Crack detection for concrete bridges with imaged based deep learning

Chunfeng Wan¹, Xiaobing Xiong¹,‡, Bo Wen², Shuai Gao¹, Da Fang¹, Caiqian Yang¹ and Songtao Xue³,⁴

¹Southeast University, Key Laboratory of concrete and prestressed concrete structure of Ministry of Education, Nanjing 210096, P. R. China
²School of Engineering Audit, Nanjing Audit University, Nanjing 211815, China
³Department of Disaster Mitigation for Structures, Tongji University, Shanghai 200092, P. R. China
⁴Department of Architecture, Tohoku Institute of Technology, Sendai, Japan

Abstract

Within the framework of intelligent bridge detection, a number of crack detection methods based on image processing techniques have been implemented. In this study, a combined novel approach with deep learning of a single shot multibox detector (SSD) and the eight neighborhood algorithm is proposed and applied to bridge crack image identification to provide an automatic method for crack detection. First, a large number of concrete crack images collected from the site were segmented and preprocessed for the establishment of a crack image dataset. Deep learning of the SSD algorithm was introduced on the training set to establish the detection model, where the model parameters were adjusted by the validation set. Sliding window technology was integrated to identify the cracks in the test set. The effects of the sliding window size and dataset size on the crack

¹Contribute equally and considered as co-first author.

Corresponding authors:
Da Fang, Southeast University, Key Laboratory of concrete and prestressed concrete structure of Ministry of Education, Nanjing 210096, P. R. China.
Email: benbenada@gmail.com

Caiqian Yang, Southeast University, Key Laboratory of concrete and prestressed concrete structure of Ministry of Education, Nanjing 210096, P. R. China.
Email: ycqjxx@seu.edu.cn

Songtao Xue, Department of Disaster Mitigation for Structures, Tongji University, Shanghai 200092, P. R. China.
Email: xue@tongji.edu.cn

Creative Commons Non Commercial CC BY-NC: This article is distributed under the terms of the Creative Commons Attribution-NonCommercial 4.0 License (https://creativecommons.org/licenses/by-nc/4.0/) which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access page (https://us.sagepub.com/en-us/nam/open-access-at-sage).
detection results were discussed. Moreover, the eight neighborhood algorithm was adopted for further crack detection correction. The results show that the configuration achieves good crack detection by the deep learning of the SSD algorithm with high precision and recall. The introduction of the eight neighborhood correction algorithm further improves the detection results by eliminating some misjudged results. Finally, the developed algorithm was placed into a portable device, with which cracks were effectively identified. The introduced method shows significantly better performance in crack detection, and the system installed on the portable device provides a way to broaden its application in the automatic crack detection of concrete bridges.

Keywords
Crack detection, deep learning, single shot multibox detector, sliding window, eight neighborhood correction algorithm

Introduction
Bridge structures experience damage accumulation and resistance attenuation under the multifactor coupling effect, and the deterioration problem after long-term service periods is increasingly prominent. Past lessons have taught us that the cause of bridge collapse is often due to the lack of scientific and timely bridge defect detection. It is critical to develop reasonable methods for bridge defect detection and regular bridge health estimation.\(^1\) As one of the most important bridge defects, cracks seriously affect the safe operation of bridges. Therefore, effective defect detection and crack identification for bridges is of great significance.

The traditional method for bridge crack detection is mainly through manual detection. Inspectors use bridge detection vehicles or bridge maintenance channels to inspect bridges at close distances. This method is time-consuming, labor-intensive, and sometimes dangerous. Automatic detection is another nondestructive method to detect cracks, which includes several technologies, such as ultrasonic surface wave testing, impact-echo, acoustic emission, optical fiber sensor network monitoring, and machine vision.\(^2\)–\(^4\) Ultrasonic surface wave testing and impact-echo are mainly used for the detection of internal defects in concrete. The acoustic emission is susceptible to noise and cannot be used to measure cracks quantitatively. Fiber optic sensor network monitoring has high measurement accuracy, a wide dynamic range, and a wide frequency bandwidth. However, these methods are complicated and expensive, and are not suitable for large bridges. Recently, crack detection by images has been welcomed for its low cost and high efficiency.

Images for crack identification can be obtained by vehicle detection systems, serpentine robots, and unmanned aerial vehicles (UAVs).\(^5\) In particular, UAVs can be used to detect parts that cannot be detected by traditional methods. Information about cracks can be acquired from the pictures obtained by cameras on UAVs. The UAVs include rotor wing UAVs, fixed wing UAVs and flapping wing UAVs. Among them, rotor wing UAVs have become a hot research topic in bridge inspections due to their good hovering photography.\(^6\) After images are acquired by drones, crack detection methods based on images mainly include the threshold segmentation method, edge detection method, region method, and matching method. Cheng proposed histogram based approaches to
identify cracks and they were verified to have good accuracy.\textsuperscript{7} Approaches based on filters such as sheet filters and Frangi filters performed well in image identification.\textsuperscript{8, 9} However, these approaches based on filters may miss some cracks when doing detection, and it would be better to consider template matching if thin cracks are important.\textsuperscript{10} Other methods such as minimal path, and Hessian-based percolation were also used to identify images.\textsuperscript{11, 12} As a new image processing method, deep learning is widely studied due to its autonomous learning.\textsuperscript{13-15}

With the profound study of deep learning, a series of neural networks focusing on crack detection have been gradually proposed. The fully convolutional network (FCN) was primarily used in crack detection.\textsuperscript{16} However, as cracks are usually part of the captured images, FCN may miss some crack images when using accurate ground truths. To solve this problem, a generative adversarial network was used.\textsuperscript{17} Due to the difficulties and challenges of crack detection tasks, many studies have focused on a series of methods for existing crack neural networks to improve the detection performance. Deep supervision was added to the neural networks to minimize segmentation errors and make the learning progress direct.\textsuperscript{18-20} Feature fusion is another widely used method that combines feature pyramids with neural networks such as hierarchical convolutional neural networks with feature preservation,\textsuperscript{19} feature pyramid and hierarchical boosting networks,\textsuperscript{20} and multimodal feature fusion networks.\textsuperscript{21} However, these methods require many parameters and significantly increase the complexity of neural networks. Researchers then focused on convolutional neural networks (CNNs) due to their rather simple structures in which fewer parameters are needed. To improve the detection accuracy of cracks on concretes, hybrid networks were derived by incorporating CNNs with long short-term memory.\textsuperscript{22} The CNN-FCN crack system applied on the texture space of a footing was proposed to detect cracks for larger structures.\textsuperscript{23} Although CNNs perform with high accuracy in controlled conditions, CNNs in industrial applications with various ranges of conditions remain challenging. To cope with this problem, a vision transformer based on CNNs was proposed and verified to have the ability to improve generalization adaptability and robustness to noisy signals.\textsuperscript{24} Instead of designing new CNNs, Han selected deep convolutional neural networks and trained them by transfer learning and a fine-tuning method.\textsuperscript{25} The seed algorithm, genetic algorithm, enhanced salp swarm algorithm and enhanced chicken swarm algorithm have also been proposed to optimize crack detection based on CNNs.\textsuperscript{26-29} In engineering, it is necessary not only to determine whether images have cracks, but also to identify other crack information. However, these methods are mainly used to classify images rather than being used for object detection. For object detection, there are one-stage and two-stage object detection. Although two-stage object detection methods such as region-CNN and faster region-CNN perform well in terms of accuracy, they are time-consuming.\textsuperscript{30-32} To address this problem, one-stage object detection methods were proposed. The single shot multibox detector (SSD) algorithm, one kind of one-stage object detection method, is based on convolutional neural networks, which use the nonmaximum suppression technique to combine highly overlapping bounding boxes into one and the hard negative mining technique to keep classes balanced during training. This method could implement the direct marking of crack images for good identification of crack strikes, which cannot be implemented in traditional convolutional neural networks.
Compared with other detection algorithms, the SSD has some advantages. With it, the object can be directly detected by using convolutional neural networks. Moreover, it uses feature maps with different scales so that objects with different scales can be detected. Last, it takes prior boxes with different scales and aspect ratios so that it can help to obtain the location of objects. These characteristics can help the SSD algorithm work well in crack identification for civil engineering structures.

In this study, a deep learning SSD algorithm was introduced and applied on the basis of the TensorFlow deep learning system to establish a detection model from a massive collection of bridge concrete crack images. Crack detection was achieved with high accuracy with a combination of the detection model and sliding window technology. With the introduction of the correction algorithm, the detection results were corrected and improved. Finally, the deep learning algorithm is implemented into a smartphone and it performs well in crack identification.

The paper is structured as follows: In Section 2, a brief introduction to SSD will be given, including the data acquisition, image processing and training parameters. Then the identification results and effects of the main factors, i.e. the sliding window size, training times and dataset size, are discussed in Section 3. Section 4 shows the implementation of the proposed algorithm for portable devices. Finally, Section 5 gives the conclusions.

**Materials and methods**

**Deep learning with the single shot multibox detector**

A single shot multibox detector (SSD) is a one-stage algorithm with fast detection speed and high accuracy in machine learning. The framework of the single shot multibox detector (SSD) algorithm, as shown in Figure 1, is based on the VGG16 framework. The convolutional feature layers to the end of the truncated base network of the SSD allow predictions of detections at multiple scales. In the SSD network, a fixed set of detection predictions is produced by the added or existing feature layer through a set of convolutional filters. A set of default bounding boxes tiling the feature map in a convolutional manner is associated with each feature map cell which is used to predict the offsets relative to the default box shapes.

![Figure 1. The framework of the SSD algorithm.](image)
When using SSD, first, sampling frames with different aspect ratios and sizes are uniformly sampled at different positions on the image, and then the features of the sampling frames are extracted through the convolutional neural network for classification. After the test pictures are detected, the category probabilities and position coordinates are directly displayed on the image. After a single test, the final test result can be obtained. This algorithm can make predictions in multiple frames at the same time.

The detection result by the SSD algorithm is shown by a box with a label and a score. For a single image, the detection effect can be judged by intersection over union (IoU). IoU is an important indicator to judge the similarity of target objects and known objects. In the training set, IoU is generated by the overlapping between the preselection box generated under the deep learning framework and the original mark box. In the test set, IoU represents the ratio of the intersection of the detection result and ground truth to their union, which can be expressed by as follows.

\[
IoU = \frac{\text{Detection Result} \cap \text{Ground Truth}}{\text{Detection Result} \cup \text{Ground Truth}} \times 100\%
\]

A comparison of IoUs with different sizes is shown in Figure 2. In the target recognition process, only when the IoU is greater than the threshold, can the target be correctly identified and marked. For this paper, the threshold is set to 0.5.

**Image acquisition and segmentation**

In this study, the crack images mainly come from two aspects, the Jing-Hang Grand Canal Extra Large Bridge and the damaged concrete beams in the laboratory. Among them, more than 300 crack images were obtained in the Jing-Hang Grand Canal Extra Large Bridge site and over 1500 crack images were collected at the laboratory. To ensure the diversity of the samples, the photos were shot at various angles and light on different concretes. Considering the effect of sample numbers on the crack detection results, the collected concrete crack images were segmented according to the sliding windows, as shown in Figure 3. Various crack patterns were selected to expand the dataset. Five thousand crack images were initially selected as the training set, and 2000 crack images were used as the validation set in the following study.

![Figure 2. The detection results with different IoUs.](image)
The collected images are in JPG format while the default image format of the SSD algorithm is the TFRecord binary format. This format can store image data and labeled data in a folder without data compression and avoiding random access, which means data can be quickly loaded into memory. The conversion of the two data formats is by Extensible Markup Language (XML), generated by LabelImg, an image annotation software. Through LabelImg, the object location can be marked in each image and the corresponding XML file is generated for each image to indicate the object location. This software not only marks multiple objects of the same type on an image, but also marks different types of objects.

**Image preprocessing**

The size of the dataset directly affects the detection results in deep learning, and collecting various samples for training can achieve good generalization. Sample data are the key to improving detection results, and data are the driving force of high-performance frameworks [23]. This study preprocesses the images by rotating, mirroring, scaling and image enhancement to amplify the dataset.

Image enhancement refers to the point operation of image pixels, and mainly includes the adjustment of brightness and contrast. The brightness adjustment is the overall increase or decrease in the pixel intensity. The contrast adjustment means that the pixel intensity in the dark portion decreases and the pixel intensity in the bright portion increases, thereby widening the brightness of a certain range in the middle.

\[
g(i, j) = \alpha f(i, j) + \beta \tag{2}
\]

where \(\alpha\) is the image contrast, and \(\beta\) is the image brightness.

**Figure 3.** Original image and segmented images: (a) original image; (b) segmented images.
Image rotation is the rotation of all the pixels of an image according to a given base point. It involves the location transformation of the image. The image size generally changes after rotation and the image beyond the display area will generally be cut off. There are also two methods of image preprocessing, image mirroring and scaling. Image mirroring contains two kinds of horizontal mirroring and vertical mirroring. For image scaling, each axis of an image is scaled at various times or the same times to obtain a new image.

Through the above various image preprocessing and dataset amplification, the number of crack image datasets is greatly increased, which offers a large number of samples for crack detection.

Parameter setting and model training

In deep learning, there are many types of hyperparameters. The weight parameters are automatically updated in the iterative process, but other types of parameters depend on manual testing, which are adjusted by the evaluation of the detection results. These parameters have different optimal values for different models. Based on the multiple analyses of the crack detection results of the validation set, the crack model was continuously adjusted. Generally the larger the batch size is, the more accurate the gradient descent direction, and the shorter the computation time. However, the memory of the training set in this paper has reached 0.89 G, and the running memory has reached 20 G during calculation. Due to the limitation of the memory of the computer, we set the batch as 4 since the value is set as the power of 2, which suffices for the GPU calculation. The initial learning rate is more suitable between 0.001 and 0.006, and the middle value 0.004 is adopted in this paper. The momentum method is designed to accelerate the gradient descent. To make the calculation converge at a suitable speed, the peak value of the update amplitude is set to 10 times as before. At this time, the momentum optimization is 0.9. To prevent overfitting, weight attenuation is adopted. To improve the accuracy of train set when training, we set the weight attention to 0.00004, which is close to 0. In addition, the ReLU activation function was used to avoid gradient disappearance during the backpropagation of the neural network.

In the deep learning of the SSD algorithm, the model training was carried out on the training set by the laboratory server for calculation. The server has 2 CPUs of an Intel Xeon CPU E5-2696 with a clock speed of 2.2 GHz. Fifty thousand training iterations were achieved. After iteration, the loss function tends to converge and stabilize with a stable value of near 4.5. The loss value could not reach the ideal value. The following analyses of the crack detection results show that the model has good detection results with high identification accuracy.

Results and discussion

Detection results

On the basis of the established crack image dataset, the deep learning of the SSD algorithm was adopted in the training set to establish the detection model, and the validation
set was used to adjust the model parameters. Finally, sliding window technology was integrated to identify the cracks in the test set.

The well-trained crack detection model was used on the test set to statistically measure the crack detection results. For the test model, there are 5000 images of the training set and 2000 images of the validation set. The number of training iterations is 50000, and the threshold value is 0.5. The size of image segmentation is $20 \times 20$ dpi with the pixel of an image of $120 \times 120$ dpi after segmentation. Four different crack forms, vertical, horizontal, slanting and crossed cracks, were identified, as illustrated in Figure 4.

![Figure 4. Four kinds of crack detection: (a) vertical crack; (b) horizontal crack; (c) slanting crack; (d) crossed crack.](image-url)
The parameters that quantitatively evaluate the detection results mainly include Precision and Recall. The calculation of the two parameters is as follows:

\[
\text{Precision} = \frac{TP}{TP + FP},
\]

\[
\text{Recall} = \frac{TP}{TP + FN},
\]

Among them, true positive (TP) represents the number of pixels in the crack region that are correctly recognized. False positive (FP) is the number of pixels that are incorrectly divided into crack regions. True negative (TN) corresponds to the number of pixels in the noncracked region that are correctly divided into non-cracks. False negative (FN) is the number of pixels in the crack area that have not been recognized. The detailed samples of the four situations are presented in Figure 5.

The detailed evaluation parameters of the detection results are listed in Table 1. It can be obtained from Figure 5 and Table 1 that good detection results exist for four kinds of cracks with an accuracy over 95% and a recall above 75%. The recall of slanting cracks is significantly lower than that of the other three cases. The main reason is that the slanting cracks form more edge cracks in the sliding window, which affects the detection results.

**Effect of sliding window size on detection results**

With the combination of the deep learning SSD algorithm and sliding window technology for crack detection, the crack range was further narrowed with the marker window. This makes the detection results more accurate than the traditional neural network method. It is known that the size of the sliding window affects the crack detection results. Here, for one crack image, different sliding window sizes, which also means different division numbers on one image (shown in Figure 6), were prepared for crack detection to investigate the effect of sliding window size on the detection results.

It is concluded from Table 2 that the identification precisions of the crack detections are all more than 80%, and the crack strikes are all well marked. In other words, the detection results show an overall good performance. What is interesting in the data is that with the increase in the division numbers, the number of FP unexpectedly increases. Taking

![Figure 5. Examples of four situations: (a) TP; (b) FN; (c) TN; (d) FP.](image-url)
the division numbers of 20 × 20 and 40 × 40 for comparison, the number of FP rises from 1 to 11 and the precision drops from 95.5% to 80%. Therefore, in the detection

Table 1. Evaluation parameters of crack detection results.

| Crack forms     | TP / (TP + FP) | Precision (%) | TP / (TP + FN) | Recall (%) |
|-----------------|----------------|---------------|----------------|------------|
| Vertical cracks | 24/25          | 96.0%         | 24/26          | 92.3%      |
| Horizontal cracks | 24/25  | 96.0%         | 24/26          | 92.3%      |
| Slanting cracks | 25/26          | 96.2%         | 25/33          | 75.8%      |
| Crossed cracks  | 43/45          | 95.6%         | 43/45          | 95.6%      |

Figure 6. Different sliding window sizes on one image: (a) division numbers of 10 × 10; (b) division numbers of 20 × 20; (c) division numbers of 30 × 30; (d) division numbers of 40 × 40.
process, considering the comprehensive analysis including the subjective visual effect and precision, the sliding window with a division number of $20 \times 20$ performs best in crack detection.

**Effect of training times on detection results**

The training of deep learning neural networks is time-consuming. When each training finishes, the weight parameters in the neural network will be updated. Such a process is also called an epoch. To explore the effect of training times on the detection results, different training times on the dataset are studied. The images with cracks are compared with images with no cracks.

It is shown in Table 3 that for these images with cracks, the greater the training times, the more cracks being detected. For these images with no cracks, images misjudged with cracks are under 10% as training times increase. In addition, when trained 10,000 times, the precision is 95.6%. When the number of training iterations is 50,000, the precision reaches a maximum of 96.9%. At 100,000 training times, it is slightly reduced to 96.1%. As training times increase, precision first increases and then decreases. The reason is that, as shown in Figure 7, when trained 10,000 times, it is underfitting since some cracks are out of detection. When trained 100,000 times, it is overfitting since part of the cracks are out of detection. When trained over 10,000 times, the precision is over 95%. The rate of recall increases with the increase in training times.

**Effect of the dataset size on the detection results**

In deep learning, the number of samples is the basis in data mining and exploring the inherent logic of data is the premise of the powerful learning ability. In addition, the deep learning model is highly dependent on the dataset size. The Google Research
Machine Perception Group pointed out that in deep learning models, the performance of visual tasks rises linearly with the increase in the amount of training data (logarithmic form).

To explore the relationship between the crack detection results and the dataset size, 100 crack images and 100 noncrack images (80 intact and 20 interference images) were selected as the test set. As illustrated in Figure 8, the interfering images may contain holes, breaks, attachments and so on. Meanwhile, various dataset sizes of the training set and validation set were prepared to discover the effect of dataset size on the detection results (Table 4).

From Table 3, it can be estimated that with the increase in the number of training sets, the crack detection results improve, showing an increase in both precision and recall. For example, when the dataset size rises from $200 + 80$ to $2000 + 800$, there is a large increase for precision from 79.6% to 96.9% and recall from 82% to 93%. When the number of training sets reaches 5000, the precision is over 95%, and the recall approaches 95%. On this basis, when the dataset size is further increased to $10000 + 4000$ by amplification, the training situation is not ideal and the loss function cannot reach a stable convergence. As a result, the detection result is not good. The reason may be that compared...
Figure 8. Test sets of crack and noncrack images: (a) noncrack images; (b) crack images.
with other objects, the cracks occupy a small area and have a single background. This feature in the dataset amplification will easily produce sample confrontation, resulting in poor detection results. The detection results produced by data amplification may be different, and the poor amplification method will produce sample confrontation, which reduces the detection results.

**Detection correction**

Good crack detection results have been obtained through deep learning of the SSD algorithm combined with sliding window technology. It is known that the cracks on the structure surface are caused by the interpenetration of microcracks. This mechanism indicates that when sliding window technology is used for small image identification, misjudgments can be eliminated and corrected based on the continuity of the crack to improve the accuracy. As a result, an eight neighborhood algorithm is introduced for the further correction of the detection results.

Taking an original crack image that has been recognized for example, the crack is tracked by the pixel value and marked by a cyan square, and the pixel value is (127, 255, 0). The specific correction process is as follows: First, the crack image with 2400 × 2400 pixels after detection is divided into $i$ regions in accordance with the size of the sliding window. Then the pixel point (127, 250, 0) is searched in the divided regions. If the point is searched in one region, there exists a crack in region $i$. An eight neighborhood algorithm is adopted to track the pixel point (127, 250, 0) in the eight neighborhoods around region $i$. If there is a tracked pixel point in the eight neighborhood, which means that there are cracks around the crack in region $i$, the detection effect is retained; if there is no pixel point, the detection effect is invalid (shown in Figure 9).

The crack detection results after eight neighborhood algorithms are demonstrated in Figure 10 and Table 5. As illustrated, through the introduction of the eight neighborhood algorithm, most of the photos that are misidentified into cracks can be eliminated to apply a correction of crack detection. The crack detection results are improved to some extent with more accuracy. This method cannot completely eliminate the misjudgment situation. As shown in Table 5, in the situation of the division number of 40 × 40, there is a continuous misjudgment situation, which cannot be eliminated in this algorithm. However, the precision with a division number of 40 × 40 increases from 80% to 95.7% after the correction algorithm. Overall, satisfactory crack detection results were achieved after a correction. The precisions of all the detection results in Table 5 rise with different ranges.

| Table 4. Crack detection results of different samples. |
|-----------------------------|-----------------|-----------------|----------------|----------------|----------------|
| Dataset                     | 200 + 80        | 500 + 200       | 1000 + 400     | 2000 + 800     | 5000 + 2000$^1$ |
| Precision                   | 79.6%           | 87.1%           | 93.8%          | 96.9%          | 96.9%          |
| Recall                      | 82%             | 88%             | 91%            | 93%            | 94%            |

$^1$5000 + 2000 indicates that the number of training sets is 5000 and the number of validation sets is 2000.
Real application in portable devices

If detection can be conducted by portable devices, it would be helpful and convenient for inspectors to discover and solve engineering problems. In this study, we tried to implement the deep learning algorithm into a smartphone, which has an Octa-core CPU and Android operating system. The Android operating system and the TensorFlow deep learning framework are both created by Google, and the compatibility of the two is much higher than that of the other operating systems and deep learning frameworks. In this paper, a crack detection model based on deep learning is first implemented on a computer, and then transferred to a smartphone through the Android platform to implement real-time crack detection.

System environment

The system environment of hardware and software, and the real-time detection application in portable devices are shown in Table 6 and Figure 11 respectively.

In view of the limitation of the computing ability of the current computer, when real-time crack detection is carried out, the whole crack will be recognized directly, and the step of segmenting and identifying the crack image and then integrating is omitted.

The detection results and effects of crack proportions

The results of cracks in real-time detection are as follows. It can be seen from Figure 12 that, the real-time detection application performs well in horizontal cracks, vertical cracks and slanting cracks.

![Figure 9. Eight neighborhood algorithm: (a) direction codes of an eight neighborhood; (b) eight neighborhood of crack.](image)

| Division numbers | 10 × 10 | 20 × 20 | 30 × 30 | 40 × 40 |
|------------------|---------|---------|---------|---------|
| Before correction | 85.7%   | 95.5%   | 88.2%   | 80%     |
| After correction  | 100%    | 100%    | 100%    | 95.7%   |
| Increasing value  | 14.3%   | 4.5%    | 11.8%   | 15.7%   |

Table 5. Comparison of precision through correction.
Figure 10. Crack detection results after correction algorithm: (a) division numbers of $10 \times 10$; (b) division numbers of $20 \times 20$; (c) division numbers of $30 \times 30$; (d) division numbers of $40 \times 40$.

Table 6. System environment of hardware and software.

| Items        | Configuration          | Server                     | Smartphone          |
|--------------|------------------------|----------------------------|---------------------|
| Hardware     | Processor              | Intel Xeon CPU E5-2696 2.20 GHz | Octa-core 1.5 GHz   |
|              | Memory                 | 26144 MB RAM               | 3 GB                |
|              | Total memory           | 143157 MB                  | 32 GB               |
|              | Type                   | –                          | Huawei Honor Series 7i |
| Software     | System                 | Win10                      | Android 6.0.1       |
|              | Program language       | Python                     | –                   |
|              | Framework              | TensorFlow, Android studio | –                   |
To verify the effect of crack proportions in real-time detection, cracks are moved from the edge to the center in the field of view. The proportion of cracks in the field of view increases from 1/8 to 1. The real-time detection results are shown in Figure 13.

Figure 13 shows that the cracks can be detected well when they are at different positions and proportions. It is also found that the larger the proportion of cracks in the field of view, the easier they can be correctly detected.

Figure 11. The real-time detection application in portable devices.

To verify the effect of crack proportions in real-time detection, cracks are moved from the edge to the center in the field of view. The proportion of cracks in the field of view increases from 1/8 to 1. The real-time detection results are shown in Figure 13.

Figure 13 shows that the cracks can be detected well when they are at different positions and proportions. It is also found that the larger the proportion of cracks in the field of view, the easier they can be correctly detected.
Deep learning using the SSD algorithm in combination with sliding window technology was introduced and adopted for bridge crack detection. Through the model training of the training set and parameter adjustment of the validation set, it has been shown that four kinds of cracks attain good detection results with a precision over 95% and a recall above 75%. This method can implement a direct mark on the crack image and the crack strikes were all well marked with the sliding window, which cannot be performed in traditional neural network algorithms.

In the investigation of the effect of sliding window size on the detection results, four different sliding window sizes with respect to four division numbers, $10 \times 10$, $20 \times 20$, $30 \times 30$ and $40 \times 40$ respectively, are taken into consideration. It is shown that the sliding

---

**Figure 12.** The real-time detection results: (a) horizontal cracks; (b) vertical cracks; (c) diagonal cracks.

**Figure 13.** The real-time detection results with different proportions: (a) 1/8; (b) 1/4; (c) 1/2; (d) 1.

**Conclusions**

Deep learning using the SSD algorithm in combination with sliding window technology was introduced and adopted for bridge crack detection. Through the model training of the training set and parameter adjustment of the validation set, it has been shown that four kinds of cracks attain good detection results with a precision over 95% and a recall above 75%. This method can implement a direct mark on the crack image and the crack strikes were all well marked with the sliding window, which cannot be performed in traditional neural network algorithms.

In the investigation of the effect of sliding window size on the detection results, four different sliding window sizes with respect to four division numbers, $10 \times 10$, $20 \times 20$, $30 \times 30$ and $40 \times 40$ respectively, are taken into consideration. It is shown that the sliding
window with a division number of $20 \times 20$ performs best in crack detection. Additionally, the effect of training times is considered. Comparing training 1000 and 10000 times, the detection results are better when trained 50000 times. In addition, it is revealed that with the increase of the number in training sets from 200 to 2000, the crack detection results improve, showing an increase in both precision and recall. However, when the number of training sets increases and reaches 5000, the precision and recall performance remain almost the same. On this basis, even when the number of training sets is further increased to 10000 or larger, the training situation is still not sufficient, and the loss function cannot reach a stable convergence. Therefore, it is essential to select the appropriate number for the training set. To eliminate the images that are misidentified into cracks, an eight neighborhood algorithm was further introduced to implement the correction of crack detection. This algorithm can largely increase the precision through correction no matter how many parts the crack image is divided into. When the division numbers are large, although some images misidentified into cracks cannot be eliminated, the crack detection results after correction are generally satisfied.

A real-time detection system on smartphones, which is based on the Android operating system and the TensorFlow deep learning framework, is developed to make detection convenient for inspectors. It shows that it performs well in crack detection when cracks are at different positions and proportions in one image. It is also found that the larger the proportion of cracks in the field of view, the easier they can be correctly detected. This system could provide a good reference for the development of a structural defect identification system installed on portable devices.

This study focuses on crack classification by the SSD algorithm, and subsequent research will focus on the length identification and width identification of cracks. In addition, the SSD algorithm is compared with other algorithms, such as YOLO, to clarify the advantages of the SSD algorithm.

**Acknowledgments**

This work is supported by the National Key Research and Development Program of China (No. 2018YFB1600200), National Natural Science Foundation of China(52078122), Key Program of Intergovernmental International Scientific and Technological Innovation Cooperation (2021YFE0112200), the Japan Society for Promotion of Science (Kakenhi No. 18K04438), the Tohoku Institute of Technology research Grant.

**Data availability**

The data used to support the findings of this study are included within the article.

**Declaration of conflicting interests**

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.
Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work is supported by the National Key Research and Development Program of China (No. 2018YFB1600200), National Natural Science Foundation of China (52078122), Key Program of Intergovernmental International Scientific and Technological Innovation Cooperation (2021YFE0112200), the Japan Society for Promotion of Science (Kakenhi No. 18K04438), the Tohoku Institute of Technology research Grant.

ORCID iDs

Xiaobing Xiong https://orcid.org/0000-0002-6429-412X
Bo Wen https://orcid.org/0000-0002-5550-9774

References

1. Moodi F and Knapton J. Research into a management system for diagnosis, maintenance, and repair of concrete structures. *J Constr Eng M Asce* 2003; 129: 555–561.
2. La HM, Gucunski N, Dana K, et al. Development of an autonomous bridge deck inspection robotic system. *J Field Robot* 2017; 34: 1489–1504.
3. Schechinger B and Vogel T. Acoustic emission for monitoring a reinforced concrete beam subject to four-point-bending. *Constr Build Mater* 2007; 21: 483–490.
4. Liu RM, Babanjad SK, Taylor T, et al. Experimental study on structural detect detection by monitoring distributed dynamic strain. *Smart Mater. Struct* 2015; 24: 1–8.
5. Potenza F, Rinaldi C, Ottaviano E, et al. A robotics and computer-aided procedure for defect evaluation in bridge inspection. *J Civ Struct Health* 2020; 10: 471–484.
6. Yang N, Zhang C and Li TH. Design of crack monitoring system for Chinese ancient wooden building based on UAV and CV. *Eng. Struct* 2021; 38: 27–39.
7. Cheng HD and Miyojim M. Novel system for automatic pavement distress detection. *J Comput Civ Eng* 1998; 12: 145–152.
8. Sato Y, et al. Tissue classification based on 3D local intensity structures for volume rendering. *IEEE T Vis Comput Gr* 2000; 6: 160–180.
9. Wirjadi O, et al. Microstructural analysis of a C/SiC ceramic based on the segmentation of X-ray phase contrast tomographic data. *Int J Mater Res* 2014; 105: 702–708.
10. Ehrig K, et al. in *International Symposium on Digital Industrial Radiology and Computed Tomography - Poster 2*. (2011).
11. Avila M, Begot S, Duculty F, et al. in *IEEE Image Proc.* 2014; 783–787.
12. Yamaguchi T and Hashimoto S. Fast crack detection method for large-size concrete surface images using percolation-based image processing. *Mach Vision Appl* 2010; 21: 797–809.
13. Bhowmick S, Nagarajaiah S and Veeraraghavan A. Vision and deep learning-based algorithms to detect and quantify cracks on concrete surfaces from UAV videos. *Sensors-Basel* 2020; 20: 1–19.
14. Liu YQ, Yeoh JK and Chua DKH. Deep learning-based enhancement of motion blurred UAV concrete crack images. *J Comput Civ Eng* 2020; 34: 1–14.
15. Mondal TG, Jahanshahi MR, Wu RT, et al. Deep learning-based multi-class damage detection for autonomous post-disaster reconnaissance. *Struct Control Hlth* 2020; 27: 1–15.
16. Liang Y, et al. in 2018 *13th IEEE Conference on Industrial Electronics and Applications (ICIEA).* 2018: 1004–1009.
17. Zhang KG, Zhang YT and Cheng HD. CrackGAN: pavement crack detection using partially accurate ground truths based on generative adversarial learning. *IEEE T Intell Transp* 2021; 22: 1306–1319.
18. Wang YY, et al. RENet: rectangular convolution pyramid and edge enhancement network for salient object detection of pavement cracks. *Measurement (Mahwah N J)* 2021; 170: 1–10.
19. Zhu QC, Dinh TH, Phung MD, et al. Hierarchical convolutional neural network with feature preservation and autotuned thresholding for crack detection. *IEEE Access* 2021; 9: 60201–60214.
20. Yang F, et al. Feature pyramid and hierarchical boosting network for pavement crack detection. *IEEE T Intell Transp* 2020; 21: 1525–1535.
21. Zhai JZ, Sun ZY, Huyan J, et al. Automatic pavement crack detection using multimodal features fusion deep neural network. *Int J Pavement Eng* 2022: 1–15. doi:10.1080/10298436.2022.2086692.
22. Andrushia D, Anand N, Neebha TM, et al. Autonomous detection of concrete damage under fire conditions. *Automat Constr* 2022; 140: 1–16.
23. Chaiyasarn K, et al. Integrated pixel-level CNN-FCN crack detection via photogrammetric 3D texture mapping of concrete structures. *Automat Constr* 2022; 140: 1–17.
24. Shamsabadi EA, et al. Vision transformer-based autonomous crack detection on asphalt and concrete surfaces. *Automat Constr* 2022; 140: 1–13.
25. Han XJ, et al. Structural damage-causing concrete cracking detection based on a deep-learning method. *Constr Build Mater* 2022; 337: 1–8.
26. Wen T, Lang H, Ding S, et al. PCDNet: seed operation based deep learning model for pavement crack detection on 3D asphalt surface. *J Transp Eng B-Pave* 2022; 148: 1–11.
27. Li JL, Liu T, Wang XF, et al. Automated asphalt pavement damage rate detection based on optimized GA-CNN. *Automat Constr* 2022; 136: 1–17.
28. Yu Y, et al. Crack detection of concrete structures using deep convolutional neural networks optimized by enhanced chicken swarm algorithm. *Structural Health Monitoring* 2022; 21: 2244–2263.
29. Yu Y, Rashidi M, Samali B, et al. Multi-Image-Feature-Based hierarchical concrete crack identification framework using optimized SVM multi-classifiers and D–S fusion algorithm for bridge structures. *Remote Sens-Basel* 2021; 13: 1–28.
30. He KM, Zhang XY, Ren SQ, et al. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE T Pattern Anal* 2015; 37: 1904–1916.
31. Ren SQ, He KM, Girshick R, et al. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE T Pattern Anal* 2017; 39: 1137–1149.
32. Liu W, et al. SSD: single shot MultiBox detector. *Computer Vision - Ecco 2016, Pt I* 2016; 9905: 21–37.
33. Smith LN. Cyclic Learning Rates for Training Neural Networks. 2017 *IEEE Winter Conference on Applications of Computer Vision (Wacv)* 2017, 2017: 464–472.