KSCE Journal of Civil Engineering (2019) 23(10):4493-4502
Copyright ⓒ2019 Korean Society of Civil Engineers
DOI 10.1007/s12205-019-0437-z

Structural Engineering
pISSN 1226-7988, eISSN 1976-3808
www.springer.com/12205

Structural Damage Detection using Deep Convolutional Neural Network and Transfer Learning

Chuncheng Feng*, Hua Zhang, Shuang Wang***, Yonglong Li****, Haoran Wang*****, and Fei Yan********

Received March 26, 2019/Revised June 27, 2019/Accepted August 4, 2019/Published Online September 3, 2019

Abstract

During the long-term operation of hydro-junction infrastructure, water flow erosion causes concrete surfaces to crack, resulting in seepage, spalling, and rebar exposure. To ensure infrastructure safety, detecting such damage is critical. We propose a highly accurate damage detection method using a deep convolutional neural network with transfer learning. First, we collected images from hydro-junction infrastructure using a high-definition camera. Second, we preprocessed the images using an image expansion method. Finally, we modified the structure of Inception-v3 and trained the network using transfer learning to detect damage. The experiments show that the accuracy of the proposed damage detection method is 96.8%, considerably higher than the accuracy of a support vector machine. The results demonstrate that our damage detection method achieves better damage detection performance.

Keywords: *hydro-junction infrastructure, damage detection, deep convolutional neural network, transfer learning, structural health monitoring, concrete surface defect*

1. Introduction

The safety and stability of hydro-junction infrastructure is important to industry and civilian security: infrastructure with serious problems can cause significant economic losses. Therefore, damage detection is critical. In recent years, researchers have focused on detecting cracks in roads, bridges, pipelines and traffic tunnels, and some traditional machine learning methods such as edge detection and a random forest model have been developed for crack detection (Nishikawa *et al*., 2012; Ng, 2014; Shi *et al*., 2016; Gui *et al*., 2017). A multiple feature classifier and a machine learning classifier were proposed for crack recognition in bridges (Prasanna *et al*., 2016). A machine learning approach is particularly suitable for classifying cracks and noncrack noise patterns that are difficult to distinguish using existing image processing algorithms (Kim *et al*., 2018). However, manually extracting the feature required by these methods is tedious. With the development of the convolutional neural network (CNN) for object classification, some researchers have applied various CNN models to identify cracks (Cha *et al.*, 2017). A fully automated tunnel assessment approach was proposed using a deep convolutional neural network (DCNN) and multi-layer perceptron (Makantasis *et al.*, 2015). A fully convolutional network model was proposed to recognize tunnel lining defects (Xue and Li, 2018). A unified crack and sealed crack detection approach was proposed that can detect and separate both cracks and sealed cracks under the same framework using transfer learning (Zhang *et al.*, 2018). A Faster

*Ph.D. Student, School of Information Engineering, Southwest University of Science and Technology, Mianyang, Sichuan 621000, China (Corresponding Author, E-mail: beidou_stars@163.com)
**Professor, School of Information Engineering, Southwest University of Science and Technology, Mianyang, Sichuan 621000, China; Intelligent Hydropower Research Institute, Sichuan Energy Internet Research Center of Tsinghua University, Chengdu, Sichuan 610000, China (E-mail: zzhh839@163.com)
***Ph.D. Student, School of Information Engineering, Southwest University of Science and Technology, Mianyang, Sichuan 621000, China; Intelligent Hydropower Research Institute, Sichuan Energy Internet Research Center of Tsinghua University, Chengdu, Sichuan 610000, China (E-mail: wangshuang@tsinghua-eiri.org)
****Ph.D. Student, School of Information Engineering, Southwest University of Science and Technology, Mianyang, Sichuan 621000, China; Intelligent Hydropower Research Institute, Sichuan Energy Internet Research Center of Tsinghua University, Chengdu, Sichuan 610000, China (E-mail: liyonglong@hotmail.com)
*****Post-doctoral Fellow, Dept. of Hydraulic Engineering, Tsinghua University, Beijing 100084, China; Intelligent Hydropower Research Institute, Sichuan Energy Internet Research Center of Tsinghua University, Chengdu, Sichuan 610000, China (E-mail: thuwhr@163.com)
******Ph.D. Student, School of Information Engineering, Southwest University of Science and Technology, Mianyang, Sichuan 621000, China (E-mail: yanfei7@163.com)

Region-based CNN structural visual inspection method was proposed to detect multiple damage types (Cha *et al*., 2018). A deep convolutional neural network was trained on the 'big data' ImageNet database, which contains millions of images, and the resulting trained model was transferred to automatically detect cracks in hot-mix asphalt and Portland cement concrete-surfaced pavement images, which also include a variety of non-crack anomalies and defects (Gopalakrishnan *et al.*, 2017). The best prediction accuracy was 92.08%, achieved by using the metaheuristic optimized edge detection algorithm and convolutional neural network for asphalt pavement crack detection (Nhat-Duc *et al.*, 2018).The active learning approach achieved a detection accuracy of 87.5% in civil infrastructure defect classification (Feng *et al*., 2017). An autoencoder framework was also proposed for structural damage identification (Pathirage *et al*., 2018). The methods mentioned above have improved the recognition accuracy of concrete damage by adopting convolutional neural networks, but the detection accuracy can still be further improved. A faster region-based concrete spalling damage detection method was proposed with an inexpensive depth sensor to quantify multiple instances of spalling (Beckman *et al*., 2019). Unmanned aerial vehicle (UAV) technology has been used to collect images for detect detection (Phung *et al.*, 2017; Li *et al*., 2017; Wang and Zhang, 2017). A crack identification strategy was proposed that combines hybrid image processing with UAV technology (Kim *et al.*, 2017). A crack detection method was proposed using a deep convolutional neural network by processing the video data collected from an autonomous UAV (Kang and Cha, 2018). There are also some new ideas for the non-destructive testing of composites using a solitary wave (Singhal *et al.*, 2017). A nondestructive evaluation (NDE) method was proposed to detect concrete surfaces using highly non-linear solitary waves (Nasrollahi *et al.*, 2017).

In this paper, we apply transfer learning and the Inception-v3 deep learning model to the damage detection task for hydro-junction infrastructure. To achieve multiple types of damage detection of hydro-junction infrastructure, we slightly altered the fully connected output layer of the network. Entire networks need to be retained to detect the damage. The advantage of deep convolutional neural networks is that they can automatically extract rich defect image features. When only a dataset with a small number of samples is available, transfer learning offers huge advantages in terms of time-consuming and accuracy. In our method, we first collect an image dataset of hydro-junction infrastructure; then, we perform image preprocessing. Next, we modify the Inception-v3 network model structure and input our collected images into the modified network to adjust its parameters using transfer learning. Our major contributions in this paper are the following:

1. We apply deep learning to classify multiple types of structural damage of hydro-junction infrastructure such as hydropower stations.
2. We combine the Inception v3 network with transfer learning to train the structural damage classifier, supporting an efficient fewer-sample detection strategy.

2. Methods

2.1 Data Collection and Preprocessing

With the explosive development of convolution neural network, some datasets are indispensable, such as ImageNet, COCO and PASCAL VOC. However, no dataset exists that contains high-quality images of hydro-junction infrastructure. Therefore, we used a UAV equipped with an HD camera, a real-time kinematic (RTK) global positioning system. The HD camera is used to capture the image. RTK is differential GPS-DGPS, which is different from traditional GPS and is used to provide stable GPS information for a UAV. As shown in Fig. 1, the UAV collects data along a fixed route and maintains a distance of ten meters from the hydropower station. Finally, we collected high-resolution images at a hydropower station in Sichuan Province, China. Fig. 1 shows the hydropower station.

After the dataset collection was complete, we obtained a total of 435 available images with a resolution of 7,952 × 5,304. To augment these images, we split each raw image into patches. Where patch consisted of 300 × 300 pixels. Finally, we obtained 18,605 300 × 300 patches. Assigning a data label is extremely important for network training. To accomplish this task, we invited experienced experts to label the data. The patches were labelled as one of five types: intact, crack, seepage, rebar exposure and spalling. Furthermore, each label type was been divided into three subsets at proportions of 8:1:1 and used for training, validation, and testing, respectively. Table 1 lists the distribution of the training, validation and testing datasets. The size of each image is 300 × 300. Fig. 2 shows the image

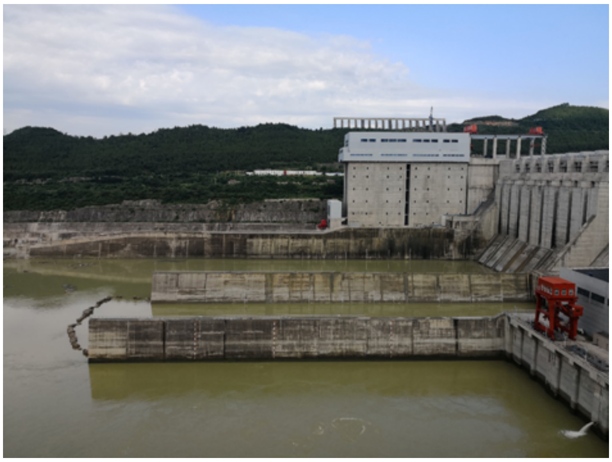

Fig. 1. A Hydropower Station

Table 1. Preparation of the Dataset

Five label types	Three Subdatasets		
	Training	Validation	Test
Crack	2,957	389	396
Intact	2,948	407	350
Spalling	2,990	365	361
Seepage	2,966	389	380
Rebar exposed	3,012	353	342

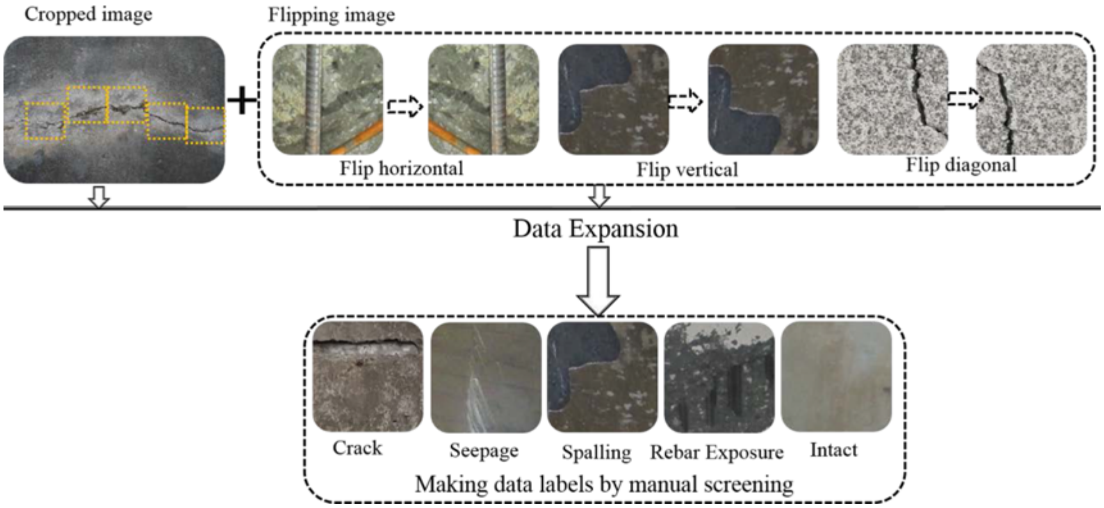

Fig. 2. Data Expansion and Making Data Labels (intact represents nondestructive type)

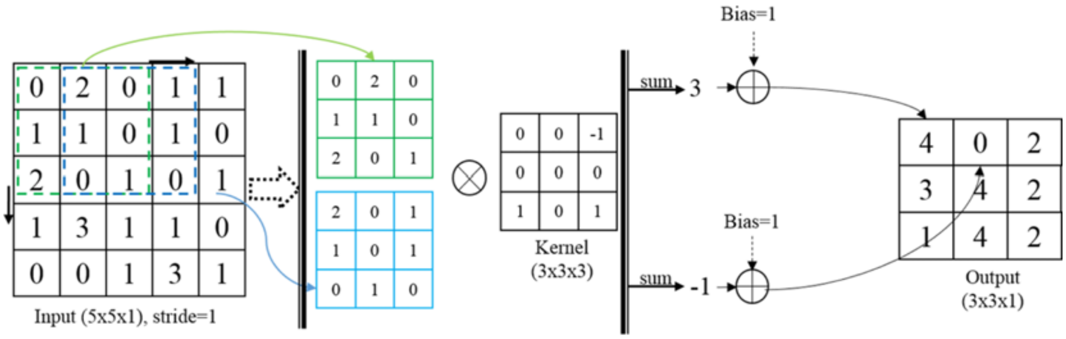

Fig. 3. The Convolutional Operator

preprocessing flow and examples of the five label types.

2.2 Convolutional Neural Network

In the 2012 ILSVRC competition, a deep convolutional neural network (DCNN) was applied to the image classification task for the first time. It improved the recognition rate by 10%, greatly surpassing the second place model and demonstrating a comprehensive advantage. A DCNN includes a feature extractor that functions automatically during the training process rather than requiring manually designed features. The DCNN feature extractor is composed of special types of neural networks whose weights are determined during the training phase (Krizhevsky *et al.*, 2012).

2.2.1 Convolution Layer

Convolution is an effective feature extraction method. Usually, a square convolutional kernel with weights is used to traverse each pixel of the input image. Each input pixel and the corresponding area of the convolution kernel are multiplied by the corresponding weights of the convolution kernel. The result of this product is summed again, adding the bias of the convolution kernel. Finally, this process results in a pixel in the output image.

$$y_i = \sum_i x_i * w_{ij} + b_i \tag{1}$$

Where x_i represents the input of each channel, w_{ij} represents the weight of each convolutional kernel, b_i represents the bias of the convolutional kernel, and y_i represents the final output.

Figure 3 shows a simple example of a convolution operation that adopts a non-zero padding mode. The size of the input array is 5 × 5 × 1. The convolutional kernel size is 3 × 3 × 1, the bias is equal to one, the stride is equal to one, and the size of the output array is 3 × 3 × 1. Eq. (2) shows the calculation of the output size:

$$output_{length} = \begin{cases} \dfrac{(input_{length} - kernel_{length} + 1)}{stride_{length}} & valid \\ \dfrac{input_{length}}{stride_{length}} & same \end{cases} \tag{2}$$

Where the *input*$_{length}$ represents the size of the input image, *kernel*$_{length}$ represents the size of the convolutional kernel, *stride*$_{length}$ indicates the stride of the convolution kernel, *valid* denotes non-zero padding and the *same* denotes zero padding.

2.2.2 Activation Layer

In general, a nonlinear activation function appears after the convolutional layer. The most common and effective activation function is the rectified linear unit (ReLU) function (Nair and Hinton, 2010). Intuitively, the gradients of the ReLU function are always zeros and ones. This approach not only solves the disappearing gradient problem that can occur during training but

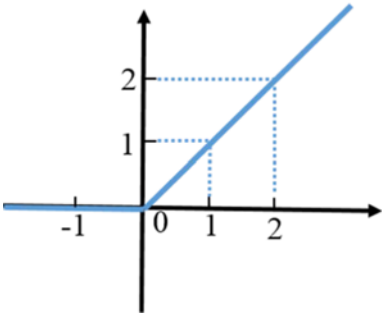

Fig. 4. The RELU Activation Function

also executes much faster and achieves better accuracy than does a model using a sigmoidal function. Fig. 4 shows an example of the ReLU function.

2.2.3 Pooling Layer

Pooling is a downsampling operation that takes a specific value as the output value from s pooling kernel region. There are two main types of pooling operations. The max pooling operator selects the maximum value from an image array's subarrays, whereas the average pooling operator selects the mean value. The advantage of the pooling layer is that it can reduce the amount of calculation and it characterizes translation invariance. Fig. 5 shows a detailed description of the pooling operator.

2.2.4 Softmax Layer

To obtain the recognition result, a softmax layer is necessary to predict the classes of input the images. The output of a softmax layer represents the probability of object recognition. Eq. (3) represents the common softmax function, which normalizes the result to between 0 and 1.

$$p(y^{(i)} = n \mid x^{(i)}; w) = \frac{1}{\sum_{j=1}^{n} e^{w_j^T x(i)}} \begin{bmatrix} e^{W_1^T x^{(i)}} \\ e^{W_2^T x^{(i)}} \\ \vdots \\ e^{W_n^T x^{(i)}} \end{bmatrix} \tag{3}$$

Where x represents the input data, n is the number of categories, m is the number of train examples, w represents a weight parameter, and $i = 1 \ldots m$, $w_n^T x^{(i)}$ are the input of the softmax layer.

2.3 Deep Convolutional Neural Network with Transfer Learning

2.3.1 The Structure of Inception-v3 Network

In the 2014 ILSVRC competition, Google presented a network in the image classification competition called GoogLeNet that achieved a recognition level equivalent to that of human beings on the ImageNet dataset. GoogLeNet innovatively included some inception modules in the network design. Inception-v3 is an improved version of GoogLeNet; its detailed structure is shown in Table 2. The Inception-v3 differs from networks such as LeNet (Lecun *et al.*, 1998) and VGG (Simonyan and Zisserman, 2014) by including a key component called an inception module. This module uses multiple sizes of receptive kernels. The output size of the convolution operation is held consistent by using zero padding. The final feature maps are obtained though filter concatenation. The inception operation

Table 2. Detailed Specification of Inception-v3 Network

Type	Kernel size/stride	Input size
Convolution	3 × 3/2	299 × 299 × 3
Convolution	3 × 3/1	149 × 149 × 32
Convolution	3 × 3/1	147 × 147 × 32
Pooling	3 × 3/2	147 × 147 × 64
Convolution	3 × 3/1	73 × 73 × 64
Convolution	3 × 3/2	71 × 71 × 80
Convolution	3 × 3/1	35 × 35 × 192
Inception module	Three modules	35 × 35 × 288
Inception module	Five modules	17 × 17 × 768
Inception module	Two modules	8 × 8 × 1,280
Pooling	8×8	8 × 8 × 2,048
Linear	Logits	1 × 1 × 2,048
Softmax	Output	1 × 1 × 1,000

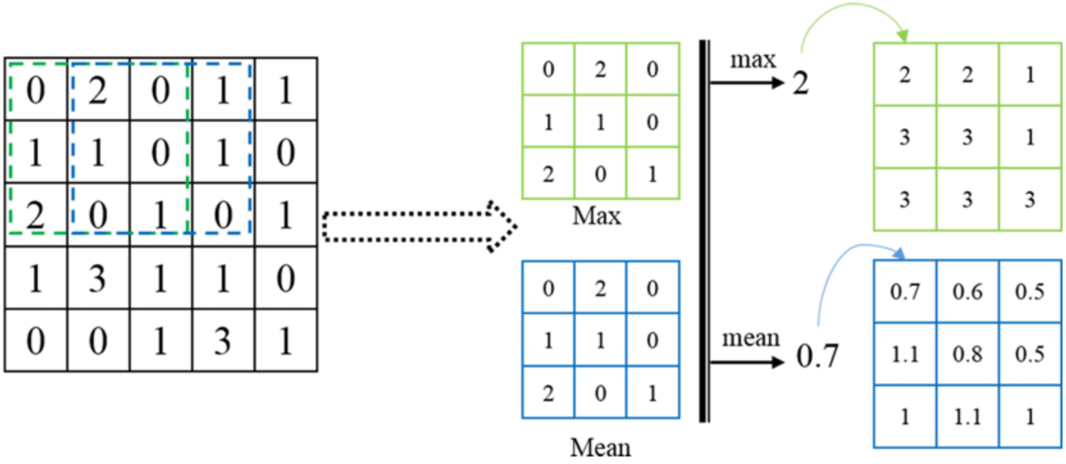

Fig. 5. The Pooling Operator

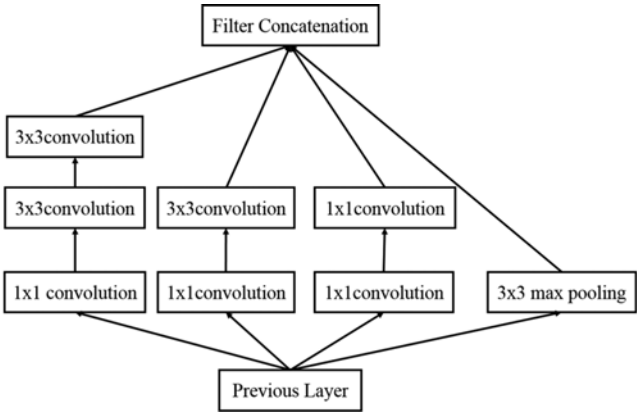

Fig. 6. Sample of the Inception Module

helps in extracting richer features from the input image. Fig. 6 shows the structure of an inception module (Szegedy *et al*., 2015).

2.3.2 Transfer Learning

Transfer learning adapts a model trained on one classification scenario to a new classification scenario through simple structural

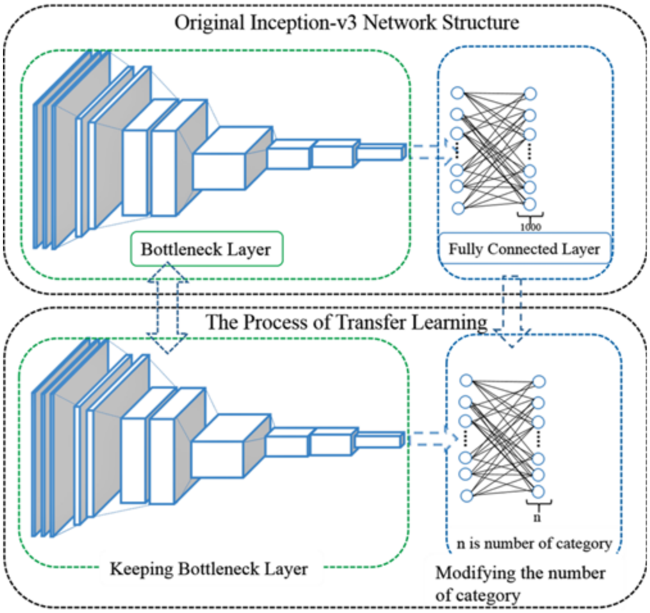
Original Inception-v3 Network Structure
1000
Bottleneck Layer
Fully Connected Layer
The Process of Transfer Learning
n
n is number of category
Keeping Bottleneck Layer
Modifying the number of category

Fig. 7. The Principle of Transfer Learning

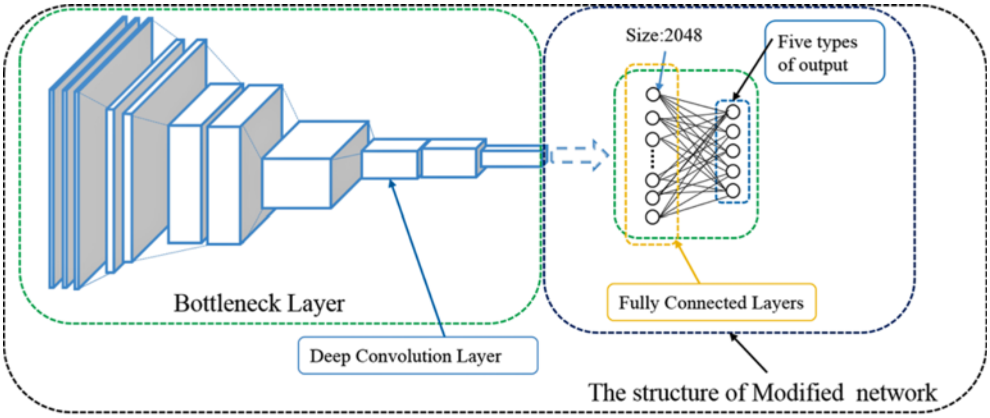

Fig. 8. The Architecture of Proposed Damage Detection Method

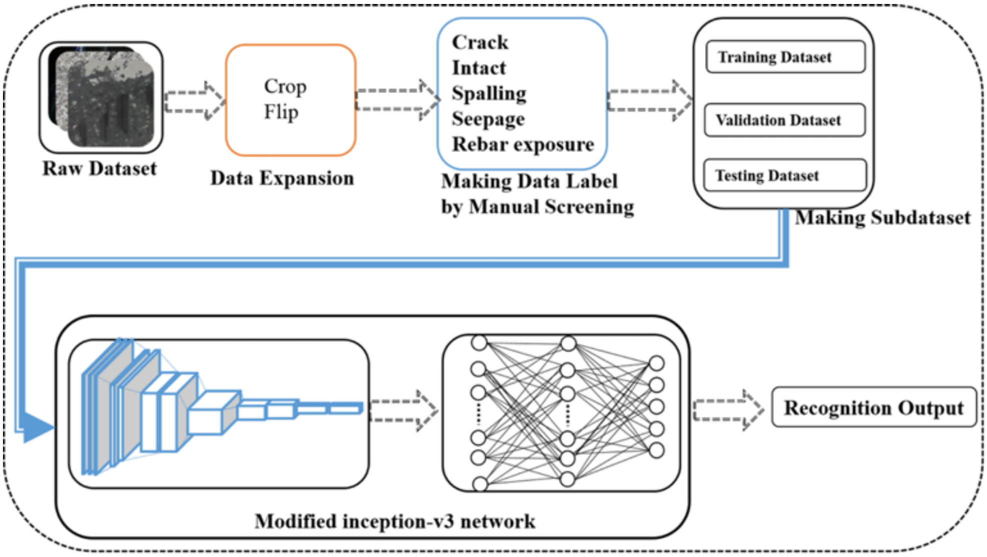

Fig. 9. Flowchart of Damage Detection for the Proposed Damage Detection Method

adjustments (Gopalakrishnan *et al.*, 2017). All the layers before the final full connection are called the bottleneck layer. The bottom layer has the advantage of performing fast feature extraction for other datasets (Donahue *et al*., 2013). As we know, the Inception-v3 network has reached 96.5% accuracy on ImageNet dataset, which includes 1.2 million annotation images. Consequently, we can assume that the network has powerful feature extraction capability for any image. However, to achieve specific object recognition tasks, we want to retain the bottleneck layer of the pre-trained Inception-v3 model and modify the network structure of the fully connected layer. The advantage of transfer learning is that it can improve the detection accuracy even when the available training dataset is small. Fig. 7 depicts the principle of transfer learning.

2.3.3 Modified Inception-v3 Network with Transfer Learning

To improve the accuracy of damage detection, we slightly altered the structure of Inception-v3 by modifying the number of fully connected neurons in the final layer to five—because we have five label types. Fig. 8 shows the architectural modifications made to Inception-v3.

A detailed flow diagram of the damage detection method is shown in the Fig. 9, including data expansion, assigning data labels, dataset generation, feature extraction and damage detection.

3. Experiments and Results

3.1 Software Environment of Experiment Setup

The experiment was performed on a computer with an Inter core i7-8750H, 16GB of random memory and an NVIDIA GTX 1060 GPU with 6GB of memory. We used the Tensorflow deep learning framework to train the network parameters of the proposed damage detection method. The GPU-accelerated software environment is as follows: CUDA-9.0, CUDNN-7.5, Tensorflow-1.9 and Python-2.7.5 from the Anacoda-2 distribution to create an independent Python environment.

3.2 Damage Detection

Before conducting network parameter training, to improve the training speed and avoid repeated feature extraction, we extracted feature maps of each image and saved them. In this way, we need only train the parameters of the fully connected layer. During training, we set the number of training epochs to 1,000, the learning rate to 0.001 and the size of batch to 100. Because the data distribution proportion was 8:1:1, we used 14873 image for training and 1903 image for validation. After the training, we used the retaining 1829 image patches to test the accuracy of the proposed damage detection method. The best test accuracy achieved was 96.8%. A loss function represents the difference between a predicted value and the correct value. When training the of network parameters, the loss function is continuously minimized by repeatedly updating all the parameters in the neural network. Finally, we obtained a highly accurate network model to perform damage detection.

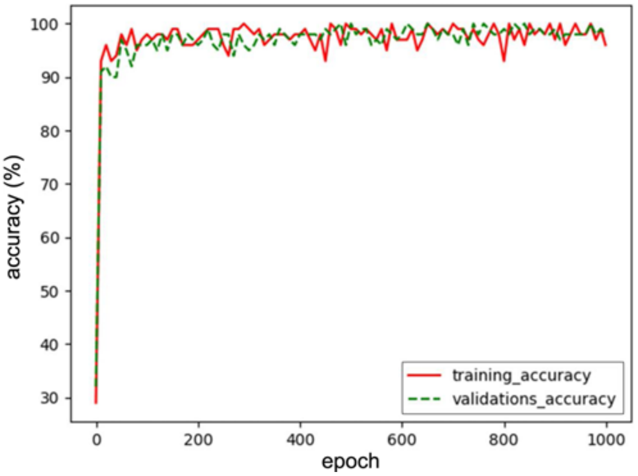

Fig. 10. Accuracy of Training and Validation Dataset for Each Epoch

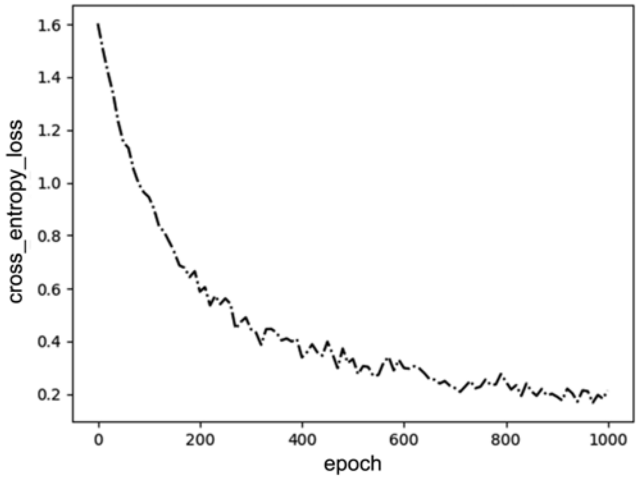

Fig. 11. Cross Entropy Loss of Training Dataset for Each Epoch

Cross entropy represents the distance between two probability distributions. The greater the cross entropy and the greater the distance between two probability distributions is, the more different the two probability distributions are. Conversely, the smaller the cross entropy and the closer the distance between two probability distributions is, the more similar the two probability distributions are. Fig. 11 shows that the cross entropy changes significantly and has a decreasing trend, showing that our network gradually converges and tends to be stable. Fig. 10 shows that the increasing trend of detection accuracy is extremely fast before the first 100 epochs; therefore, the network is highly efficient.

After completing network training, we saved the network parameters by a freezing the final model. Then, we use the final model to process the test dataset. Fig. 12 shows the results of some test images. We choose the maximum probability value as the final detection result in the detection result of each image. As we can see, except for crack_0001, which is misidentified as seepage, all others are correctly detected. This error occurred because the features of crack_0001 are similar to those of seepage. In addition, the network's predictions for crack_0005,

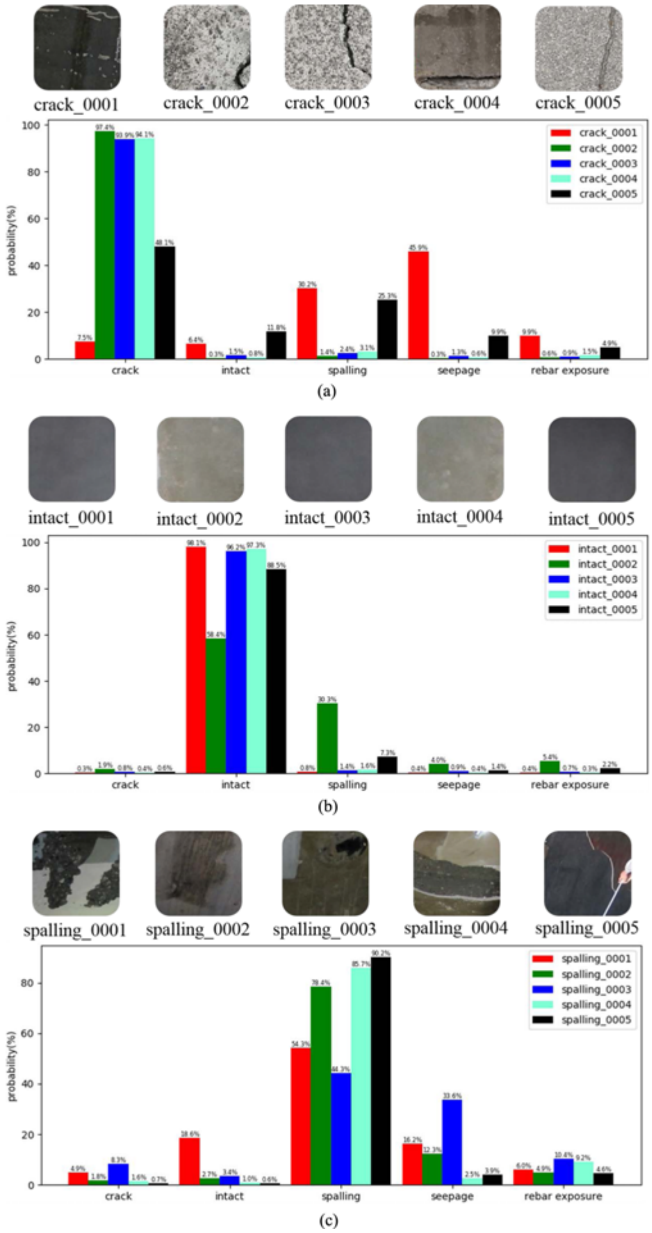

Fig. 12. Sample of Damage Detection for Five Label Type: (a) Crack, (b) Intact, (c) Spalling, (d) Seepage, (e) Rebar Exposure

intact_0002, spalling_0003 and seepage_0005 have low confidence, although the detection result of these images are correct because the features of these images are not obvious.

3.3 Comparative Study

Common machine learning classifiers, such as support vector machines, requires manual image feature selection. To demonstrate the performance of our deep convolutional neural network with transfer learning. We compared the result of a support vector machine (SVM) model with that of our method, as shown in Tables 3 and 4. To perform this comparison, we used a convenient machine learning library, SVMUTIL. SVMUTIL includes the SVM feature extraction algorithm for image classification.

Accuracy is a common index of detection accuracy. To understand the details of the detection results, we can use a confusion matrix to intuitively express the number of correct and incorrect results of damage detection. As shown in Tables 3 and 4, the first line label represents the prediction type, and the first column represents the ground truth.

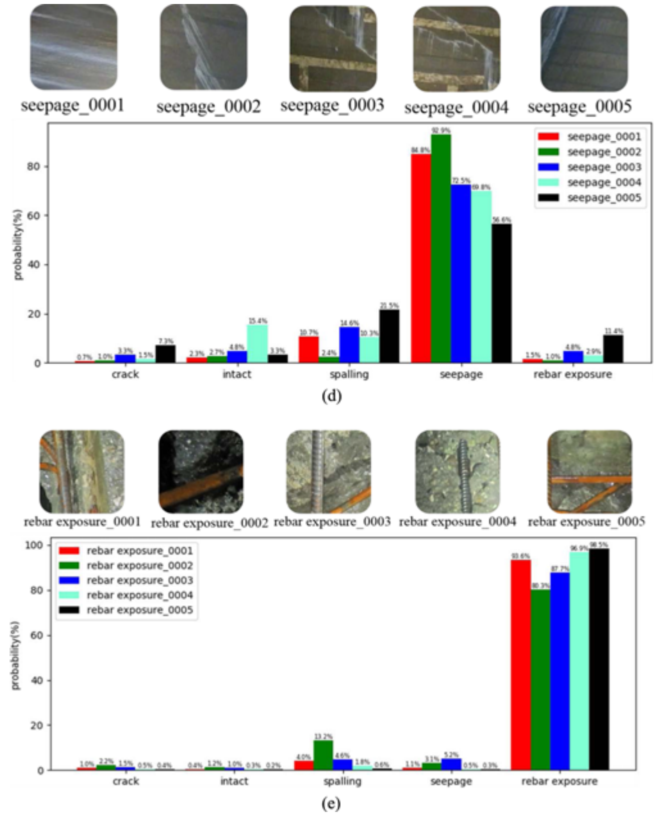

Fig. 12. (continued)

Table 3. Confusion Matrix of the Proposed Damage Detection Method

Truth label	Prediction label				
	Crack	Intact	Spalling	Seepage	Rebar exposure
Crack	395	0	1	0	0
Intact	0	342	0	8	0
Spalling	1	1	346	8	5
Seepage	0	1	7	372	0
Rebar exposure	0	1	26	0	315

Table 4. Confusion Matrix of SVM

Truth label	Prediction label				
	Crack	Intact	Spalling	Seepage	Rebar exposure
Crack	376	11	0	6	3
Intact	0	307	28	4	11
Spalling	17	14	209	67	35
Seepage	13	117	89	126	35
Rebar exposure	75	58	56	69	103

Table 3 shows the confusion matrix for the proposed damage detection method, and Table 4 shows the confusion matrix for the SVM method. We can clearly determine the number of misclassifications and the prediction accuracy for each damage type. Although, the SVM method achieves high detection accuracy for the cracks and intact classes, its accuracy on the spalling and seepage classes is very low. Its overall average accuracy is 61.2%.

There are several indicators for measure the quality of a damage detection method. The sensitivity metric is also called recall and its calculation formula is shown in Eq. (4). *TP* indicates that the ground truth label is positive and that the prediction result was also positive, *FN* represents outcomes in which the ground truth label is positive but the prediction result is negative. Second, the positive prediction value is called precision which represents the positive accuracy of the predicted results. *TN* represents a ground-truth label that is negative and whose predicted label is also negative. *FP* represents a ground-truth label that is negative but whose predicted result is positive. Eq. (5) shows the calculation of precision. Finally, for a real actual detection task, both precision and recall should be comprehensively considered; thus the F_1 value is a necessary factor. Eq. (6) shows the F_1 calculation.

$$recall = \frac{TP}{(TP + FN)} \tag{4}$$

$$precision = \frac{TP}{(TP + FP)} \tag{5}$$

$$F_1 = \frac{(2 \times recall \times precision)}{(recall + precision)} \tag{6}$$

4. Discussion of Results

From the results of these experiments, we can see that the proposed damage detection method achieves high performance. As shown in Table 3, there are only 59 images misclassified by the proposed damage detection method, which translates to an accuracy is 96.8%. As shown in Tables 5 and 6, we compared our damage detection method with an SVM model on multiple parameter indicators. The *TPR* values of the cracks and intact classes are high when using the SVM method, but its results on the remaining three types are poor. In terms of *recall, precision* and F_1 for each defect types, our damage detection method is higher than SVM. The SVM cannot effectively extract image features, while a deep convolution neural network can extract high dimensional feature information from an image through a large number of convolution kernel parameters. Table 7 shows the average accuracy of the proposed damage detection method and the SVM method on our test dataset. For each damage type the corresponding model accuracy is listed, indicating the proportion of correct predictions to the total number of each damage type. The average accuracy on all damage types is taken as the average damage detection accuracy. We can see that the proposed damage detection method is much more accurate than SVM. Concrete damage image from hydropower stations have high noise, similar backgrounds, complex textures, and inconspicuous features. Thus it is necessary to apply a network that can extract high-dimensional texture features well. The Inception-v3 uses multi-scale convolution kernel in its convolutional layers and then stacks the multiple convolution results. This approach increases the width of the network and makes it more suitable for multi-scale targets. Because the distribution and shapes of concrete damage are not fixed, Inception v3 can efficiently extracts damage features such as crack and spalling. Every inception module has a batch normalization that can accelerate convergence.

Table 5. Recall, Precision and F_1 of the Proposed Damage Detection Method

Label type	Detection indicator		
	Recall	Precision	F_1
Crack	0.997	0.997	0.997
Intact	0.977	0.991	0.983
Spalling	0.958	0.910	0.933
Seepage	0.978	0.958	0.967
Rebar_exposure	0.921	0.984	0.951

Table 6. Recall, Precision and F_1 of the SVM Method

Label type	Detection indicator		
	Recall	Precision	F_1
Crack	0.949	0.781	0.856
Intact	0.877	0.605	0.716
Spalling	0.578	0.547	0.562
Seepage	0.331	0.463	0.386
Rebar_exposure	0.301	0.550	0.389

Table 7. Comparison of Average Accuracy

Method	Average accuracy
Proposed method	96.8%
SVM	61.2%

In our research, we do not pay attention to the location of detected damage within input image. The recognition and classification of multiple types of damage is our main attentive focus. On the basis of the situation analysis for practical engineering applications, the actual position of detected damage in the hydropower station is necessary for later damage repair. Our each raw image contains information about the global position system (GPS), such as latitude, longitude and altitude. In the test phase of the network, we only need to obtain the type of input image and use the GPS information of the input image to locate the detected damage in the hydropower station. The Faster R-CNN method was used to detect damage (Cha *et al*., 2017). They have focused on the locating accuracy of detected damage within input image and the real-time simultaneous detection of multiple types of damages. However, the focus of our detection work is on the multiple types of damage recognition and classification. The GPS information of input image is applied to the positioning of detected damage in the hydropower infrastructure, which is different from the work of the paper mentioned above and our work.

When we collect images, the weather and lighting will affect the quality of the image, which results in the features of images that are not obvious, and the detection accuracy will be reduced. Some ideas may improve the accuracy for these images. First, we increase the number of these images by rotating, mirroring, and flipping so that these images can be trained more times. Second, changing these images of brightness and contrast by traditional image processing technology to enhance texture features. In future work, we will improve the network of structures to enhance the feature extraction capabilities of the network. Convolutional neural networks can use the GPU for accelerated training and processing. Therefore, the proposed damage detection method is time-saving. However, there is currently no unified image dataset for hydro-junction infrastructure so it is difficult to make a comprehensive comparison.

5. Conclusions

In this paper, we proposed a damage detection method using a deep convolution neural network with transfer learning for hydro-junction infrastructure. Our dataset has five label types: crack, seepage, spalling, intact and rebar exposure. We use the Inception-v3 network as the basic network, and the Inception-v3 network helps to extract richer feature of the image. Our dataset is relatively small. At the same time, transfer learning is especially suitable for small datasets. Therefore, we adopted an Inception-v3 and applied transfer learning to detect damage to hydro-junction infrastructure. Our model achieves a higher detection accuracy than does an SVM model; its final test accuracy was 96.8% on our test dataset.

Acknowledgements

The work described in this paper was supported by the Sichuan Energy Internet Research Center of Tsinghua University.

ORCID

Chuncheng Feng https://orcid.org/0000-0003-1822-7257

References

Beckman, G. H., Polyzois, D., and Cha, Y. J. (2019). "Deep learning-based automatic volumetric damage quantification using depth camera." *Automation in Construction*, Elsevier, Vol. 99, No. 1, pp.114-124, DOI: 10.1016/j.autcon.2018.12.006.

Cha, Y., J., Choi, W., and Büyüköztürk, O. (2017). "Deep learning-based crack damage detection using convolutional neural networks." *Computer-Aided Civil and Infrastructure Engineering*, wiley, Vol. 32, No. 5, pp. 361-378, DOI: 10.1111/mice.12263.

Cha, Y. J., Choi, W., Suh, G., Mahmoudkhani, S., and Büyüköztürk, O. (2018). "Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types." *Computer-Aided Civil and Infrastructure Engineering*, Wiley, Vol. 33, No. 9, pp. 731-747, DOI: 10.1111/mice.12334.

Donahue, J., Jia, Yangqing., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., and Darrell, T. (2013). "DeCAF: A deep convolutional activation feature for generic visual recognition." *arXiv:1310.1531v1*.

Feng, C., Liu, M. Y., Kao, C. C., and Lee, T. Y. (2017). "Deep active learning for civil infrastructure defect detection and classification." *International Workshop on Computing in Civil Engineering 2017*, ASCE, Seattle, Washington, pp. 298-306, DOI: 10.1061/9780784480823.036.

Gopalakrishnan, K., Khaitan, S. K., Choudhary, A., and Agrawal, A. (2017). "Deep convolutional neural networks with transfer learning for computer vision-based data-driven pavement distress detection." *Construction and Building Materials*, Elsevier, Vol. 157, No. 1, pp. 322-330, DOI: 10.1016/j.conbuildmat.2017.09.110.

Gui, G., Pan, H., Lin, Z., Li, Y., and Yuan, Z. (2017). "Data-driven support vector machine with optimization techniques for structural health monitoring and damage detection." *KSCE Journal of Civil Engineering,* KSCE, Vol. 21, No. 2, pp. 523-534, DOI: 10.1007/s12205-017-1518-5.

Kang, D. and Cha, Y. J. (2018). "Autonomous uavs for structural health monitoring using deep learning and an ultrasonic beacon system with geo-tagging." *Computer-Aided Civil and Infrastructure Engineering*, Wiley, Vol. 33, No. 10, pp. 885-902, DOI: 10.1111/mice.12375.

Kim, H., Ahn, E., Shin, M., and Sim, S, H. (2018). "Crack and noncrack classification from concrete surface images using machine learning." *Structural Health Monitoring*, SAGE, Vol. 18, No. 3, pp. 725-738, DOI: 10.1177/1475921718768747.

Kim, H., Lee, J., Ahn, E., Cho, S., Shin, M., and Sim, S.-H. (2017). "Concrete crack identification using a uav incorporating hybrid image processing." *Sensors*, MDPI, Vol. 17, No. 9, p. 2052, DOI: 10.3390/s17092052.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). "Imagenet classification with deep convolutional neural networks." *In Advances in Neural Information Processing Systems,* NIPS, USA, pp. 1097-1105, DOI: 10.1145/3065386.

Lecun, Y. L., Bottou, L., and Bengio, Y. (1998). "Gradient-based learning applied to document recognition." *Proceedings of the IEEE*, IEEE, Vol. 86, No. 11. pp. 2278-2324, DOI: 10.1109/9780470544976.ch9.

Li, X., Yang, Q., Chen, Z., Luo, X., and Yan, W. (2017). "Visible defects detection based on uav-based inspection in large-scale photovoltaic systems." *IET Renewable Power Generation*, IET, Vol. 11, No. 10, pp. 1234-1244, DOI: 10.1049/iet-rpg.2017.0001.

Makantasis, K., Protopapadakis, E., Doulamis, A., Doulamis, N., and Loupos, C. (2015). "Deep convolutional neural networks for efficient vision based tunnel inspection." *IEEE International Conference on Intelligent Computer Communication and Processing*, ICCP, Cluj-Napoca, Romania, pp. 335-342, DOI: 10.1109/iccp.2015.7312681.

Nair, V. and Hinton, G. E. (2010). "Rectified linear units improve restricted Boltzmann machines." *Proc. of the 27th International Conference on Machine Learning.,* ICML, Haifa, Israel.

Nasrollahi, A., Deng, W., Rizzo, P., Vuotto, A., and Vandenbossche, J. M. (2017). "Nondestructive testing of concrete using highly nonlinear solitary waves." *Nondestructive Testing and Evaluation*, Taylor & Francis, Vol. 32, No. 4, pp. 381-399, DOI: 10.1080/10589759.2016.1254212.

Ng, C. T. (2014). "On the selection of advanced signal processing techniques for guided wave damage identification using a statistical approach." *Engineering Structures*, Elsevier, Vol. 67, No. 1, pp. 50-60, DOI: 10.1016/j.engstruct.2014.02.019.

Nhat-Duc, H., Nguyen, Q.-L., and Tran, V.-D. (2018). "Automatic recognition of asphalt pavement cracks using metaheuristic optimized edge detection algorithms and convolution neural network." *Automation in Construction*, Elsevier, Vol. 94, No. 1, pp. 203-213, DOI: 10.1016/j.autcon.2018.07.008.

Nishikawa, T., Yoshida, J., and Sugiyama, T. (2012). "Concrete crack detection by multiple sequential image filtering." *Computer-Aided Civil and Infrastructure Engineering*, Wiley, Vol. 27, No. 1, pp. 29-47, DOI: 10.1111/j.1467-8667.2011.00716.x.

Pathirage, C. S. N., Li, J., Li, L., Hao, H., Liu, W., and Ni, P. (2018). "Structural damage identification based on autoencoder neural networks and deep learning." *Engineering Structures*, Elsevier, Vol. 172, No. 1, pp. 13-28, DOI: 10.1016/j.engstruct.2018.05.109.

Phung, M. D., Quach, C. H., Dinh, T. H., and Ha, Q. (2017). "Enhanced discrete particle swarm optimization path planning for uav vision-based surface inspection." *Automation in Construction*, Elsevier, Vol. 81, No. 1, pp. 25-33, DOI: 10.1016/j.autcon.2017.04.013.

Prasanna, P., Dana, K. J., Gucunski, N., Basily, B. B., La, H. M., Lim, R. S., and Parvardeh, H. (2016). "Automated crack detection on concrete bridges." *IEEE Transactions on Automation Science and Engineering*, IEEE, Vol. 13, No. 2, pp. 591-599, DOI: 10.1109/tase.2014.2354314.

Shi, Y., Cui, L., Qi, Z., Meng, F., and Chen, Z. (2016). "Automatic road crack detection using random structured forests." *IEEE Transactions on Intelligent Transportation Systems*, IEEE, Vol. 17, No. 12, pp. 3434-3445, DOI: 10.1109/tits.2016.2552248.

Simonyan, K. and Zisserman, A. (2014). "Very deep convolutional networks for large-scale image recognition." *arXiv:1409.1556v6*.

Singhal, T., Kim, E., Kim, T. Y., and Yang, J. (2017). "Weak bond detection in composites using highly nonlinear solitary waves." *Smart Materials and Structures*, IOP, Vol. 26, No. 5, pp. 055011, DOI: 10.1088/1361-665X/aa6823.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., and Rabinovich, A. (2015) "Going deeper with convolutions." *Proc of the IEEE Conference on Computer Vision and Pattern Recognition.*, CVPR, Boston, pp. 1-9, DOI: 10.1109/cvpr.2015.7298594.

Wang, L. and Zhang, Z. (2017). "Automatic detection of wind turbine blade surface cracks based on uav-taken images." *IEEE Transactions on Industrial Electronics*, IEEE, Vol. 64, No. 9, pp. 7293-7303, DOI: 10.1109/TIE.2017.2682037.

Xue, Y. and Li, Y. (2018). "A fast detection method via region-based fully convolutional neural networks for shield tunnel lining defects." *Computer-Aided Civil and Infrastructure Engineering*, Wiley, Vol. 33, No. 8, pp. 638-654, DOI: 10.1111/mice.12367.

Zhang, K., Cheng, H. D., and Zhang, B. (2018). "Unified approach to pavement crack and sealed crack detection using preclassification based on transfer learning." *Journal of Computing in Civil Engineering*, ASCE, Vol. 32, No. 2, pp. 04018001, DOI: 10.1061/(ASCE)CP.1943-5487.0000736.