Smart Structures and Systems, *Vol. 29, No. 1 (2022) 141-151*
https://doi.org/10.12989/sss.2022.29.1.141

Smartphone-based structural crack detection using pruned fully convolutional networks and edge computing

X.W. Ye [a], Z.X. Li [b] and T. Jin*

Department of Civil Engineering, Zhejiang University, Hangzhou 310058, China

(*Received April 15, 2021, Revised July 14, 2021, Accepted August 6, 2021*)

Abstract. In recent years, the industry and research communities have focused on developing autonomous crack inspection approaches, which mainly include image acquisition and crack detection. In these approaches, mobile devices such as cameras, drones or smartphones are utilized as sensing platforms to acquire structural images, and the deep learning (DL)-based methods are being developed as important crack detection approaches. However, the process of image acquisition and collection is time-consuming, which delays the inspection. Also, the present mobile devices such as smartphones can be not only a sensing platform but also a computing platform that can be embedded with deep neural networks (DNNs) to conduct on-site crack detection. Due to the limited computing resources of mobile devices, the size of the DNNs should be reduced to improve the computational efficiency. In this study, an architecture called pruned crack recognition network (PCR-Net) was developed for the detection of structural cracks. A dataset containing 11000 images was established based on the raw images from bridge inspections. A pruning method was introduced to reduce the size of the base architecture for the optimization of the model size. Comparative studies were conducted with image processing techniques (IPTs) and other DNNs for the evaluation of the performance of the proposed PCR-Net. Furthermore, a modularly designed framework that integrated the PCR-Net was developed to realize a DL-based crack detection application for smartphones. Finally, on-site crack detection experiments were carried out to validate the performance of the developed system of smartphone-based detection of structural cracks.

Keywords: deep learning; edge computing; fully convolutional networks; structural crack detection; structural health monitoring

1. Introduction

Bridges are crucial infrastructure for providing smooth-flowing traffic for public transportation over valleys, rivers, highways and railways (Ni *et al*. 2010, 2012, Li *et al*. 2020). Most of the bridges are expected to serve a long period of up to 50 years or even 100 years (Ye *et al*. 2013). However, the field environment of bridges is hostile to the desired length of service lifetime (Jang *et al*. 2019, Ye *et al*. 2021). Thus, regular inspection is critical to the health of in-service bridges. Cracks are one of the most commonly detected kinds of damage and they take up a large proportion of the maintenance work (Ye *et al*. 2012, Spencer *et al*. 2019, Mondal and Jahanshahi 2020). During the whole lifetime of a bridge, the cracks in it might be induced by multiple negative impacts including incorrect design, irregular construction, repeated vehicle load, temperature change, acid rain, fatigue effect, etc. (Gresil *et al*. 2013, Ryu *et al*. 2020). Therefore, the inspection of cracks is extremely important for the health of a bridge. Yet, due to the multiple causes, complicated bridge structures and scattered distribution of bridge locations, the detection of cracks in bridges remains a major challenge (Alipour *et al*. 2019, Xu *et al*. 2019, Ye *et al*. 2019a).

The traditional ways of inspecting bridges rely greatly on manual work. The inspectors have to reach the inspection area, and cracks on the surface are recognized by the naked eye. These kinds of inspection methods are labor-intensive, empirical, expensive and unsafe for the inspectors. Many engineers and scholars have devoted a lot of efforts to developing autonomous crack inspection approaches. Similar to the manual visual inspection, these approaches are mostly vision-based and they involve two steps, i.e., the phase of acquiring the images and that of processing those images. In the process of image acquisition, mobile devices such as cameras, smartphones and drones are typically used as sensing platforms to acquire images. To some extent, the use of mobile devices such as drones with high-resolution cameras reduces the difficulty, such as the reduction in the demand for the inspector to reach the inspection area. Navigation techniques could help the drones or other robotic devices to move around the target autonomously. However, the acquiring and gathering of images of the structures is time-consuming, and this inevitably delays the process of inspecting for cracks. Also, the relocation of cracks will be extremely difficult if the positioning systems of the mobile devices are not accurate enough. Meanwhile, the present mobile devices such as smartphones and some drones have high-performance CPUs embedded within them which

*Corresponding author, Ph.D.,
E-mail: cetaojin@zju.edu.cn
[a] Ph.D., Professor, E-mail: cexwye@zju.edu.cn
[b] M.Sc. Student

Copyright © 2022 Techno-Press, Ltd.

ISSN: 1738-1584 (Print), 1738-1991 (Online)

could also be used as computing platforms and not only sensing platforms.

As for the phase of image processing to detect cracks, the studies could roughly be divided into three branches: IPT, machine learning (ML)-based methods, and DL-based methods (Hakim and Razak 2014, Ye *et al*. 2019b). The IPTs are not reliable when the images contain noise motifs such as spots, welding lines, markers, etc. (Abdel-Qader *et al*. 2003). The ML methods require handcrafted criterion and designs that are not robust enough when faced with complicated crack forms and noise motifs (Fujita and Hamamoto 2011). In recent years, the theory and practice of DL have obtained breakthrough achievements. They are robust and autonomous approaches for the detection of cracks (Bao *et al*. 2019, Spencer *et al*. 2019, Tang *et al*. 2019). However, the application of DL methods is usually carried out by means of expensive hardware in a fixed indoor environment such as servers or workstations (Li and Zhao 2020). The DNNs are often large in size and demand high performance hardware. Yet, to inspect cracks, on-site detection is an urgent need that would reduce the picturing, storage and transmission of crack images and promptly locate and evaluate the detected cracks. Definitely, due to the enormous demand of computation, the platform for the process of training should place the priority on performance. However, the platform for the process of testing should take mobility into consideration for practical application. Many mobile devices could be more than just sensing platforms, in fact they could also be computing platforms (Li and Zhao 2019). Besides, for the DNNs to run smoothly and fast on mobile devices which only have relatively limited computation resources, the method for the reduction and optimization of the size of the model should be developed to fit the mobile platforms.

In this study, an architecture of a deep neural network called the pruned crack recognition network (PCR-Net) was proposed for the detection of cracks. A bridge crack dataset was established for training and validation. The dataset contains 11000 elaborately labeled images with a 256×256 resolution. The PCR-Net was based on the U-net and pruned by a pruning method to reduce the overall size of the network. Then, the proposed PCR-Net was compared with the IPTs and two DNNs for the validation of the performance. Afterwards, a modularly designed framework was proposed to integrate the PCR-Net so as to develop a DL-based application for smartphones for the detection of cracks. Finally, on-site testing for the detection of cracks was conducted to evaluate the performance of this smartphone application.

2. Development of the PCR-Net

2.1 Establishment of the dataset

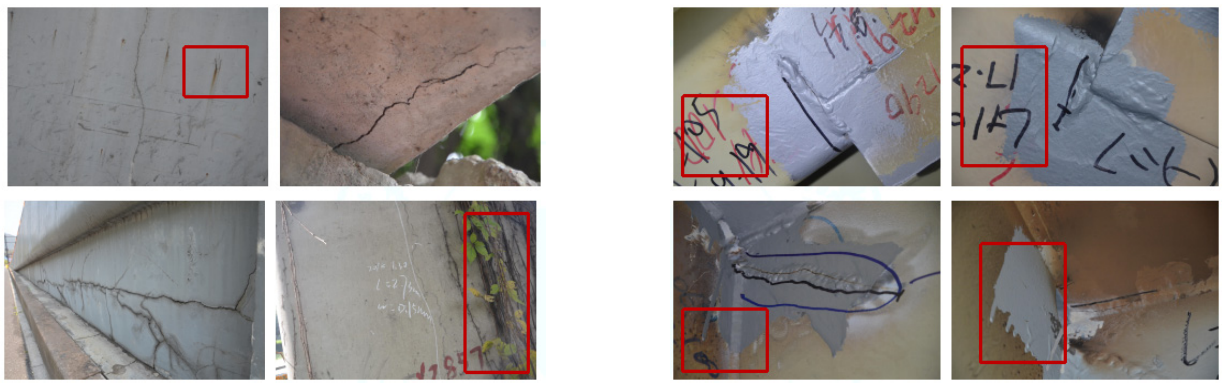
The dataset was established from 1200 images pictured uring on-site bridge inspection. Among them, 200 images were provided by the committee of the 1st International Project Competition for Structural Health Monitoring (IPC-SHM, 2020). The other 1000 images were collected from the inspection of more than 50 bridges over a period of two years. The typical original images are shown in Fig. 1. As shown in Fig. 1, the images contain plenty of commonly seen noise motifs in bridge inspection that will help to improve the robustness of DNNs.

The dataset is one of the most important factors for deep learning-based tasks. Adequate numbers and a rich diversity of training images could reduce overfitting during the training process and improve the robustness of DNNs. For bridge crack detection tasks, there are plenty of noise motifs such as spots and welding lines. Thus, it is important and absolutely necessary to collect an abundant number of images from in-service bridges for the detection of cracks with strong performance. A dataset of images of cracks in bridges was established from raw images illustrated in Fig. 1, which contains 11000 carefully labeled images with a 256 × 256 resolutions, shown in Fig. 2. Among them, 10000 images were adopted for training, 1000 images were adopted for validation and extra full-sized images were adopted for testing.

2.2 Model architecture and training strategy

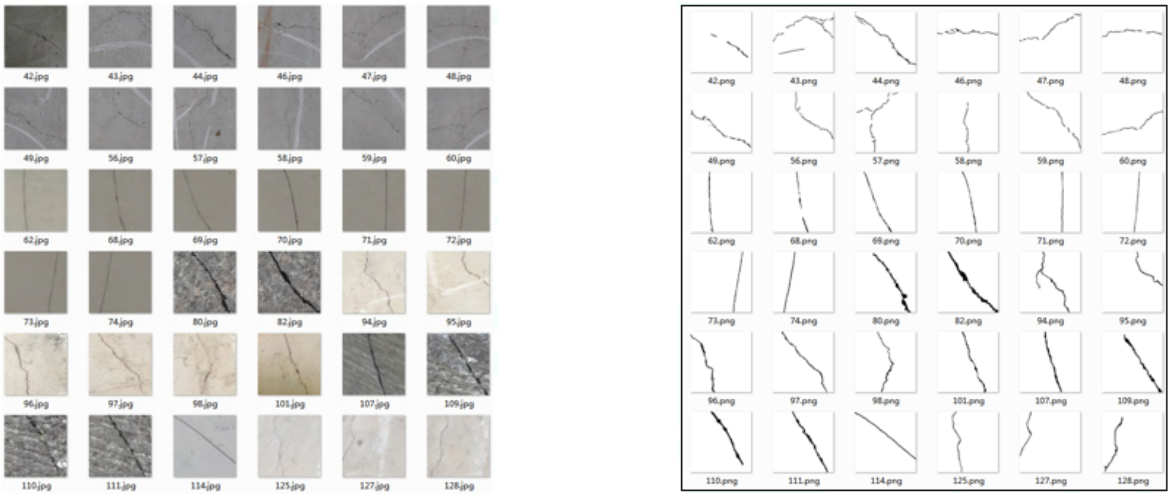
2.2.1 Model architecture

For the detection of cracks, it is important to extract the shape of cracks, which will be helpful for further evaluation of the degree of damage for target structures. Therefore, semantic segmentation is more useful than classification in



(a) Raw images from inspection (b) Raw images from IPC-SHM

Fig. 1 Raw training images with multiple noise motifs



(a) Original images (b) Image labels

Fig. 2 Training images with labels

this task. The fully convolutional network (FCN) was a powerful tool for semantic segmentation. The FCN proposed the idea of classification in a pixel-wise level that will be suitable for semantic segmentation in the task of recognizing cracks. For the detection of cracks, the FCN will process each pixel in the raw image to identify whether it belongs to a crack or not, and the detected pixels jointly form the crack area. U-net was a well-known architecture for semantic segmentation proposed by Ronneberger *et al.* (2015). It has been adopted by many research groups for all kinds of tasks regarding semantic segmentation. In this study, the idea of U-shaped architecture was adopted from a project based on U-net by Liu (2019).

One challenge in the training of DNNs is the problem of vanishing gradient that will worsen when the neural network goes deeper. In order to tackle this problem, the idea of residual networks (ResNet) was proposed. The ResNet introduced a residual block by building shortcut connections to link layers that are not directly connected. By means of these connections, the loss could be propagated steadily from the output end to the input end. In this study, the residual block was adopted to avoid problems of vanishing gradient.

Another major challenge for the training of DNNs is the existence of numerous parameters. Too many parameters require large datasets for the training process; otherwise, the network will get into an over-fitting situation. Also, too many parameters demand high performance hardware that reduces the portability of the network. For tasks regarding the recognition of cracks, it is virtually a binary classification problem, i.e., whether a pixel belongs to a crack or not. However, many DNNs have been proposed and trained to classify 1000 classifications like the VGG in the ImageNet Challenge. Thus, those DNNs might have surplus learning capacity that will raise the difficulty of training. Given this consideration, a pruning method was introduced to reduce the size of our network. Based on the U-Net, the ResNet and the pruning method, an architecture called the pruned crack recognition network (PCR-Net) was proposed. The base architecture for pruning operations in order to build the PCR-Net is shown in Fig. 3.

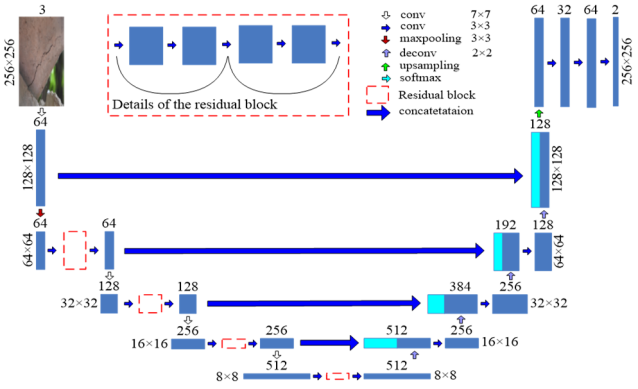


Fig. 3 The base architecture for pruning

2.2.2 Loss function

For problems of binary classification, the cross entropy loss is widely used as the loss function. The performance was verified by many research groups. It is defined by

$$H(p,q) = -\sum_{i=1}^{k} p(i)\log q(i) \tag{1}$$

where $H(p, q)$ is the cross entropy loss for the modification of weight values, $p(i)$ stands for the label of each image pixel, and $q(i)$ stands for the prediction of classification for each image pixel. The prediction of the pixel class is conducted by using the sigmoid function, which is defined by

$$\sigma(z) = \frac{1}{1+e^{-z}} \tag{2}$$

where z stands for the pixel value of the output result of the neural network. Accordingly, $\sigma(z)$ ranges from 0 to 1. It stands for the probability of a certain category.

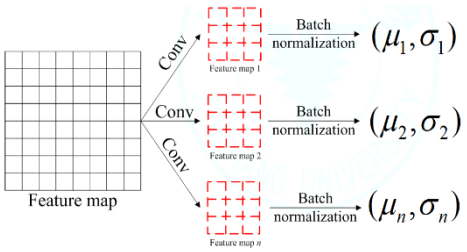


Fig. 4 Pruning process

2.2.3 Optimization algorithm

In this study, the Adam method, proposed by Kingma and Ba (2014), is adopted for the training process. The Adam algorithm can be expressed by

$$\theta_t \leftarrow \theta_{t-1} + \Delta\theta_t \tag{3}$$

where t and t–1 are the time step, θ is the training parameter for modification. $\Delta\theta$ is the modification of θ, it can be expressed by

$$\Delta\theta_t = -\varepsilon \frac{\hat{s_t}}{\sqrt{\hat{r_t}} + \delta} \tag{4}$$

$$\hat{s_t} \leftarrow \frac{s_t}{1-\rho_1} \tag{5}$$

$$\hat{r_t} \leftarrow \frac{r_t}{1-\rho_2} \tag{6}$$

where ε is the learning rate, $\hat{s}$ is the modified first order momentum, $\hat{r}$ is the modified second order momentum, δ is a constant value, ρ_1 is the decay coefficient of the first order momentum, and ρ_2 is the decay coefficient of the second order momentum. The values of s and r are calculated as

$$s_t \leftarrow \rho_1 s_{t-1} + (1-\rho_1) g_t \tag{7}$$

$$r_t \leftarrow \rho_2 r_{t-1} + (1-\rho_2) g_t^2 \tag{8}$$

$$g_t = \frac{1}{m} \nabla_\theta \sum_{i=1}^{m} L(f(x_i;\ \theta),\ y_i) \tag{9}$$

where g_t is the mean value of the gradient, m is the number of images in a batch size, y_i is the label values, $L(f(x_i;\ \theta),\ y_i)$represents the loss function, and $f(x_i;\ \theta)$ is the predicted probability.

2.2.4 Pruning strategy

Nowadays, a DNN architecture could have as many as dozens of hidden layers, which guarantees the automatic learning of features for tasks of recognition. However, due to the lack of interpretability, the proper number of layers is more of a testing problem. A fixed network architecture might contain too many layers which are not helpful for the tasks of recognition, when the training dataset is not large enough. The over-fitting problem will reduce the performance in the process of testing. In order to reduce the size of the PCR-Net, a pruning method for network slimming proposed by Liu *et al*. (2017) was applied. It was adopted to eliminate the kernels in the hidden layers that make little contribution to the network. Unlike the original way, pruning was implemented in the layer level, but not the whole network architecture. The pruning process is illustrated in Fig. 4.

2.2.5 Trainng strategy

Due to the complicated features in an image from in-service bridges, a trained DNN will still make mistakes in recognition. Like the way human beings do, learning from mistakes is a good way to improve the performance of the network. There are totally 27000 cropped images from IPC-SHM. Among them, 2000 images contain cracks and 25000 images do not have cracks. There is no point in taking every cropped image for training, since some images have little contribution to the learning of noise motifs. Also, too many images demand high performance hardware. Therefore, the network will be trained with images containing cracks and then it will be used to process all the images without cracks. The predicted results will be applied to pick out the images that are helpful to the training. Then, the images containing cracks and the chosen images without cracks are applied in order to train the network. The overall training strategy is shown in Fig. 5.

2.3 The process of training the PCR-Net

Several indices are adopted for the evaluation of the performance of DNNs. In this study, the accuracy rate,

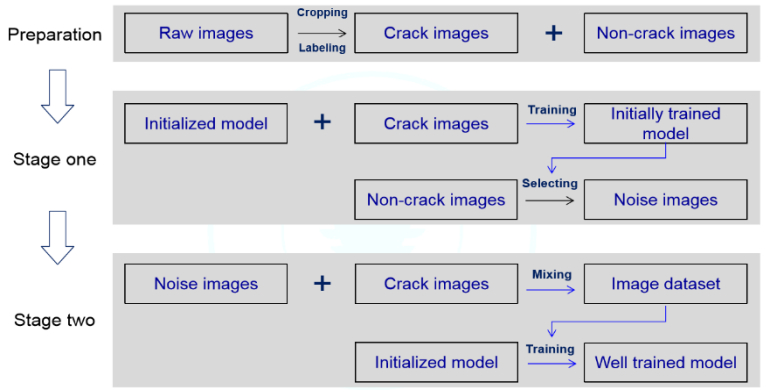


Fig. 5 Workflow of the training strategy

precision rate, recall rate, mean intersection over union (IoU) and F-measure are taken for evaluation. These indices are defined as follows.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{10}$$

$$Precision = \frac{TP}{TP + FP} \tag{11}$$

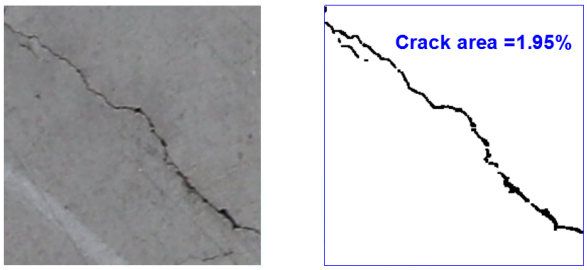
$$Recall = \frac{TP}{TP + FN} \tag{12}$$

$$F_\beta = (1 + \beta^2)\frac{Precision + Recall}{\beta^2 \times Precision + recall} \tag{13}$$

$$MIoU = \frac{1}{j}\sum_{1}^{j}\frac{TP}{TP + FN + FP} \tag{14}$$

where *TP* stands for the true positive, *TN* stands for the true negative, *FP* stands for the false positive, *FN* stands for the false negative. β is a coefficient of the trade-off between recall and precision. *j* stands for the number of categories.

Accuracy stands for the capacity to predict the right categories including the background in tasks regarding semantic segmentation. However, in this task regarding the recognition of cracks, most of the image pixels are not cracks, as shown in Fig. 6. If only the network could predict all the pixels as the background, the accuracy would be high, so this index could not represent a good performance alone. The precision rate stands for the probability of correct classification in the predicted classification of crack pixels. The recall rate stands for the probability of correct classification among all the crack pixels. F_β stands for a weighted average for reflecting the precision and recall rate, β is the weighted value that can be predefined. MeanIoU rate stands for the overlap of the predicted classification and the ground truth label of crack pixels.



(a) Training image (b) Image label

Fig. 6 Example of the proportion of crack area

Table 1 Test setup for the selection of the best model

Test number	Model	Dataset
1	Base model	7800 crack images
2	PCR-Net1	7800 crack images
3	PCR-Net2	7800 crack images
4	Base model	7800 crack images + 3200 disturbing images
5	PCR-Net1	7800 crack images + 3200 disturbing images
6	PCR-Net2	7800 crack images + 3200 disturbing images

In order to select the best model which comprehensively balances the recognition performance and recognition speed, a base model was selected and 6 tests were conducted to select the best architecture and training dataset among them, as shown in Table 1. In Table 1, the base model is the U-net shown in Fig. 2. PCR-Net1 means the architecture that was pruned one time after trained and fine-tuned later, and PCR-Net2 means the architecture was pruned twice. PCR-Net1 (initialized) means the model has the same architecture as PCR-Net1 but the parameter was initialized and the model was trained from scratch.

The training process was conducted on a server, the hardware and software information of the server and desktop is listed in Table 2.

The training was conducted with 7800 crack images first, since these pixel-wise labeled images have already contained cracks and noise motifs. The model architectures are listed in Table 1. The comparison evaluated by five indices is shown in Table 3. In Table 3, all of the models were trained by crack images with labels. The indices were based on the test for 1000 images with 256 × 256 resolutions and the process time was counted in the test for

Table 2 Details of the high performance server

Platform	Item	Unit	Version
Server	Hardware	CPU	2×Intel(R) Xeon(R) Silver4215R CPU@3.20 GHz
		GPU	NVIDIA RTX 3090/ GDDR5X 24 GB
		RAM	64 GB
	Software		Windows 10 Professional
			Pytorch 1.7.1
			Python 3.8.5
			Opencv 4.4.0
Desktop	Hardware	CPU	Intel(R) Core(TM) i7-9700 CPU@3.00 GHz
		RAM	16 GB
	Software		Windows 10 Professional
			Pytorch 1.6.1
			Python 3.7.7
			Opencv 4.3.0

Table 3 Comparison among different pruning levels based on 7800 images

Model	MeanIoU	Precision	Recall	Accuracy	F1
Basic model	0.3833	0.5173	0.5968	0.9834	0.5542
PCR-Net1	0.3750	0.4727	0.6447	0.9814	0.5455
PCR-Net2	0.4231	0.5649	0.6277	0.9852	0.5946

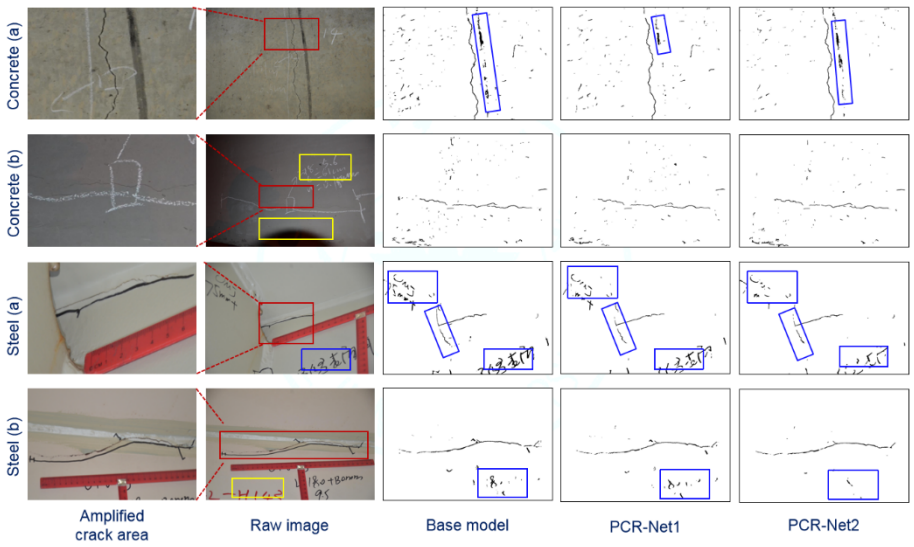


Fig. 7 Comparison of the models with crack images only

20 raw images with 5152 × 3864 resolutions on a desktop. One pruning operation will delete 30% of the convolution kernel in the left half of the base model. For a layer that has 64 channels, pruning one time means 44 channels left and pruning twice means 30 channels left. By doing this, the size of the architecture is smaller. The training time of the base model in one epoch is 79 seconds, that of the PCR-Net1 is 59 seconds and that of the PCR-Net2 is 45 seconds. The multiple indices show that the performance was not satisfied by crack images only. While the accuracy was satisfiable, the MeanIOU, precision, recall and F1 were not good. This is due to the lack of training with the noise motifs that are not close to the crack area on the raw image cracks. The performance of image test is shown in Fig. 7.

The test result shows that all of the three architectures successfully eliminate the disturbance of the black marks and rules. The black letters are considered as cracks but the red letters are not, which indicates the color of the pixel affects the results of the recognition. It is better to use color images than grayscale images while collecting crack images.

The base model has the largest size and the PCR-Net2 has the smallest size. The base model was affected mostly by the letters and the PCR-Net2 was least affected. Yet, the PCR-Net2 missed a lot of the crack pixels and the recognized cracks were not continuous. By comparison, the base model tends to be bold and the PCR-Net2 is the most conservative in predicting the existence of cracks.

As Fig. 7 shows, some of the letters, the welding lines and the tiny gap between the U-shaped rib and the diaphragm are considered to be cracks. It means the trained networks are not good at recognizing those tiny black lines. It is necessary to train the network with more images containing noise motifs. Given this consideration, disturbing images without cracks were cropped from the IPC-SHM dataset. The total number of cropped noise images is 25000. The 25000 images without cracks were processed by the trained PCR-Net1. As many as 3200 images leading to poor results of recognition were selected for training, the typical ones are shown in Fig. 8 and those

Fig. 8 Selected noise images

with little noise motifs were abandoned.

After the selection of 3200 images without cracks but containing plenty of noise motifs, the six models with different pruning levels were trained again by all of the 11000 images. As shown in Table 4, when the reduction rate increases, the evaluation indices (MeanIoU, precision, recall, accuracy and F1) are basically reduced.

By adding disturbing images without cracks, all the indices are much higher except the accuracy, which means the architectures are better in recognizing the crack area. As shown in Fig. 9, the results of the recognition were generally much better than those shown in Fig. 7. Almost all of the letters and the tiny gaps were eliminated. However, the base model still mistakenly identified some letters. This might be induced by the large size of the model that has a problem of over-fitting. By comparison, PCR-Net2 recognizes fewer parts of cracks, but mistakenly considered a welding line as a crack. Altogether, the PCR-Net1 trained by 11000 images was selected as the architecture for the recognition of cracks in this study.

2.4 Comparison with other crack recognition methods

2.4.1 Comparison with IPTs

The IPTs were adopted as the approaches for tasks of crack recognition in the past. In this study, three

Table 4 Comparison of the performance of the detection of cracks of models with different pruning levels

Model	MeanIoU	Precision	Recall	Accuracy	F1	Reduction rate	Model size (Mil.)
Basic model	0.5962	0.7993	0.7012	0.9880	0.7470	0%	14.9
PCR-Net1	0.6022	0.8061	0.7042	0.9883	0.7517	30%	9
PCR-Net2	0.5834	0.8002	0.6831	0.9877	0.7369	51%	6.1
PCR-Net3	0.5683	0.7618	0.6912	0.9882	0.7248	66%	4.7
PCR-Net4	0.5621	0.7669	0.6780	0.9881	0.7197	76%	4
PCR-Net5	0.5456	0.7537	0.6639	0.9875	0.7060	83%	3.7

pervasively-used IPTs were adopted for comparison with the PCR-Net1. They are the Canny detection method, the LOG detection method, and the Sobel edge detection method. The indices are listed in Table 6 and their performance in the detection of cracks is illustrated in Fig. 10. According to Table 5, the PCR-Net1 is much better than the IPTs in terms of the indices. As shown in Fig. 10, the IPT detection methods are also able to recognize the cracks,

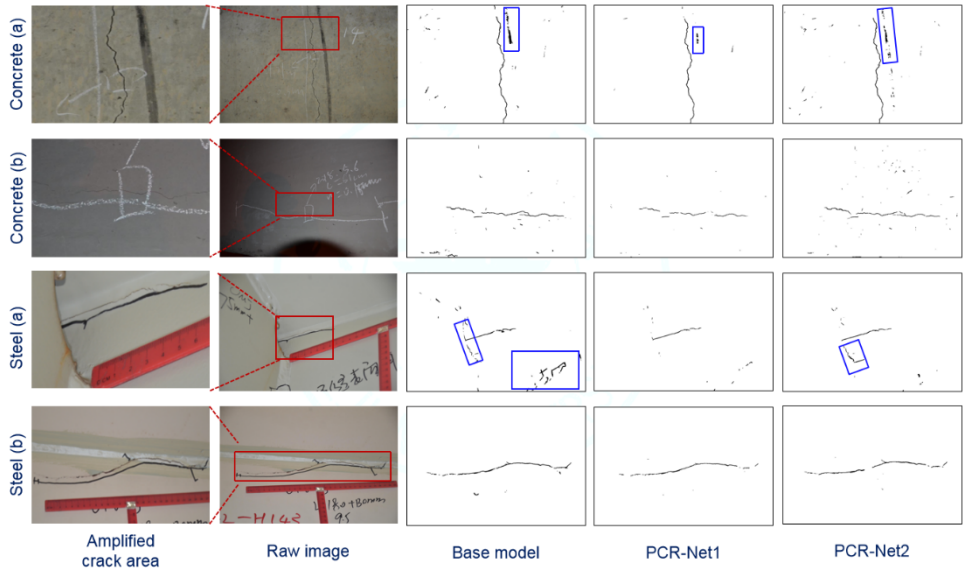


Fig. 9 Comparison of the models

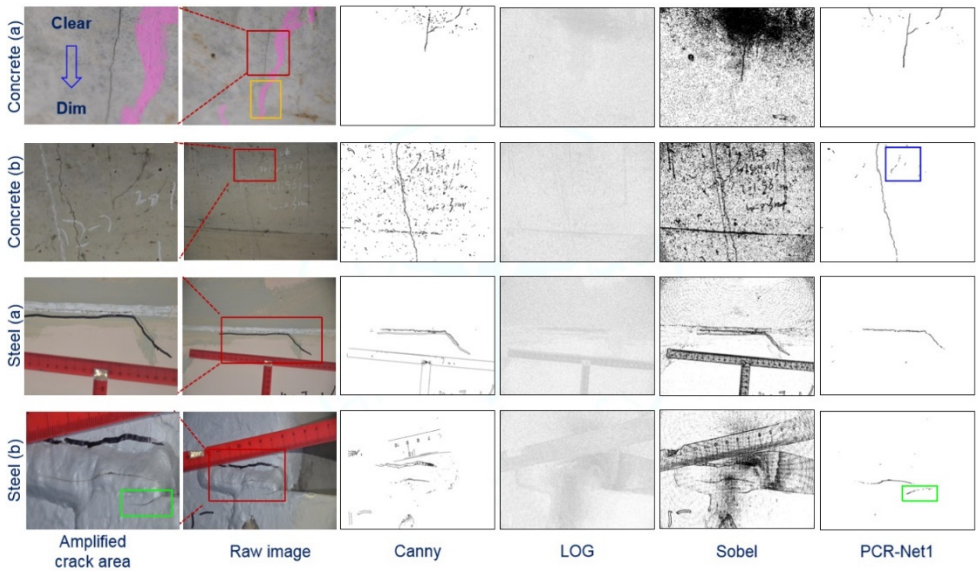


Fig. 10 Comparison between PCR-Net1 and IPTs

Table 5 Comparison of indices among PCR-Net1 and IPTs

Model	MeanIoU	Precision	Recall	Accuracy	F1
PCR-Net1	0.6022	0.8061	0.7042	0.9883	0.7517
Canny	0.0319	0.0411	0.1251	0.9348	0.0618
LOG	0.0117	0.0141	0.0654	0.9052	0.0231
Sobel	0.0546	0.0921	0.1184	0.9648	0.1036

Table 6 Comparison of indices among PCR-Net1 and other DNNs

Model	MeanIoU	Precision	Recall	Accuracy	F1
PCR-Net1	0.6022	0.8061	0.7042	0.9883	0.7517
FCN (VGG-based)	0.6095	0.8187	0.7047	0.9886	0.7574
Deeplab V3	0.5196	0.8311	0.5809	0.9865	0.6838
PCR-Net1	0.6022	0.8061	0.7042	0.9883	0.7517

but many noise motifs are also considered to be cracks. Thus, for images from in-service bridges, there are too many noise motifs for which the IPT methods are not robust enough to accurately detect cracks. Seen from the detection of the raw image of steel (b), there are actually two cracks and one of them was not marked by the inspector. This is most likely because the two cracks are close together, so one marker is enough to locate them. Also, it is possible that human inspectors miss the crack due to carelessness. Whatever the reason is, this tiny and short crack was still detected by the robust PCR-Net1. It can be observed in the image of concrete (a) that the blur level of images will affect the result of the detection. When the image goes dimmer, the deep learning-based approach and the IPTs are all affected. Interestingly, the influence of image blur has a similar impact on the Canny, Sobel and PCR-Net1, as the parts with cracks that were detected are close to each other, especially in the Sobel and PCR-Net1. For the image of concrete (b), a crack-like line, perhaps a bit of spider silk stuck by dust, is misidentified as a crack. It is a challenging task to suppress this kind of noise motifs, which requires more raw images to be collected from inspection of various kinds of aged in-service bridges.

2.4.2 Comparison with DNNs

The selected PCR-Net1 was also compared with other DNNs for the evaluation of performance. There were plenty of network architectures proposed, and in this task, the architecture of FCN was selected for the task of recognizing pixel-wise cracks. The comparison of indices between PCR-Net1 and other DNNs is listed in Table 6. Compared by means of the five indices, the performance in detection on the part of the PCR-Net1 and the VGG-based FCN is close and slightly better than the Deeplab V3.

Table 7 Comparison of computation efficiency among different models

Model	Model size (Mil.)	Efficiency (Sec/image)	
		Server	Desktop
PCR-Net1	9	2.65	27.91
FCN (VGG-based)	18.9	3.00	57.13
Deeplab V3	178	4.42	146.52

The comparison of computation efficiency on the server and the desktop is also presented in Table 7. When the detection is conducted on the server, the speed of the PCR-Net1 is 1.13 times that of the VGG-based FCN and 1.67 times that of the Deeplab V3. However, as presented in Table 8, when the detection is conducted on the desktop that is not a high-performance platform, the speed of the PCR-Net1 is 2.05 times that of the VGG-based FCN and 5.25 times that of the Deeplab V3. The results indicate the reduction in the size of the model can greatly benefit the implementation of DNN models on less high-performance platforms.

3. Field validation of the application of on-site crack detection

Cracks are sparsely distributed on the on-site bridge surfaces, and getting close to them is sometimes quite difficult. The cracks are not difficult to reach when they are not high. However, when they are located in high places like the piers of the overpass bridges, expensive equipment is necessary and the ground traffic will inevitably be disturbed. In some cases, the piers are so high that drones are needed to take crack images. The situation will be more difficult when it comes to large-scale bridges such as cable-stayed bridges and suspension bridges.

Also, even if the crack images are collected, usually, they are gathered for processing by the servers in the office rooms. There is a time gap between collection and the stage of processing that delays the work of inspecting cracks. With the development of high-performance mobile devices such as mobile phones and drones, it is now possible to deploy the DNNs on mobile devices to conduct on-site recognition of cracks. The PCR-Net1 was deployed on a mobile phone to explore the potential of mobile detection.

3.1 Development of smartphone-based crack detection system

The training process was to obtain a well-trained version of the to-be-deployed version of the PCR-Net1, shown in Fig. 11. Then, a modular design based on on-site application of crack detection was developed for detecting cracks. The framework of the system is shown in Fig. 12. The system was modularly designed which consisted of the input module, the recognition module and the storage module. The input module could call on the camera of the smartphone to acquire real-time images of the structures for real-time detection of cracks. The recognition module is the core of the application that integrates the trained PCR-Net1 model. The model could be replaced conveniently if a better model were to be trained, without changing the other two modules. The storage module could record the raw crack images, the predicted result of the detection of cracks and the location of crack detection. The operation system is

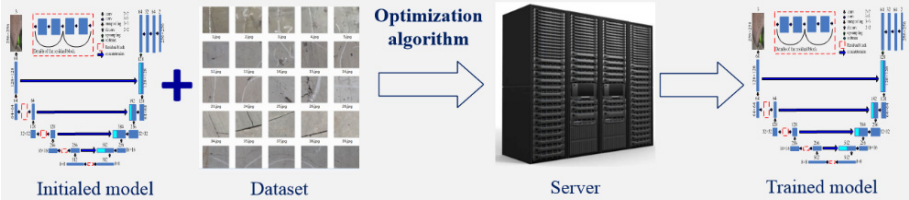


Fig. 11 Training progress of the PCR-Net1

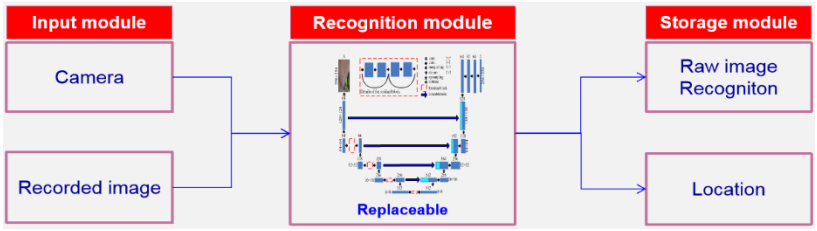
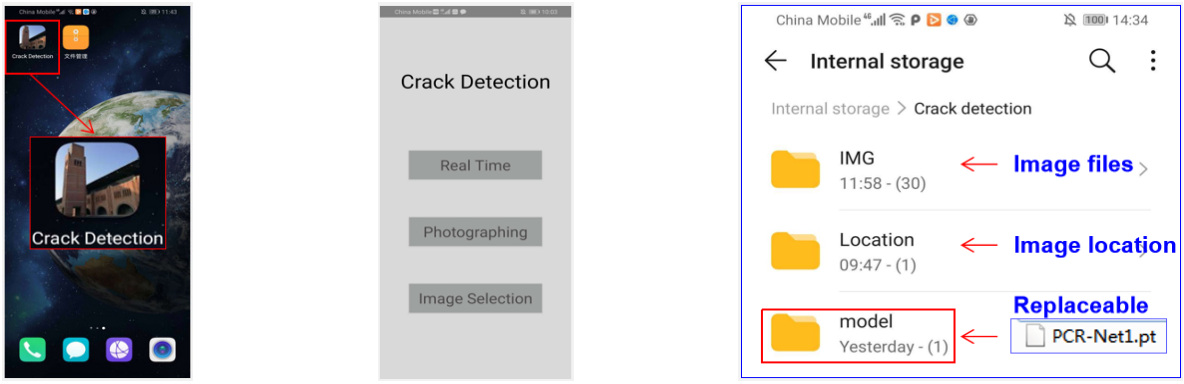


Fig. 12 Framework of the mobile crack detection system based on modular design



(a) APP (b) Interface (c) File system

Fig. 13 Application for the detection of cracks

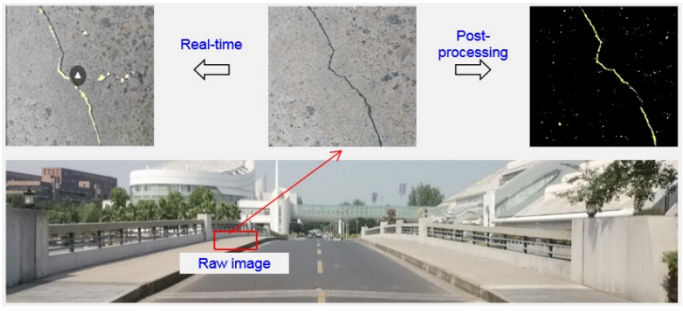


Fig. 14 The detection of cracks via mobile phone on Yangming Bridge, Hangzhou, China

Android 9.0, the CPU is Kirin 980 and the RAM is 8 G.

Benefited by the modularly designed framework, the detection of cracks could be conducted in real-time mode and post-processing mode. In the real-time mode, the application directly calls on the camera of the smartphone to obtain real-time images to be transmitted to the recognition module for crack detection. In the post-processing mode, the application could call on the camera to take pictures of the detected area and conduct crack detection on the recorded image. Also, the application could select images that are not taken by the application to detect cracks.

3.2 Field validation of the application of on-site crack detection

The application developed for the detection of cracks is illustrated in Fig. 13. It could be conveniently installed on smartphones with an Android system. On the interface of the app, three choices, including real time recognizing,

photographing, and image selection, are provided to fulfill the aforementioned functions. The file system stores the raw images, the results of the recognition, image locations and the model architecture. The images will be downsized to 768 × 768 resolutions before they are processed.

The testing of field validation was conducted on a three-span bridge called Yangming Bridge, Hangzhou, China. The concrete pavement that contains plenty of spots was inspected under a sunny environment. Both the real-time mode and the post-processing mode were tested to validate the convenience and thoroughness of the application, illustrated in Fig. 14. In the real-time model of the detection of cracks, even though the camera of the smartphone was not steady when held by a hand, the crack area was still successfully detected. In post-processing testing, the crack area was detected more accurately than the real-time detection. The time for the detection of cracks of each image is 2.35 seconds based on the field validation for 20 images.

4. Conclusions

In this study, an architecture called PCR-Net1 was developed for the task of recognizing cracks. A dataset containing 11000 images with 256 × 256 pixel resolutions was established for the training purpose. A pruning method was adopted to reduce the size of the base model. Comparative studies of the performance of detecting cracks were conducted among the PCR-Nets with different pruning levels, the IPTs and other DNN models. Further, an application for the detection of cracks was developed to accommodate mobile detection in the field. Based on this study, several conclusions can be addressed as follows:

(i) The robustness against disturbing noise motifs is significant to the performance of a DNN model. According to the results of the training process, by adding disturbing noise images to the training process, the average improvement in the MeanIoU, Precision, Recall, Accuracy and F1 for the PCR-Net of different pruning levels was 52.93%, 58.59%, 11.32%, 0.51% and 33.03% respectively. Therefore, it is essential to collect disturbing images from the real world situations to improve the performance of the detection of structural cracks;

(ii) In terms of the performance of the detection of structural cracks, the detection of edges is easily deceived by various disturbing noise motifs, while the deep-learning based approaches are more robust, accurate, objective and automatic. The comparison among the DNN models indicates that the PCR-Net1 has a similar performance in detection with the FCN, and they outperform the Deeplab V3 slightly. Notably, precision for the PCR-Net1, the FCN and the Deeplab V3 are 0.8061, 0.8187 and 0.8311 respectively, which shows that larger DNNs with more learnable parameters tend to be conservative in their prediction; and

(iii) In order to facilitate the on-site detection of cracks with mobile devices, a modularly designed framework integrating the PCR-Net1 was proposed for the development of a smartphone-based application for the detection of cracks. On-site testing of the detection of cracks was carried out in order to validate the convenience and applicability of the developed application.

Acknowledgments

The work described in this paper was jointly supported by the National Natural Science Foundation of China (Grant Nos. 52178306, 51822810 and 51778574), and the Zhejiang Provincial Natural Science Foundation of China (Grant No. LR19E080002). The authors would like to thank the organizations of the International Project Competition for SHM (IPC-SHM 2020) ANCRiSST, Harbin Institute of Technology (China), and the University of Illinois at Urbana-Champaign (USA) for their generously providing the invaluable data from actual structures. The authors also would like to thank the chairs of IPC-SHM 2020 Prof. Hui Li, and Prof. Billie F. Spencer Jr. for their leadership in the competition.

References

Abdel-Qader, I., Abudayyeh, O. and Kelly, M.E. (2003), "Analysis of edge-detection techniques for crack identification in bridges", *J. Comput. Civil Eng.*, **17**(4), 255-263. https://doi.org/10.1061/(ASCE)0887-3801(2003)17:4(255)

Alipour, M., Harris, D.K. and Miller, G.R. (2019), "Robust pixel-level crack detection using deep fully convolutional neural networks", *J. Comput. Civil Eng.*, **33**(6), 04019040. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000854

Bao, Y.Q., Tang, Z.Y. and Li, H. (2019), "Compressive-sensing data reconstruction for structural health monitoring: a machine-learning approach", *Struct. Health Monitor.*, **19**(1), 293-304. https://doi.org/10.1177/1475921719844039

Fujita, Y. and Hamamoto, Y. (2011), "A robust automatic crack detection method from noisy concrete surfaces", *Mach. Vision Appl.*, **22**(2), 245-254. https://doi.org/10.1007/s00138-009-0244-5

Gresil, M., Yu, L., Shen, Y. and Giurgiutiu, V. (2013), "Predictive model of fatigue crack detection in thick bridge steel structures with piezoelectric wafer active sensors", *Smart Struct. Syst., Int. J.*, **12**(2), 97-119. https://doi.org/10.12989/sss.2013.12.2.097

Hakim, S.J.S. and Razak, H.A. (2014), "Modal parameters based structural damage detection using artificial neural networks - a review", *Smart Struct. Syst., Int. J.*, **14**(2), 159-189. https://doi.org/10.12989/sss.2014.14.2.159

Jang, J., Shin, M., Lim, S., Park, J. and Paik, J. (2019), "Intelligent image-based railway inspection system using deep learning-based object detection and weber contrast-based image comparison", *Sensors*, **19**(21), 4738. https://doi.org/10.3390/s19214738

Kingma, D.P. and Ba, J.L. (2015), "Adam: a method for stochastic optimization", *Proceedings of the 3rd International Conference on Learning Representations*, San Diego, CA, USA. (CD-ROM)

Li, S.Y. and Zhao, X.F. (2019), "Image-based concrete crack detection using convolutional neural network and exhaustive search technique", *Adv. Civil Eng.*, 2019. https://doi.org/10.1155/2019/6520620

Li, S.Y. and Zhao, X.F. (2020), "Automatic crack detection and measurement of concrete structure using convolutional encoder-decoder network", *IEEE Access*, **8**, 134602-134618. https://doi.org/10.1109/ACCESS.2020.3011106

Li, G., Ma, B., He, S.H., Ren, X.L. and Liu, Q.W. (2020), "Automatic tunnel crack detection based on u-net and a convolutional neural network with alternately updated clique", *Sensors*, **20**(3), 717. https://doi.org/10.3390/s20030717

Liu, Q. (2019), U-Net Implementation in PyTorch. Retrieved from https://github.com/Qiuyan918/Unet_Implementation_PyTorch/blob/master/Unet_Implementation_PyTorch.ipynb

Liu, Z., Li, J., Shen, Z., Huang, G., Yan, S. and Zhang, C. (2017), "Learning efficient convolutional networks through network slimming", *Proceedings of 2017 IEEE International Conference on Computer Vision*, *Venice*, Italy. (CD-ROM) https://doi.org/10.1109/ICCV.2017.298

Mondal, T.G. and Jahanshahi, M.R. (2020), "Autonomous vision-based damage chronology for spatiotemporal condition assessment of civil infrastructure using unmanned aerial vehicle", *Smart Struct. Syst., Int. J.*, **25**(6), 733-749. https://doi.org/10.12989/sss.2020.25.6.733

Ni, Y.Q., Ye, X.W. and Ko, J.M. (2010), "Monitoring-based fatigue reliability assessment of steel bridges: analytical model and application", *J. Struct. Eng.*, **136**(12), 1563-1573. https://doi.org/10.1061/(ASCE)ST.1943-541X.0000250

Ni, Y.Q., Ye, X.W. and Ko, J.M. (2012), "Modeling of stress spectrum using long-term monitoring data and finite mixture distributions", *J. Eng. Mech.*, **138**(2), 175-183. https://doi.org/10.1061/(ASCE)EM.1943-7889.0000313

Ronneberger, O., Fischer, P. and Brox, T. (2015), "U-net: convolutional networks for biomedical image segmentation", *Proceedings of the 18th International Conference on Medical Image Computing and Computer Assisted Intervention*, Munich, Germany. (CD-ROM) https://doi.org/10.1007/978-3-319-24574-4_28

Ryu, E., Kang, J., Lee, J., Shin, Y. and Kim, H. (2020), "Automated detection of surface cracks and numerical correlation with thermal-structural behaviors of fire damaged concrete beams", *Int. J. Concrete Struct. Mater.*, **14**(1), 12. https://doi.org/10.1186/s40069-019-0387-3

Spencer Jr., B.F., Hoskere, V. and Narazaki, Y. (2019), "Advances in computer vision-based civil infrastructure inspection and monitoring", *Engineering*, **5**(2), 199-222. https://doi.org/10.1016/j.eng.2018.11.030

Tang, Z.Y., Chen, Z.C., Bao, Y.Q. and Li, H. (2019), "Convolutional neural network-based data anomaly detection method using multiple information for structural health monitoring", *Struct. Control. Health Monitor.*, **26**(1), e2296. https://doi.org/10.1002/stc.2296

Xu, Y., Wei, S.Y., Bao, Y.Q. and Li, H. (2019), "Automatic seismic damage identification of reinforced concrete columns from images by a region-based deep convolutional neural network", *Struct. Control. Health Monitor.*, **26**(3), e2313. https://doi.org/10.1002/stc.2313

Ye, X.W., Ni, Y.Q., Wong, K.Y. and Ko, J.M. (2012), "Statistical analysis of stress spectra for fatigue life assessment of steel bridges with structural health monitoring data", *Eng. Struct.*, **45**, 166-176. https://doi.org/10.1016/j.engstruct.2012.06.016

Ye, X.W., Ni, Y.Q., Wai, T.T., Wong, K.Y., Zhang, X.M. and Xu, F. (2013), "A vision-based system for dynamic displacement measurement of long-span bridges: algorithm and verification", *Smart Struct. Syst., Int. J.*, **12**(3-4), 363-379. https://doi.org/10.12989/sss.2013.12.3_4.363

Ye, X.W., Jin, T. and Yun, C.B. (2019a), "A review on deep learning-based structural health monitoring of civil infrastructures", *Smart Struct. Syst., Int. J.*, **24**(5), 567-585. https://doi.org/10.12989/sss.2019.24.5.567

Ye, X.W., Jin, T. and Chen, P.Y. (2019b), "Structural crack detection using deep learning-based fully convolutional networks", *Adv. Struct. Eng.*, **22**(16), 3412-3419. https://doi.org/10.1177/1369433219836292

Ye, X.W., Jin, T., Ang, P.P., Bian, X.C. and Chen, Y.M. (2021), "Computer vision-based monitoring of the 3-D structural deformation of an ancient structure induced by shield tunneling construction", *Struct. Control. Health Monitor.*, **28**(4), e2702. https://doi.org/10.1002/stc.2702