

Article

# Applying Deep Learning to Continuous Bridge Deflection Detected by Fiber Optic Gyroscope for Damage Detection

Sheng Li <sup>1,\*</sup>, Xiang Zuo <sup>2</sup>, Zhengying Li <sup>2</sup> and Honghai Wang <sup>1</sup>

<sup>1</sup> National Engineering Laboratory for Fiber Optic Sensing Technology, Wuhan University of Technology, Wuhan 430070, China; wanghh@whut.edu.cn

<sup>2</sup> School of Information Engineering, Wuhan University of Technology, Wuhan 430070, China; zuoxiang@whut.edu.cn (X.Z.); zhyli@whut.edu.cn (Z.L.)

\* Correspondence: lisheng@whut.edu.cn

Received: 4 January 2020; Accepted: 6 February 2020; Published: 8 February 2020



**Abstract:** Improving the accuracy and efficiency of bridge structure damage detection is one of the main challenges in engineering practice. This paper aims to address this issue by monitoring the continuous bridge deflection based on the fiber optic gyroscope and applying the deep-learning algorithm to perform structural damage detection. With a scale-down bridge model, three types of damage scenarios and an intact benchmark were simulated. A supervised learning model based on the deep convolutional neural networks was proposed. After the training process under ten-fold cross-validation, the model accuracy can reach 96.9% and significantly outperform that of other four traditional machine learning methods (random forest, support vector machine, k-nearest neighbor, and decision tree) used for comparison. Further, the proposed model illustrated its decent ability in distinguishing damage from structurally symmetrical locations.

**Keywords:** bridge damage detection; fiber optic gyroscope; deep learning; convolutional neural network

## 1. Introduction

Dynamic modal analysis has been the most commonly used approach for structural damage detection in civil engineering [1–3]. The use of wavelet, Hilbert–Huang transform, and other signal processing methods are also the conventional choices for structural damage detection that directly analyze the perturbation of vibration signals [4]. Various structural non-destructive testing approaches [5–7] are also significant means for detecting structural damage. Over the last decade, machine-learning algorithms have been used to address a wide range of vibration-based damage detection problems [8,9]. Although most of these techniques are based on vibration responses and such approaches still dominate the diagnosis and prognosis of structural health monitoring [10], feature extraction processes heavily relying on handcrafted intervention prior to damage classification [11] have often become major challenges that limit the effectiveness of various methods.

With the ability of automatic feature extraction and classification, deep convolutional neural networks (CNN) have been explored to address the range of difficulties in such following areas as computer vision [12,13], speech recognition [14], natural language processing [15], medical image processing [16], pathological signal classification [17,18], mechanical fault diagnosis [19–21], impact evaluation of natural disasters on infrastructure systems [22], and structural damage detection [23–25].

Most of the research efforts on deep CNN-based structural damage detection are essentially associated with the supervised learning processes. In this emerging area, Cha et al. [23] pioneered the deep CNN study of damage detection for cracks in concrete structures, and subsequently, Cha et al. [26]

further expanded the detection objectives of structural damage based on the faster Region-CNN (R-CNN). The recent study based on R-CNN to quantify the identified concrete spalling damage in terms of volume was reported in [27]. Xue and Li [28] established a fully convolutional neural networks model to classify the concrete tunnel lining defects. Other image-based researches on structural damage detection using deep learning were reported in [29–31]. In addition to two-dimensional convolution operations on structural images, one-dimensional convolution operations which usually spend a considerably cheaper computational cost than that of recurrent neural networks [32], are employed by researchers to perform the signal-based structural damage detection. The structural vibration signal as a typical type of one-dimensional time series [33] data is used to perform deep CNN-based structural damage detection. For instance, Abdeljaber et al. [18] proposed a method for detecting structural damage using one-dimensional CNN for multi-nodal vibration testing of steel frames. Lin et al. [25] simulated the vibration response of simply supported beams under various damage scenarios and proposed a procedure for detecting the categories of structural damage using the one-dimensional CNN. Huang et al. [34] analyzed the mechanical operation process through vibration signal by constructing a one-dimensional CNN.

Although deep CNN technology to some extent relieves the heavy pre-processing on the raw data or feature crafting for the damage detection when using vibration signals, the analysis of bridge damage detection is more complex comparing to that of simple structures, which needs more support of structural responses. In other words, compared with the amount of the degree of structural freedom, the scale of available vibration sensors used for bridge structural damage detection are often finite or even insufficient. To obtain as much structural dynamic information as possible in the case of limited measurement points, sensor optimization layout [35] is generally considered, which results in a decrease in damage detectability of complex structures. Therefore, a novel type of structural response which can easily cover the whole and local test requirement and provide enough structural information for the analysis of damage detection by using deep CNN should be attempted and explored. In this paper, we aimed at the multi-dimensional type of signal and chose a test technique for continuous curve mode of deformation based on fiber optic gyroscope (FOG) to produce continuous deflection of bridge. Detailed fundamental principles of the FOG-based testing technique were reported in [36–38]. A corresponding sample set based on supervised learning techniques was established, and a specific one-dimensional CNN model was proposed to automate feature extraction and classification. Specifically, the scheme procedure for the production of structural damage scenarios based on deformation responses was elaborated. The deformation responses and corresponding output labels were established through data augmentation and pre-processing. Furthermore, architectures and algorithms of the proposed one-dimensional CNN, and the partition rules of the dataset used for method verification were discussed. Finally, the performance of the proposed approach was compared with the results of other pattern recognition methods, all of which were conducted under the ten-fold cross-validation [39].

## 2. Design and Implementation of Structural Damage Scenarios

### 2.1. Experimental Platform and Instrumentation

A scale-down model of cable-stayed bridge was used as the experimental platform to represent the responses due to the simulation damage. The model shown in Figure 1 with the main span of 9.7 m, tower height of 3.46 m, deck width of 0.55 m, and 56 stay cables, was manufactured at a scale of 1:40. Figure 2 further illustrated a structure diagram and a physical structure of the device dedicated to measuring continuous deflection of the bridge model, which was triggered by a remote controller to perform mobile measurement. When the test device [36] moved along the surface of the bridge model, data of continuous deflection can be collected at sampling rate 150 Hz. Since the measurement period based on the motion carrier was relatively short, the continuous deflection obtained chronologically at each time was regarded as a multi-dimensional variable acquired at the

same moment, and the deflection of main span was chosen as the structural deformation response used for subsequent analysis.



Figure 1. Experimental platform for testing continuous deflection.

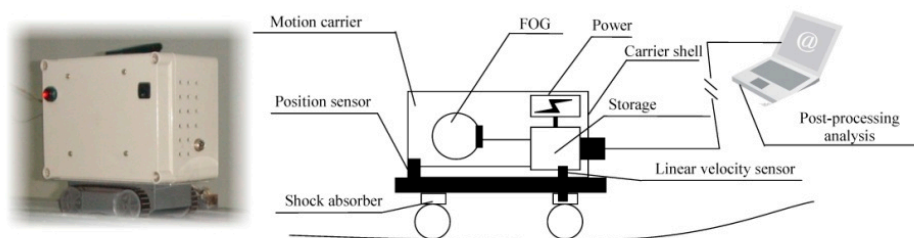


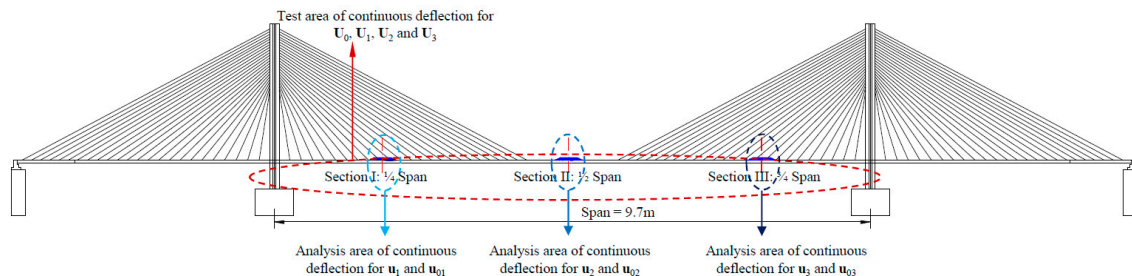
Figure 2. Measuring device integrated in a motion carrier.

## 2.2. Experimental Design and Procedures

The change in structural geometry can reflect a certain degree of transformation of interior mechanical properties. Further, structural damage is one reason for the change of interior mechanical properties of structure. Therefore, the different damage scenarios of the structure theoretically have corresponding structural deformation states. Continuous deflection can provide the dense deformation information, which can present more abundant structural response information than other finite point-based geometry measurement methods [40–42]. In the context of an experiment based on supervised learning, it was assumed that the change in the continuous deflection of bridge was only due to the result of structural damage. A metal pad (42.8L × 12.8W × 0.2H cm) with slope at both ends was used to simulate structural deformation caused by damage rather than physically destroying the structure [43]. The pad as an obstacle was placed on the movement path of the measuring device to simulate the deformation caused by structural damage. Compared with the situation without the pad, the measuring device can capture responses of the continuous deflection of bridge under the disturbance of the pad. This localized continuous deflection caused by the influence of the pad was clearly the most important of the continuous deflection of the entire bridge. Using such local responses instead of the global deflections can undoubtedly simplify the training process of the following supervised learning algorithm.

By this way, as shown in Figure 3, when the pad was not placed, the corresponding continuous deflection of the bridge was defined as  $U_0$ . For each of the three damage scenarios, one pad was placed at a position each time, and therefore,  $U_1$ ,  $U_2$ , and  $U_3$  can be obtained. Here,  $U_0$ ,  $U_1$ ,  $U_2$ , and  $U_3$  as raw data of continuous deflection represented four types of simulated structure states, respectively. To improve the training efficiency and save the computational overhead of the supervised learning,  $U_1$ ,  $U_2$ , and  $U_3$  were truncated to  $u_1$ ,  $u_2$ , and  $u_3$ . Such truncated selection in the areas affected by the pad can be estimated through both the original testing curves and the dimension of pad in the context of experiment based on supervised learning. In the intact and three damage scenarios, the actual

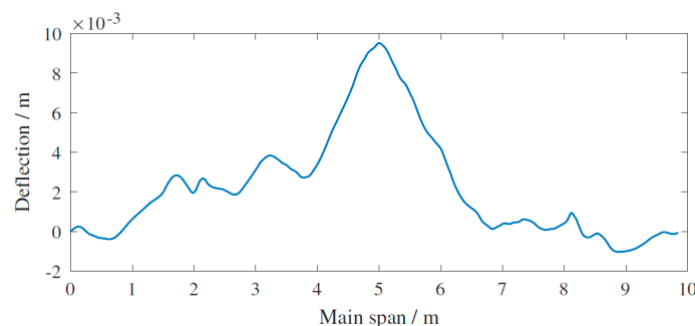
benchmarks of  $\mathbf{u}_1$ ,  $\mathbf{u}_2$ , and  $\mathbf{u}_3$  were  $\mathbf{u}_{01}$ ,  $\mathbf{u}_{02}$ , and  $\mathbf{u}_{03}$ , respectively. A common baseline for the three damage scenarios was defined to facilitate analysis. The weights of  $\mathbf{u}_{01}$ ,  $\mathbf{u}_{02}$ , and  $\mathbf{u}_{03}$  were regarded as equal and their average  $\mathbf{u}_0$  was designated as the nominal benchmark of  $\mathbf{u}_1$ ,  $\mathbf{u}_2$  and  $\mathbf{u}_3$ . The following work utilized  $\mathbf{u}_0$ ,  $\mathbf{u}_1$ ,  $\mathbf{u}_2$ , and  $\mathbf{u}_3$  to conduct the damage detection based on deep CNN algorithm.



**Figure 3.** Locations of the metal pad used to assist in simulated damage scenarios.

### 2.3. Raw Samples of Continuous Deflection of Bridge

The spatial resolution in motion direction refers to the adjacent sampling interval of the device in Figure 2. This parameter is determined by the wheel diameter and the reticle of the rotational speed code wheel, and is approximately 1.48 mm. For the continuous deflection of the main span, taking intact scenario for an example shown in Figure 4, the deformation response of  $\mathbf{U}_0$  consisted of sequence data of 6554 dimensions which depicted the length of main span of 9.7 m. Due to the high spatial resolution, the continuous curve clearly reflected a certain degree of pre-camber applied at the main span. Moreover, the continuous curve revealed that the experiment platform did not exhibit completely symmetrical structural deformation owing to the handcrafted control for the cable force.



**Figure 4.** Deformation of main span depicted by continuous deflection.

The local continuous deflection curves of  $\mathbf{u}_0$ ,  $\mathbf{u}_1$ ,  $\mathbf{u}_2$  and  $\mathbf{u}_3$  were shown in Figure 5. Each type of the local continuous deflection curve contained 390 dimensions of sequence data. The coverage length of the area affected by the pad was considered to be the primary basis for determining the length of the local continuous deflection. Moreover, through preliminary data observation, the length of the region having the largest influence range among the three disturbance positions of the pad was selected, rounded, and defined as the final truncated length, which guaranteed the consistency of multiple sets of sample dimensions. The continuous curve mode test technique was used to separately collect the structural response of the scale-down bridge model under intact and simulated structural damage, and five groups of  $\mathbf{U}_0$ ,  $\mathbf{U}_1$ ,  $\mathbf{U}_2$ , and  $\mathbf{U}_3$ , were collected, respectively. Therefore, five groups of  $\mathbf{u}_0$ ,  $\mathbf{u}_1$ ,  $\mathbf{u}_2$ , and  $\mathbf{u}_3$  corresponded to four types of structural conditions, namely intact, damage $_{1/4}$ , damage $_{1/2}$ , and damage $_{3/4}$ , and these were used as raw samples to conduct the following study.

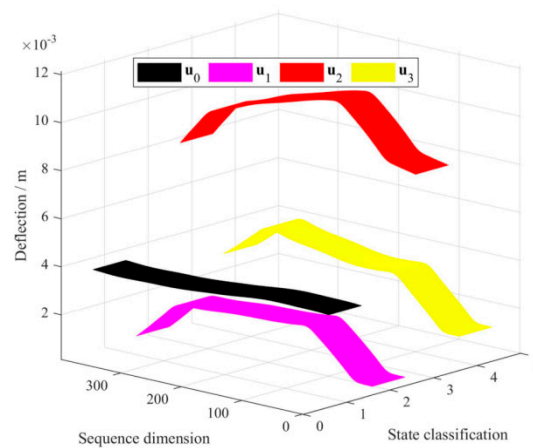


Figure 5. Truncated deformations depicted by continuous deflections.

### 3. Detection Methodology Based on Deep CNN

#### 3.1. Data Augmentation and Pre-processing

Data augmentation and pre-processing are two essential tasks before carrying out deep learning. The former is always the first choice to boost the performance of a deep network. For image recognition based on deep CNN, there are a wide range of ways to perform data augmentation [12,44,45]. However, the above approaches are not suitable for signal-based pattern classification when using deep CNN algorithm. As shown in Figure 6a, dividing the raw acquisition signals into the same sub-fragment directly is common means of data augmentation [24,25]. It can be seen from Figure 6b that, for fragments of the same length as that in Figure 6a, the overlapping zone set in the adjacent fragments causes the amount of the fragment  $m$  to be larger than  $n$  shown in Figure 6a, which effectively increases the amount of data size.

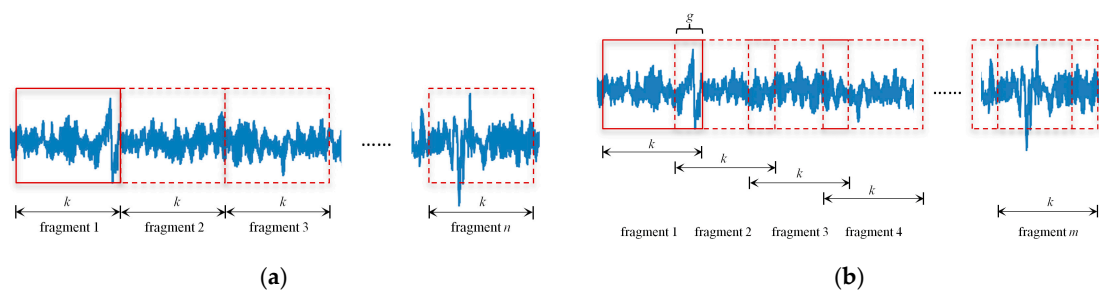
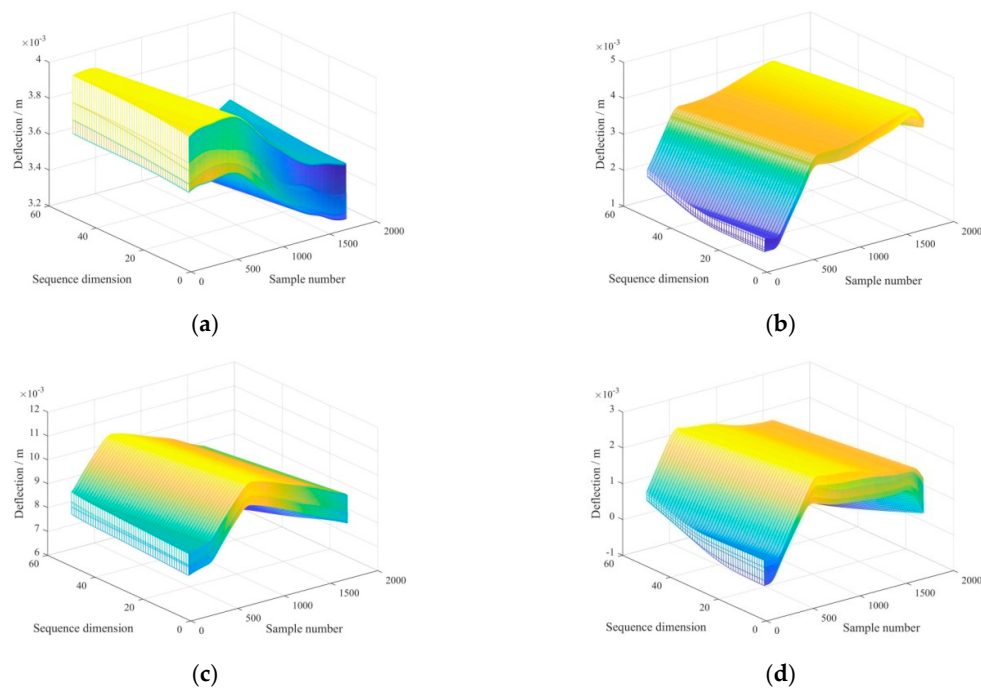


Figure 6. Data augmentation of (a) common means and (b) adopted operation.

Since the original experimental samples were small, the overlapping zone was taken as  $g = 1$ . It was obvious that the larger the value of  $k$  in Figure 6, the smaller the number of fragments after data augmentation and vice versa, which also indicated the greater number of fragments needed more computational overhead of model training. With the consideration of a tradeoff result between the training objective of model and the computational overhead, the length of fragment was set as  $k = 50$ , followed by the 390-dimensional original sequence becoming 341 50-dimensional sequence samples. For the five raw groups of  $\mathbf{u}_0$ ,  $\mathbf{u}_1$ ,  $\mathbf{u}_2$ , and  $\mathbf{u}_3$ , after data augmentation, the sample set of  $\mathbf{u}'_0$ ,  $\mathbf{u}'_1$ ,  $\mathbf{u}'_2$ , and  $\mathbf{u}'_3$ , each including 1705 samples, corresponding to Figure 7a–d, were shown by mesh graphics.





**Figure 7.** Sample set represented by mesh graphics through data augmentation for (a) intact, (b) damage<sub>1/4</sub>, (c) damage<sub>1/2</sub>, and (d) damage<sub>3/4</sub>.

To eliminate the difference in the deflection amplitudes of four types of samples in Figure 7 and boost a better classification effect [46,47], a type of min-max normalization [48] expressed in Equation (1) is used to normalize all the amplitudes to the range of 0~1.

$$\mathbf{u}_i'' = \frac{\mathbf{u}_i' - \min(\mathbf{u}_i')}{\max(\mathbf{u}_i') - \min(\mathbf{u}_i')}, (i = 0, 1, 2, 3 \dots) \quad (1)$$

As shown in Table 1, raw and truncated represented the continuous deflections of the test area and analysis area shown in Figure 3, respectively. After data augmentation and normalization based on the truncated stage, each category of the four-dataset including intact and three types of simulated damage contained 1705 samples. The four types of state, namely,  $\mathbf{u}_0''$ ,  $\mathbf{u}_1''$ ,  $\mathbf{u}_2''$ , and  $\mathbf{u}_3''$  were used as input data, in which  $\mathbf{u}_0''$  represented the intact baseline and the rest three represented different damage scenarios. One-hot form was used to describe the output labels corresponding to the four categories, meaning that the label vector was generated by the rule that the vector had all zero elements except the position  $j$ , where  $j$  was the type number of structural state.

**Table 1.** The dataset details of training and test.

Processing Stage	Variable				Dimension of Each Variable	Total Samples
Raw	$\mathbf{U}_0$	$\mathbf{U}_1$	$\mathbf{U}_2$	$\mathbf{U}_3$	6554	4 × 5 = 20
Truncated	$\mathbf{u}_0$	$\mathbf{u}_1$	$\mathbf{u}_2$	$\mathbf{u}_3$	390	
Augmentation	$\mathbf{u}'_0$	$\mathbf{u}'_1$	$\mathbf{u}'_2$	$\mathbf{u}'_3$	50	4 × 1705 = 6820
Normalization	$\mathbf{u}''_0$	$\mathbf{u}''_1$	$\mathbf{u}''_2$	$\mathbf{u}''_3$		

The entire measurement process was performed under stable temperature field, and the data of this study came from actual measurements, which already contained noise disturbances existing in the indoor environment. Therefore, extra interferences of simulated noise and temperature effect were not further considered here.

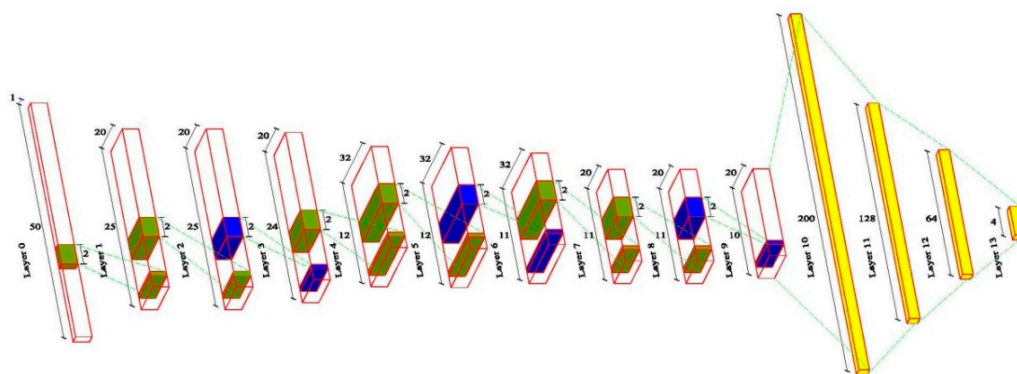
### 3.2. Descriptions of the Proposed CNN Architecture

Table 2 gave the details of the proposed CNN structure through the trial-and-error under the current computing resource configuration. The model structure was inspired by Cifar-10 [49], in which operations of convolution and pooling were not pairwise used. Figure 8 showed the graphical representation of CNN structure with 50 input sample lengths where the green, blue, and yellow referred to the kernel size, max-pooling, and fully connected layer, respectively.

**Table 2.** The details of CNN structure.

Layers	Type	No. of Neurons (Output Layers)	Kernel Size	Stride	Padding	Activation
0-1	Convolution	25 × 20	2	2	Same	PReLU
1-2	Convolution	25 × 20	2	1	Same	PReLU
2-3	Max-pooling	24 × 20	2	1	Valid	—
3-4	Convolution	12 × 32	2	2	Same	PReLU
4-5	Convolution	12 × 32	2	1	Same	PReLU
5-6	Max-pooling	11 × 32	2	1	Valid	—
6-7	Convolution	11 × 20	2	1	Same	PReLU
7-8	Convolution	11 × 20	2	1	Same	PReLU
8-9	Max-pooling	10 × 20	2	1	Valid	—
9-10	Flatten	200	—	—	—	—
10-11	Dense	128	—	—	—	PReLU
11-12	Dense with dropout	64	—	—	—	PReLU
12-13	Dense	4	—	—	—	Softmax

Note: PReLU—parametric rectified linear unit.



**Figure 8.** The proposed deep CNN architecture.

Layer 0 as the input layer in Figure 8 was convolved with a kernel of size 2 to produce Layer 1. The convolution and cross-correlation were used interchangeably in deep learning [50], which can be described as:

$$f(i) = \sum_{n=1}^N s(i+n)k(n) \quad (2)$$

where  $s$  is input signal,  $k$  is filter, and  $N$  is the number of elements in  $s$ . The output vector  $f$  is the cross-correlation of  $s$  and  $k$ . Next, Layer 1 was convolved with a same kernel size to produce Layer 2. After two times of convolution, a max-pooling of size 2 was applied to every feature map (Layer 3). By repeating the above operations two times, other four convolutional layers and two max-pooling layers were created. In Layer 9, the neurons were then fully connected to 200 neurons in Layer 10 by flatten. Eventually, Layer 10 was fully connected to 128 neurons in Layer 11 and Layer 11 was fully connected to 64 neurons in Layer 12. Finally, Layer 12 was connected to the last layer (Layer 13) with 4 output neurons which represented intact, damage<sub>1/4</sub>, damage<sub>1/2</sub> and damage<sub>3/4</sub>.

Because the gradient of the left side of rectified linear unit (ReLU) [51] as shown in (3) is always zero, the activation operation may become invalid during training process if the weights updated by a large gradient become zero after being activated.

$$f(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (3)$$

The Leaky ReLU method [52] is a good alternative to address such problem by considering a parameter  $\alpha$  in (4),

$$f(x) = \begin{cases} x, & x \geq 0 \\ \alpha x, & x < 0 \end{cases} \quad (4)$$

where  $\alpha$  is usually set to a small number, and once  $\alpha$  is set, its value will keep constant. This allows a small, non-zero gradient when the unit is not active. The parametric rectified linear unit (PReLU) [53] which has the same mathematical expression to Leaky ReLU, takes this idea further by making the coefficient  $\alpha$  into a parameter that is learnt along with the other neural network parameters. Since it was not necessary to consider how to specify  $\alpha$ , PReLU was used to take the place of Leaky ReLU in this work as an activation function for the convolutional layers (1, 2, 4, 5, 7 and 8) and two fully connected layers (11 and 12).

Further, the Softmax function was used to compute the probability distribution of the four output classes, which can be expressed as follows:

$$p_k = \frac{e^{x_k}}{\sum_{i=1}^n e^{x_i}} \quad (5)$$

where  $x_k$  is the input of last layer,  $n$  is the number of output nodes and output values of  $p_k$  are between 0 and 1 and their sum equals to one. Equation (5) was used for Layer 13 in Figure 8 to predict which category the input signals (intact, damage<sub>1/4</sub>, damage<sub>1/2</sub>, or damage<sub>3/4</sub>) belonged to.

Compared with shallow neural networks, deep CNN as a more complicated model contains more hidden layers and more weights, and is particularly prone to overfitting. In the proposed deep CNN, a dropout rate of 0.35 was used before the classification layer (Layer13) as shown in Table 2, which together with early stopping [54] mentioned in the following, effectively suppressed incidence of overfitting during all training processes.

### 3.3. Training Setting

Ten percent of the total dataset was used for test, while the rest 90% was divided into two parts, namely, training (80%) and validation (20%). The reason for validation was to evaluate the performance of the model for each epoch and prevent overfitting.

Because the cross entropy function is much more sensitive to the error, the learning rules derived from the cross entropy function generally yield better performance. Here, categorical cross entropy was used as the objective function to estimate the difference between original and predicted damage types, expressed as follows:

$$J = \sum_{i=1}^k [-d_i \ln(y_i) - (1 - d_i) \ln(1 - y_i)] \quad (6)$$

where  $J$  is the cross entropy,  $y_i$  is the output of prediction class,  $d_i$  is the original class in the training data, and  $k$  is the number of output nodes.

To minimize the above objective function, adaptive moment estimation (Adam) was selected as the optimization algorithm. It calculated an adaptive learning rate for each parameter and stored both an exponentially decaying average of past squared gradients and an exponentially decaying average



of past gradients [55]. Details about the training parameters in this work are given in Table 3, in which the early stopping technique was used to control training epochs and further avoid overfitting, and the parameters set in Adam were based on the suggestion in [56].

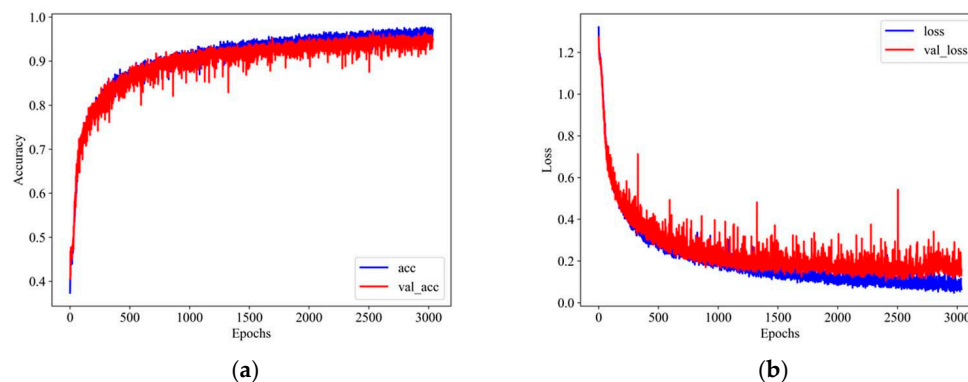
**Table 3.** The training parameters of CNN structure in this work.

Batch Size	Epoch	Patience in Earliestopping	Adam			
			Initial Learning Rate	$\beta_1$	$\beta_2$	$\epsilon$
128	5000	500	0.001	0.9	0.009	$1.0 \times 10^{-8}$

Moreover, a ten-fold cross-validation approach was used in this study, the purpose of which was to reduce the sensitivity of algorithm performance to data partitioning and to obtain as much valid information as possible from the enhanced data. First, all the prepared dataset was randomly divided into ten equal parts. Nine out of ten parts of the total were used to train the proposed deep CNN while the remaining one-tenth dataset were used to test the performance of the model. This strategy was repeated ten times by shifting the training and test dataset. The accuracies reported in the paper were the average values obtained from ten evaluations.

#### 4. Results and Discussion

The proposed CNN model was implemented by Python package Tensorflow and Keras [57]. The average training runtime of each fold for the proposed model was approximately 15 minutes, which was run on a GPU core (GTX 1080 Ti) with twelve 2.20 GHz processors (Intel Xeon E5-2650 v4). According to the setting in Table 3, the training processes showed that in the initial 500 epochs, the convergence speed was rather quickly for all of the dataset from the ten-fold cross-validation, but it still needed approximately 3000 to 4500 epochs to reach the best performance based on the patience rule set in early stopping. The typical training process regarding accuracy and loss represented by fold 2 is shown in Figure 9, which stopped at the epochs of 3036.



**Figure 9.** Typical training processes about (a) accuracy and (b) loss.

The confusion matrix cross all ten-fold was presented in Figure 10a. It was observed that 98.3% of  $\mathbf{u}_0''$  signals were correctly classified as intact. Moreover, 1.7% of  $\mathbf{u}_0''$  were erroneously classified as other damage categories. Further, a high percentage of 98.4% of  $\mathbf{u}_1''$  signals were correctly classified as damage<sub>1/4</sub> with 1.3% of  $\mathbf{u}_1''$  wrongly classified as damage<sub>3/4</sub>. For  $\mathbf{u}_2''$  the accuracy rate for damage<sub>1/2</sub> reached 96.8% with 2.9% of  $\mathbf{u}_2''$  wrongly predicted as damage<sub>3/4</sub>. Similarly, 94.2% of  $\mathbf{u}_3''$  signals were correctly classified as damage<sub>3/4</sub> with 5.8% wrongly classified as intact (1.1%), damage<sub>1/4</sub> (1.9%), and damage<sub>1/2</sub> (2.8%).

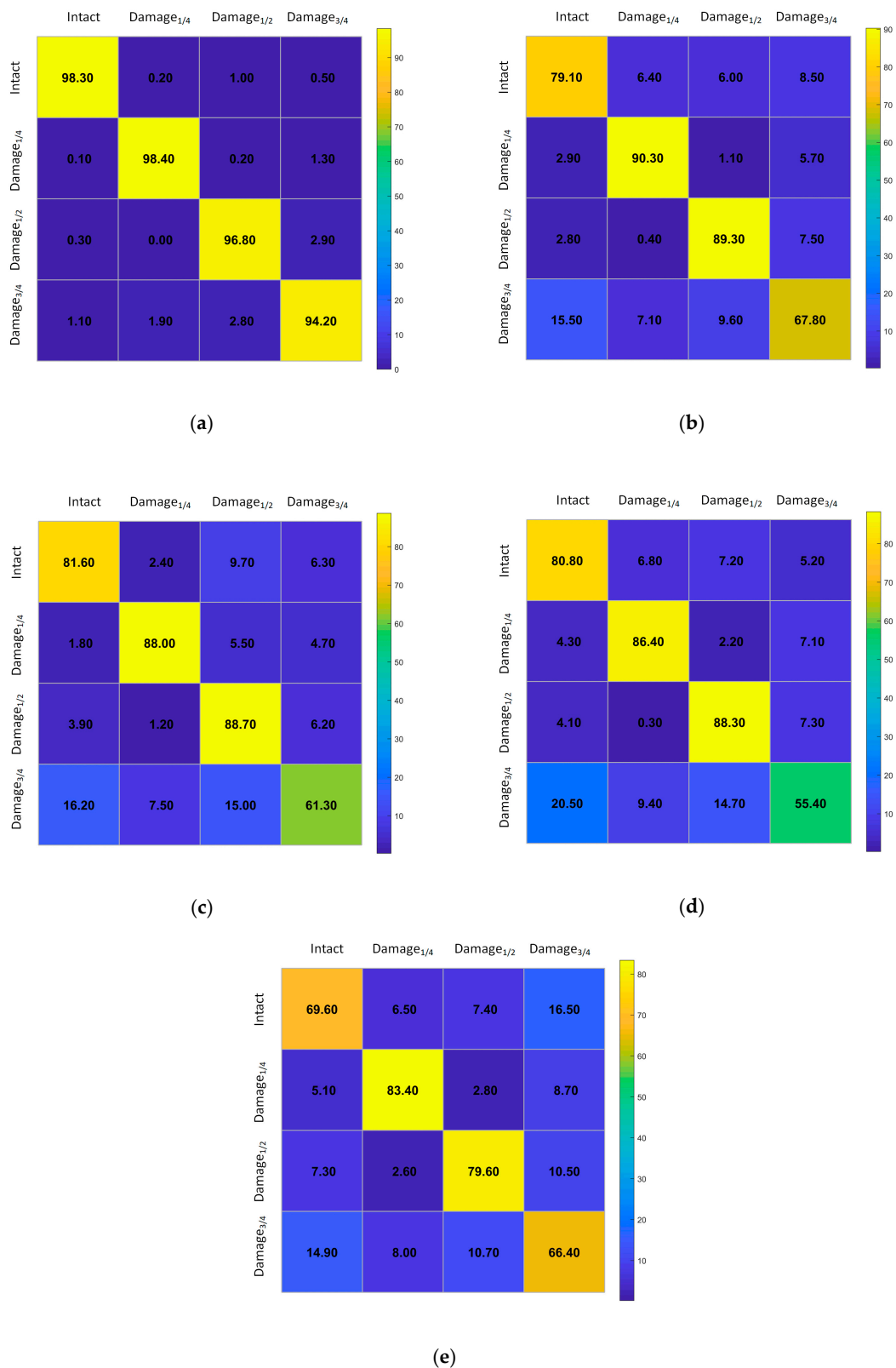
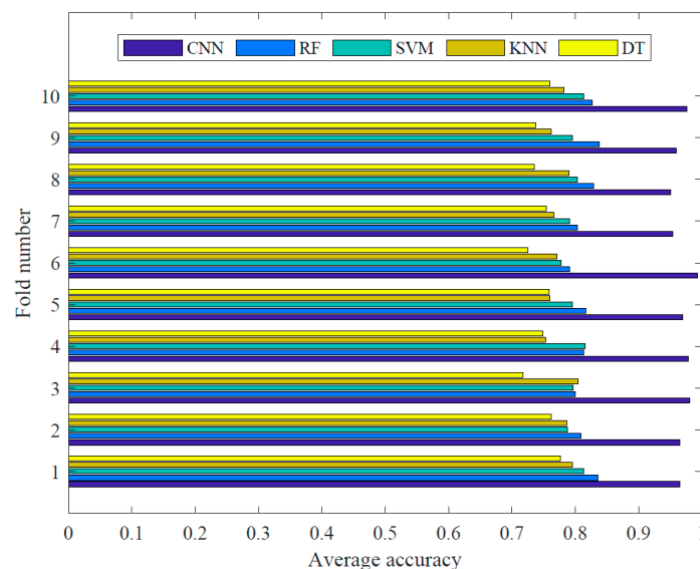


Figure 10. The confusion matrices of (a) CNN, (b) random forest, (c) support vector machine, (d) k-nearest neighbor, and (e) decision trees.

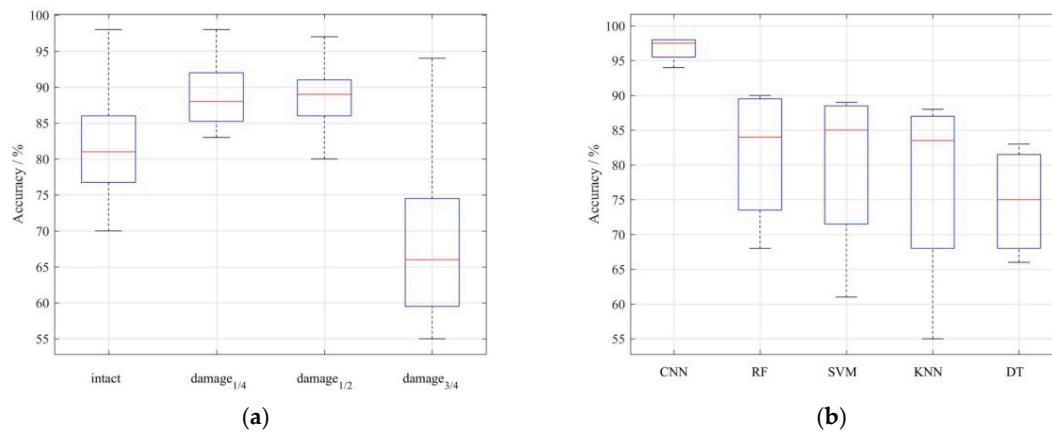
Furthermore, to evaluate the capability in each fold of cross-validation, average accuracy results shown in Figure 11 for different classes were compared between the proposed model and other

four pattern recognition methods. When the samples were directly used to classify without heavy consideration regarding features extraction, the accuracy of automatic detection for proposed CNN model (96.9%) was obviously better than that of random forest (RF) (81.6%), support vector machine (SVM) (79.9%), k-nearest neighbor (KNN) (77.7%), and decision trees (DT) (74.8%). Here, the allocation of dataset of the four comparison methods was consistent with the proposed deep CNN algorithm. To fully compete with the proposed model, the most decent key hyperparameters set in sklearn [58] for RF, SVM, KNN and DT were derived through trial-and-error. To further quantify the effect of classifiers, Figure 10b–e show confusion matrices of the other four methods, respectively. It was observed that the best accuracy in various comparison methods can reach to 90.3% as shown in Figure 10b, which was still inferior to the lowest accuracy 94.2% as shown in Figure 10a.



**Figure 11.** Comparisons of average accuracy in each fold of cross-validation.

Next, as shown in Figure 12a, for all five methods, the classification effects on damage<sub>1/4</sub> obviously outperformed the results of the other three categories. Moreover, the detection results of damage<sub>3/4</sub> were the worst in all methods, having a direct influence on the average accuracy of various ways. Further, as shown in Figure 12b, the classification imbalance presented in the confusion matrix was most severe when KNN was used as the classifier. This phenomenon may be related to the relatively lower algorithm complexity of KNN [59,60] compared with other methods mentioned in this work. Only the proposed approach based on deep CNN effectively mitigated this imbalance, although the accuracy of damage<sub>3/4</sub> in Figure 10a was still slightly less than the other three classes. The limited data samples should be a major aspect for such imbalance. In addition, current tests were all from the one-way results and lack of the data from the opposite direction. This may introduce a cumulative system error to the results of the structural response. Further, only slight pre-processing was carried out for the original dataset, which reduced the learning ability of each method mentioned in this paper. Actually, as shown in Table 4, the other four machine learning methods for comparison had fewer key parameters to consider in the process of balancing training accuracy and training error than the proposed method. This weak complexity, determined by the principles of the algorithm, resulted in a poor predictive effect on training and validation. Therefore, under the same circumstance, the proposed approach clearly demonstrated better overall performance in automatic feature extraction than other comparison means.



**Figure 12.** Accuracy distribution based on (a) damage category and (b) detection method.

**Table 4.** Key parameters set in the comparison methods.

RF	SVM	KNN	DT
FS criteria = gini Number of DT = 150	Kernel = rbf gamma = 10 C = 10	K = 6 DM = euclidean	FS criteria = entropy

Note: FS—Feature selection, gamma—The influence of kernel radius, C—The penalty parameter, K—k value defined in KNN, DM—Distance metric.

## 5. Conclusions

A deep learning CNN model with 11 trainable hidden layers was proposed to automatically extract and classify the bridge damage represented by the continuous deflection of bridge. Although current research on the use of FOG-based test technique to detect the damage of a scale-down bridge model through deep learning is just a pilot study, the following conclusions can be drawn:

- (1) In the case where it is easy to measure the FOG-based continuous deflection of the target structure, it is convenient to build structural deformation database that can provide sufficient training samples for deep learning-based damage detection.
- (2) Based on the data preparation strategies adopted in this work, one-dimensional convolution operation can effectively extract the detailed features of bridge deflection after a slight data pre-processing.
- (3) The deep CNN-based method as a classifier has at least 15.3% accuracy advantage over other traditional methods mentioned in this paper in distinguishing different types of bridge deformation modes.
- (4) Even if the same level structural damage occurs at a symmetrical position, the proposed method can still achieve satisfactory results with a deviation of only 4.2% for the recognition accuracy of damage at the symmetrical position.
- (5) For an actual bridge with a complete deformation monitoring database, the advantage of deep learning on automatic extracting of features of large-scale database can be exploited to search the damage or provide the preliminary diagnostic findings. Moreover, since the FOG-based measurement system has higher test accuracy for larger distributed deflections [61], the proposed method should be more suitable for long-span bridges.

**Author Contributions:** Data curation, X.Z.; Funding acquisition, S.L.; Methodology, S.L.; Project administration, Z.L.; Resources, H.W.; Supervision, H.W.; Writing—original draft, X.Z.; Writing—review & editing, S.L. All authors have read and agreed to the published version of the manuscript

**Funding:** This research was funded by the National Natural Science Foundation of China grant number 61875155.

**Acknowledgments:** The research work reported in this paper was supported by the National Engineering Laboratory for Fiber Optic Sensing Technology, Wuhan University of Technology.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Moughty, J.J.; Casas, J.R. A state of the art review of modal-based damage detection in bridges: Development, challenges, and solutions. *Appl. Sci.* **2017**, *7*, 510. [[CrossRef](#)]
2. Sarah, F.; Hejazi, F.; Rashid, R.S.M.; Ostovar, N. A review of dynamic analysis in frequency domain for structural health monitoring. *IOP Conf. Ser.: Earth Environ. Sci.* **2019**, *357*, 012007. [[CrossRef](#)]
3. Sabato, A.; Niezrecki, C.; Fortino, G. Wireless MEMS-based accelerometer sensor boards for structural vibration monitoring: A review. *IEEE Sens. J.* **2017**, *17*, 226–235. [[CrossRef](#)]
4. Amezcua-Sanchez, J.P.; Adeli, H. Signal processing techniques for vibration-based health monitoring of smart structures. *Arch. Comput. Method Eng.* **2016**, *23*, 1–15. [[CrossRef](#)]
5. Tao, Y.H.; Fitzgerald, A.J.; Wallace, V.P. Non-contact, Non-destructive testing in various industrial sectors with terahertz technology. *Sensors* **2020**, *20*, 712. [[CrossRef](#)]
6. Tashakori, S.; Baghalian, A.; Unal, M.; Fekrmandi, H.; Senyurek, V.Y.; McDaniel, D.; Tansel, I.N. Contact and non-contact approaches in load monitoring applications using surface response to excitation method. *Measurement* **2016**, *89*, 197–203. [[CrossRef](#)]
7. Tashakori, S.; Baghalian, A.; Senyurek, V.Y.; Unal, M.; McDaniel, D.; Tansel, I.N. Implementation of heterodyning effect for monitoring the health of adhesively bonded and fastened composite joints. *Appl. Ocean Res.* **2018**, *72*, 51–59. [[CrossRef](#)]
8. Farrar, C.R.; Worden, K. *Structural Health Monitoring: A Machine Learning Perspective*; John Wiley and Sons: New York, NY, USA, 2013; pp. 17–43.
9. Burgos, D.A.T.; Vargas, R.C.G.; Pedraza, C.; Agis, D.; Pozo, F. Damage identification in structural health monitoring: A brief review from its implementation to the use of data-driven applications. *Sensors* **2020**, *20*, 733. [[CrossRef](#)]
10. Webb, G.T.; Vardanega, P.J.; Middleton, C.R. Categories of SHM deployments: Technologies and capabilities. *J. Bridge Eng.* **2015**, *20*, 04014118. [[CrossRef](#)]
11. Pan, H.; Azimi, M.; Yan, F.; Lin, Z. Time-frequency-based data-driven structural diagnosis and damage detection for cable-stayed bridges. *J. Bridge Eng.* **2018**, *23*, 04018033. [[CrossRef](#)]
12. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. In Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS 2000), Lake Tahoe, NV, USA, 3–6 December 2012.
13. Koziarski, M.; Cyganek, B. Image recognition with deep neural networks in presence of noise-dealing with and taking advantage of distortions. *Integr. Comput. Aided Eng.* **2017**, *24*, 337–349. [[CrossRef](#)]
14. Abdel-Hamid, O.; Mohamed, A.; Jiang, H.; Penn, G. Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Kyoto, Japan, 25–30 March 2012.
15. Hu, B.; Lu, Z.; Li, H.; Chen, Q. Convolutional neural network architectures for matching natural language sentences. In Proceedings of the 28th Annual Conference on Neural Information Processing Systems (NIPS 2014), Montreal, QC, Canada, 8–13 December 2014.
16. Wang, G. A perspective on deep imaging. *IEEE Access* **2016**, *4*, 8914–8924. [[CrossRef](#)]
17. Acharya, U.R.; Fujita, F.; Oh, S.L.; Hagiwara, Y.; Tan, J.H.; Adam, M. Application of deep convolutional neural network for automated detection of myocardial infarction using ECG signals. *Inf. Sci.* **2017**, *415–416*, 190–198. [[CrossRef](#)]
18. Acharya, U.R.; Oh, S.L.; Hagiwara, Y.; Tan, J.H.; Adeli, H. Deep convolutional neural network for the automated detection and diagnosis of seizure using EEG signals. *Comput. Biol. Med.* **2018**, *100*, 270–278. [[CrossRef](#)] [[PubMed](#)]
19. Chen, Z.; Li, C.; Sanchez, R. Gearbox fault identification and classification with convolutional neural networks. *Shock Vib.* **2015**, *2015*. [[CrossRef](#)]
20. Guo, X.; Chen, L.; Shen, C. Hierarchical adaptive deep convolution neural network and its application to bearing fault diagnosis. *Measurement* **2016**, *93*, 490–502. [[CrossRef](#)]



21. Janssens, O.; Slavkovikj, V.; Vervisch, B.; Stockman, K.; Loccufier, M.; Verstockt, S.; Van de Walle, R.; Van Hoescke, S. Convolutional neural network based fault detection for rotating machinery. *J. Sound Vib.* **2016**, *377*, 331–345. [[CrossRef](#)]
22. Nabian, M.A.; Meidani, H. Deep learning for accelerated reliability analysis of transportation networks. *Comput. Aided Civ. Infrastruct. Eng.* **2018**, *33*, 443–458. [[CrossRef](#)]
23. Cha, Y.; Choi, W.; Büyüköztürk, O. Deep learning-based crack damage detection using convolutional neural networks. *Comput. Aided Civ. Infrastruct. Eng.* **2017**, *32*, 361–378. [[CrossRef](#)]
24. Abdeljaber, O.; Avci, O.; Kiranyaz, M.S.; Gabbouj, M.; Inman, D.J. Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks. *J. Sound Vib.* **2017**, *388*, 154–170. [[CrossRef](#)]
25. Lin, Y.; Nie, Z.; Ma, H. Structural damage detection with automatic feature-extraction through deep learning. *Comput. Aided Civ. Infrastruct. Eng.* **2017**, *32*, 1025–1046. [[CrossRef](#)]
26. Cha, Y.; 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](#)]
27. Beckman, G.H.; Polyzois, D.; Cha, Y. Deep learning-based automatic volumetric damage quantification using depth camera. *Autom. Constr.* **2019**, *99*, 114–124. [[CrossRef](#)]
28. Xue, Y.; Li, Y. A fast detection method via region-based fully convolutional neural networks for shield tunnel lining defects. *Comput. Aided Civ. Infrastruct. Eng.* **2018**, *33*, 638–654. [[CrossRef](#)]
29. Gao, Y.; Mosalam, K.M. Deep transfer learning for image-based structural damage recognition. *Comput. Aided Civ. Infrastruct. Eng.* **2018**, *33*, 748–768. [[CrossRef](#)]
30. Kang, D.; Cha, Y.J. Autonomous UAVs for structural health monitoring using deep learning and an ultrasonic beacon system with geo-tagging. *Comput. Aided Civ. Infrastruct. Eng.* **2018**, *33*, 885–902. [[CrossRef](#)]
31. Zhang, L.; Zhou, G.; Han, Y.; Lin, H.; Wu, Y. Application of Internet of Things technology and convolutional neural network model in bridge crack detection. *IEEE Access* **2018**, *6*, 39442–39451. [[CrossRef](#)]
32. Lipton, Z.C.; Berkowitz, J.; Elkan, C. A Critical Review of Recurrent Neural Networks for Sequence Learning. Available online: <https://arxiv.org/abs/1506.00019> (accessed on 10 December 2019).
33. Bagnall, A.; Lines, J.; Bostrom, A.; Large, J.; Keogh, E. The great time series classification bake off: A review and experimental evaluation of recent algorithmic advances. *Data Min. Knowl. Discov.* **2017**, *31*, 606–660. [[CrossRef](#)]
34. Huang, S.; Tang, J.; Dai, J.; Wang, Y. Signal status recognition based on 1DCNN and its feature extraction mechanism analysis. *Sensors* **2019**, *19*, 2018. [[CrossRef](#)]
35. Chang, M.; Pakzad, S. Optimal sensor placement for modal identification of bridge systems considering number of sensing nodes. *J. Bridge Eng.* **2014**, *19*, 04014019. [[CrossRef](#)]
36. Li, S.; Hu, W. A novel bridge curve mode measurement technique based on FOG. *Optik* **2015**, *126*, 3442–3445. [[CrossRef](#)]
37. Yang, D.; Wang, L.; Hu, W.; Zhang, Z.; Fu, J.; Gan, W. Singularity detection in FOG-based pavement data by wavelet transform. In Proceedings of the 25th International Conference on Optical Fiber Sensors, Jeju, Korea, 24–28 April 2017.
38. Li, L.; Tang, J.; Gan, W.; Hu, W.; Yang, M. The continuous line-shape measurement of bridge based on tri-axis fiber optic gyro. In Proceedings of the 25th International Conference on Optical Fiber Sensors, Jeju, Korea, 24–28 April 2017.
39. Zemmour, E.; Kurtser, P.; Edan, Y. Automatic parameter tuning for adaptive thresholding in fruit detection. *Sensors* **2019**, *19*, 2130. [[CrossRef](#)] [[PubMed](#)]
40. Liu, Y.; Deng, Y.; Cai, C.S. Deflection monitoring and assessment for a suspension bridge using a connected pipe system: A case study in China. *Struct. Control Health Monit.* **2015**, *22*, 1408–1425. [[CrossRef](#)]
41. Bogusz, J.; Figurski, M.; Grzegorz, N.; Marcin, S.; Wrona, M. GNSS-based multi-sensor system for structural monitoring applications. Proceedings of the Institute of Navigation International Technical Meeting 2012 (ITM 2012), Newport Beach, CA, USA, 30 January–1 February 2012.
42. Hou, X.; Yang, X.; Huang, Q. Using inclinometers to measure bridge deflection. *J. Bridge Eng.* **2005**, *10*, 564–569. [[CrossRef](#)]
43. Yang, D.; Wang, L.; Hu, W.; Ding, C.; Gan, W.; Liu, F. Trajectory optimization by using EMD- and ICA-based processing method. *Measurement* **2019**, *140*, 334–341. [[CrossRef](#)]

44. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2015), Boston, MA, USA, 7–12 June 2015.
45. Deep Image: Scaling up Image Recognition. Available online: <http://www.academia.edu/download/55673602/97789723e45650442cfaedff17e1c11a5080.pdf> (accessed on 16 September 2019).
46. Han, J.; Kamber, M.; Pei, J. *Data Mining: Concepts and Techniques*, 3rd ed.; Morgan Kaufmann: Waltham, MA, USA, 2011; pp. 83–124.
47. Jahan, A.; Edwards, K.L. A state-of-the-art survey on the influence of normalization techniques in ranking: Improving the materials selection process in engineering design. *Mater. Des.* **2015**, *65*, 335–342. [CrossRef]
48. Patro, S.G.K.; Sahu, K.K. Normalization: A Preprocessing Stage. Available online: <https://arxiv.org/abs/1503.06462> (accessed on 16 September 2019).
49. Krizhevsky, A. Learning Multiple Layers of Features from Tiny Images. Available online: <http://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf> (accessed on 10 December 2019).
50. Ketkar, N. *Deep Learning with Python: A Hands-On Introduction*, 1st ed.; Apress: New York, NY, USA, 2017; pp. 63–66.
51. Nair, V.; Hinton, G.E. Rectified linear units improve restricted Boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning (ICML 2010), Haifa, Israel, 21–25 June 2010.
52. Maas, A.L.; Hannun, A.Y.; Ng, A.Y. Rectifier nonlinearities improve neural network acoustic models. In Proceedings of the 30th International Conference on Machine Learning (ICML'13), Atlanta, GA, USA, 16–21 June 2013.
53. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. *arXiv* **2015**, arXiv:1502.01852.
54. Caruana, R.; Lawrence, S.; Giles, L. Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. In Proceedings of the 14th Annual Neural Information Processing Systems Conference (NIPS 2000), Denver, CO, USA, 27 November–2 December 2000.
55. Ruder, S. An Overview of Gradient Descent Optimization Algorithms. Available online: <https://arxiv.org/abs/1609.04747> (accessed on 16 September 2019).
56. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. Available online: <https://arxiv.org/abs/1412.6980> (accessed on 10 December 2019).
57. Keras. Available online: <http://github.com/keras-team/keras> (accessed on 10 December 2019).
58. Hackeling, G. *Mastering Machine Learning with Scikit-Learn*, 1st ed.; Packt Publishing: Birmingham, UK, 2014; pp. 97–185.
59. Kubat, M. *An Introduction to Machine Learning*, 2nd ed.; Springer: New York, NY, USA, 2017; pp. 43–56.
60. Noi, P.T.; Kappas, M. Comparison of random forest, k-nearest neighbor, and support vector machine classifiers for land cover classification using sentinel-2 imagery. *Sensors* **2018**, *18*, 18. [CrossRef]
61. Li, S.; Hu, W.; Yang, Y.; Liu, F.; Gan, W. Research of FOG-based measurement technique for continuous curve modes of long span bridge. *Bridge Constr.* **2014**, *44*, 69–74.

