Hindawi
Advances in Civil Engineering
Volume 2022, Article ID 1813821, 15 pages
https://doi.org/10.1155/2022/1813821

Research Article

Intelligent Crack Detection and Quantification in the Concrete Bridge: A Deep Learning-Assisted Image Processing Approach

Licun Yu,[1,2] **Shuanhai He**,[1] **Xiaosong Liu**,[3] **Shuqing Jiang**,[3] **and Shuiying Xiang**[3]

[1]*School of Highway, Chang'an University, Xi'an 710064, China*
[2]*CCCC First Highway Consultants Co., Ltd., Xi'an 710075, China*
[3]*State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an 710071, China*

Correspondence should be addressed to Licun Yu; 727705858@qq.com

Received 25 December 2021; Revised 24 January 2022; Accepted 9 February 2022; Published 3 March 2022

Academic Editor: Nhat-Duc Hoang

We proposed a modified concrete bridge crack detector based on a deep learning-assisted image processing approach. Data augmentation technology is introduced to extend the limited dataset. In our proposed method, the bounding box for the crack is detected by YOLOv5. Then, the image covered by the bounding box is processed by the image processing techniques. Compared with the conventional image processing-based crack detection method, the deep learning-assisted image processing approach leads to higher detection accuracy and lower computation cost. More precisely, the mask filter is employed to remove handwritten marks, and the ratio filter is adopted to eliminate speckle linear noises. When a single crack is detected by several bounding boxes, we proposed a novel fusion method to merge these bounding boxes. Furthermore, we proposed a connected component search approach based on the crack trend of the area to improve the connection accuracy. With the crack detector, the cracks that are wider than 0.15 mm can be correctly detected, quantified, and visualized. The detection absolute error of the crack width is less than 0.05 mm. Thus, we realized fast and precise detection and quantification of bridge crack based on the practical engineering dataset.

1. Introduction

Most of the bridges built in the world are concrete bridges. In the process of service, it is necessary to detect the bridge regularly to facilitate the development of corresponding maintenance countermeasures [1]. Crack is one of the main diseases of concrete bridges, which has become the most important content of concrete bridge detection and maintenance. Traditional manual detection methods have the disadvantages of inaccuracy and low efficiency. Therefore, using image processing or deep learning (DL) techniques to detect bridge crack has become a research hotspot [2].

Some bridge crack detectors were realized based on traditional digital image processing methods. For instance, the gray threshold segmentation method was adopted to extract the crack according to the gray difference between the crack area and the background [3, 4]. Besides, the Canny iterative method was used to detect the edge feature of the crack according to the linear feature of the crack [5]. In addition, the Otsu method and multiple filtering in image processing were employed to detect cracks of concrete structures [6]. Moreover, an improved image segmentation algorithm based on the Chan–Vese (C–V) model was provided for crack extraction [7]. All the above-mentioned algorithms could lead to good experimental results, but the background of the crack image was simple, and no obstacle was considered. These algorithms may not be applicable for bridge crack detection with complex background.

In recent years, many researchers adopted the DL to locate and classify bridge diseases and achieved good performance. For example, Lei Z. et al. used a convolution neural network (CNN) to detect road cracks. They divided the complete images into several small image blocks and then classified the small image blocks to complete the extraction of cracks [8]. Cha et al. used the CNN and the target detection network of a faster region-based convolutional

neural network (Faster R-CNN) to realize the detection of building damage, including cracks [9, 10]. Li et al. adopted the YOLO to perform the detection of concrete cracks [11]. Zhang et al. further used the target detection network of the YOLOv3 and migration learning to complete the damage detection of cracks in concrete bridges [12]. Hoang and Nguyen constructed an automatic model to detect and classify asphalt pavement crack. It could detect cracks and wa able to recognize the types of pavement cracks, including the longitudinal, transverse, and alligator cracks [13]. Meng studied an image segmentation algorithm for concrete cracks based on CNN. Compared with other methods, the algorithm achieved higher detection accuracy and generalization ability [14]. Ding et al. established a concrete crack identification model based on an improved Mask R-CNN. The results showed that the average prediction accuracy of concrete crack identification was better than the YOLOv4 method [15]. Hu et al. studied a pavement crack detection method based on YOLOv5. The experimental results showed that the detection accuracy of the YOLOv5 series models was above 85% [16]. Note that, with the above works, one can detect the location of the cracks but cannot quantify the cracks. In practical engineering applications, the quantitative result of crack is very important, which can be directly used to analyze the stress status of the structure, and thus is an important basis for bridge maintenance decision. Moreover, the entire learning process of the present DL is a dimension reduction process. The crack width of a concrete bridge is usually narrow compared to the image. Hence, the crack information may be lost during the learning process.

The fully convolution neural network (FCN) has been widely studied to realize pixel-level crack detection and measurement. FCN is an end-to-end, pixel-to-pixel convolutional network for semantic segmentation. An FCN is composed of downsampling and upsampling parts. The downsampling part mainly includes convolutional layers and pooling layers. The upsampling part includes deconvolution layers [17]. For example, Yang et al. employed the FCN to realize the pixel-level crack classification and measurement. The results showed that the accuracy of segmentation was 97.96%, and the relative quantification error was within 24.01% for crack width [18]. Li et al. utilized the U-Net to realize the location of concrete cracks in the tunnel. The segmentation accuracy was 92.25%, and the relative quantification error was within 18.57% for crack width [19]. They further adopted the segmentation network of FCN and Naïve Bayes data fusion (NB-FCN) to realize the location of concrete cracks. The cracks were quantified by introducing postprocessing with accuracy being 93.2% [20].

For the concrete bridge crack in practical engineering, the length is usually long and the width is narrow. Usually, the crack width may be less than 0.5 mm. Thus, the crack detector is more sensitive to the quantification error in the width direction. With the present mainstream camera resolution, a practical microcrack may contain only a few pixels. Even the FCN can recover the original image from the feature map with the deconvolution layers, the original scale information of the narrow crack width cannot be recovered according to the information theory. Note that it is hard to use high-resolution images as inputs for a neural network due to the limited computation power. If digital image processing is used for detection and quantification, the image plane needs to be smooth enough with less noise, which may not be practical in the engineering environment. Therefore, the digital image processing method or DL method cannot solve the microcrack task detection well separately.

In this paper, in order to solve the microcrack detection task in practical engineering, we proposed a modified concrete bridge crack detector based on a DL-assisted image processing approach. The computational novelty is threefold. First, the modular split design is considered. More specifically, the latest YOLOv5 DL method is utilized for crack classification and location, and the digital image processing method is employed for crack quantification. Second, we propose a novel digital image quantification and crack recovery method by using a region connected component search algorithm based on the crack trend. Third, the narrowest crack width that can be detected reaches 0.15 mm, and the absolute error can be controlled within 0.05 mm.

The rest of the paper is organized as follows. In Section 2, the network architecture of the modified bridge crack detector based on the DL-assisted image processing approach is presented. The detection, quantification, and visualization algorithms are described. In Section 3, the establishment of bridge damage dataset, offline data augmentation, training process, and crack classification and location results obtained by the YOLOv5 algorithm are introduced in detail. We analyze crack connected component optimization methods, including the mask filter used to remove handwritten marks, the ratio filter adopted to eliminate speckle linear noises, the fusion of the same crack detection bounding boxes, the regional adaptive threshold segmentation, and the connected components search approach based on the crack trend of area. Besides, we compare the crack quantification accuracy with and without these optimization methods. A comparison between the calculated crack width based on the connected components and the results measured by a bridge engineer using the crack width gauge is given. Finally, conclusions are drawn.

2. Methodology

For practical bridge damage detection, it is necessary to know whether there is damage, where is the damage, and what is the size of the damage. Here, we proposed a modified bridge crack detector based on the DL-assisted digital image processing technique. The microcrack task in practical engineering is focused on.

2.1. Architecture of the Crack Detection Network Based on the DL-Assisted Image Processing Approach. The overall architecture of the damage detection network based on the DL-assisted image processing approach is shown in Figure 1. It includes two modular functional parts. At first, the region with crack is identified through the DL-based detection

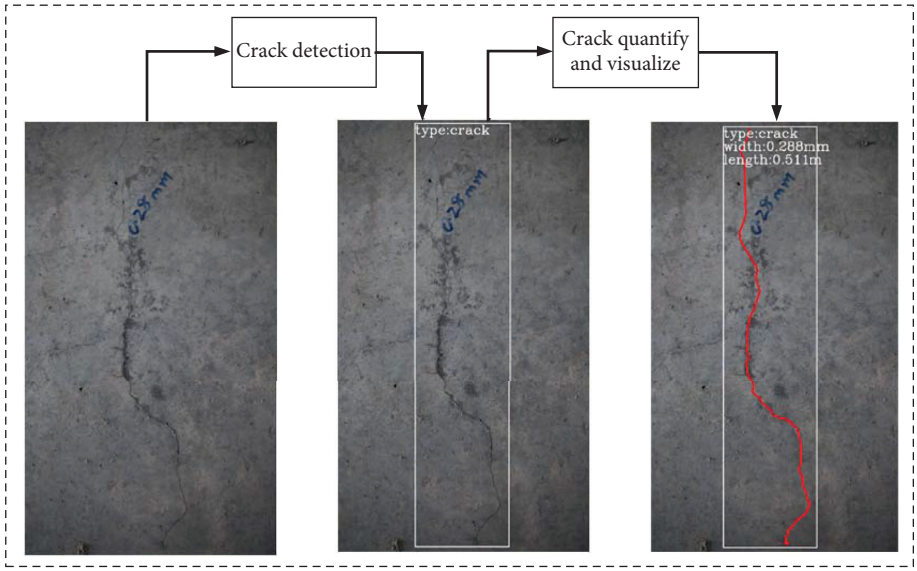


Figure 1: The overall flow of the modified concrete bridge crack detector.

network. Here, YOLOv5 is employed to get the predicted bounding boxes. After that, according to the predicted bounding boxes, the digital image processing techniques are adopted to quantify the length and width of the cracks, and the crack details are further visualized in the image.

2.2. Detection Algorithm. We use the target detection method to determine whether there is a crack in the image and where is the crack. There are many excellent neural networks in the field of target detection [21], such as Faster R-CNN [22], SDD [23], and EfficientDet [24]. We used the latest network of the series of YOLO [25–28], that is, the YOLOv5 network. Note that there is no corresponding paper on this network at present, but the code has been open. It is worth mentioning that the detection method here can be any network due to the modular design. When a subsequent network that is more suitable for our detection task appears, it can be adopted.

In the following, we introduce the YOLOv5 network. The overall structure is shown in Figure 2. It can be divided into three parts: backbone layer, neck layer, and head layer. In the input part, the long edge of the original image was first scaled to 608, and the short edge was scaled proportionally and filled to 608 with 0 pixels. Thus, the input size is $608 \times 608 \times 3$ for the backbone. The backbone is responsible for the feature extraction, and the neck layer performs the multiscale feature fusion. In the head layer, the bounding box is obtained by predicting the coordinate offset, and the loss is calculated by means of generalized intersection over union (GIOU) between the predicted box and the ground truth box. Bounding box classification and regression is directly finished in the head layer.

More precisely, the backbone layer is responsible for feature extraction of input image data, which is divided into Focus structure and CSPNet structure. The cross-stage partial network (CSPNet) is used in the backbone layer, which can extract abundant information features from the input image [29]. CSPNet can be combined with other networks such as ResNet [30] and ResNeXt [31]. Two kinds of convolution kernels are employed. One is 3×3 and is used for feature extraction. The other convolution kernel has a size of 1×1 and is used to flexibly control the depth of feature maps. The output feature map of the backbone layer is $19 \times 19 \times 512$. Here, the role of the Res unit is to increase the depth of the network and avoid the gradient vanishing or gradient explosion caused by the increase of the depth of the proposed detection network. The neck layer uses a path aggregation network (PANET) [32] and is mainly used to generate a feature pyramid. PANET is based on Mask R-CNN and feature pyramid network (FPN) frameworks [33, 34]. The convolution kernel of the neck layer is the same as that of the backbone layer. In the spatial pyramid pooling (SPP) structure, multiple pooling layers are used. The size of the pooling core is 5×5, 9×9, and 13×13, and the padding is 2, 4, and 6, respectively. After the operation of the neck layer, the output feature map scales are $19 \times 19 \times 255$, $38 \times 38 \times 255$, and $76 \times 76 \times 255$. The head layer is a universal detection layer, which is the same as that of YOLOv3 and YOLOv4 [27, 28]. In the head layer, bounding boxes are generated and classified. There are three output heads, whose strides are 8, 16, and 32, respectively. The receptive field of a small-scale feature map is larger, which is used to detect large targets. The receptive field of a large-scale feature map is smaller, which is used to detect small targets. The three-scale feature maps are jointly used to improve the accuracy of the network. For these three output heads, each output feature map has 19×19, 38×38, or 76×76 cells, respectively. Each cell generates 3 bounding boxes; each bounding box contains 4 positional parameters, 1 confidence value, and 80 classes (classes = 80 of the COCO dataset is used by default, and crack is one of the classes). Therefore, the depth of the output

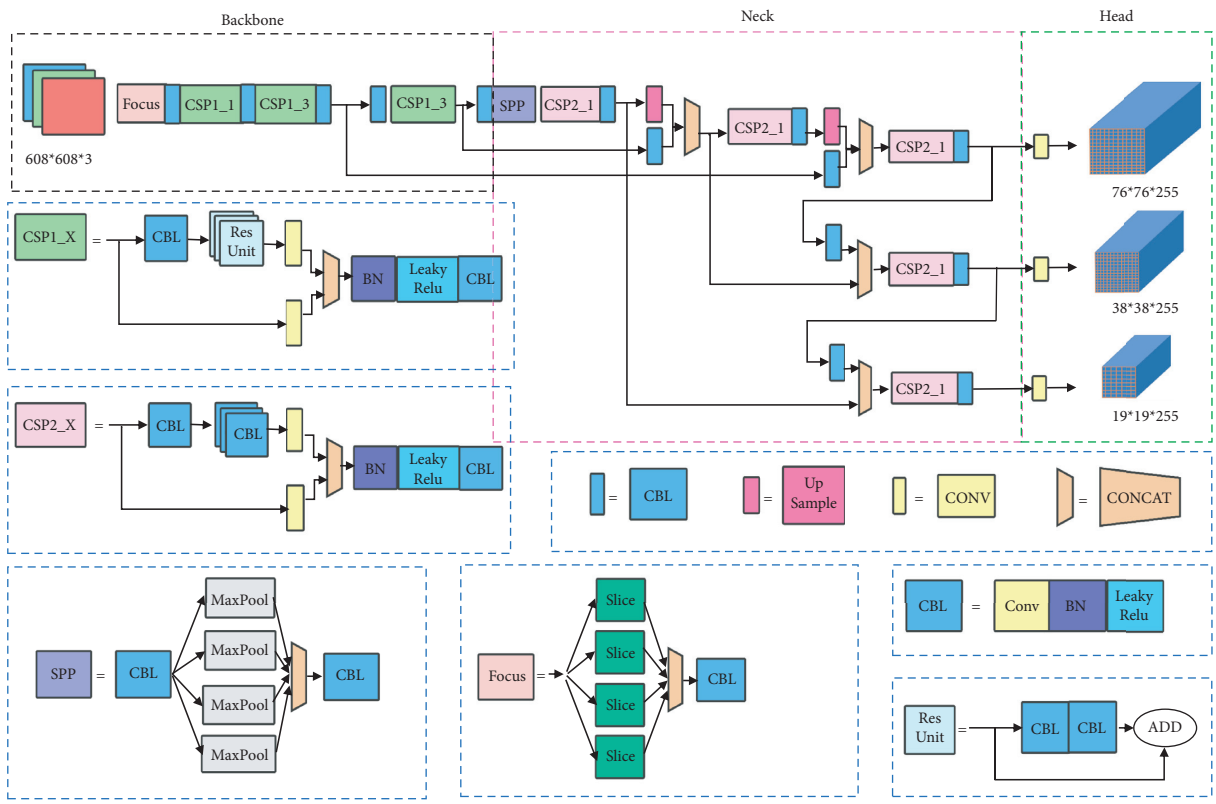


Figure 2: The overall architecture of the detection network based on YOLOv5.

feature map is $(4+1+80)\times 3 = 255$. The default confidence threshold in YOLOv5 is 0.45. The bounding box is judged as a positive sample, namely, crack, when its confidence is greater than 0.45; otherwise, it is judged as noncrack.

In the process of training, YOLOv5 uses GIOU [35] to calculate loss, which can be calculated as

$$\begin{aligned} \mathrm{IOU} &= \frac{|A \cap B|}{|A \cup B|}, \\ \mathrm{GIOU} &= \mathrm{IOU} - \frac{|A_c - U|}{A_c}, \end{aligned} \tag{1}$$

where A and B represent the predicting box and ground truth box, respectively, A_c represents the smallest enclosing rectangle of two rectangular boxes, and U represents the union of two rectangular boxes; that is, $A \cup B$.

2.3. Quantification and Visualization Algorithm. We propose a method to realize the crack quantification and visualization. The processing steps are presented in Figure 3. Through the target detection network, we get the bounding box of the crack in the original image, as shown in Figure 3(a). Because the resolution of the original image is high, in order to reduce the amount of calculation, we first crop the original image to only retain the area with cracks according to the bounding box, as shown in Figure 3(b). Note that there are many noises in the crack area, such as the salt and pepper noise in actual shooting, the concrete bridge surface handwritten marks, dirt, weeds, and shadow occlusion noises. In order to extract more pixel-level information about cracks, we extract the binary image of cracks through image processing techniques. The main steps include median filtering and graying, CLAHE and local segmentation, and image binary conversion [36–38]. The image obtained after median filtering and graying is shown in Figure 3(c). We carry out CLAHE processing on the image and then divide the image into multiple square regions from top to bottom with the short side of the image as the side length and calculate the redundant nonsquare parts separately. For each region, local segmentation is performed as shown in Figure 3(d). Then, the gray distribution of all pixels in each square area is calculated.

Binarization is performed in each square region by adaptive threshold segmentation. In our dataset, the resolution of the image taken by the camera is 8688×4888. The camera takes images from a distance of 50 cm. The focal length of the camera is 35 mm. We randomly selected 50 images from the dataset. For these 50 images, the number of pixels occupied by the crack width is recorded manually by the bridge engineering experience. The results show that the number of pixels occupied by the crack width is less than 12 pixels. Therefore, in each square area, the maximum number of pixels belonging to the crack is $12\sqrt{2}a$, where a is the side length (pixel numbers) of the square area. In the image histogram of each square area, the gray value corresponding to the pixel number of $12\sqrt{2}a$ is used as the threshold of image binarization [39]. If the gray value of a given pixel is larger than the threshold, it is judged as background; otherwise, it is judged as crack. The binary image of each square

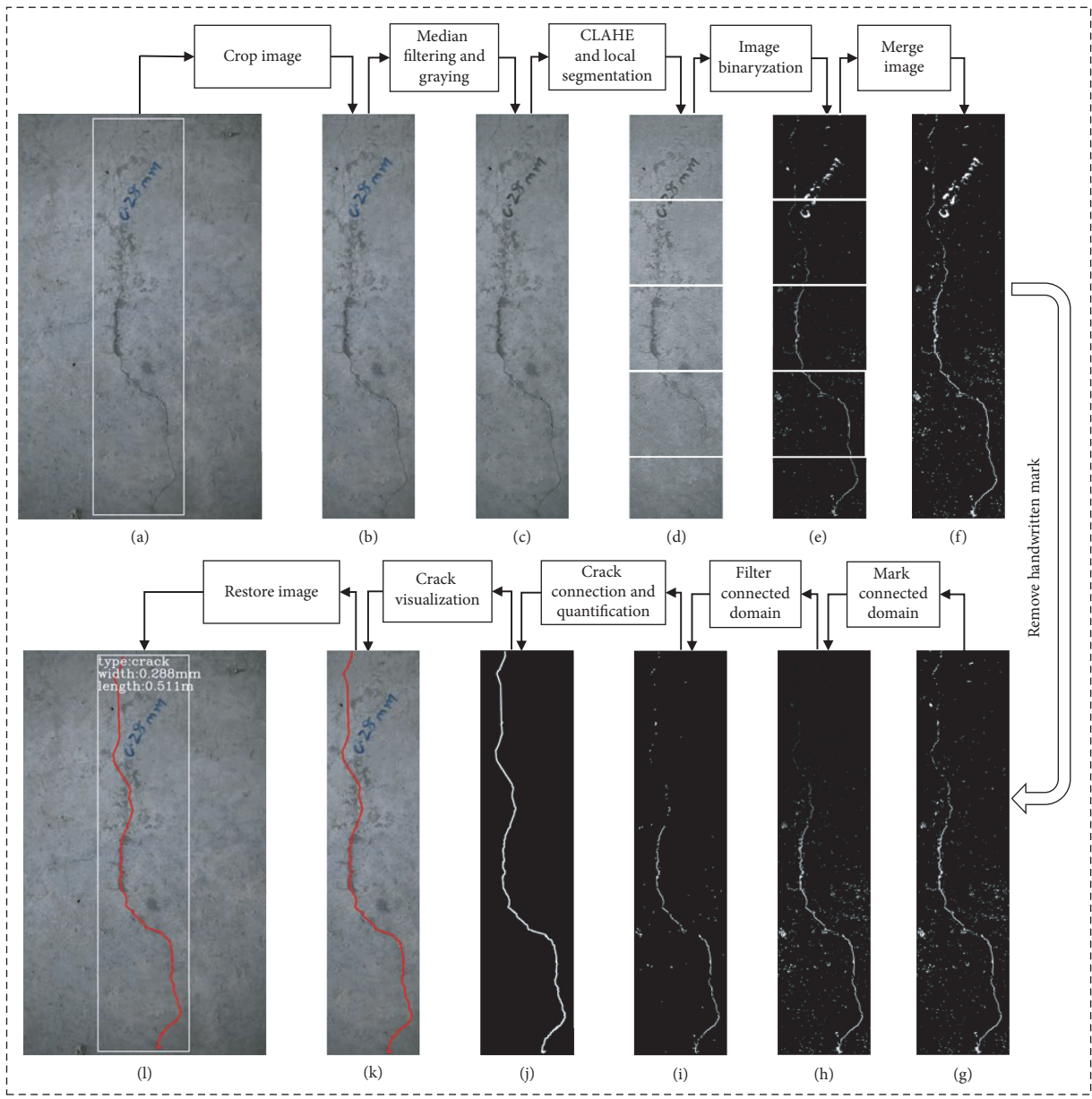


Figure 3: The flowchart of crack quantification and visualization.

area is shown in Figure 3(e). Here, the black pixel 0 is the background, and the white pixel 255 is the potential crack. After threshold segmentation for each square region, these small regions are combined to get a complete binary image of the crack region, as shown in Figure 3(f). It can be seen that there are still some handwritten marks, noises, spots, and linear noises in the image. In order to better deal with these noncrack noises, we removed handwritten marks with a mask filter and filtered speckle linear noises with a ratio filter. The filtered results are shown in Figure 3(g).

Through the connected component labeling algorithm [40, 41], the binary image obtained after removing the handwritten marks is transformed into the corresponding connected component graph, as shown in Figure 3(h). It can be seen from the connected component diagram that the connected component belonging to the crack is approximately linear, and the length is generally much larger than the width. Therefore, line-like noise is further filtered out by the aspect ratio feature of the connected component, and speckled noise is filtered out by the area feature, as shown in Figure 3(i). It can be observed that there are some intermittent connected component fragments, and there are still some unremoved noises. In order to make the visible cracks more complete and clear and in line with the actual trend of cracks, we propose a region connected component search algorithm based on the connected component of cracks. The binary image of cracks after the connection is shown in Figure 3(j).

After obtaining the contour coordinates of the crack binary image, the length of the crack is calculated according to the obtained coordinates, and the average width of the crack is calculated by dividing the area of the connected

component by the length of the crack. When the contour is drawn in the image of the crack region, the image visualization task of the crack is realized. The visualization results are shown in Figure 3(k). Finally, the visualization image is mapped back to the original image according to the location of the bounding box so that the visualization of the crack on the original image is realized. Meanwhile, the results of the crack length and width are displayed on the original image, as shown in Figure 3(l).

3. Experiments and Results

In the present work, the experiments were performed on a computer with the following configuration: two Intel Xeon(R) E5-2620v4@2.1 GHz CPUs; 64 GB Random Access Memory; and NVIDIA Tesla P40 24 GB GPU. This method was implemented based on the DL framework PyTorch [42]. The experiments were conducted by modifying the open-source library of YOLOv5 and OpenCV [43].

3.1. Experiments

3.1.1. Dataset. We manually collected and labeled the images to establish a dataset for training and testing the network. Three types of bridge damage, that is, crack, spall, and rebar, are included in the dataset. 1000 digital photos with 8688×4888 pixels are taken by a Canon EOS 5DS R camera. All the images used in this study were acquired from the periodic inspection for bridges by the CCCC First Highway Consultants Co., Ltd. Note that the shooting distance and focal length were fixed during the image acquisition process, which was helpful for quantifying the damage. Here, we focus only on the concrete cracks. We took cracks as targets, marked ground truth, and produced a dataset. The number of images with the crack in the dataset is 487, and the dataset is divided into three parts, training set, verification set, and test set. The verification set and test set, respectively, account for 10% of all the images, and the training set accounts for 80% of all the dataset.

In order to speed up the convergence of the network, we made statistics on the average value and standard deviation of the pixels of the three channels of the image in the dataset and obtained the following results: $\text{mean}_{R,G,B} = [114.86, 111.59, 109.38]$ and $\text{std}_{R,G,B} = [21.14, 21.38, 22.04]$.

After feeding the image into the network, the above two sets of values are used for normalization by the Z-score standardization method.

3.1.2. Offline Data Augmentation. As the amount of data collected in the actual engineering was limited, we adopted data augmentation to prevent the problem of overfitting in the training process. Here, mosaic, random rotation, random cropping, Gaussian noise, and manual exposure are used [28, 44]. Data-augmented samples of the images in the crack dataset are shown in Figure 4.

3.1.3. Training. During the network training, batch processing was used to increase the training speed of the network, the learning rate was set to 0.01, and the batch size of both network training and validation was 16. A total of 200 training cycles were trained. Figure 5 shows the performance of the network in the training process. The loss curve of the training set and the validation set in the network training process, as well as the precision, recall, and mean average precision (mAP) curve on the validation set, are presented [45–47]. It can be seen that the training algorithm converged rapidly, and high mAP can be obtained.

3.2. Results

3.2.1. Crack Detection Results. In order to quantify the performance of the network in the target detection phase, data from the validation set were used to verify the network performance, and the results are presented in Table 1. Here, 88 images with 162 targets are detected. When the IoU threshold is 0.5, the calculated mAP is mAP@0.5 = 0.987. When the threshold of IoU ranges from 0.5 to 0.95 with a step of 0.05, the mean value of mAP is mAP@0.5: 0.95 = 0.987.

Figure 6 presents the network detection results of cracks in the experiment. We can find that, with the detection network, the cracks can be successfully detected. But for some cases, a single crack is detected as multiple bounding boxes.

3.2.2. Crack Quantification and Visualization Results. (1) Noise Removal.

(1) Mask filter removes handwritten marks.

Here, we remove the handwritten marks to reduce the connection error caused by connecting to the handwritten marks. After the crack binary image is extracted, the color (blue) information of the handwritten mark is extracted in the HSV (Hue, Saturation, Value) color space to obtain the mask of the blue marker. Note that the extracted mask also contains other speckle noises, which are much smaller than the area of handwritten marks. Therefore, image morphology operation, image corrosion, and image expansion operations can be carried out on the mask to remove these speckle noises, leaving only the part of handwritten marks. As shown in Figure 7, the binary image is multiplied by the corresponding pixel points of the mask after pixel inversion to obtain the crack binary image with handwritten marks removed. The visual results of cracks before and after the removal of handwritten marks are also presented. It can be seen that, with the mask filter, the connection to the handwritten mark is avoided in Figure 7(d).

(2) Ratio filter removes noise.

In order to remove the noncrack noise through the crack morphological features and retain the crack pixel information as much as possible, we have made statistics on the width and length scale information in the crack connected component diagram.

Figure 4: Data-augmented samples of the images in the crack dataset: (a) original image, (b) mosaic, (c) random rotation, (d) random cropping, (e) Gaussian noise, and (f) manual exposure.

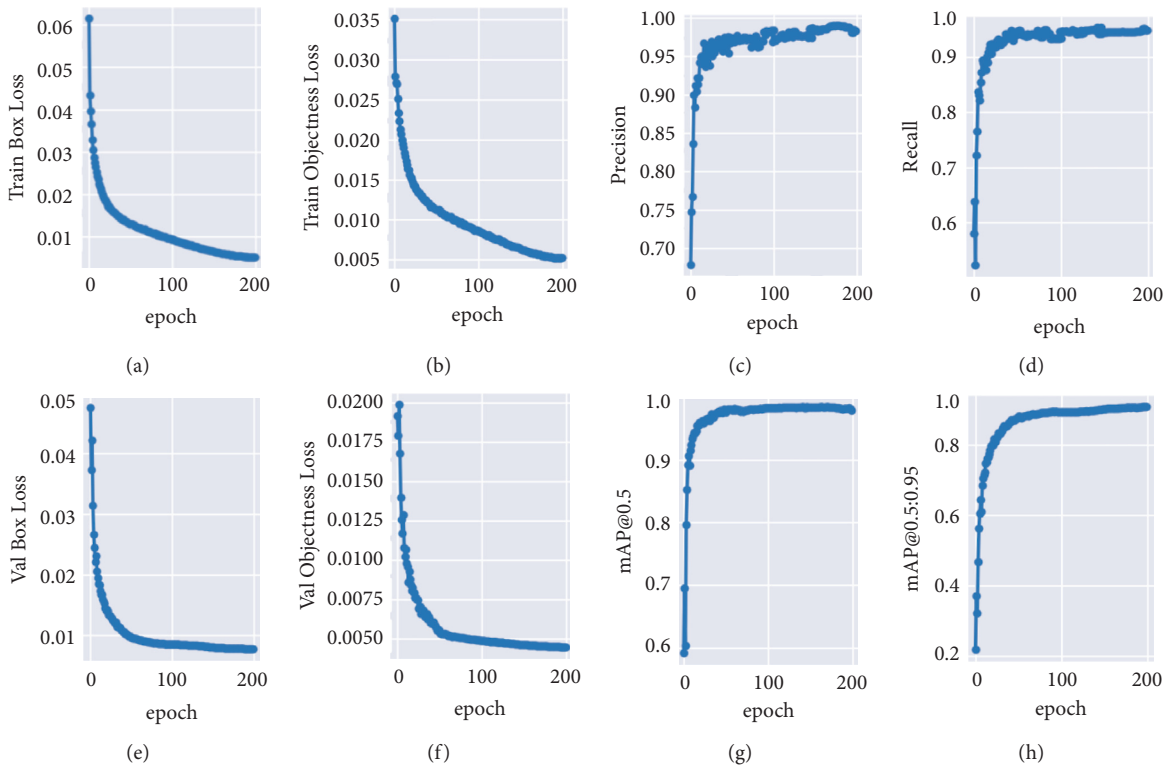
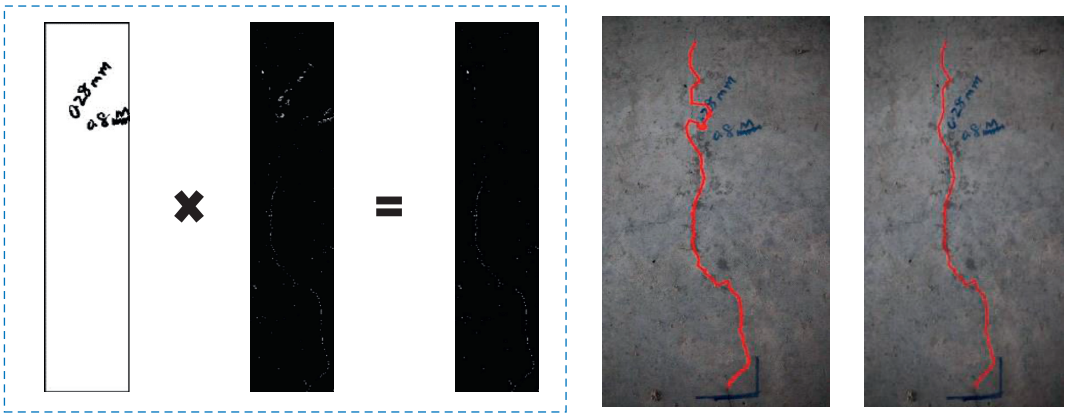


Figure 5: The performance of the network in the training process: (a) train box loss, (b) train objectness loss, (c) precision, (d) recall, (e) val box loss, (f) val objectness loss, (g) when the IoU threshold is 0.5, the calculated mAP (map@0.5), and (h) when the threshold of IoU ranges from 0.5 to 0.95 with a step of 0.05, the mean value of mAP (mAP@0.5 : 0.95).

Table 1: Detection result.

Images	Labels	Precision	Recall	mAP@0.5	mAP@0.5 : 0.95
88	162	0.92	0.975	0.987	0.987

Figure 6: The network detection results of cracks.



(a) (b) (c) (d) (e)

Figure 7: Method of removing handwritten marks and the visual results of cracks before and after the removal of handwritten marks: (a) mask after pixel inversion, (b) crack binary image, (c) crack binary image after removing handwritten marks, (d) visualization with handwritten marks not removed, and (e) visualization with removing handwritten marks.

Figure 8(a) is the original crack binary image extracted through image processing, and Figure 8(d) is the corresponding width-length ratio distribution of all connected components. It can be seen that the width-length ratio of all connected components in the extracted crack binary image is dispersive. Figure 8(b) is the ideal crack binary image, which is the expected binary diagram of cracks after removing

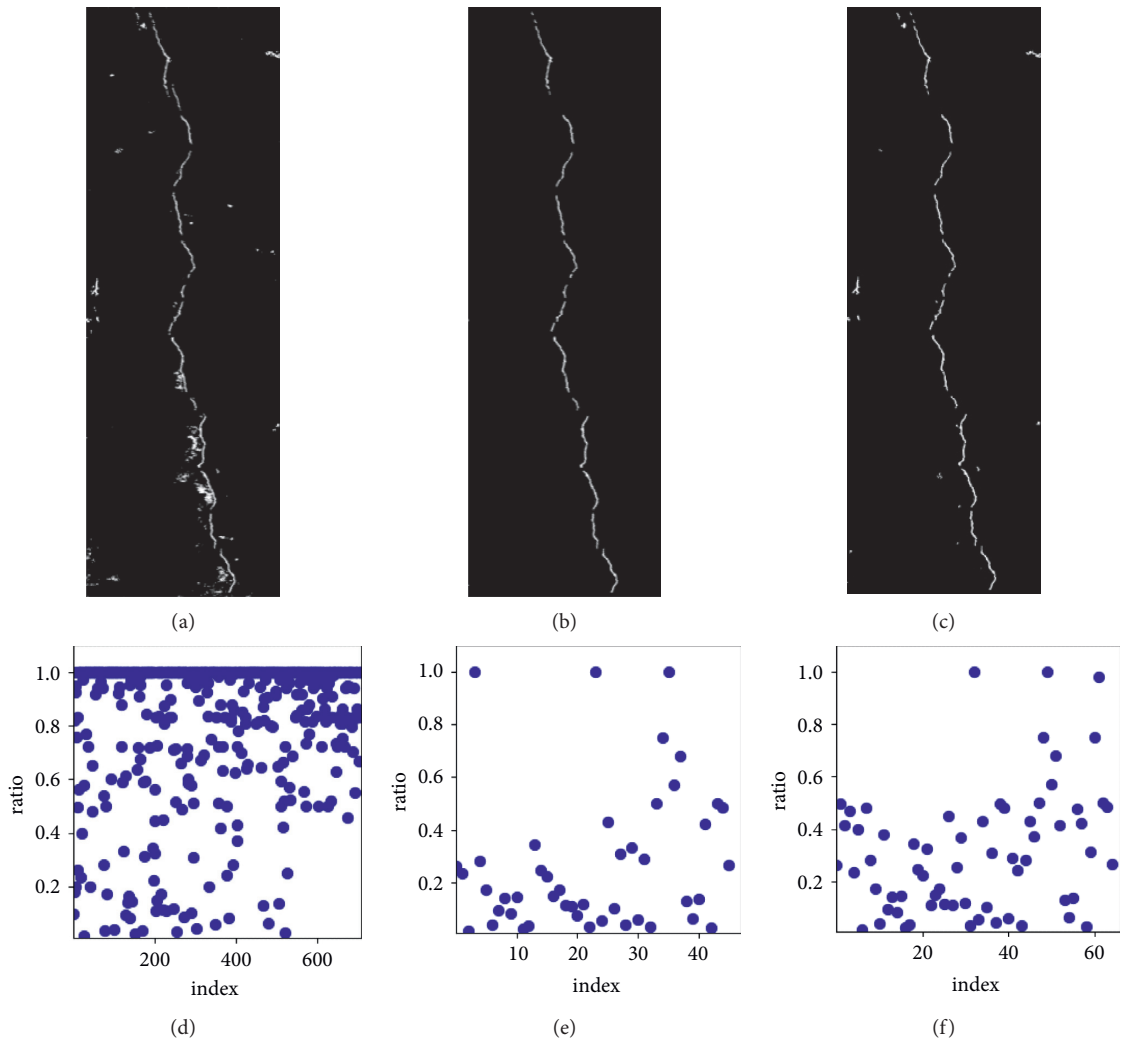


Figure 8: The process of filtering the connected domain to eliminate the noncrack noise: (a) the original crack binary image, (b) the ideal crack binary image, (c) the actual crack binary image, and (d–f) the width-length ratio of the connected domains corresponding to (a), (b), and (c).

noise. However, it is impossible to separate these real connected components completely by the available connected component information. Therefore, it is desirable to retain more connected components that are likely to belong to cracks and remove those connected components that are less likely to belong to cracks. According to the usual crack shape, the length of the connected component belonging to the crack is generally much larger than the width. Thus, the connected component with a small width-length ratio is more likely to belong to the crack. Figure 8(e) shows the width-length ratio of the connected component corresponding to the ideal crack binary image. It can be seen that the ratio of the majority of connected components belonging to cracks is less than 0.6. According to Figure 8(e), we sorted the connected components in the extracted crack binary image by the width-length ratio in ascending order and selected the first 60% connected components. Figure 8(c) is the actual crack binary image obtained after filtering according to width-length ratio characteristics. Figure 8(f) displays the width-length ratio corresponding to Figure 8(c). Obviously, most of the cracks that can maintain crack morphology are left after filtering.

(2) Fusion of Multiple Bounding Boxes for the Same Crack. When the detection network is used to identify the cracks, a crack in the image may have multiple bounding boxes because of the disconnection of the cracks or the inaccurate network identification. In that case, the visualized cracks are not continuous, and the number of cracks cannot be counted correctly. Therefore, we further perform a fusion operation for multiple bounding boxes.

We sort multiple bounding boxes detected in an image in ascending order according to the y coordinate of the upper left point and save them in the list, as shown in Figure 9(a). The coordinate of the upper left point of the first bounding box A is (x_1, y_1), and the coordinate of the lower right point of the first bounding box is $(x_{1'}, y_{1'})$. For the remaining bounding boxes with the coordinates of the upper left point

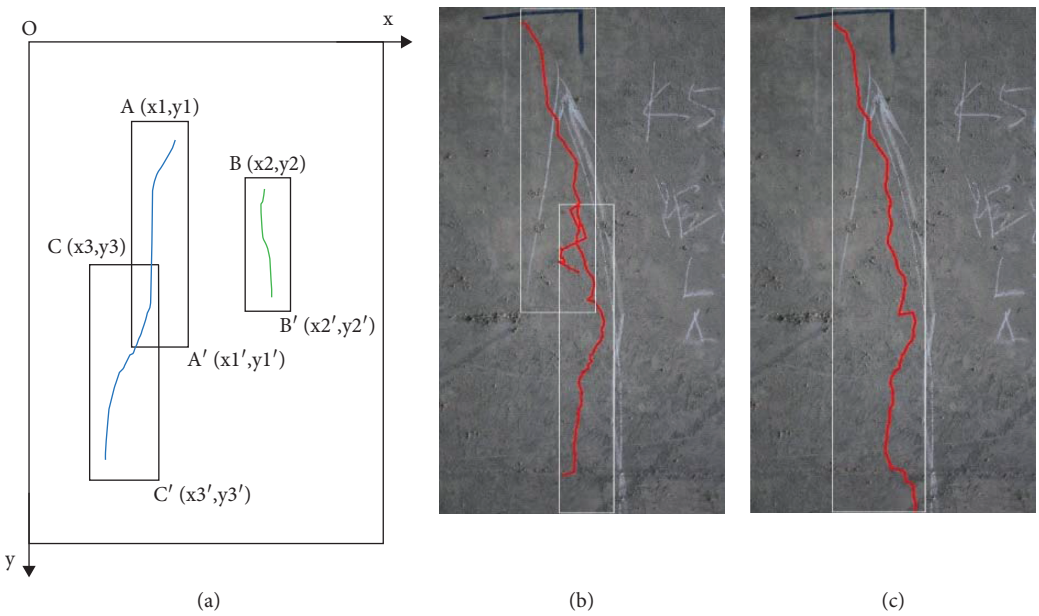


Figure 9: A visual comparison of multiple bounding boxes in the same crack before and after fusion: (a) bounding boxes' diagram, (b) multiple bounding boxes before fusion, and (c) fusional bounding box.

(x_i, y_i), we examine whether the following condition is satisfied:

$$x_1 - \Delta x \le x \le x_{1'}, \quad y_1 \le y \le y1', \tag{2}$$

where $\triangle x$ is the width of the first detection box, $\triangle x = |x_{1'} - x_1|$.

If a bounding box meets the condition, the bounding box and the first bounding box overlap and need to be fused. For example, the bounding box covers C and C' are overlapped with the bounding box covers A and A'. If no bounding box satisfies (2) for the first bounding box, then we repeat the process from the second bounding box until all the bounding boxes are checked. The aim of bounding box fusion is to find the minimum outer rectangle of the two bounding boxes. Specifically, we adopt the coordinates of the upper left point and the lower right point of the two bounding boxes and return the vertex coordinates of the fusional bounding box. When the bounding boxes are fused, we delete the original two bounding boxes from the list and add the fusional bounding box to the top of the list. A visual comparison of multiple bounding boxes corresponding to the same crack before and after fusion is shown in Figures 9(b) and 9(c). Obviously, in Figure 9(b), two bounding boxes are detected for a single crack, while only a single fusional bounding box exists in Figure 9(c).

(3) Crack Connection Strategy. As shown in Figure 3(i), the filtered connected component image contains some intermittent connected component fragments and some unremoved noises. In order to make the visible cracks more complete and clearer, we need to connect these connected components in line with the actual trend of the cracks. Meanwhile, we consider the method to avoid the connection to noise.

We propose a region connected component search algorithm based on the crack trend. The flowchart of the proposed algorithm is shown in Figure 10(a). The first step is to initialize the index and vertex coordinates of each connected component. The second step is to update the connected component and select the connected component with the largest aspect ratio in the image as the current connected component. In the third step, we consider a downward connection from the current connected component and build a square search box according to the lower vertex of the current connected component. The width of the search box is equal to the width of the image. The schematic diagram of the connection is shown in Figure 10(b). The fourth step is to find the target connected component to be connected in the search box. In the search box, the distance d and the angle θ between the lower vertex of the current connected component and the upper vertex of the next candidate connected component are calculated. Then, the confidence degree c can be calculated as follows:

$$c = d \times \sin\frac{\theta}{2}. \tag{3}$$

We can choose the candidate connected component with the smallest c as the target connected component.

If such a target connected component is found in the search box, a white line is used to connect the lower vertex of the current connected component and the upper vertex of the target connected component. The width of the white line is the average width of the current connected component and the target connected component. After the connection is complete, we delete the index of the current connected component, update the target connected

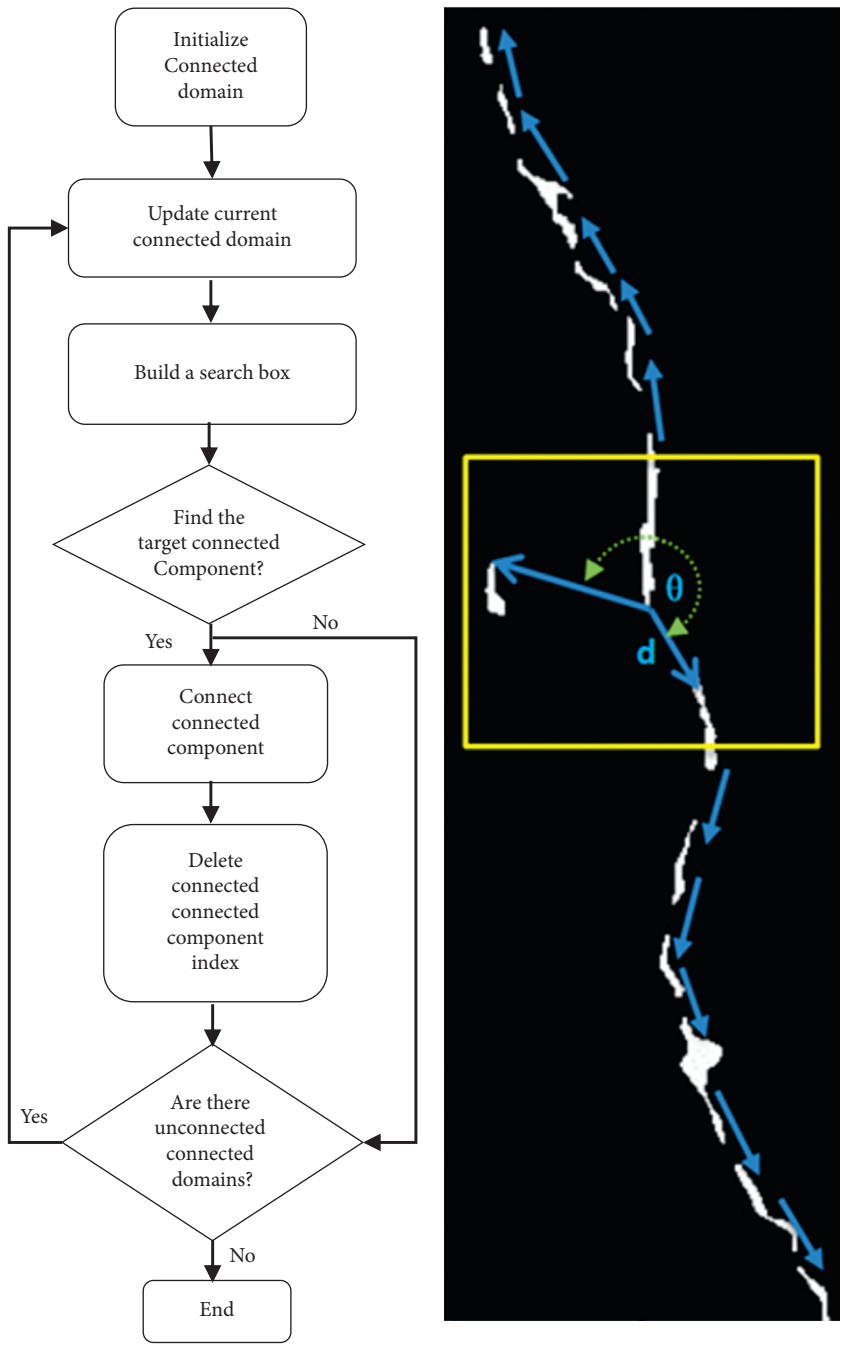


Figure 10: (a) Flowchart and (b) schematic diagram of the crack connection algorithm.

component as the current connected component, and continue the downward connection. If such target connected component cannot be found in the search box, we directly delete the current connected component index, select the next connected component index as the novel current connected component index, and continue the downward connection. When the connection is connected to the index of the last connected component, if there is no other unconnected connected component in the dictionary set of the connected component, the connection process ends. In the present application scenario, a single longitudinal extension of the crack is considered, which is the most common one. So far, the downward connection from the initial connected component is completed. Correspondingly, the upward connection from the initial connected component can also be performed in a similar way.

(4) Crack Quantization Result. In the following, we calculate the actual length and width of the crack from the target connected component. We assume that the total number of connected components selected to represent

Table 2: Quantitative results of crack width.

Crack number	The width measured by a crack width gauge (mm)	Quantized mean width $\overline{w}_c$ (mm)	Absolute error (mm)	Relative error (%)
#101	0.40	0.39	−0.01	−2.5
#102	0.30	0.28	−0.02	−6.7
#103	0.26	0.29	0.03	11.5
#104	0.31	0.34	0.03	9.7
#105	0.29	0.27	−0.02	−6.9
#106	0.18	0.19	0.01	5.6
#107	0.45	0.47	0.02	4.4
#108	0.40	0.37	−0.03	−7.5
#109	0.40	0.38	−0.02	−5.0
#110	0.40	0.41	0.01	2.5
#111	0.31	0.34	0.03	9.7
#112	0.35	0.39	0.04	11.4

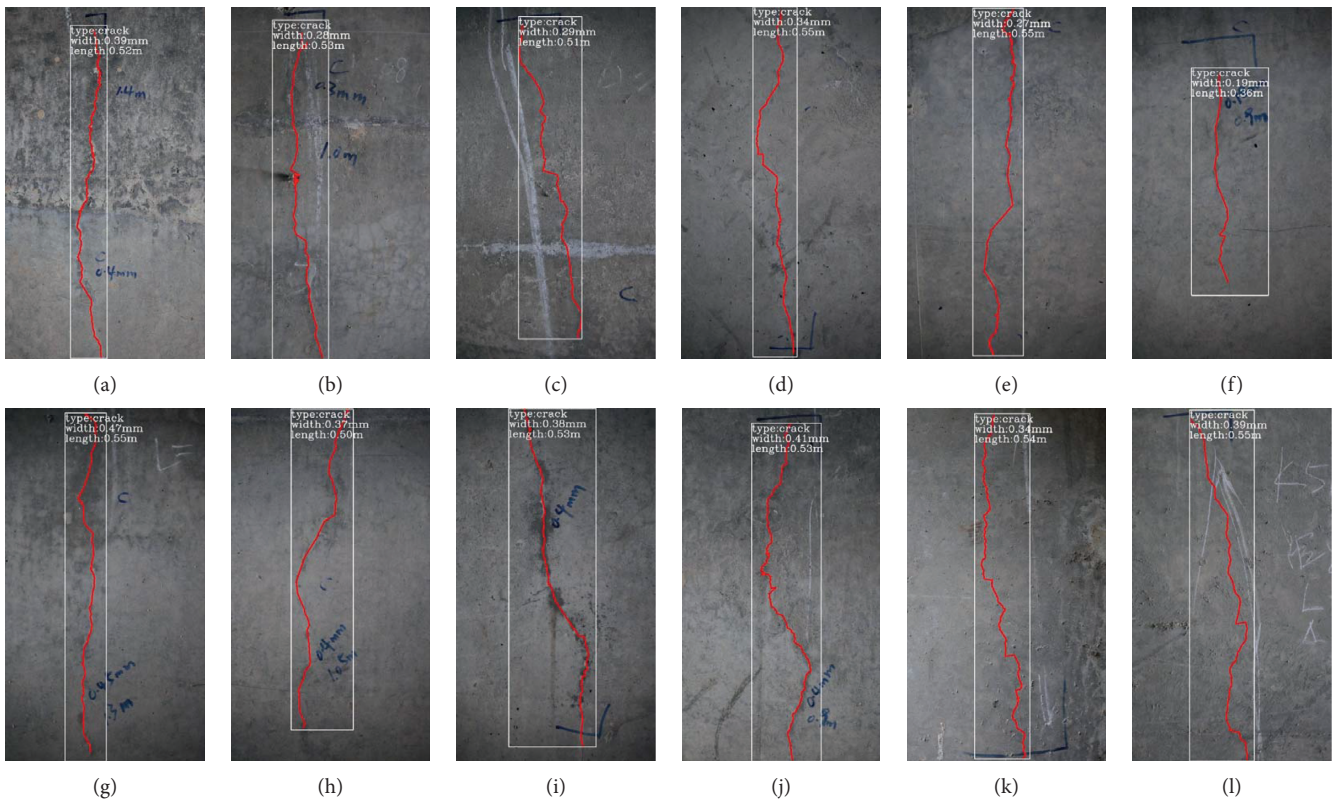


Figure 11: Graph of crack quantization results: (a) #101, (b) #102, (c) #103, (d) #104, (e) #105, (f) #106, (g) #107, (h) #108, (i) #109, (j) #110, (k) #111, and (l) #112.

the actual crack is N. The area of the i-th connected component is S_i, and the length is l_i. The area of the connected component is just the number of pixels in the connected component.

For quantification of the crack length, the length of the connected domain is the number of pixels after the skeleton extraction of the connected domain. The actual length of the crack L is equal to the sum of the length of the entire connected domains times the actual distance δ_y of each pixel in the vertical direction of the image, namely,

$$L \cong \delta_y \sum_{i=1}^{N} l_i. \tag{4}$$

For quantification of the crack width, the width of the i-th connected component is calculated as $w_i = (s_i/l_i)$. Thus, the average width of the N filtered connected components can be calculated as follows:

$$\overline{w}_d = \frac{\sum_{i=1}^{N} w_i}{N}. \tag{5}$$

Therefore, the average width of the crack $\overline{w}_c$ is equal to the average width of the entire connected component times the actual distance δ_x of each pixel in the horizontal direction of the image, namely,

$$\overline{w}_c \cong \delta_x \frac{\sum_{i=1}^{N} (s_i/l_i)}{N}. \tag{6}$$

Here, the resolution of the image taken by the camera is 8688×4888. The camera takes images from a distance of 50 cm. The focal length of the camera is 35 mm. The comparison between the calculated $\overline{w}_c$ and the crack width measured by a crack width gauge in the actual bridge detection engineering is shown in Table 2. It can be seen that the widths of cracks in actual concrete bridges are generally less than 0.5 mm. The absolute error of the crack width obtained by our proposed crack quantification method is −0.03~0.04 mm, and the relative error is −7.5% to 11.5%. For the cracks above 0.15 mm, the absolute error of the width quantification results is less than 0.05 mm. The crack quantification results and visualization results are shown in Figure 11. It can be observed that good measurement accuracy is obtained.

4. Conclusion

In conclusion, we proposed a method of concrete bridge crack detection and quantification based on a DL-assisted image processing approach. The detection and the quantification phases are separately designed, which can easily replace the target detection network with an advanced DL algorithm. The target detection network based on DL is used to determine whether there is a crack in the image and then extract the crack area. In addition, we proposed a new digital image quantification and crack visualization method by introducing the region connected component search algorithm based on the crack trend.

In our dataset, we collected 487 original images. Among them, 399 images were processed by offline data augmentation to obtain 3365 images as a training set, and the remaining 88 images were used for testing. The proposed DL-assisted digital image crack detection and quantification approach achieved 92.0% precision, 97.5% recall, and 98.7% mAP@0.5 in the detection stage for the testing set. With the subsequent noise processing and connected component connection, the narrowest crack that can be detected, quantified, and visualized is about 0.15 mm, and the absolute error is within 0.05 mm. These results show that the proposed crack detection and quantification method can improve the detection efficiency and reduce the detection cost, which is helpful and valuable for the intelligent development of bridge detection.

Note that there are still several directions that need to be improved. First, the crack merging, which is desired for crack statistics and evaluation, is not carried out in our solution. Second, the current dataset is still small, and a larger dataset may further improve the accuracy of the proposed algorithm in practical application.

Data Availability

All the data are available from the corresponding author (727705858@qq.com) upon request.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (2021YFB2801900, 2021YFB2801901, 2021YFB2801902, and 2021YFB2801904) and the National Natural Science Foundation of China (nos. 61974177 and 61674119).

References

[1] Y. Li and C. Wang, "Research advances in long-life of worldwide bridges and corresponding reflections," *Bridge Construction*, vol. 49, pp. 17–23, 2019.

[2] I. Abdel-Qader, O. Abudayyeh, and M. E. Kelly, "Analysis of edge-detection techniques for crack identification in bridges," *Journal of Computing in Civil Engineering*, vol. 17, no. 4, pp. 255–263, 2003.

[3] H. Oh, N. Garrick, and L. Achenie, "Segmentation algorithm using iterative clipping for processing noisy pavement images," in *Proceedings of the Imaging Technologies: 2nd International Conference on Techniques and Application in Civil Engineering*, pp. 138–147, Davos, Switzerland, May 1997.

[4] S. Yang, L. T. Shao, X. X. Guo, X. Liu, and B. Y. Zhao, "A crack segmentation approach using the combination of gray thresholds and fractal feature," *Advanced Materials Research*, vol. 487, pp. 622–626, 2012.

[5] S. Liang, J. Xing, L. Xie, and J. Wang, "An adaptive threshold-based Canny algorithm for crack detection," *Microcomputer & Its Applications*, vol. 36, pp. 35–37, 2017, 41.

[6] A. M. A. Talab, Z. Huang, F. Xi, and L. Haiming, "Detection crack in image using Otsu method and multiple filtering in image processing techniques," *Optik*, vol. 127, no. 3, pp. 1030–1033, 2016.

[7] 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.

[8] Z. Lei, F. Yang, D. Zhang, and J. Ying, "Road crack detection using deep convolutional neural network," in *Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP)*, pp. 3708–3712, Phoenix, AZ, USA, 25-28 September 2016.

[9] Y.-J. Cha, W. Choi, and O. Büyüköztürk, "Deep learning-based crack damage detection using convolutional neural networks," *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, no. 5, pp. 361–378, 2017.

[10] Y.-J. Cha, W. Choi, G. Suh, S. Mahmoudkhani, and O. Büyüköztürk, "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.

[11] W. Li, Z. Shen, and P. Li, "Crack detection of track plate based on YOLO," in *Proceedings of the 2019 12th International*

Symposium On Computational Intelligence and Design (Iscid), pp. 15–18, Hangzhou, China, 14-15 December 2019.

[12] C. Zhang, C. c. Chang, and M. Jamshidi, "Concrete bridge surface damage detection using a single-stage detector," *Computer-Aided Civil and Infrastructure Engineering*, vol. 35, no. 4, pp. 389–409, 2020.

[13] N.-D. Hoang and Q.-L. Nguyen, "A novel method for asphalt pavement crack classification based on image processing and machine learning," *Engineering with Computers*, vol. 35, no. 2, pp. 487–498, 2019.

[14] X. Meng, "Concrete crack detection algorithm based on deep residual neural networks," *Scientific Programming*, vol. 2021, Article ID 3137083, 7 pages, 2021.

[15] Y. Ding, S.-X. Zhou, H.-Q. Yuan et al., "Crack identification method of steel fiber reinforced concrete based on deep learning: a comparative study and shared crack database," *Advances in Materials Science and Engineering*, vol. 2021, Article ID 9934250, 10 pages, 2021.

[16] G. X. Hu, B. L. Hu, Z. Yang, L. Huang, and P. Li, "Pavement crack detection method based on deep learning models," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 5573590, 13 pages, 2021.

[17] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 640–651, 2015.

[18] X. Yang, H. Li, Y. Yu, X. Luo, T. Huang, and X. Yang, "Automatic pixel-level crack detection and measurement using fully convolutional network," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 12, pp. 1090–1109, 2018.

[19] G. Li, B. Ma, S. He, X. Ren, and Q. Liu, "Automatic tunnel crack detection based on U-Net and a convolutional neural network with alternately updated clique," *Sensors*, vol. 20, no. 3, pp. 717–738, 2020.

[20] G. Li, Q. Liu, S. Zhao, W. Qiao, and X. Ren, "Automatic crack recognition for concrete bridges using a fully convolutional neural network and Naïve Bayes data fusion based on a visual detection system," *Measurement Science and Technology*, vol. 31, Article ID 075403, 2020.

[21] L. Jiao, F. Zhang, F. Liu et al., "A survey of deep learning-based object detection," *IEEE Access*, vol. 7, pp. 128837–128868, 2019.

[22] R. Girshick, "Fast R-CNN," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1440–144807–13, Santiago, Chile, December 2015.

[23] W. Liu, D. Anguelov, D. Erhan et al., "SSD: single shot MultiBox detector," in *Proceedings of the 14th European Conference on Computer Vision (ECCV)*, pp. 21–3711-14, Amsterdam, The Netherlands, October 2016.

[24] M. Tan, R. Pang, and Q. Le, "EfficientDet: scalable and efficient object detection," in *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10778–10787, Seattle, WA, USA, 14-19 June 2020.

[25] 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. 779–78827-30, Las Vegas, NV, USA, June 2016.

[26] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6517–652521-26, Honolulu, HI, USA, July 2017.

[27] J. Redmon and A. Farhadi, "YOLOv3: an incremental improvement," 2018, https://arxiv.org/abs/1804.02767.

[28] A. Bochkovskiy, C. Wang, and H. Liao, "YOLOv4: optimal speed and accuracy of object detection," 2020, https://arxiv.org/abs/2004.10934.

[29] C. Wang, H. Liao, Y. Wu, P. Chen, and I. Yeh, "CSPNet: a new backbone that can enhance learning capability of CNN," in *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1571–1580, Seattle, WA, USA, 14-19 June 2020.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–77827-30, Las Vegas, NV, USA, June 2016.

[31] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5987–5995, Honolulu, HI, USA, 21-26 July 2017.

[32] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8759–876818-23, Salt Lake City, UT, USA, June 2018.

[33] T. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 936–94421-26, Honolulu, HI, USA, July 2017.

[34] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988, Venice, Italy, 22-29 Oct. 2017.

[35] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, and S. Savarese, "Generalized intersection over union: a metric and a loss for bounding box regression," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 658–66615-20, Long Beach, CA, USA, June 2019.

[36] W. Wang, M. Wang, H. Li, H. Zhao, and K. Wang, "Pavement crack image acquisition methods and crack extraction algorithms: a review," *Journal of Traffic and Transportation Engineering*, vol. 6, pp. 7–28, 2019.

[37] K. Zuiderveld, "Contrast limited adaptive histogram equalization," *Graphics Gems*, pp. 474–485, Academic Press, Boston, MA, USA, 1994.

[38] N. Kong, H. Ibrahim, and S. Hoo, "A literature review on histogram equalization and its variations for digital image enhancement. International Journal of Innovation," *Management and Technology*, vol. 4, p. 386, 2013.

[39] H. Kim, E. Ahn, S. Cho, M. Shin, and S.-H. Sim, "Comparative analysis of image binarization methods for crack identification in concrete structures," *Cement and Concrete Research*, vol. 99, pp. 53–61, 2017.

[40] L. Di Stefano and A. Bulgarelli, "A simple and efficient connected components labeling algorithm," in *Proceedings of the 10th International Conference on Image Analysis and Processing*, pp. 322–32727-29, Venice, Italy, September 1999.

[41] K. Wu, E. Otoo, and K. Suzuki, "Optimizing two-pass connected-component labeling algorithms," *Pattern Analysis & Applications*, vol. 12, no. 2, pp. 117–135, 2008.

[42] A. Paszke, S. Gross, F. Massa, A. Lerer, and S. Chintala, "PyTorch: an imperative style, high-performance deep learning library," 2019, https://arxiv.org/abs/1912.01703.

[43] Z. Xu, B. Xu, and G. Wu, "Canny edge detection based on Open CV," in *Proceedings of the 2017 13th IEEE International Conference on Electronic Measurement & Instruments (ICEMI)*, pp. 53–56, Yangzhou, China, 20-22 October 2017.
[44] B. Zoph, E. D. Cubuk, G. Ghiasi, T.-Y. Lin, J. Shlens, and Q. V. Le, "Learning data augmentation strategies for object detection," in *Proceedings of the 16th European Conference on Computer Vision (ECCV)*, pp. 566–583, Glasgow, UK, 23–28 August, 2020.
[45] R. Kohavi and F. Provost, "Glossary of terms," *Machine Learning*, vol. 30, pp. 271–274, 1998.
[46] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: a retrospective," *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, 2015.
[47] K. Oksuz, B. C. Cam, E. Akbas, and S. Kalkan, "Localization recall precision (LRP): a new performance metric for object detection," in *Proceedings of the 15th European Conference on Computer Vision (ECCV)*, pp. 521–537, Munich, Germany, 8–14 September, 2018.