

Article

# An Intelligent Classification Model for Surface Defects on Cement Concrete Bridges

Jinsong Zhu <sup>1,2,\*</sup>  and Jinbo Song <sup>1</sup> <sup>1</sup> School of Civil Engineering, Tianjin University, Tianjin 300072, China; jxjtxylqx@126.com<sup>2</sup> Ministry of Education, Key Laboratory of Coast Civil Structure Safety (Tianjin University), Tianjin 300072, China

\* Correspondence: jszhu@tju.edu.cn

Received: 16 January 2020; Accepted: 28 January 2020; Published: 2 February 2020



**Abstract:** This paper mainly improves the visual geometry group network-16 (VGG-16), which is a classic convolutional neural network (CNN), to classify the surface defects on cement concrete bridges in an accurate manner. Specifically, the number of fully connected layers was reduced by one, and the Softmax classifier was replaced with a Softmax classification layer with seven defect tags. The weight parameters of convolutional and pooling layers were shared in the pre-trained model, and the rectified linear unit (ReLU) function was taken as the activation function. The original images were collected by a road inspection vehicle driving across bridges on national and provincial highways in Jiangxi Province, China. The images on surface defects of cement concrete bridges were selected, and divided into a training set and a test set, and preprocessed through morphology-based weight adaptive denoising. To verify its performance, the improved VGG-16 was compared with traditional shallow neural networks (NNs) like the backpropagation neural network (BPNN), support vector machine (SVM), and deep CNNs like AlexNet, GoogLeNet, and ResNet on the same sample dataset of surface defects on cement concrete bridges. Judging by mean detection accuracy and top-5 accuracy, our model outperformed all the contrastive methods, and accurately differentiated between images with seven classes of defects such as normal, cracks, fracturing, plate fracturing, corner rupturing, edge/corner exfoliation, skeleton exposure, and repairs. The results indicate that our model can effectively extract the multi-layer features from surface defect images, which highlights the edges and textures. The research findings shed important new light on the detection of surface defects and classification of defect images.

**Keywords:** transfer learning; bridge defects; convolutional neural network (CNN); intelligent classification

## 1. Introduction

Recent years has seen a growing popularity of nondestructive evaluation (NDE) in the evaluation of infrastructural aging. One of the most promising NDE techniques for bridge defects is deep learning. This technique has various advantages: the portability of devices and the swiftness of data collection and processing. Hence, deep learning enjoys a great application potential in fast defect detection of real-world bridges.

Deep learning is an important field of machine learning [1,2]. Unlike traditional methods for image recognition, deep learning, which mimics the visual perception of humans, expresses the image features in an abstract form rather than actively establish the key features. Based on deep learning, artificial intelligence (AI) [3,4] is realized in the computing system by setting up artificial neural networks (ANNs). Below is a brief review of relevant studies on the application of the artificial neural networks (ANNs) in image classification tasks.

Gopalakrishnan et al. [5] introduced deep migration learning to detect surface cracks of roads made of hot mix asphalt (HMA) and Portland cement concrete (PCC), and observed the best detection effect on the single-layer neural network (NN) after feature training. Based on a convolutional neural network (CNN), Gulgec et al. [6] conducted finite-element simulation of cracked gusset plate connections in steel bridges, and differentiated between defected and healthy bridge samples. Wang et al. [7] proposed a CNN method based on the sliding window, which uses deep learning frameworks from AlexNet and GoogLeNet to classify bridge damages.

Chen and Jahanshahi [8] trained the CNN with the track defects in the Photometric Stereo Images Database. The defects are cavities on the track surface, which indicate further surface degradation before track breakage. Cha et al. [9] proposed an NN to classify images on bridge surfaces, and compares the expected reflection features of defected bridge surfaces and those of intact bridge surfaces. Specifically, the classical CNN is trained in a purely supervised manner, and the influence of different regularization methods is discussed including unsupervised layered pre-training and enhancement of bridge defect datasets.

Wu et al. [10] developed a CNN-based approach to detect fatigue cracks from real crack images at triangular plate joints, which are the most fatigue-prone joints of steel bridges. By this approach, the image datasets are collected from previous fatigue tests and bridge checks under uncontrolled settings. Next, the pre-trained benchmark CNN architecture on large general image datasets is adopted to transfer the deep learning features to bridge crack classification tasks. Zhang et al. [11] put forward a novel road crack detection algorithm based on deep learning and adaptive image segmentation. First, the deep CNN was trained to judge the presence of cracks in images. Then, the crack-containing images were smoothed by a bilateral filter to minimize the number of noise pixels. Lastly, the cracks were extracted from pavement through adaptive thresholding.

Pan et al. [12] designed a machine learning algorithm that adapts well to actual scenarios. Fan et al. [13] detected structural defects by machine learning. The signals were collected via nondestructive tests, and the defectiveness was evaluated based on the signals. Ding et al. [14] identified bridge cracks using Welch's power spectrum and a generalized regression neural network (GRNN). Guan et al. [15] used a surface acoustic wave and the NN to identify pavement cracks of bridges. To determine the material damages of bridge surfaces, Yam et al. [16] identified the vibration of composite structures through wavelet transform and deep learning-based NN.

Peng et al. [17] decomposed the gray images of various roads into approximate and detailed sub-images through wavelet transform, created a feature vector based on energy features of the wavelet, and computed the gray level co-occurrence matrix (GLCM) features of detailed sub-images. Lastly, the backpropagation neural network (BPNN) was optimized, using eight characteristic parameters, and was trained to achieve the best results. Sharif Razavian et al. [18] summed up the merits of the CNN over the standard NN, which include the ability to capture the grid topology of road surfaces, the need for fewer computing iterations due to neuron sparsity and the pooling operation, and the excellent results of road defect detection.

With the aid of the CNN, Soukup and Huber-Mork [19] found the defects on railway surfaces, which enhanced the manufacturing quality of railway workpieces. Qiao et al. [20] designed a practical method to detect defects of bridge surfaces, which integrates the CNN with pattern recognition. Shang L. [21] combined image processing and the CNN to detect defects on bridge surfaces. In light of deep learning, Yang and Zhao [22] proposed a CNN-based defect detection algorithm of steel surfaces, which establishes a CNN model and datasets to extract and detect steel surface defects automatically. Prasanna et al. [23] achieved 90% detection accuracy through thousands of crack tests on bridge surfaces.

German et al. [24] integrated thresholding and template matching to measure the spalling degree of post-seismic bridges, and observed that the integrated strategy achieved 81.1% in mean accuracy and 80.2% in sensitivity on horizontally-exposed concrete. Chen et al. [25] adopted the deep NN to detect different types of defects. Four independent classifiers were trained by one or several types

of bridge surface defects. The trained classifiers realized an accuracy of 79.5% and a sensitivity of 65.0%. The results fully demonstrate the classification ability of deep NN. You et al. [26] proposed a deep learning method that directly estimates bridge defects from outdoor images without relying on expensive customized sensors. Imani, M. et al. [27] proposed a Bayesian decision making framework for control of Markov Decision Processes (MDPs) with unknown dynamics and large, possibly continuous, state, action, and parameter spaces in data-poor environments. Xie S. et al. [28] constructed a classification rule for nonstationary data through linear discriminant analysis (LDA), using a linear Gaussian state space model. Imani M. et al. [29] proposed an approach for finite-horizon control of partially-observed Boolean dynamical systems (POBDS) with uncertain, continuous controlled input and infinite observation space. Imani M. et al. [30] derived an optimal Bayesian estimator for damage state and parameters, and created deep learning features based on the CNN and a recurrent neural network (RNN). The CNN and the RNN were connected with shortcuts. The former captures the full view of the detection target, while the RNN mimics the transfer of human attention. The extension of the CNN to the RNN makes defect detection more accurate, which provides a novel deep learning approach in the industrial field.

Through the above analysis, this paper attempts to develop an intelligent classification system for surface defects on the cement concrete bridge. First, a complete dataset of concrete bridge surface defects was established, and the tags of seven classes of defects were set up according to the current code in China. Next, the sample images were preprocessed through morphology-based weight adaptive denoising. Lastly, the parameters of the VGG-16, which is a classic CNN, were optimized to realize intelligent identification and classification of surface defects.

## 2. Sampling and Image Preprocessing

### 2.1. Sampling

The data samples were collected by a road inspection vehicle (Figure 1) of the Highway Engineering Test and Inspection Center, Jiangxi Province, China. The vehicle drove at a mean speed of 60 km/h along the roads in Jiangxi Province, including national highways like G105, G319, and G356 and provincial highways like 219, 221, 222, and 312. A total of 600,000 concrete bridge images were taken by charge-coupled device (CCD) line-scan cameras (2048×2000 pixels) mounted on the vehicle.



**Figure 1.** Road inspection vehicle.

Out of the 600,000 images, 40,610 images with clear features were extracted manually, according to the seven classes of defects mentioned in the current code in China (Figures 1 and 2). This creates a dataset of surface defects on cement concrete bridges. The dataset contains the images in seven classes of defects, including 3000 normal images, 8090 images of cracking, 9790 images of plate fracturing, 3320 images of corner rupturing, 4180 images of edge/corner exfoliation, 6900 images of skeleton exposure, and 5240 images of repairs. From the extracted images, 90% were randomly selected and allocated to the training set, and the remaining 10% were allocated into the test set. The division between the two sets and the number of images in each defect class are shown in Figure 3.

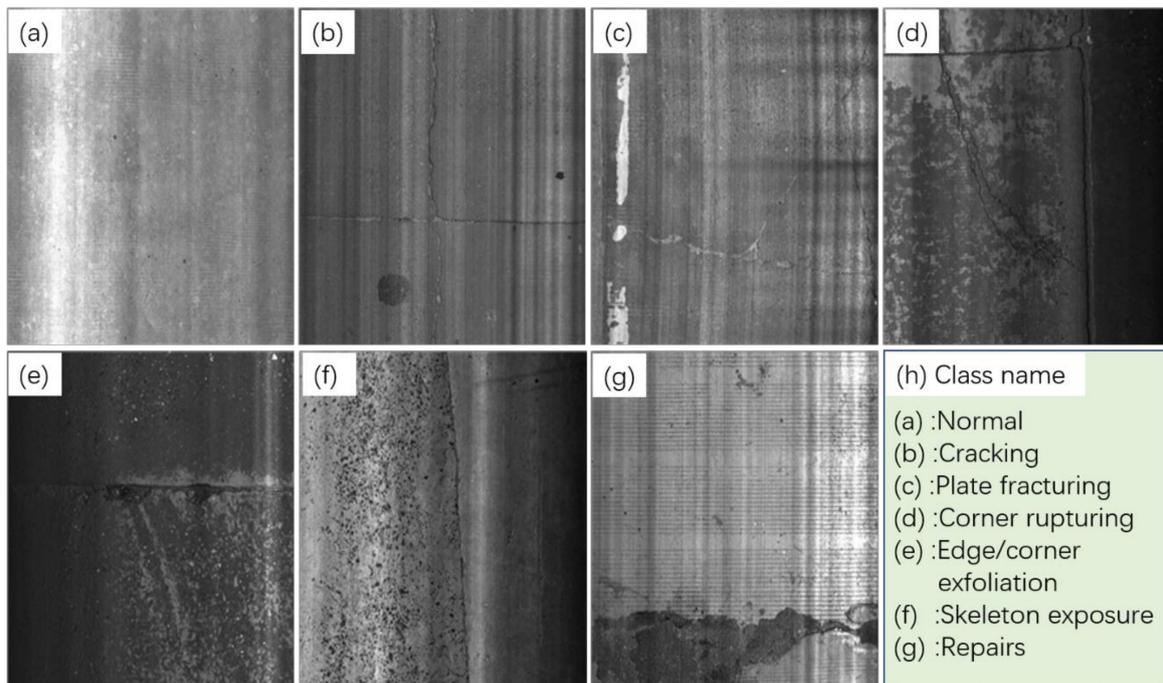


Figure 2. Sample images and seven defect classes.

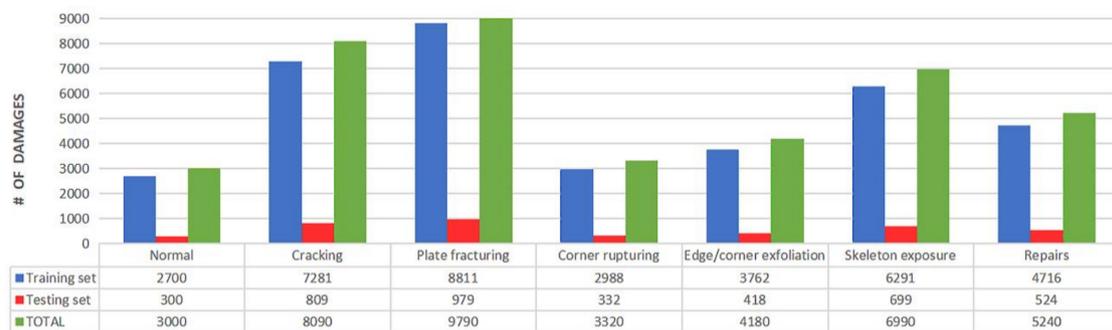


Figure 3. Number of images in each defect class.

### 2.2. Image Preprocessing

During the collection and transmission, digital images are affected by noises arising from camera elements and an external environment. To eliminate the noise impacts, the CNN was introduced to preprocess the collected images. Considering the difficulty in differentiating between defects, the authors developed a morphology-based weight adaptive denoising method for the collected images. The developed method integrates basic morphological transformations such as corrosion, expansion, opening, and closing operations. Several shapes were selected to characterize the seven classes of defects, and used to distinguish between the images in light of the set theory. Normally, the positive and negative impulse noises can be filtered out from the images through open and close operations, respectively. In this paper, the two operations were combined into a morphology-based open-close cascade filter (1) and a morphology-based close-open cascade filter (2).

$$OC(m(x)) = (m \circ h \bullet h)(x) \tag{1}$$

$$CO(m(x)) = (m \bullet h \circ h)(x) \tag{2}$$

On this basis, the structural elements with the selected shapes were combined with the adaptive weight algorithm into a composite filter (Figure 4) to preprocess the collected images.

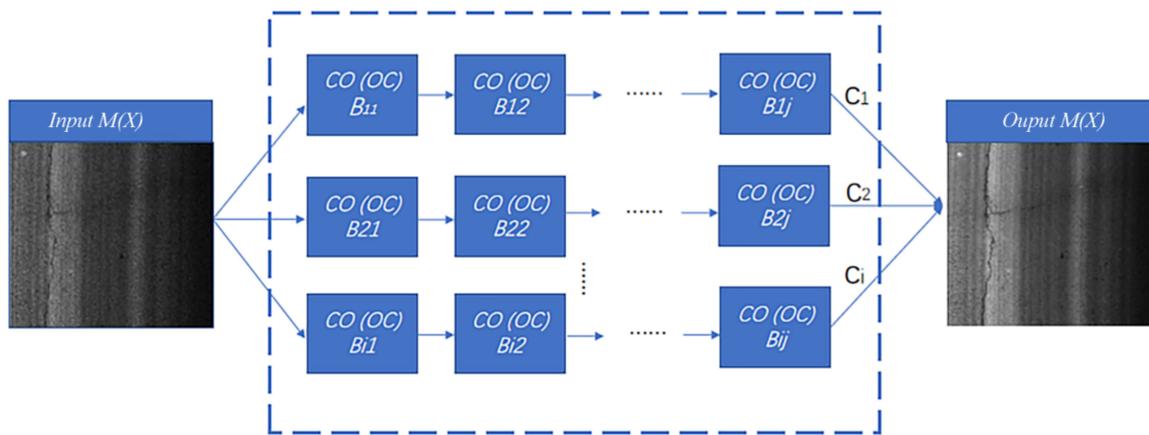


Figure 4. The composite filter for image preprocessing.

In Figure 4,  $C_i$  is the weight of the  $i$ -th structural element and  $M(x)$  is the input image. For any image inputted, the outputs  $m_n(X)$ ,  $n = 1, 2, \dots, i$  of the composite filter will be provided as an output image.

$$M(x) = \sum_{n=1}^i c_i m_n(x) \tag{3}$$

The set theory was employed to analyze each image. The morphology-based filter was selected, considering the geometric features of the target image. The purpose is to perform effective and efficient filtering of the original image without losing any information, which makes it easier for the convolutional kernel to extract features.

### 3. Basic Principles of Deep Learning

#### 3.1. The CNN

The CNN is an emerging deep ANN capable of convolutional computation, which offers a representative algorithm of deep learning. Mimicking the visual perception of humans, the CNN supports supervised learning, unsupervised learning, and semi-supervised learning. Over the years, this network has been successfully applied to such fields as natural language processing, voice recognition, and image classification.

The CNN model generally consists of the convolutional layer, the pooling layer, the fully connected layer, and the Softmax classification layer. Among them, the convolutional and pooling operations can be repeated many times to optimize the filtering effect. The relatively mature CNN models include LeNet, AlexNet, GoogLeNet, VGG-16Net, and ResNet. All of them have performed well in the field of artificial intelligence (AI).

##### 3.1.1. Convolutional Layer

There is a kernel on each convolutional layer. During the operation, the kernel processes different areas of the input image according to the preset parameters (e.g. the step length), which outputs a convoluted feature map. Through the convolutional layer, the features can be extracted from the input image through the following convolutional computing.

$$y^{conv} = F \left( \sum_{j=0}^{J-1} \sum_{i=0}^{I-1} x^{a+i,d+j} w^{ij} + b \right) \tag{4}$$

$(0 \leq a \leq A, 0 \leq d \leq D)$

where  $x$  is a 2D vector of the input image,  $A$  and  $D$  are the length and width of the 2D vector, respectively,  $w$  is the convolutional kernel,  $j$  and  $i$  are the length and width of the kernel, respectively,  $b$  is the bias of each element of the kernel,  $y^{conv}$  is the convolutional result, and  $F$  is the activation function.

### 3.1.2. Pooling Layer

After the feature extraction through the convolutional layer, the resulting feature map is transferred to the pooling layer for further feature extraction and information filtering. The common pooling methods include max pooling, average pooling, and global average pooling. The most classic method is max pooling. For instance, take a 1D  $6 \times 6$  input matrix without padding. The max pooling with a filter size of  $2 \times 2$  and a step length of 2 will output a 1D  $3 \times 3$  matrix. Therefore, the feature map is reduced by half in size through max pooling, while the dimensions of the information channels remain unchanged. Figure 5 explains the max pooling method of this paper.

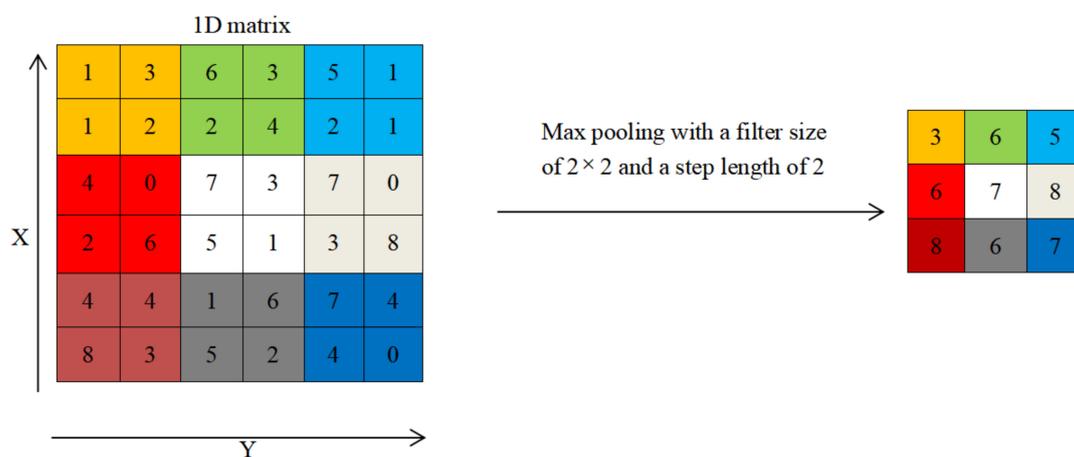


Figure 5. The max pooling process.

### 3.1.3. Fully Connected Layer

The fully connected layer appears at the end of the CNN. The output of convolution/pooling is flattened into a single vector of values, with each representing a probability that a certain feature belongs to a label. In this layer, each neuron is connected to every neuron in the superior layer, such that the distributed features extracted and learned in the previous layers can be mapped into the tag space. In other words, all the features previously obtained are consolidated into a unified output, which reduces the burden of subsequent classification tasks. To improve the nonlinear expression, the fully connected layer is usually activated by an activation function. Figure 6 presents the structure of a relatively simple, fully connected layer.

## 3.2. Activation Function

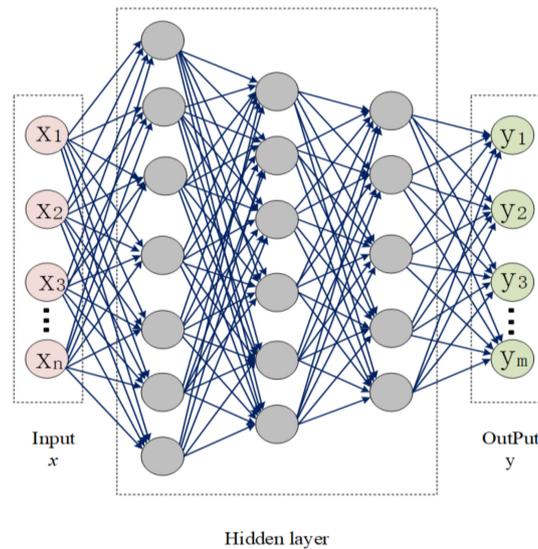
In the neural network (NN) architecture, some nonlinear functions are often introduced to activate some inputs or outputs. These functions are known as activation functions. The overlapping of these functions will improve the nonlinear fitting of the CNN. Typical activation functions include sigmoid function, tanh function, and ReLU function.

### 3.2.1. Sigmoid Function

The sigmoid function can be expressed by the equation below.

$$f(x) = \frac{1}{1 + e^{-x}} \tag{5}$$

The sigmoid function is one of the earliest and most popular activation functions. However, there are two clear defects with this function. First, this function can easily lead to vanishing gradient, which makes it impossible for neurons to accept signals or transfer weight and data. Second, the output of sigmoid function (Figure 7a) is not based on zero but the interval of [0, 1], i.e., the function always outputs positive results. Thus, the weight gradients are either all positive or all negative, which adds to the optimization difficulty.



**Figure 6.** The structure of the fully connected layer.

### 3.2.2. Tanh Function

The tanh function expression can be expressed by the equation below.

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{6}$$

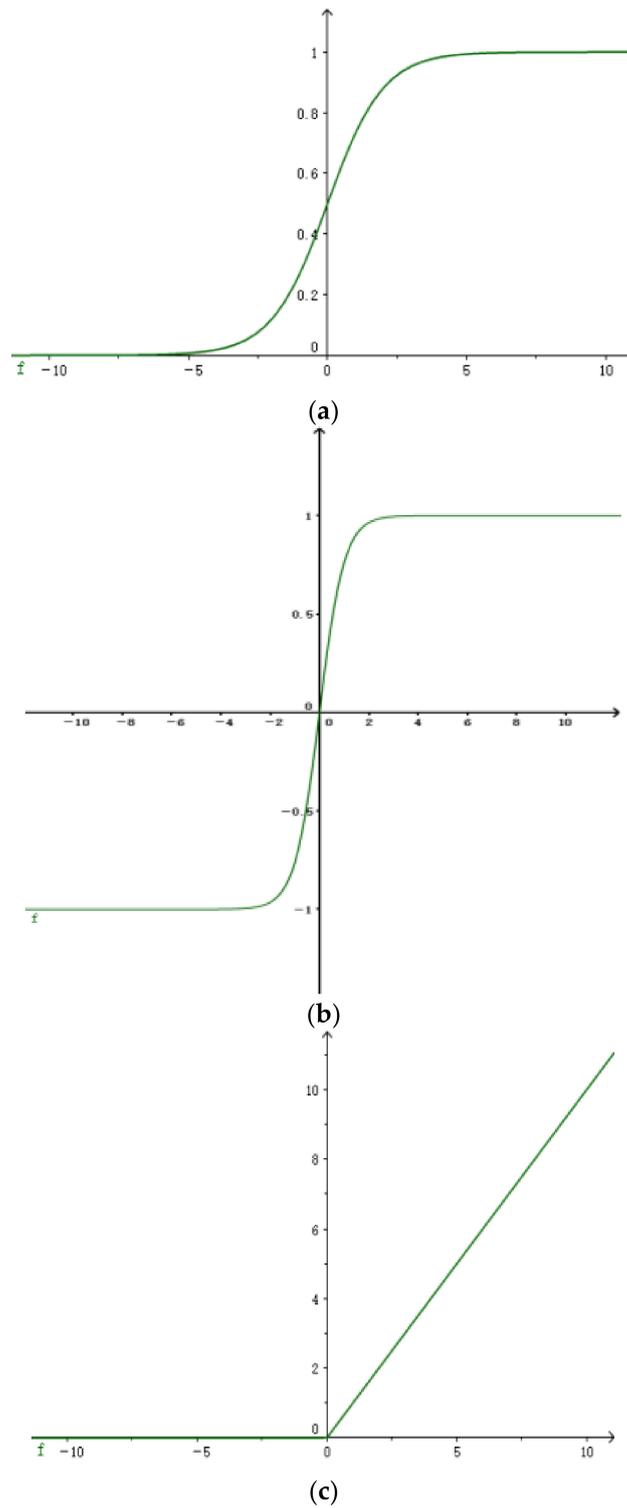
As shown in Figure 7b, the tanh function controls the output within [−1, 1], which overcomes the second defect of sigmoid function and facilitates the optimization process. Nevertheless, the tanh function also faces the problem of a vanishing gradient.

### 3.2.3. ReLU Function

The rectified linear unit (ReLU) function can be expressed by the equation below.

$$f_{ReLU}(x) = \max\{0, x\} \tag{7}$$

Compared with a sigmoid function and tanh function, the ReLU function (Figure 7c) solves the vanishing gradient and speeds up the convergence. Therefore, this function has a small error in the back-propagation process.



**Figure 7.** Geometric diagram of typical activation functions. (a) Sigmoid function. (b) Tanh function. (c) ReLU function.

### 3.3. Transfer Learning

Transfer learning refers to transferring the acquired knowledge in a certain field to the related disciplines, which eliminates the need to restart the learning of the entire knowledge system. If applied in image classification, transfer learning mainly relocates the knowledge acquired from the training of the original task to the application of the target task.

In this paper, the surface defects of cement concrete bridges are detected based on the VGG-16. Specifically, the VGG-16 was trained on the ImageNet dataset, and the weights and parameters of the trained model were migrated to our model. Thus, our model could achieve a better detection effect on cement concrete bridges.

#### 4. Model Construction and Experiments

##### 4.1. Construction and Optimization of the Visual Geometry Group Network-16 (VGG-16) Model

The VGG-16 is a CNN with great strength in image classification. A typical VGG-16 consists of 13 convolutional layers and three fully connected layers. The relatively large kernels are replaced with two or three  $3 \times 3$  kernels. Through the replacement, the nonlinear fitting ability will be enhanced, such that the same convolutional effect can be achieved with fewer parameters.

In this case, each input image is subjected to feature extraction on the convolutional layer with a step length of 1. Then, the resulting feature map received max pooling with the step length of 2 and the pooling window of  $2 \times 2$ . The planar size of the feature map was halved after max pooling. Through multiple convolutions, pooling, and ReLU activation, the half-size feature map was inputted into three fully connected layers, whose channel number was 4096, 4096, and 1000, respectively. The greater the number of channels, the lower the spatial resolution. Thus, the input image could be converted into the classification vector smoothly in terms of the dimension. Lastly, the obtained data were classified by a Softmax classifier with 1,000 tags. The proposed VGG-16 model is illustrated in Figure 8.

The VGG-16 convolutional neural network model contains a large number of weight parameters, which are trained by a large image database of ImageNet, so it has a strong ability of deep feature learning and excellent performance in image classification and target extraction detection (Figure 9). The depth of the configurations increases from the left (A) to the right (E), as more layers are added (the added layers are shown in bold). The convolutional layer parameters are denoted as “conv<receptive field size> - <number of channels>”. The rectified linear unit (ReLU) activation function is not shown for brevity.

In this paper, the principles of fine tuning and transfer learning are implemented to save cost and time of the entire network. The mature VGG-16 model and trained parameters were directly improved to recognize the seven classes of surface defects on cement concrete bridges. The improved network can be divided into convolutional, pooling, fully connected, and Softmax classification layers.

To further optimize the network performance, the ReLU function was adopted as the activation function. The three fully connected layers of the VGG-16 model were reduced into two, and the Softmax classifier with 1,000 tags was changed into a Softmax classification layer with seven tags. Through these fine tunings, the VGG-16 model can efficiently and accurately recognize the surface defects of cement concrete bridges.

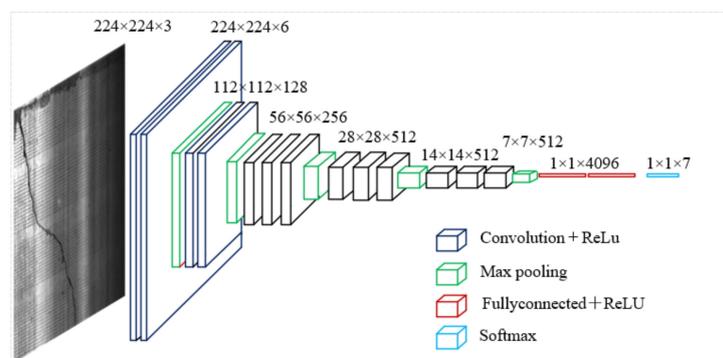


Figure 8. Visual geometry group network-16 (VGG-16) transfer learning model.

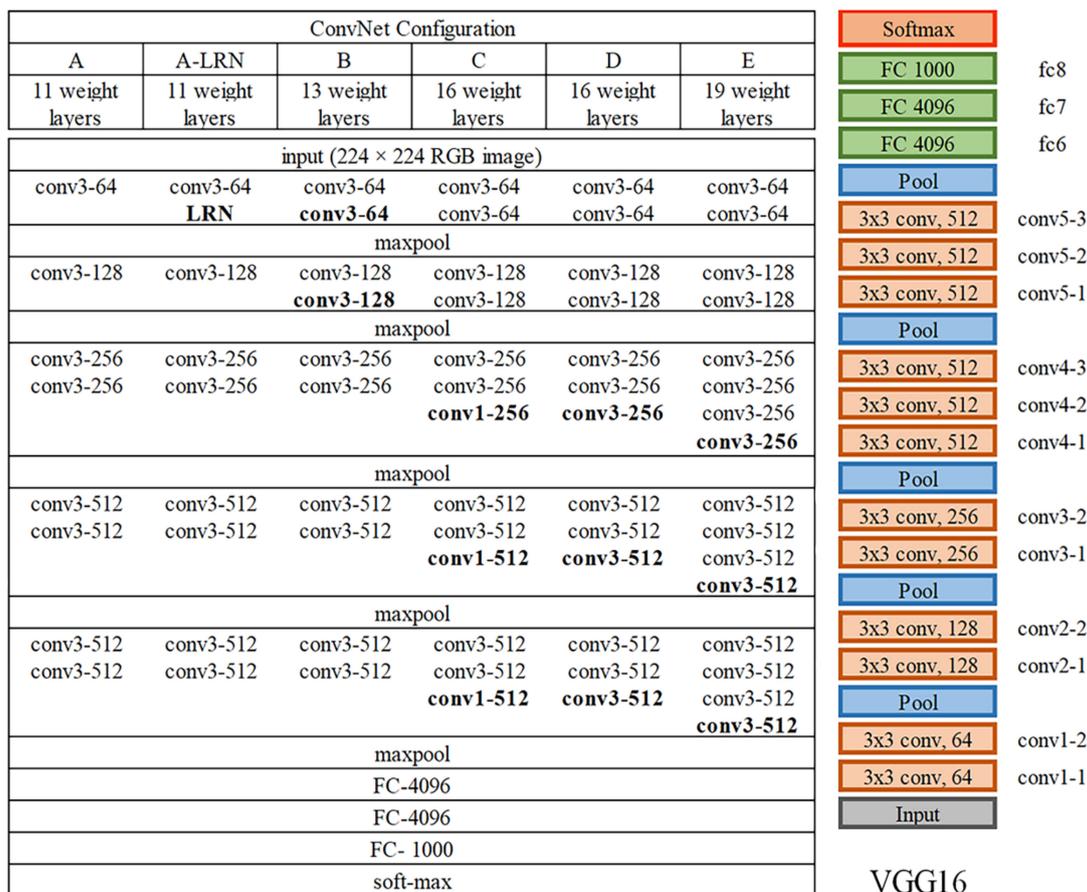


Figure 9. Structure of our visual geometry group network-16 (VGG-16) model.

Overall, our VGG-16 model has a total of 16 convolutional or fully connected layers, including five groups of convolutional layers and three fully connected layers:  $16 = 2+2+3+3+3+3$ .

#### 4.2. Experimental Steps

The parameters of the pre-trained VGG-16 model were migrated to the surface defect detection model for cement concrete bridges, which follows the principles of fine tuning and transfer learning. In addition, the fully connected layers and Softmax classification layer of the VGG-16 model were adjusted. Our experiments were carried out in the following steps.

Step 1. Inputting the surface defect samples of cement concrete bridges.

The images of the seven defect classes were randomly extracted from the database of defect samples of cement concrete bridges, and inputted as the training set.

Step 2. Preprocessing the input images.

The images are polluted by noises in collection and transfer. In this case, the morphology-based weight adaptive denoising method is applied to enhance the image features and eliminate the useless noises from the images. Then, the input images were scaled and cropped into the standard size of  $224 \times 224$ .

Step 3. Constructing the surface defect model.

The multi-class logistic regression objectives were optimized through backpropagation with small batch gradient descent with momentum (Batch size: 256. Momentum: 0.9). To void overfitting, the

weight attenuation was regularized (L2 penalty factor:  $5 \times 10^4$ ) and the first two fully connected layers were discarded (discard ratio: 0.5). The learning rate was initialized as  $10^{-2}$ , and then reduced by 10 times when the accuracy of the test set no longer improved.

Step 4. Adjusting the VGG-16 model

Based on the pre-trained VGG-16 model, the three fully connected layers were reduced to two and the Softmax classifier with 1,000 tags was replaced with a Softmax classification layer with seven tags.

Step 5. Fine tuning and transfer learning

The pre-trained parameters of the VGG-16 were migrated to the 13 convolutional layers and pooling layers of the surface defect detection model for cement concrete bridges through fine tuning and transfer learning.

Step 6. Training the surface defect detection model

Before training, the momentum, learning rate, and training time were configured manually. The parameters of the first 13 convolutional and pooling layers were kept constant, and the model was randomly initialized to obtain the initial loss. The model was then iteratively optimized through error backpropagation and gradient descent, which aims to train the parameters of the subsequent two fully connected layers and the Softmax classification layer with seven tags. The optimization was terminated until the value of the loss function reached the minimum. At the termination, the weights and biases of the model were optimal, which correspond to the best modelling effect.

Step 7. Testing the surface defect detection model

The remaining images of the seven defect classes were extracted from the database and used to verify the established surface defect detection model. The accuracy and efficiency of the model were tested through the steps in Figure 10.

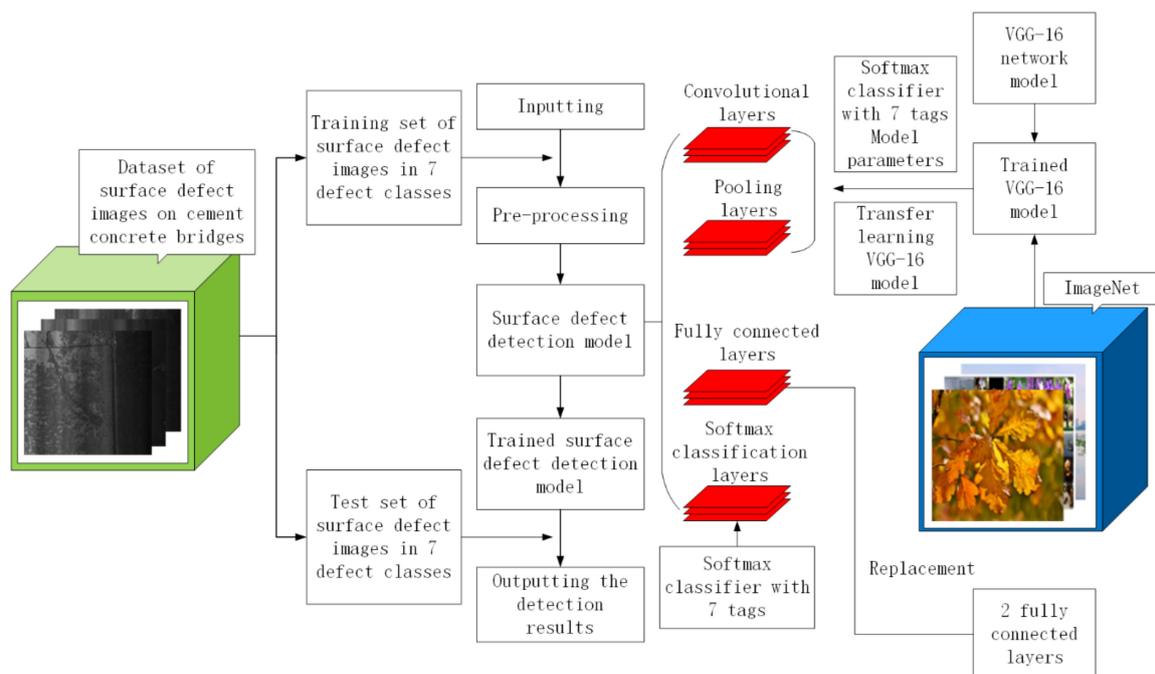


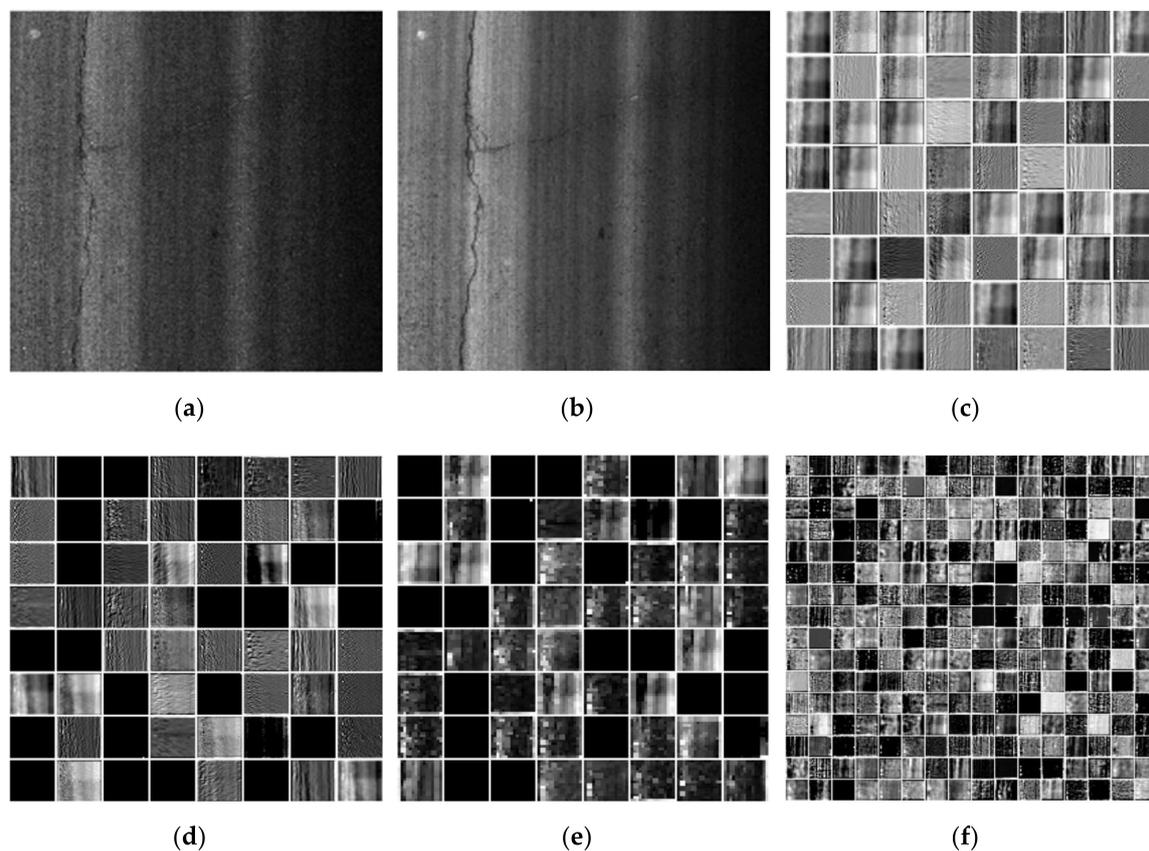
Figure 10. Experimental flow.

## 5. Experimental Results and Analysis

Our experiments were carried out on a computer with an NVIDIA GeForce MX130 GPU, an Intel Core i7-8550U CPU (4 cores and 8 threads), two 4GB DDR3 memories. The computer runs on Windows 10. The improved VGG-16 model for surface defect detection was trained, tested, and simulated on Matlab2018a.

### 5.1. Model Training Strategy and Test Analysis

The selected training set was divided into several small batches for training. The stochastic gradient descent and error backpropagation were adopted to solve the negative gradient of each sub-training set and to adjust the model parameters continuously such as to iteratively optimize the model. Through the optimization, a VGG-16 model with good detection performance of surface defects was obtained. The size of the sub-training set, momentum, number of iterations, initial learning rate, and the adjustment factor of the learning rate were set to 100 images, 0.8, 4,000 times, 100 iterations, and 0.88, respectively. Figure 11 displays the original image and the images in each step of pre-processing. The final results are provided in Figure 12.



**Figure 11.** The original image and the images in each step of pre-processing. (a) Original image. (b) Preprocessed image. (c) Convolutional layer. (d) rectified linear unit (ReLU) 1-1. (e) Max pooling layer 1-1. (f) Fully connected layer 1.

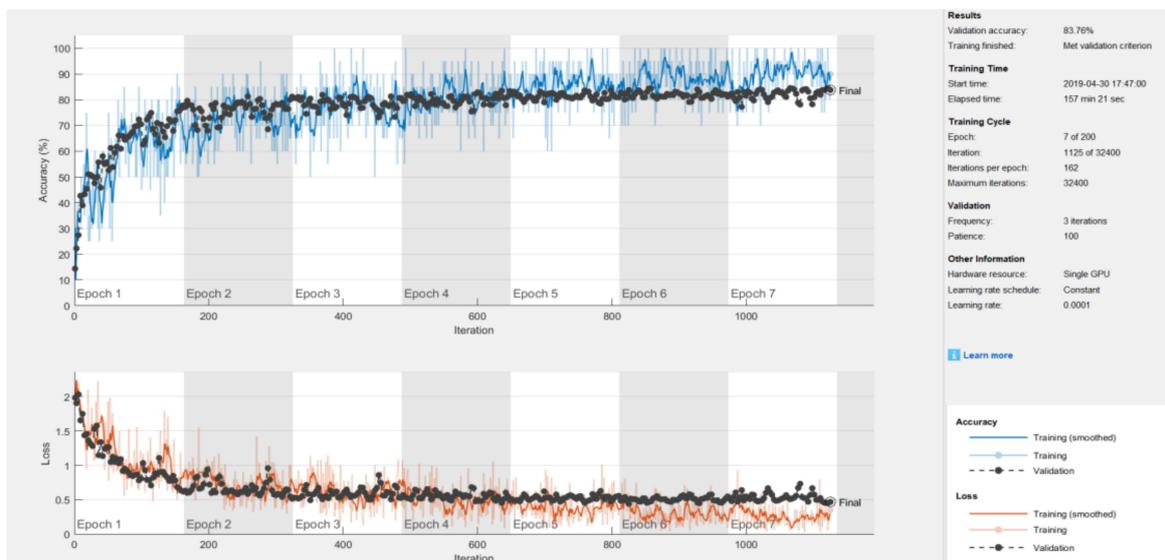


Figure 12. The accuracy and loss of the model.

### 5.2. Comparative Experiments

To verify its performance, the proposed model was compared with traditional shallow NNs like the backpropagation neural network (BPNN) and support vector machine (SVM) and deep CNNs like AlexNet, GoogLeNet, and ResNet on the same sample dataset of surface defects on cement concrete bridges.

With relatively few layers, the shallow NNs are good at solving simple problems. However, the functions of the shallow NNs do not have enough expression ability to tackle complex real-world problems. By contrast, deep CNNs provide excellent tools for complex problems, but their robustness varies from problem to problem.

Therefore, the above contrastive models and our model were trained and tested at the same ratio between training and test samples. All the models were tested under the framework of Matlab2018a. The results of these models are compared in Table 1.

Table 1. Comparison of the six models on surface defect detection.

Recognition Model	Normal	Cracks	Plate Fracturing	Corner Rupturing	Edge/ Corner Exfoliation	Skeleton Exposure	Repairs	Mean Accuracy	Top 5-Accuracy
BPNN	59.72	60.01	57.65	61.23	58.96	60.82	58.75	59.59	70.68
SVM	61.54	62.37	60.23	64.58	61.66	64.39	61.36	62.30	80.56
AlexNet	79.32	81.51	78.31	82.96	79.89	81.65	82.16	80.83	92.32
GoogLeNet	78.46	84.65	80.24	83.61	80.52	80.88	81.53	81.41	95.49
ResNet	81.64	84.99	80.55	84.02	81.64	81.36	81.65	82.33	97.85
Our model	82.98	84.96	81.06	84.67	83.03	85.45	84.17	83.76	98.96

As shown in Table 1, deep NNs clearly outperformed shallow ones in the detection of surface defects on cement concrete bridges. Among the deep NNs, our model achieved better detection results than AlexNet, GoogLeNet, and ResNet. Hence, our model is highly suitable and robust for detecting the surface defects on cement concrete bridges.

### 6. Conclusions

This paper mainly develops a surface defect detection model for cement concrete bridges based on transfer learning and the convolutional neural network (CNN). The collected images were preprocessed by a morphology-based weight adaptive denoising method. The VGG-16, which is a classic CNN, was fine-tuned by reducing the three fully connected layers into two and making the Softmax layer

with seven tags as the final classification layer. The fine-tuned model was trained and tested with characteristic images of seven classes of defects including normal, cracks, fracturing, plate fracturing, corner rupturing, edge/corner exfoliation, skeleton exposure, and repairs. Lastly, our model was compared with other NNs, using the same ratio between training and test sets. The comparison demonstrates the excellence of our model in defect detection, generalization, and robustness.

**Author Contributions:** Conceptualization, J.Z. and J.S.; methodology, J.Z.; writing—original draft preparation, J.Z.; writing—review and editing, J.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** The National Key Research and Development Program of China (Grant No.: 2018YFB1600300 and 2018YFB1600301), the National Science Foundation of China (Grant No.: 51578370), and the Transportation Science and Technology Development Planning Project, Tianjin, China (Grant No.: G2018-29) supported this work. Any opinions, findings, conclusions, and recommendations mentioned in this work are of the authors and do not necessarily reflect those of the sponsors.

**Conflicts of Interest:** The authors declare there is no conflicts of interest regarding the publication of this paper.

## References

- Meng, W.L.; Mao, C.Z.; Zhang, J.; Wen, J.; Wu, D.H. A fast recognition algorithm of online social network images based on deep learning. *Traitement Signal* **2019**, *36*, 575–580. [[CrossRef](#)]
- Lakshmipathi, A.N.; Battula, B.P. Deep convolutional neural networks for product recommendation. *Ing. Syst. d'Inform.* **2018**, *23*, 161–172. [[CrossRef](#)]
- Neelapu, R.; Devi, G.L.; Rao, K.S. Deep learning based conventional neural network architecture for medical image classification. *Traitement Signal* **2018**, *35*, 169–182. [[CrossRef](#)]
- Teidj, S.; Khamlichi, A.; Driouach, A. Identification of Beam Cracks by Solution of an Inverse Problem. *Procedia Technol.* **2016**, *22*, 86–93. [[CrossRef](#)]
- Gopalakrishnan, K.; Khaitan, S.K.; Choudhary, A.; Agrawal, A. Deep Convolutional Neural Networks with transfer learning for computer vision-based data-driven pavement distress detection. *Constr. Build. Mater.* **2017**, *157*, 322–330. [[CrossRef](#)]
- Gulgec, N.S.; Takáč, M.; Pakzad, S.N. Structural damage detection using convolutional neural networks. In *Model Validation and Uncertainty Quantification*; Springer: Cham, Switzerland, 2017; Volume 3, pp. 331–337.
- Wang, K.C.; Zhang, A.; Li, J.Q.; Fei, Y.; Chen, C.; Li, B. Deep learning for asphalt pavement cracking recognition using convolutional neural network. In Proceedings of the International Conference on Highway Pavements and Airfield Technology 2017, Philadelphia, PA, USA, 27–30 August 2017; pp. 166–177. [[CrossRef](#)]
- Chen, F.C.; Jahanshahi, M.R. NB-CNN: Deep learning-based crack detection using convolutional neural network and Naïve Bayes data fusion. *IEEE Trans. Ind. Electron.* **2017**, *65*, 4392–4400. [[CrossRef](#)]
- Cha, Y.J.; Choi, W.; Suh, G.; Mahmoudkhani, S.; Büyüköztürk, O. Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types. *Comput. Aided Civ. Infrastruct. Eng.* **2018**, *33*, 731–747. [[CrossRef](#)]
- Wu, L.; Mokhtari, S.; Nazef, A.; Nam, B.; Yun, H.B. Improvement of crack-detection accuracy using a novel crack defragmentation technique in image-based road assessment. *J. Comput. Civ. Eng.* **2014**, *30*, 04014118. [[CrossRef](#)]
- Zhang, L.; Yang, F.; Zhang, Y.D.; Zhu, Y.J. Road crack detection using deep convolutional neural network. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 3708–3712.
- Pan, L.Q.; Tu, K.; Su, Z.P.; Yang, J.L.; Li, H.W. Crack detection in eggs using computer vision and BP neural network. *Trans. Chin. Soc. Agric. Eng.* **2007**, *5*, 028. [[CrossRef](#)]
- Fan, R.; Bocus, M.J.; Zhu, Y.; Jiao, J.; Wang, L.; Ma, F.; Chen, S.S.; Liu, M. Road Crack Detection Using Deep Convolutional Neural Network and Adaptive Thresholding. In Proceedings of the 2019 IEEE Intelligent Vehicles Symposium (IV), Paris, France, 9–12 June 2019.
- Ding, T.; Lu, W.; Zhang, C.; Du, J.; Ding, W.; Zhao, X. Eggshell Crack Identification Based on Welch Power Spectrum and Generalized Regression Neural Network (GRNN). *Food Sci.* **2015**, *36*, 156–160.
- Guan, J.; Shen, Z.; Xu, B.; Lu, J.; Ni, X. Surface crack detection for Al plate using the surface acoustic waves and neural network identification. *Lasers Mater. Process. Manuf. II* **2005**, *5629*, 466–474.

16. Yam, L.H.; Yan, Y.J.; Jiang, J.S. Vibration-based damage detection for composite structures using wavelet transform and neural network identification. *Compos. Struct.* **2003**, *60*, 403–412. [[CrossRef](#)]
17. Peng, H.; Wen, Y.X.; Wang, Q.H.; Wang, S.C.; Wu, L.L. Crack Detection in Eggs with Multi-level Wavelet Transform and BP Neural Network. *Trans. Chin. Soc. Agric. Mach.* **2009**, *40*, 170–174.
18. Sharif Razavian, A.; Azizpour, H.; Sullivan, J.; Carlsson, S. CNN features off-the-shelf: An astounding baseline for recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Columbus, OH, USA, 24–27 June 2014; pp. 806–813.
19. Soukup, D.; Huber-Mörk, R. Convolutional neural networks for steel surface defect detection from photometric stereo images. In *International Symposium on Visual Computing*; Springer: Cham, Switzerland, 2014; pp. 668–677.
20. Qiao, L.; Zhao, E.; Liu, J.; Cheng, B. Research on Workpiece Defect Detection Method Based on CNN. *Comput. Sci.* **2017**, *S2*, 248–253.
21. Shang, L.; Yang, Q.; Wang, J.; Li, S.; Lei, W. Detection of rail surface defects based on CNN image recognition and classification. In Proceedings of the 20th International Conference on Advanced Communication Technology (ICACT), Chuncheon-si Gangwon-do, Korea, 11–14 February 2018; pp. 45–51.
22. Yang, Y.X.; Zhao, M. CNN-based strip steel surface defect detection. *Heavy Mach.* **2019**, *2*, 25–29.
23. Prasanna, P.; Dana, K.J.; Gucunski, N.; Basily, B.B.; La, H.M.; Lim, R.S.; Parvardeh, H. Automated crack detection on concrete bridges. *IEEE Trans. Autom. Sci. Eng.* **2014**, *13*, 591–599. [[CrossRef](#)]
24. German, S.; Brilakis, I.; DesRoches, R. Rapid entropy-based detection and properties measurement of concrete spalling with machine vision for post-earthquake safety assessments. *Adv. Eng. Inform.* **2012**, *26*, 846–858. [[CrossRef](#)]
25. Chen, F.; Liu, M.Y.; Kao, C.C.; Lee, T.Y. Deep active learning for civil infrastructure defect detection and classification. *Comput. Civ. Eng.* **2017**, 298–306.
26. You, Y.; Lu, C.; Wang, W.; Tang, C.K. Relative CNN-RNN: Learning relative atmospheric visibility from images. *IEEE Trans. Image Process.* **2018**, *28*, 45–55. [[CrossRef](#)]
27. Imani, M.; Ghoreishi, S.F.; Braga-Neto, U.M. Bayesian control of large MDPs with unknown dynamics in data-poor environments. Proceedings of Advances in Neural Information Processing Systems, Montréal, QC, Canada, 2–8 December 2018; pp. 8146–8156.
28. Xie, S.; Imani, M.; Dougherty, E.R.; Braga-Neto, U.M. Nonstationary linear discriminant analysis. In Proceedings of the 51st Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, USA, 29 October–1 November 2017; pp. 161–165.
29. Imani, M.; Braga-Neto, U.M. Finite-horizon LQR controller for partially-observed Boolean dynamical systems. *Automatica* **2018**, *95*, 172–179. [[CrossRef](#)]
30. Imani, M.; Dougherty, E.R.; Braga-Neto, U. Boolean Kalman filter and smoother under model uncertainty. *Automatica* **2020**, *111*, 108609. [[CrossRef](#)]

