL     | 80.59        | 61.69        | 86.25       | 89.96          |
| F + L + DL     | 82.72        | 65.96        | 90.41       | 89.48          |
| F + C + L + DL | <b>83.23</b> | <b>66.96</b> | 90.47       | 89.51          |

The bold font is to highlight that the optimized algorithm in this paper performs better than any of the other optimizations.



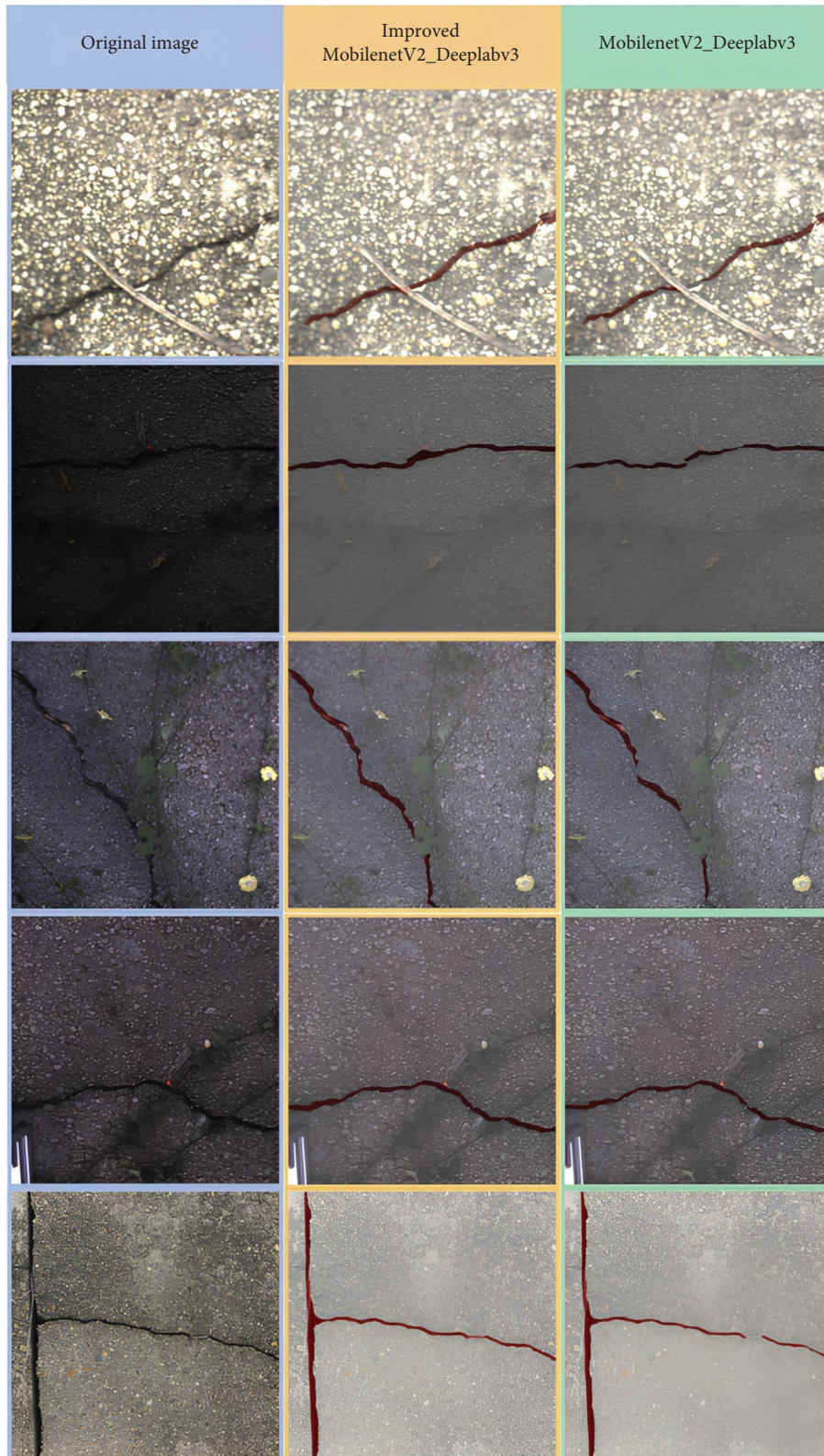


FIGURE 15: Comparison of original MobileNetV2\_DeepLabV3 and improved MobileNetV2\_DeepLabV3 crack segmentation results.

TABLE 5: Comparison of network results.

| Method                          | MIoU (%)     | IoU for crack (%) | Total params (MB) | Train time (s) | Frames per second (FPS) |
|---------------------------------|--------------|-------------------|-------------------|----------------|-------------------------|
| U-Net                           | 79.16        | 58.87             | 2905.95           | 40,560         | 7.3                     |
| PSPNet                          | 68.93        | 38.41             | 7575.59           | 7,980          | 14.7                    |
| FCN                             | 56.5         | 39.10             | 4121.82           | 35,400         | 9.1                     |
| BiSeNetV2                       | 61.35        | 24.34             | 240.16            | 1,496          | 83.3                    |
| ENet                            | 76.53        | 54.17             | 292.83            | 1,583          | 58.8                    |
| MobileNetV2_DeepLabV3+          | 81.12        | 62.29             | 1247.93           | 14,880         | 9.9                     |
| Improved MobileNetV2_DeepLabV3+ | <b>83.23</b> | <b>66.96</b>      | <b>1277.72</b>    | <b>14,940</b>  | <b>9.7</b>              |

The bold font indicates that the performance of the optimized algorithm in this paper is the best in the table.

TABLE 6: Physical width measurement results of examples.

| Example | Calculation (mm) | Ground truth (mm) | Relative error (%) |
|---------|------------------|-------------------|--------------------|
| 1       | 7.583            | 7.40              | 2.47               |
| 2       | 8.327            | 8.75              | 4.83               |
| 3       | 6.102            | 6.20              | 1.58               |
| 4       | 5.748            | 5.5               | 4.51               |
| 5       | 4.64             | 3.64              | 11.22              |

training time by 60 seconds and decreased the FPS by a mere 0.3. Thus, the improved MobileNetV2\_DeepLabV3+ has a higher recognition effect while ensuring a lighter weight and is more suitable for real-time detection. The improved method is effective.

To better visualize the effect, the segmentation results of the original MobileNetV2\_DeepLabV3 algorithm and the improved MobileNetV2\_DeepLabV3 algorithm were compared. Owing to the limited space, only certain representative segmentation results are given, as shown in Table 4. To render the cracks more visible, the background brightness of the graphs in Table 6 has been increased. As can be observed from Figure 14, the cracks identified by the original MobileNetV2\_DeepLabV3 were incoherent and intermittent, and the cracks were thinner, indicating that the recognition was not effective.

*4.5. Crack Width Accuracy Assessment.* To verify the accuracy of the method used in Section 3, five samples were considered for comparing the maximum crack width measured by the method, described in Section 3, with the actual maximum crack width. The comparison results are shown in Table 6.

In two previous studies [5, 57], the relative error rates in the calculation of the fracture image converted into the maximum width of the real fracture were 1.20%–9.09% and 13.27%–24.01%, respectively. Considering their data as a reference, it can be concluded from Table 6 that the relative error rate of the measurement was 1.58%–11.22%, which showed the feasibility of the method and the accuracy rate can be guaranteed.

## 5. Conclusion

This study proposes a dam crack detection method based on the improved lightweight MobileNetV2\_DeepLabV3. It utilizes a dataset of 1560 dam crack images collected from

smartphones and depth cameras for training, with 1404 images used for validation and 760 images for testing purposes. Although this dataset is relatively large, it is important to acknowledge the possibility of under-representation, which may limit the generalization ability of the results.

The “LabelMe” tool was employed as the image annotation tool, while a depth camera was used to identify the maximum crack width in the crack images. However, it is important to note that both of these steps involve human manipulation and are susceptible to subjective factors and errors. These factors can potentially impact the accuracy and reliability of the results. It is crucial to consider these limitations when interpreting the findings of the study, as the presence of subjective factors and errors can affect the accuracy of the results.

To mitigate interference from noise, light, shadows, and other factors during segmentation, modifications were made to the parameters of the atrous spatial pyramid pooling (ASPP) module in the DeepLabV3+ network structure. In addition, a multifeature fusion structure was used instead of the parallel structure in ASPP, enabling the network to capture richer crack features. Subsequently, segmentation experiments were performed.

The experimental results demonstrate that the improved MobileNetV2\_DeepLabV3+ algorithm achieves better segmentation performance compared to the original MobileNetV2\_DeepLabV3+ algorithm. It shows an increase of 2.11% in the mean intersection over union (MIoU) and a 4.67% increase in the intersection over union (IoU) for crack segmentation. In comparison to U-Net, PSPNet, and FCN algorithms, the proposed algorithm reduces computation by 2–7 times and training time by 5,900–25,000 seconds. This higher efficiency makes it more suitable for engineering applications and allows for its integration into drones or other machinery and equipment to achieve automatic crack detection. However, it is recommended to consider other advanced crack detection



algorithms for comparison to provide a more comprehensive assessment of the method's advantages and limitations in terms of performance. In terms of real crack identification, the relative error rate between the actual maximum crack width calculated by the proposed method and the measured maximum crack width was found to be in the range of 1.58%–11.22%. This demonstrates the feasibility of the method.

For future work, it is suggested to expand the dataset by collecting images with an increased complexity and various types of background images of dam cracks. In addition, efforts should be made to improve the accuracy of the proposed method for direct applicability to UAV equipment, enabling the realization of automatic detection of dam cracks.

### Data Availability

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

### Conflicts of Interest

The authors declare that they have no conflicts of interest.

### Acknowledgments

This research was funded by the National Natural Science Foundation of China (52368028), China Postdoctoral Science Foundation (Grant no. 2021M690765), Natural Science Foundation of Guangxi Province (Grant no. 2021GXNSFAA220045), Science and Technology Planning Project of Guangzhou (Grant no. 202102080269), and Systematic Project of Guangxi Key Laboratory of Disaster Prevention and Engineering Safety (Grant no.2021ZDK007).

### References

- [1] Y. Tang, L. Li, C. Wang et al., "Real-time detection of surface deformation and strain in recycled aggregate concrete-filled steel tubular columns via four-ocular vision," *Robotics and Computer-Integrated Manufacturing*, vol. 59, pp. 36–46, 2019.
- [2] Y. Tang, Z. Chen, Z. Huang, Y. Nong, and L. Li, "Visual measurement of dam concrete cracks based on U-net and improved thinning algorithm," *Journal of Experimental Mechanics*, vol. 37, pp. 209–220, 2022.
- [3] L. Graham, "A procedure for estimating loss of life caused by dam failure," *Dam Safety Office*, vol. 10, 1999.
- [4] T. P. Rich, "Lessons in social responsibility from the Austin dam failure," *International Journal of Engineering Education*, vol. 22, pp. 1287–1296, 2006.
- [5] Y. Liu and J. K. W. Yeoh, "Robust pixel-wise concrete crack segmentation and properties retrieval using image patches," *Automation in Construction*, vol. 123, Article ID 103535, 2021.
- [6] G. Li, S. He, Y. Ju, and K. Du, "Long-distance precision inspection method for bridge cracks with image processing," *Automation in Construction*, vol. 41, pp. 83–95, 2014.
- [7] W. Cao, Q. Liu, and Z. He, "Review of pavement defect detection methods," *IEEE Access*, vol. 8, pp. 14531–14544, 2020.
- [8] M. R. Jahanshahi and S. F. Masri, "Adaptive vision-based crack detection using 3D scene reconstruction for condition assessment of structures," *Automation in Construction*, vol. 22, pp. 567–576, 2012.
- [9] J. Li, T. Liu, X. Wang, and J. Yu, "Automated asphalt pavement damage rate detection based on optimized GA-CNN," *Automation in Construction*, vol. 136, Article ID 104180, 2022.
- [10] L. Wu, X. Lin, Z. Chen, P. Lin, and S. Cheng, "Surface crack detection based on image stitching and transfer learning with pretrained convolutional neural network," *Structural Control and Health Monitoring*, vol. 28, no. 8, Article ID e2766, 2021.
- [11] F. Barrarat, K. Rayane, B. Helifa, and I. K. Lefkaier, "Characterization of subsurface cracks in eddy current testing using machine learning methods," *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, vol. 35, no. 6, Article ID e2876, 2022.
- [12] L. Liu, W. Ouyang, X. Wang et al., "Deep learning for generic object detection: a survey," *International Journal of Computer Vision*, vol. 128, no. 2, pp. 261–318, 2020.
- [13] T. Han, T. Zhou, Y. Xiang, and D. Jiang, "Cross-machine intelligent fault diagnosis of gearbox based on deep learning and parameter transfer," *Structural Control and Health Monitoring*, vol. 29, no. 3, Article ID e2898, 2022.
- [14] J. Pal, S. Sikdar, and S. Banerjee, "A deep-learning approach for health monitoring of a steel frame structure with bolted connections," *Structural Control and Health Monitoring*, vol. 29, no. 2, p. e2873, 2022.
- [15] H. Perez and J. H. M. Tah, "Deep learning smartphone application for real-time detection of defects in buildings," *Structural Control and Health Monitoring*, vol. 28, no. 7, p. e2751, 2021.
- [16] M. Zarski, B. Wojcik, K. Ksiazek, and J. A. Miszczak, "Finicky transfer learning-A method of pruning convolutional neural networks for cracks classification on edge devices," *Computer-Aided Civil and Infrastructure Engineering*, vol. 37, no. 4, pp. 500–515, 2022.
- [17] B. Kim and S. Cho, "Automated vision-based detection of cracks on concrete surfaces using a deep learning technique," *Sensors*, vol. 18, no. 10, p. 3452, 2018.
- [18] K. Gopalakrishnan, S. K. Khaitan, A. Choudhary, and A. Agrawal, "Deep Convolutional Neural Networks with transfer learning for computer vision-based data-driven pavement distress detection," *Construction and Building Materials*, vol. 157, pp. 322–330, 2017.
- [19] A. Ji, X. Xue, Y. Wang, X. Luo, and W. Xue, "An integrated approach to automatic pixel-level crack detection and quantification of asphalt pavement," *Automation in Construction*, vol. 114, Article ID 103176, 2020.
- [20] X. Zhang, D. Rajan, and B. Story, "Concrete crack detection using context-aware deep semantic segmentation network," *Computer-Aided Civil and Infrastructure Engineering*, vol. 34, no. 11, pp. 951–971, 2019.
- [21] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, Las Vegas, NV, USA, June 2016.
- [22] S. Teng, Z. Liu, G. Chen, and L. Cheng, "Concrete crack detection based on well-known feature extractor model and the YOLO\_v2 network," *Applied Sciences*, vol. 11, no. 2, p. 813, 2021.
- [23] C. Yang, J. Chen, Z. Li, and Y. Huang, "Structural crack detection and recognition based on deep learning," *Applied Sciences*, vol. 11, no. 6, p. 2868, 2021.

- [24] Z. Yu, Y. Shen, and C. Shen, "A real-time detection approach for bridge cracks based on YOLOv4-FPM," *Automation in Construction*, vol. 122, Article ID 103514, 2021.
- [25] H. Yao, Y. Liu, X. Li, Z. You, Y. Feng, and W. Lu, "A detection method for pavement cracks combining object detection and attention mechanism," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 11, pp. 22179–22189, 2022.
- [26] Y. J. Cha, W. Choi, G. Suh, S. Mahmoudkhani, and O. Buyukozturk, "Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 9, pp. 731–747, 2018.
- [27] H. Maeda, Y. Sekimoto, T. Seto, T. Kashiyama, and H. Omata, "Road damage detection and classification using deep neural networks with smartphone images," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 12, pp. 1127–1141, 2018.
- [28] Y. Wu, Y. Qin, Y. Qian, F. Guo, Z. Wang, and L. Jia, "Hybrid deep learning architecture for rail surface segmentation and surface defect detection," *Computer-Aided Civil and Infrastructure Engineering*, vol. 37, no. 2, pp. 227–244, 2022.
- [29] L. Fan, H. Zhao, Y. Li, S. Li, R. Zhou, and W. Chu, "RAO-UNet: a residual attention and octave UNet for road crack detection via balance loss," *IET Intelligent Transport Systems*, vol. 16, no. 3, pp. 332–343, 2022.
- [30] J. Chen and Y. He, "A novel U-shaped encoder-decoder network with attention mechanism for detection and evaluation of road cracks at pixel level," *Computer-Aided Civil and Infrastructure Engineering*, vol. 37, no. 13, pp. 1721–1736, 2022.
- [31] J. Zhao, F. Hu, and W. Qiao, "A modified U-net for crack segmentation by Self-Attention-Self-Adaption neuron and random elastic deformation," *Smart Structures and Systems*, vol. 29, pp. 1–16, 2022.
- [32] Y. Xu, Y. Bao, Y. Zhang, and H. Li, "Attribute-based structural damage identification by few-shot meta learning with inter-class knowledge transfer," *Structural Health Monitoring*, vol. 20, no. 4, pp. 1494–1517, 2021.
- [33] C. V. Dung and L. D. Anh, "Autonomous concrete crack detection using deep fully convolutional neural network," *Automation in Construction*, vol. 99, pp. 52–58, 2019.
- [34] O. Ronneberger, P. Fischer, and T. Brox, "U-net: convolutional networks for biomedical image segmentation," in *Proceedings of the 18th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 234–241, Munich, Germany, October 2015.
- [35] 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.
- [36] H. Ju, W. Li, S. Tighe, Z. Xu, and J. Zhai, "CrackU-net: a novel deep convolutional neural network for pixelwise pavement crack detection," *Structural Control and Health Monitoring*, vol. 27, p. e2551, 2020.
- [37] X. Cui, Q. Wang, J. Dai, Y. Xue, and Y. Duan, "Intelligent crack detection based on attention mechanism in convolution neural network," *Advances in Structural Engineering*, vol. 24, Article ID 136943322098663, 2021.
- [38] S. E. Park, S. H. Eem, and H. Jeon, "Concrete crack detection and quantification using deep learning and structured light," *Construction and Building Materials*, vol. 252, pp. 119096–3699, 2020.
- [39] M. Khayatizad, L. De Pue, and W. De Waele, "Detection of corrosion on steel structures using automated image processing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, pp. 2814–2823, 2021.
- [40] J. S. Lee, S. H. Hwang, I. Y. Choi, and Y. Choi, "Estimation of crack width based on shape-sensitive kernels and semantic segmentation," *Structural Control and Health Monitoring*, vol. 27, no. 4, p. e2504, 2020.
- [41] Y. Xu, Y. Fan, Y. Bao, and H. Li, "Task-aware meta-learning paradigm for universal structural damage segmentation using limited images," *Engineering Structures*, vol. 284, Article ID 115917, 2023.
- [42] Y. Xu, Y. Fan, and H. Li, "Lightweight semantic segmentation of complex structural damage recognition for actual bridges," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, 2023.
- [43] 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 15th European Conference on Computer Vision (ECCV)*, pp. 833–851, Munich, Germany, September 2018.
- [44] D. Xi, Y. Qin, and Z. Wang, "Attention Deeplabv3 model and its application into gear pitting measurement," *Journal of Intelligent and Fuzzy Systems*, vol. 42, no. 4, pp. 3107–3120, 2022.
- [45] H. Peng, C. Xue, Y. Shao et al., "Semantic segmentation of litchi branches using DeepLabV3+Model," *IEEE Access*, vol. 8, pp. 164546–164555, 2020.
- [46] B. Ayhan and C. Kwan, "Tree, shrub, and grass classification using only RGB images," *Remote Sensing*, vol. 12, no. 8, p. 1333, 2020.
- [47] Z. Zhou, Z. Song, L. Fu, F. Gao, R. Li, and Y. Cui, "Real-time kiwifruit detection in orchard using deep learning on Android™ smartphones for yield estimation," *Computers and Electronics in Agriculture*, vol. 179, Article ID 105856, 2020.
- [48] M. Sandler, A. G. Howard, A. Z. M. Zhu, and L. Chen, "Inverted residuals and linear bottlenecks: mobile networks for classification, detection and segmentation," pp. 4510–4520, 2018, <https://arxiv.org/abs/1801.04381>.
- [49] P. Nagraath, R. Jain, A. Madan, R. Arora, P. Kataria, and J. Hemanth, "SSDMNV2: a real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2," *Sustainable Cities and Society*, vol. 66, Article ID 102692, 2021.
- [50] S. Jiang and J. Zhang, "Real-time crack assessment using deep neural networks with wall-climbing unmanned aerial system," *Computer-Aided Civil and Infrastructure Engineering*, vol. 35, no. 6, pp. 549–564, 2020.
- [51] X. Xiang, Y. Zhang, and A. El Saddik, "Pavement crack detection network based on pyramid structure and attention mechanism," *IET Image Processing*, vol. 14, no. 8, pp. 1580–1586, 2020.
- [52] Y. Pan and L. Zhang, "Dual attention deep learning network for automatic steel surface defect segmentation," *Computer-Aided Civil and Infrastructure Engineering*, vol. 37, no. 11, pp. 1468–1487, 2022.
- [53] M. Yang, K. Yu, and C. Zhang, "DenseASPP for semantic segmentation in street scenes," in *Proceedings of the 31st IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3684–3692, Salt Lake City, UT, USA, June 2018.

- [54] C. Chen and P. Shen, "Research on crack width measurement based on binocular vision and improved deeplabv3+," *Applied Sciences*, vol. 13, 2023.
- [55] Z. Chen, D. Ting, R. Newbury, and C. Chen, "Semantic segmentation for partially occluded apple trees based on deep learning," *Computers and Electronics in Agriculture*, vol. 181, Article ID 105952, 2021.
- [56] H. Fu, D. Meng, W. Li, and Y. Wang, "Bridge crack semantic segmentation based on improved Deeplabv3+," *Journal of Marine Science and Engineering*, vol. 9, no. 6, p. 671, 2021.
- [57] W. Choi and Y. J. Cha, "SDDNet: real-time crack segmentation," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 9, pp. 8016–8025, 2020.