LETTER

A Two-Stage Crack Detection Method for Concrete Bridges Using Convolutional Neural Networks

Yundong LI[†a)], ***Member***, **Weigang ZHAO**[††b)], **Xueyan ZHANG**[†], ***and*** **Qichen ZHOU**[†], ***Nonmembers***

SUMMARY Crack detection is a vital task to maintain a bridge's health and safety condition. Traditional computer-vision based methods easily suffer from disturbance of noise and clutters for a real bridge inspection. To address this limitation, we propose a two-stage crack detection approach based on Convolutional Neural Networks (CNN) in this letter. A predictor of small receptive field is exploited in the first detection stage, while another predictor of large receptive field is used to refine the detection results in the second stage. Benefiting from data fusion of confidence maps produced by both predictors, our method can predict the probability belongs to cracked areas of each pixel accurately. Experimental results show that the proposed method is superior to an up-to-date method on real concrete surface images.

key words: *crack detection, two-stage predictors, convolutional neural networks, bridge inspection*

1. Introduction

Bridge inspection is highly important to the quantitative assessment of the deterioration and damage for concrete bridges. Traditionally, crack detection is conducted by skilled inspectors by means of scaffolding, lifting and other protective equipment. Human visual inspection is time-consuming and dangerous, especially for large-span, high-pier bridges. Therefore, computer-vision based approaches are hopeful to replace human inspection for crack detection. A summary of crack detection approaches can be found in [1], in which approaches are roughly divided into pre-processing, feature-based, model-based, pattern-based and 3D reconstruction types. From the view of feature design, crack detection approaches can be categorized into two parts: handcrafted-feature based and learned-feature based. Handcrafted-based methods use low level image features including intensity, gradient, statistical variables, and etc. Common handcrafted-based methods include edge detector [2], LoG [3], percolation [4], etc. Most of these methods have been intensively focused on crack detection under uncomplex conditions in which cracks are high contrast regions against nearly uniform background [5]. However, the concrete surface images of real bridges always contain noise and clutters, as well as image quality is affected by illumination variations. Therefore, it's a challenging task to separate cracks from background for a real concrete bridge. Fujita et al. presented a multi-scale and multi-stage approach to detect cracks from noisy surface images, in which median filter, multi-scale line filter and probabilistic relaxation were used to emphasize cracks, and remove noise and shadings [6]. Some learned-feature based approaches using shallow features extracted via SVM [5], PCA [7], K-means [8] were studied. Prasanna et al. proposed a Spatially Tuned Robust Multifeature (STRUM) classifier to detect cracks [5]. Curve fitting was used to identify points in cracked areas within a small patch. Further, STRUM classifier based on SVM was used to remove false fitting results. Compared with the shallow networks, features extracted with deep networks are more discriminative and robust to noise. Qian et al. presented a pavement crack detection algorithm based on Sparse Autoencoder (SAE) and tensor voting [9]. Schmugge et al. proposed a deep neural network based method which was used in crack detection for nuclear power plant [10]. In their scheme, images were divided into patches of 224×224 pixels. Convolutional Neural Networks (CNN) was used to classify each patch into crack or non-crack categories. However, these two methods are block-wised, and can't locate cracks at a pixel level.

This Letter presents a two-stage crack detection method using CNN, in an attempt to identify the defects in concrete surface images under complex conditions. Our idea is inspired by the following facts: (1) Concrete surface images are highly spatial correlated. Whether a pixel is belonged to cracked areas is depended on its context at a local scale. (2) Cracks have a typical topology at a global scale. Therefore, it's reasonable to combine the local context with the global information to improve detection accuracy. The proposed method consists of two stages. At the first stage, a predictor of small receptive field is used, which predicts the probability of each pixel using the local context centering the pixel. At the second detection stage, a predictor is used to refine the detection results using a large receptive field.

This research has two contributions. First, we propose a pixel-wise crack detection method using deep learning. Second, we fuse both global and local context information to improve detection accuracy. To our best knowledge, no previous work has done this before. Compared with the conventional methods using handcrafted and shallow features, the advantage of the proposed method is that features extracted with CNN are more discriminative, and the approach is more robust to noise and clutters under complex condi-

Manuscript received July 12, 2018.
Manuscript publicized September 5, 2018.

[†]The authors are with School of Electronic and Information Engineering, North China University of Technology, Beijing, 100144 China.
[††]The author is with the Institute of Structure Health Monitoring and Control, Shijiazhuang Tiedao University, Shijiazhuang, 050043 China.
a) E-mail: liyundong@ncut.edu.cn
b) E-mail: zhaoweig2002@163.com (Corresponding author)

DOI: 10.1587/transinf.2018EDL8150

tions. Compared with the block-wised deep learning methods, the proposed method can detect cracks pixel-wisely, in which the crack locations are more accurate.

2. Two-Stage Crack Detection Method

The proposed method consists of two stages. In the first stage, a small patch centering each pixel is fed into the predictor to obtain a probability whether the pixel belongs to cracked areas. The probabilities of all pixels comprise the confidence map of the input image. In the second stage, a bigger patch tiled from the confidence map produced by the first predictor instead of the raw data is fed into the second predictor to obtain a confidence map again. Subsequently, the two confidence maps are fused to get the final confidence map which can enhance cracked areas, as well as depress noise and clutters. At last, binary segmentation with a fixed threshold is exploited to locate the cracks accurately, and isolated noisy points are removed in the post-processing. The overall scheme is shown in Fig. 1.

A predictor is essentially a classifier which can divide a pixel into crack or non-crack categories using the context information centering the pixel. The context of a pixel is defined as a rectangular area centering this pixel, with the width of w and the height of h. Whether a pixel belongs to cracked areas is related to its context because bridge images are highly spatial correlated. The predictor of the first stage is constructed based on Convolutional Neural Networks in our scheme. The input is an 18×18 pixels image, which is corresponding to the size of a patch centering each pixel. Besides the input layer, convolutional layers and sub-sampling layers are alternatively stacked. Convolutional kernel size is 3×3, and max-pooling is adopted in sub-sampling layers. After convolutional layers, two full connection layers are concatenated to classify the features

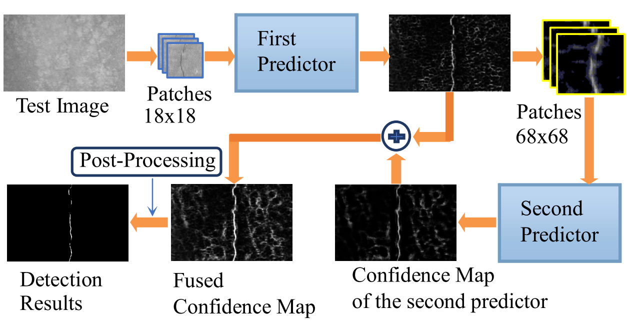

Fig. 1 Overall of the proposed method.

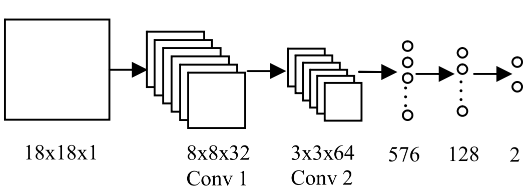

Fig. 2 Structure of the first predictor.

extracted via convolutional layers into cracks or non-cracks. It is noted that Sigmoid activation function instead of ReLU is used in the output layer. In order to accelerate the training convergence, batch normalization is conducted following each convolutional layer and full connection layer. The structure of the first predictor is shown in Fig. 2.

Another predictor of a bigger receptive field is concatenated to the first predictor to refine the detection results in the second stage. The structure of the second predictor is similar with that of the first predictor except the input image is 68×68, convolutional kernel is 5×5, and 3 convolutional layers are included. Here a large patch size and a large kernel size are used because we want to capture the spatial topology of cracks. Benefiting from a bigger receptive field and a larger kernel size, the second predictor can gather more global information which conforms to the topology of cracks. Therefore, the detection results can be refined to decrease the effect of noise and clutters by means of the global context information.

3. Experimental Results

To evaluate the performance of the proposed method, we compared it with an up-to-date method called STRUM proposed by Prasanna et al. [5]. The traditional Canny edge detection method was also included as a baseline. The testing code was implemented under the framework of Pytorch using python language. The testing computer is configured with Intel i7 processor with 3.2 GHz frequency, 64 GB memory and GPU GT730. The training parameters of CNN are presented here: learning rate is 0.001, momentum is 0.9, batch size is 100, and ReLU is used as activate function. Both the predictors were trained for 200 epochs.

Dataset included 60 images were collected from different bridges, in which 45 images were used to build training set, and 15 images were used as test images. The training set used to train the first predictor includes 56,000 positive samples and 270,000 negative samples, while for the second predictor it includes 48,000 positive samples and 240,000 negative samples.

Parts of experimental results are shown in Fig. 3. Figure 3 (a) is a test image containing noise and clutters. The spatial resolution is 475×290. Figure 3 (b) is the confidence map produced by the first predictor. Figure 3 (c) is the confidence map of the second predictor. Figure 3 (d) is the data fusion by adding Fig. 3 (b) to Fig. 3 (c). Figure 3 (e) is the final detection result obtained via binary segmentation and removing isolated noisy points. Figure 3 (f) is the curve fitting result of STRUM, Fig. 3 (g) is the final detection result of STRUM method. Detection results of Canny edge detector are also included. A 3×3 median filter is firstly applied on the test image, then the test images are detected via Canny edge detector. The high and low threshold are 0.3 and 0.12 respectively. The result is shown in Fig. 3 (h). The edge detection result of Fig. 3 (h) is further post-processed to remove isolated noisy points, and filled to obtain the crack area. The detected edges are filled because our goal is to lo-

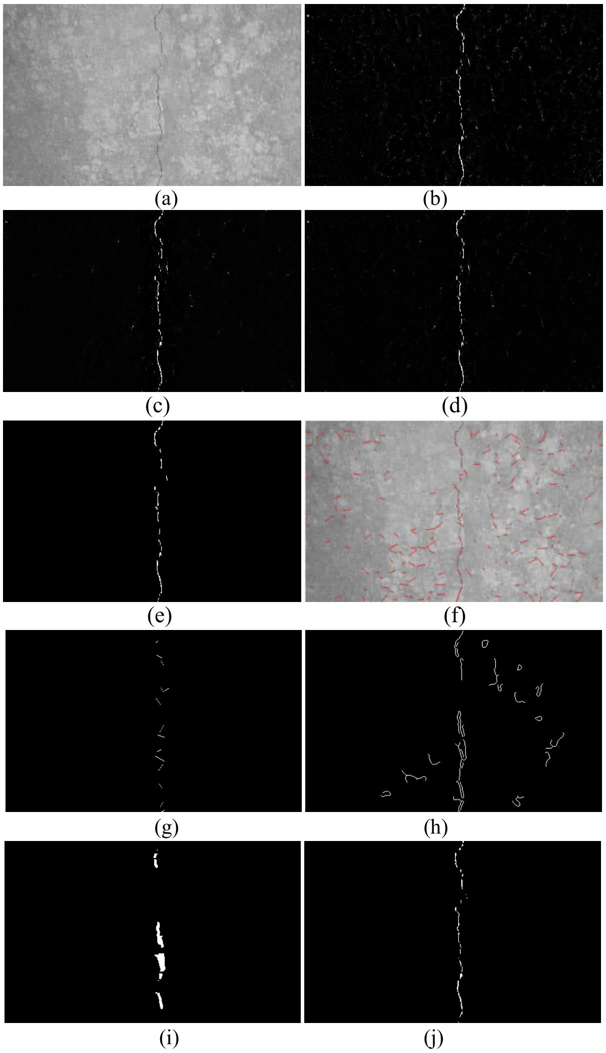

Fig. 3 Detection results of the proposed method, STRUM, and Canny methods. (a) Original test image. (b) Confidence map of the first predictor. (c) Confidence map of the second predictor. (d) Data fusion of (b) and (c). (e) Final segmentation results. (f) Curve fitting of STRUM. (g) Detection results of STRUM method. (h) Detection result of Canny edge detector with 3×3 median filter. (i) Final result of Canny edge detector after post-processing. (j) Ground truth.

cate the crack areas in the surface images. The final result of Canny is shown in Fig. 3 (i). At last the ground truth labeled manually is shown in Fig. 3 (j).

The results shown in Fig. 3 clearly illustrate that the proposed method outperforms STRUM method and the Canny edge detector.

Measurement metrics such as accuracy, precision and sensitivity [5] are further employed to quantify the detection performance. The overall performance of STRUM, Canny edge detector, the predictor of the first stage, and the proposed two-stage predictors are shown in Table 1, in which the metric values are the average of all the 15 test images. Furthermore, the curves of accuracy, precision and sensitivity of all the test images are shown in Fig. 4, Fig. 5 and Fig. 6 respectively.

Table 1 Overall performance comparison.

Methods	Accuracy (%)	Precision (%)	Sensitivity (%)
STRUM	98.81	40.34	16.45
Canny Edge Detector	96.84	34.46	69.80
The First Predictor	99.38	69.67	**79.15**
Two-stage Predictors	**99.55**	**78.49**	78.21

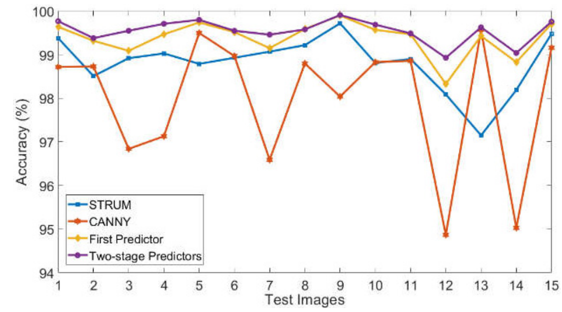

Fig. 4 Accuracy comparison of different methods.

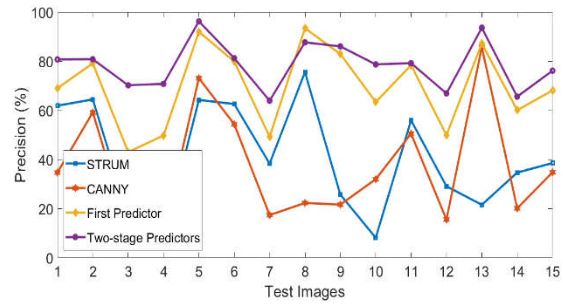

Fig. 5 Precision comparison of different methods.

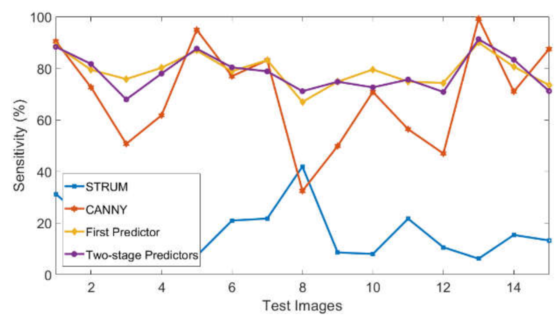

Fig. 6 Sensitivity comparison of different methods.

From Table 1 we can see that the detection performance has been significantly improved by the proposed two-stage predictors compared with STRUM method and Canny edge detector. The proposed method also performs well than the first predictor. The detection precision has been improved from 69.67% to 78.49%, meanwhile, it can get comparable sensitivity compared with the first predictor. A higher precision means that the detection results are less interfered by the noise and clutters since precision is defined as the ratio of the correctly detected pixels to the totally detected pixels.

4. Conclusions

In this letter, we propose an innovative algorithm using deep learning to deal with challenging issues in crack detection for real bridges. A two-stage prediction method consists of two predictors is built based on Convolutional Neural Networks. The confidence maps of the first stage predictor and the second stage predictor are fused to improve the crack detection accuracy for real concrete surface images. Future work will investigate the application of the proposed method to a crack automatic detection system.

Acknowledgments

This research is supported by Beijing Natural Science Foundation (4182020), and Key Laboratory for Health Monitoring and Control of Large Structures, Shijiazhuang, 050043.

References

[1] C. Koch, K. Georgieva, V. Kasireddy, B. Akinci, and P. Fieguth, "A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure," Advanced Engineering Informatics, vol.29, no.2, pp.196–210, 2015.
[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] R.S. Lim, H.M. La, and W.H. Sheng, "A robotic crack inspection and mapping system for bridge deck maintenance," IEEE Trans. Autom. Sci. Eng., vol.11, no.2, pp.367–378, 2014.
[4] T. Yamaguchi and S. Hashimoto, "Fast crack detection method for large-size concrete surface images using percolation-based image processing," Mach. Vision Appl., vol.21, no.5, pp.797–809, 2010.
[5] P. Prasanna, K.J. Dana, N. Gucunski, B.B. Basily, H.M. La, R.S. Lim, and H. Parvardeh, "Automated Crack Detection Concrete Bridges," IEEE Trans. Autom. Sci. Eng., vol.13, no.2, pp.591–599, 2016.
[6] Y. Fujita and Y. Hamamoto, "A robust automatic crack detection method from noisy concrete surfaces," Machine Vision and Applications, vol.22, no.2, pp.245–254, 2011.
[7] I. Abdel-Qader, S. Pashaie-Rad, O. Abudayyeh, and S. Yehia, "PCA-based algorithm for unsupervised bridge crack detection," Advances in Engineering Software, vol.37, no.12, pp.771–778, 2006.
[8] D. Lattanzi and G.R. Miller, "Robust automated concrete damage detection algorithms for field applications," J. Comput. Civ. Eng.. vol.28, no.2, pp.253–262, 2014.
[9] B. Qian, H.M. Tang, and W. Xu, "Pavement crack detection based on sparse autoencoder," Transactions of Beijing Institute of Technology, vol.35, no.8, pp.800–804, 2015.
[10] S.J. Schmugge, L. Rice, N.R. Nguyen, J. Lindberg, R. Grizzi, C. Joffe, and M.C. Shin, "Detection of cracks in nuclear power plant using spatial-temporal grouping of local patches," Applications of Computer Vision, IEEE, pp.1–7, 2016.