Structural Damage Detection Based on Real-Time Vibration Signal and Convolutional Neural Network

Zhiqiang Teng 1, Shuai Teng 1, Jiqiao Zhang 1, Gongfa Chen 1,* and Fangsen Cui 2

1 School of Civil and Transportation Engineering, Guangdong University of Technology, Guangzhou 510006, China; tzq1217@sina.cn (Z.T.); 1114ai0805@sina.com (S.T.); zhangjq@gdut.edu.cn (J.Z.)
2 Institute of High Performance Computing, Agency for Science, Technology and Research, Singapore 138632, Singapore; cuifs@ihpc.a-star.edu.sg
* Correspondence: gongfa.chen@gdut.edu.cn; Tel.: +86-136-6248-3527

Received: 8 June 2020; Accepted: 7 July 2020; Published: 9 July 2020

Abstract: The traditional methods of structural health monitoring (SHM) have obvious disadvantages such as being time-consuming, laborious and non-synchronizing, and so on. This paper presents a novel and efficient approach to detect structural damages from real-time vibration signals via a convolutional neural network (CNN). As vibration signals (acceleration) reflect the structural response to the changes of the structural state, hence, a CNN, as a classifier, can map vibration signals to the structural state and detect structural damages. As it is difficult to obtain enough damage samples in practical engineering, finite element analysis (FEA) provides an alternative solution to this problem. In this paper, training samples for the CNN are obtained using FEA of a steel frame, and the effectiveness of the proposed detection method is evaluated by inputting the experimental data into the CNN. The results indicate that, the detection accuracy of the CNN trained using FEA data reaches 94% for damages introduced in the numerical model and 90% for damages in the real steel frame. It is demonstrated that the CNN has an ideal detection effect for both single damage and multiple damages. The combination of FEA and experimental data provides enough training and testing samples for the CNN, which improves the practicability of the CNN-based detection method in engineering practice.

Keywords: structural damage detection; real-time vibration signal; convolutional neural network; finite element analyses; steel frame

1. Introduction

Structural damage detection (SDD) is an important measure to avoid accidents of bridges in service. Structural damages, such as surface cracks, surface fall-off, and aging, generally exist and thus change the mass and stiffness of the structure [1]. The traditional SDD method, visual inspection by experienced engineers, is subjective, time-consuming, and laborious. To offset the disadvantages of the visual inspection method, vibration-based SDD methods have been developed [2,3]. As structural damages will certainly cause changes in the structural stiffness and mass and then affect the natural frequencies and mode shapes of the structure [4,5], some early research suggested natural frequencies as a damage index [6,7], however, natural frequencies have been proved to be insensitive to the damage locations [8]. Compared with natural frequencies, mode shapes are more capable of detecting the local locations [9]. The derivatives of the mode shapes, e.g., the mode curvature, modal strain energy, and strain mode, were also proposed as damage indexes and some encouraging results have been achieved [10,11]. However, these mode-based methods are vulnerable to the impact of the measurement environment. As the vibration responses of the structure reflect the structural damage information, thus the structural damage may be detected by vibration signals directly [12,13]. Relevant research shows that the vibration signal-based damage
detection methods contain more damage information than modal parameters [14] and overcome the
dependence of detection accuracy on specific modes. However, signal-based methods involve a lot
of signal processing work, which requires a powerful data processing tool, such as machine learning
algorithms.

Classical machine learning algorithms include support vector machines (SVMs), a decision tree
(DT), artificial neural networks (ANNs), and so on [15,16]. While the research results show that
SVMs and DTs are difficult to implement for large-scale training samples in complex situations, on
the other hand, ANNs can easily overcome the above problems [17]. As a powerful data processing
tool, ANNs can automatically extract the information of structural damages in signals and map this
information to the structural damage states. Therefore, introducing an ANN into the field of damage
detection has great application prospects.

As a powerful data processor, ANNs have been widely used in pattern recognition, prediction
and estimation, automatic control as well as the field of SDD [18]. An ANN is composed of many
artificial neurons, which are essentially nonlinear functions. The inputs and outputs of the ANN are
mapped by the nonlinear functions [19,20]. However, the traditional ANN is time-consuming and
over-fitting, which limits its application in practical engineering [21,22]. In recent years, the
convolutional neural network (CNN) has developed rapidly in various fields (e.g., image
classification, object recognition, and so on.) [23,24], and it has been successfully applied to crack
detection [25] in civil engineering. However, the image-based detection method cannot evaluate the
inner structural damages or damages in inaccessible locations. The combination of a CNN and
vibration-based SDD indexes can overcome the limitations of the above image-based methods. Ma et
al. [26] used a one-dimensional CNN to detect damages in the numerical model of a steel beam.
Their results show that a CNN based on acceleration signals can detect damages with an accuracy of
94.1%. Abdeljaber et al. [27] carried out SDD experiments using loosened bolts to simulate the
damage of a steel frame and accurately detected the location of the loosened bolts. Zhang et al. [28]
detected structural stiffness and mass changes in T-shaped steel beams and real-world steel bridges.
Teng et al. [29] used a CNN to detect structural damages by combining structural dynamic
responses with modal parameters and proved that introducing structural dynamic responses can
effectively improve the damage detection effect. Though it is encouraging to use CNNs to detect
structural damages in the above research, however, it is not clear whether a CNN trained by
numerical simulations is applicable to detect damages in a real structure from vibration
measurements. As for real-world structures, it is difficult to obtain a large number of network
training samples from actual experimental measurements, which can be effectively compensated by
the finite element analyses (FEA). It is expected that the combination of experimental measurements
and FEA can significantly improve the practical applications of CNNs in SDD.

In this paper, a SDD method combining numerical simulations with experimental
measurements is proposed. A three-dimensional steel frame is taken as the structural model, the
CNN architecture is designed and the training samples are obtained by the FEA of the steel frame.
Finally, the experimental data is inputted into the FEA-trained CNN to validate its detection
effectiveness.

2. Methods

In this paper, the acceleration that was obtained from measuring the steel frame model and
used as the CNN input to detect structural damages. The CNN could automatically extract damage
features from these signals without analyzing the indicators like traditional detection methods. The
overview of the proposed method was organized as follow: (1) got the structural response data
through the FEA and vibration experiments; (2) trained the CNN by using the samples obtained
from the FEA; (3) tested the trained CNN using single damage, multiple damages, and combined
datasets.

This paper used a steel frame beam as the research object (Figure 1a). The steel frame had a
length, width, and height of 9.912 m, 0.354 m, and 0.354 m, respectively. The steel frame consisted of
355 rods; each rod had a hollow circular cross section with an external radius of 0.005 m and
thickness of 0.002 m. The two ends of the steel frame were fixed. Damages were introduced in 9 rods (numbered 1 to 9 in Figure 1b) in the steel frame. The response signals of the 13 measurement points (labelled as A1 to A13) on the bottom chords were used as the inputs of the CNN samples. The 4 excitation points (F1–F4) were on the top chords (Figure 1b). In this paper, the acquisition frequency of the response signals in the numerical simulations and vibration experiments was 100 Hz, and the collection time was 8 s for each excitation.

2.1. Numerical Simulations

The software package ABAQUS (SIMULIA Inc, Providence, RI, USA) was used to build the finite element model (FEM) of the steel frame used in the vibration experiments (Figure 2). The density, elastic modulus, Poisson’s ratio, and modal damping were 37800 kg/m³, 212 GPa, 0.028, and 0.02, respectively. It was assumed that the structural damage level was proportional to the reduction of the elastic modulus, i.e., the damage of 50% was simulated by reducing the elastic modulus of the rod to its half.

Each of the 355 rods was regarded as an element. To obtain the training samples for various damage scenarios studied in this paper, python scripts were used to automatically analyze the damage scenarios and extract vibration signals in batch in ABAQUS [29]. The specific steps included: 1. Use ABAQUS to establish the FEM and specify each rod material properties, then generate the input file of the intact structure. 2. Read the input file of the intact structure and modify the material property of the damaged rod to generate the input file for a damage scenario and submit it for FEA. 3. Extract response signals (accelerations) of the concerned rods under the intact state and various damage scenarios.

2.2. Vibration Experiments

Figure 1 showed the steel frame used in the experiments. The damage was introduced in any one of the 9 rods by cutting off 50% of its cross-sectional area (Figure 3a).
Experimental equipment (Figure 3) included a dynamic data acquisition instrument (JM3840, Jing-Ming Technology Inc., Yangzhou, China), 13 accelerometers, a hammer, and a laptop. As the frame weighs 135 kg, while the accelerometers weigh 0.52 kg and the cable 0.84 kg, therefore, the effect of the added mass (of the accelerometers and cable) on the measurement results can be ignored. The steel frame was excited by the hammer and the accelerometers were used to obtain the vibration signals of the 13 measurement points.

2.3. CNN Samples

The CNN samples were from two sources, numerical simulations and vibration experiments, which are described in the following.

Numerical simulations: four datasets (A, B, C, D) with a single damage in a rod, double damages simultaneously in 2 rods, triple damages simultaneously in 3 rods and mixed partial samples of three above datasets.

For the single damage dataset, there were 10 scenarios (9 damage locations + 1 intact structure). The acceleration time history signals of the 13 measurement points for these 10 scenarios were used as the inputs of the CNN samples, and for the corresponding CNN output, the intact structure was set to 0, the damage on Rod 1 set to 1, the damage on Rod 2 set to 2, and so on.

For the dataset of damages simultaneously in 2 rods, any two of the 9 rods were randomly selected, hence, there are 36 ($C_9^2$) scenarios. The acceleration signals of the 13 measurement points for the 36 scenarios were used as the CNN inputs, correspondingly 1, 2, ..., 36 were set as the CNN outputs respectively.

For the dataset of simultaneous damages in 3 rods, any three of the 9 rods were randomly selected. There were 84 scenarios ($C_9^3$), and the acceleration signals of the 13 measurement points for the 84 scenarios were used as the CNN inputs, correspondingly, 1, 2, ..., 84 were set as the CNN outputs respectively.

Dataset D included the intact structure plus 7 damage scenarios, which were (1) damage in Rod 1, (2) damage on Rod 5, (3) damage in Rod 9, (4) damages simultaneously in Rod 1 and Rod 5, (5) damages simultaneously in Rod 1 and Rod 9, (6) damages simultaneously in Rod 5 and Rod 9, and (7) damages simultaneously in Rod 1, Rod 5, and Rod 9. The acceleration signals of the 13
measurement points for the 8 scenarios were used as the CNN input, correspondingly, 0, 1, 2, ..., 7 were set as the CNN outputs.

Vibration experiments: dataset E was consistent with the damage scenarios of dataset D in the numerical simulation; the vibration signals of 8 structural states were used as the CNN inputs and 0, 1, 2, ..., 7 were set as the CNN outputs.

In the numerical simulations, the excitations were applied at 4 locations (F1, F2, F3, and F4), in turn, 5 times; for each excitation, the response was collected for 8 s (with the acquisition frequency of 100 Hz) at the 13 measurement points, therefore, a data matrix of $16000 \times 13$ was collected. A sliding window (with the size of $10 \times 13$) was used to slide down the data matrix with a step each time, so that 15,991 samples were produced for each damage scenario (Figure 4). Tables 1–3 list the sample numbers of each dataset.

![Sample acquisition](image)

**Figure 4.** Sample acquisition.

| Datasets | Damage Scenarios | Samples | Total |
|----------|-----------------|---------|-------|
| A        | State 0         | 15,991  |       |
|          | State 1         | 15,991  |       |
|          |                 |         | 159,910 |
|          | State 9         | 15,991  |       |
| B        | State 1         | 15,991  |       |
|          | State 2         | 15,991  |       |
|          |                 |         | 575,676 |
|          | State 36        | 15,991  |       |
| C        | State 1         | 15,991  |       |
|          | State 2         | 15,991  |       |
|          |                 |         | 1,343,244 |
|          | State 83        | 15,991  |       |
|          | State 84        | 15,991  |       |

**Table 1.** Samples of the datasets A, B, and C.

| Damage Scenarios | Samples | Damage Scenarios | Samples |
|------------------|---------|-----------------|---------|
| State 0          | 15,991  | State 4         | 15,991  |
| State 1          | 15,991  | State 5         | 15,991  |
| State 2          | 15,991  | State 6         | 15,991  |
| State 3          | 15,991  | State 7         | 15,991  |
| Total            |         |                 | 127,928 |
Table 3. Samples of the dataset E.

| Damage Scenarios | Samples | Damage Scenarios | Samples |
|------------------|---------|------------------|---------|
| State 0          | 791     | State 4          | 791     |
| State 1          | 791     | State 5          | 791     |
| State 2          | 791     | State 6          | 791     |
| State 3          | 791     | State 7          | 791     |
| Total            | 6,328   |                  |         |

2.4. Convolutional Neural Network

In this paper, the CNN architecture for classification is shown in Figure 5. The CNN architecture was based on the steel frame and the location of the acceleration measurement points. The CNN included two convolution layers (the first layer had 30 convolution kernels with the size and stride being $5 \times 5$ and 1; the second layer had 60 convolution kernels with the size and stride being $2 \times 2$ and 1), a pooling layer (the size and stride being $3 \times 3$ and 3), a fully connection layer, and an output layer.

![Figure 5: Convolutional neural network (CNN) architecture.](image)

The convolution process was to multiply each element of the convolution kernel with the corresponding element of each sub matrix of the input matrix and then sum them to get an element in a feature matrix as shown in Equation (1):

$$ f(i) = \sum_{n=1}^{U_k} S(i+n)K(n) $$  \hspace{1cm} (1)

where the function $S$ is the input, the function $K$ is the convolution kernel, $U_k$ was the number of elements in the convolution region, and $i$ was the number of moves of the convolution kernel. Then, for the convolution kernel slides with a fixed step size, the process was repeated until all elements of the input matrix were involved; finally, it forms the feature matrix (Figure 6). The pooling layer plays a role in reducing the dimension of the input layer. Generally, there were two types of pooling: maximum pooling and average pooling (Figure 7). In this paper, maximum pooling was adopted as it performs better than average pooling [30].
Activation functions enhance the CNN learning ability. The commonly used activation functions include Sigmoid, tanh, and ReLU (Rectified Linear Unit) (Figure 8). In this paper, ReLU was used, because it behaves better in computations compared with Sigmoid and tanh [31].

Softmax was the output layer of the CNN, which was used to carry out multi-classification and output the final results predicted by the CNN. The softmax calculation is shown in Formula (2); where $Z$ was a vector composed of $k$ elements, and $\sigma(z)$ was the probability distribution of each element of the vector.
\[ \sigma(z) = \frac{\exp(Z_j)}{\sum_{k=1}^{K} \exp(Z_k)} \]  

(2)

where \( j = 1, \ldots, K \), \( Z = (Z_1, Z_2, \ldots, Z_K) \in \mathbb{R}^K \), \( 0 \leq \sigma(z) \leq 1 \), \( \sum_{j=1}^{K} \sigma(z) = 1 \). The above formula was applied to solve the problem of multi-label classification in the CNN. The prediction probability of class \( j \) for a given sample vector \( x \) and weighted vector \( w \) was

\[ P(y = j|x) = \frac{\exp(x^T w_j)}{\sum_{k=1}^{K} \exp(x^T w_k)} \]  

(3)

where \( j = 1, \ldots, K \), \( 0 \leq P(y = j|x) \leq 1 \), \( \sum_{j=1}^{K} P(y = j|x) = 1 \), \( x \) was the output of the fully connected layer, and \( w \) was the probability of the connection weights between the predicted output and actual output.

In order to update the network weights efficiently, the classical stochastic gradient descent method is generally used to optimize the network. This paper adopts a more effective optimization method, adaptive motion estimation (Adam), so as to achieve a more effective recognition effect. Adam is a combination of AdaGrad (gradient algorithm) and Rmsprop (root mean square prop) [32], which combines the advantages of both algorithms: their ability to maintain an adaptive level of learning for each parameter.

2.5. Structural Damage Detection

In this paper, the CNN was designed in MATLAB (MathWorks Inc, Natick, MA, USA) and the CNN internal parameters were adjusted to achieve the ideal detection results. The CNN parameters were shown in Table 4.

| Layer | Type         | Kernel Number | Kernel Size | Stride | Padding | Activation |
|-------|--------------|----------------|-------------|--------|---------|------------|
| 1     | Input        | None           | None        | None   | None    | None       |
| 2     | Convolution  | 30             | [5 5]       | [1 1]  | 0       | ReLU       |
| 3     | Max Pooling  | None           | [3 3]       | [3 3]  | 0       | None       |
| 4     | Convolution  | 60             | [2 2]       | [1 1]  | 0       | ReLU       |
| 5     | FC           | None           | None        | None   | None    | None       |
| 6     | Softmax      | None           | None        | None   | None    | None       |
| 7     | Output       | None           | None        | None   | None    | None       |

In this paper, five datasets were studied. The CNN architecture was the same for all damage scenarios. Datasets A, B, C, and D were studied with the samples obtained from the numerical simulations. The vibration signals (accelerations) obtained from the numerical simulations were used for network training and testing (15,200 training samples and 791 testing samples) for each damage scenario. Dataset E was to provide experimental testing samples to validate the applicability of the CNN trained using numerical simulations.

Normalization was widely used in data processing because it can keep data in a range and make the data from different sources comparable. In this paper, the acceleration of the measuring points was normalized using Formula (4):
where \( x \) and \( y \) are the values before and after normalization, respectively, and \( b \) and \( a \) are the maximum and minimum values of the sample data, respectively. In this paper, the acceleration was normalized into the range \([-1, 1]\) along the time direction. The normalization method was shown in Figure 9. After the normalization, the data were inputted into the CNN for damage detection, which is shown in Figure 10.

\[
y = \frac{x-a}{b-a} \tag{4}
\]

3. Results

3.1. Damage Classification Based on Numerical Simulations

Four datasets (A, B, C, D), described in Section 2.3, were used for training and testing the CNN, and the results were as follows.

Figure 11 and Table 5 show the training process and testing results for dataset A. Figure 11 shows that the CNN converged after about 200 iterations and the detection accuracy of validation and training was above 98%. Table 5 shows that the detection accuracy of the damaged rods in all locations was above 90%, except for the cases of the intact structure (85.3%) and damage in Rod 7 (87.1%). The overall detection accuracy was 96%.
Figure 11. Training process of dataset A.

Table 5. Detection results of dataset A.

| Amount | 
|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|-------|
|        | 0      | 1      | 2      | 3      | 4      | 5      | 6      | 7      | 8      | 9      | Total |
|        | 0      | 675    | 10     | 0      | 0      | 65     | 0      | 0      | 41     | 0      | 0      | 791   | 85.3  |
|        | 1      | 5      | 766    | 0      | 0      | 14     | 0      | 0      | 4      | 2      | 0      | 791   | 96.8  |
|        | 2      | 0      | 0      | 791    | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 791   | 100   |
|        | 3      | 0      | 0      | 2      | 789    | 0      | 0      | 0      | 0      | 0      | 0      | 791   | 99.7  |
| Actual | 4      | 40     | 27     | 0      | 0      | 724    | 0      | 0      | 0      | 0      | 0      | 791   | 91.5  |
| damage | 5      | 0      | 0      | 0      | 0      | 0      | 791    | 0      | 0      | 0      | 0      | 791   | 100   |
| location| 6      | 0      | 0      | 0      | 0      | 1      | 0      | 790    | 0      | 0      | 0      | 791   | 99.9  |
|        | 7      | 66     | 5      | 0      | 0      | 31     | 0      | 0      | 689    | 0      | 0      | 791   | 87.1  |
|        | 8      | 2      | 1      | 0      | 0      | 0      | 0      | 0      | 788    | 0      | 0      | 791   | 99.6  |
|        | 9      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 791   | 100   |
| Total  | 788    | 809    | 793    | 789    | 835    | 791    | 790    | 734    | 790    | 791    | 7910  |       | 96    |

Table 6 shows that in the scenarios of simultaneous damages in any two members, the detection accuracy of damage locations in 36 scenarios was above 90%, except one case (17), and the overall detection accuracy was 94.6%.

Figure 12 and Table 6 show the training process and testing results for dataset B. Figure 12 shows that the CNN converged after 120 iterations and the accuracy of validation and training was above 95%. Table 6 shows that, in the scenarios of simultaneous damages in any two members, the detection accuracy of damage locations in 36 scenarios was above 90%, except one case (17), and the overall detection accuracy was 94.6%.

Figure 12. Training process of dataset B.
Table 6. Detection results of dataset B.

| Damage State | Predicted Number | Total | %  | Damage State | Predicted Number | Total | %  |
|--------------|------------------|-------|----|--------------|------------------|-------|----|
| 1            | 726              | 791   | 91.8| 19           | 715              | 791   | 90.4|
| 2            | 744              | 791   | 94.1| 20           | 780              | 791   | 98.6|
| 3            | 715              | 791   | 90.4| 21           | 783              | 791   | 99.0|
| 4            | 754              | 791   | 95.3| 22           | 736              | 791   | 93.0|
| 5            | 727              | 791   | 91.9| 23           | 726              | 791   | 91.8|
| 6            | 697              | 791   | 88.1| 25           | 727              | 791   | 91.9|
| 7            | 727              | 791   | 91.9| 26           | 700              | 791   | 88.5|
| 8            | 783              | 791   | 99.0| 27           | 781              | 791   | 98.7|
| 9            | 721              | 791   | 91.2| 28           | 727              | 791   | 91.9|
| 10           | 778              | 791   | 98.4| 29           | 788              | 791   | 99.6|
| 11           | 773              | 791   | 97.7| 30           | 776              | 791   | 98.1|
| 12           | 732              | 791   | 92.5| 31           | 712              | 791   | 90.0|
| 13           | 783              | 791   | 99.0| 32           | 787              | 791   | 99.5|
| 14           | 779              | 791   | 98.5| 33           | 783              | 791   | 99.0|
| 15           | 745              | 791   | 94.2| 34           | 736              | 791   | 93.0|
| 16           | 780              | 791   | 98.6| 35           | 697              | 791   | 88.1|
| 17           | 773              | 791   | 97.7| 36           | 790              | 791   | 99.9|
| **Total**    |                  |       |     | **Total**    |                  |       |    |
|              |                  |       |     |              |                  |       | 94.6|

Figure 13 and Table 7 show the training process and testing results for dataset C. Figure 13 shows that the CNN converged at about 200 iterations and the accuracy of validation and testing reached 95%. Table 7 shows that, in the scenarios of simultaneous damages in three rods, the detection results of damage locations were also ideal in 84 scenarios and the overall detection accuracy was above 95.8%.

![Figure 13. Training process of dataset C.](image_url)

Table 7. Detection results of dataset C.

| Damage State | Predicted Number | Total  | %  | Damage State | Predicted Number | Total  | %  |
|--------------|------------------|--------|----|--------------|------------------|--------|----|
| 1            | 758              | 791    | 95.8| 43           | 788              | 791    | 99.6|
| 2            | 714              | 791    | 90.3| 44           | 707              | 791    | 89.4|
| 3            | 769              | 791    | 97.2| 45           | 784              | 791    | 99.1|
| 4            | 754              | 791    | 95.3| 46           | 788              | 791    | 99.6|
Figure 14 and Table 8 were the training process and testing results for dataset D. Figure 14 shows that the CNN converged after 70 iterations and the accuracy of validation and training almost reached 100%. Table 8 illustrates that, in the case where both a single damage and multiple damages were involved, the detection accuracy was over 97%.
Figure 14. Training process of dataset D.

Table 8. Detection results of dataset D.

| Amount | Prediction Damage Location | Total | %  |
|--------|----------------------------|-------|----|
| 0      | 773 0 16 0 2 0 0 0         | 791   | 97.7 |
| 1      | 0 770 1 0 0 20 0 0         | 791   | 97.3 |
| 2      | 32 0 757 0 2 0 0 0         | 791   | 95.7 |
| Actual | 3 0 0 1 774 16 0 0 0 0 0   | 791   | 97.9 |
| damage | 4 2 0 0 49 740 0 0 0 0     | 791   | 93.6 |
| location | 5 0 33 0 0 0 0 758 0 0 0   | 791   | 95.8 |
| 6      | 3 1 0 0 0 0 787 0           | 791   | 99.5 |
| 7      | 0 0 0 0 0 0 0 0 0 0 0 0 791 | 791   | 100  |
| Total  | 810 804 775 823 760 778 787 791 | 6328  | 97.2 |

3.2. Structural Damage Classification under Vibration Experiments

The CNN was trained with the samples obtained by the FEA, and the acceleration signals obtained from the experiments were inputted into the trained CNN for damage detection. Figure 15 is an acceleration signal extracted from the experiments (the acceleration signals of 13 points were extracted). Table 8 is the testing results of damage detection for the experimental input.

Figure 15. Acceleration time-history curve.
Table 9 showed that, for the eight damage scenarios (intact structure + 7 damage scenarios), the detection accuracy for cases No.0 (intact structure), No.6 (double damages) and No.7 (triple damages) was above 93%, the detection accuracy for other damage scenarios was above 85%, and the overall detection accuracy was 90.1%.

| Amount | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | Total | %   |
|--------|----|----|----|----|----|----|----|----|-------|-----|
| 0      | 742| 4  | 33 | 0  | 12 | 0  | 0  | 0  | 791   | 93.8|
| 1      | 41 | 687| 9  | 0  | 0  | 54 | 0  | 0  | 791   | 86.9|
| 2      | 56 | 17 | 691| 7  | 20 | 0  | 0  | 0  | 791   | 87.3|
| Actual | 3  | 15 | 0  | 19 | 702| 45 | 10 | 0  | 791   | 88.7|
| damage | 4  | 27 | 0  | 7  | 78 | 679| 0  | 0  | 791   | 85.8|
| location| 5  | 9  | 69 | 2  | 8  | 0  | 693| 10 | 0     | 791 | 87.6|
| 6      | 0  | 12 | 0  | 0  | 0  | 38 | 741| 0  | 791   | 93.6|
| 7      | 0  | 0  | 0  | 0  | 0  | 5  | 21 | 765| 791   | 96.7|
| All    | 890| 789| 761| 795| 756| 800| 772| 765| 6328  | 90.1|

4. Discussion and Conclusions

In this paper, the applicability of CNNs in SDD was investigated. The training samples of the CNN were obtained using FEA; the trained CNN has an ideal detection effect on the testing samples obtained from the numerical model; the detection effectiveness was also validated by the samples obtained from the vibration experiments. The detection accuracy of the proposed method was above 90%.

In Section 3.1, the detection of a single damage, double damages, triple damages and mixed damage scenarios was investigated using the CNN trained with the samples obtained from numerical simulations of the steel frame. It was found that the network converges after only 100 to 200 iterations and the training accuracy reaches 94%.

In Section 3.2, the vibration signals obtained from vibration experiments of the steel frame were used as the inputs of the CNN to validate the applicability of the CNN trained using FEA data. The detection results were ideal, and the detection accuracy was over 90%. Generally, it is unrealistic to train a CNN through real structural measurements, as it is difficult to obtain adequate damage samples for the engineering structures in service. To simulate structural damages by loosening joint bolts, increasing the structural mass, and other methods in the experiments [27] provides some hints for the creation of structural damage scenarios, but these man-made “damages” are time-consuming and laborious and limited to certain types. On the other hand, the FEA can simulate a variety of damage scenarios and generate a large number of training samples. The results show that these samples can predict the actual structural damages, which makes up for the disadvantages of the experimental method in generating damage scenarios.

In this paper, data normalization was used to make the data from numerical simulations and vibration experiments comparable. After dimensionless normalization, these two kinds of data were distributed in the same range without changing the inherent damage features of the original signals.

From the above discussions, the following conclusions can be drawn:
1. The CNN can accurately map the vibration signals (acceleration) to the structural damage state; it has an ideal detection effect for a single damage and multiple damages.

2. The effective combination of numerical simulations and vibration experiments makes the CNN-based damage detection method more applicable in engineering practice. The use of FEA can generate a large number of CNN training samples.

There are still some shortcomings in this paper: 1. the structural model is a steel frame used in a laboratory and the damage scenarios are man-made; 2. it is noticed that the results of this method are based on numerical simulations and experimental measurements. In practical engineering, the
measuring environment has a significant impact on the data collection process, which is not covered in this paper, further discussions with respect to environmental effects are referred to in the work of Ubertini’s group [33,34].

**Author Contributions:** Z.T. contributed to the paper in original draft preparation, methodology, investigation, formal analysis. S.T. contributed to the paper in Validation, investigation, formal analysis. J.Z. contributed to the paper in Writing—review, editing and supervision. G.C. contributed to the paper in conceptualization, methodology, investigation. F.C. contributed to the paper in conceptualization, methodology, review and editing. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was partially supported by the project (No. 2016YFE0117200) of the China–Singapore Joint Research Programme.

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

**References**

1. Das, S.; Saha, P.; Patro, S.K. Vibration-based damage detection techniques used for health monitoring of structures: A review. *J. Civ. Struct. Health Monit.* 2016, 6, 477–507.
2. Chang, K.C.; Kim, C.W. Modal-parameter identification and vibration-based damage detection of a damaged steel truss bridge. *Eng. Struct.* 2016, 122, 156–173.
3. Farahani, R.V.; Penumadu, D. Damage identification of a full-scale five-girder bridge using time-series analysis of vibration data. *Eng. Struct.* 2016, 115, 129–139.
4. Pandey, A.K.; Biswas, M.; Samman, M.M. Damage detection from changes in curvature mode shapes. *J. Sound Vib.* 1991, 145, 321–332.
5. Salawu, O.S. Detection of structural damage through changes in frequency: A review. *Eng. Struct.* 1997, 19, 718–723.
6. Cawley, P.; Adams, R.D. The location of defects in structures from measurements of natural frequencies. *J. Strain Anal. Eng. Des.* 1979, 14, 49–57.
7. Chen, Y.; Hou, X.B.; Li, W.; Zhang, X.H. Applications of different criteria in structural damage identification based on natural frequency and static displacement. *Sci. China Technol. Sci.* 2016, 59, 1746.
8. Döhlen, M.; Hille, F.; Mevel, L.; Rücker, W. Structural health monitoring with statistical methods during progressive damage test of S101 Bridge. *Eng. Struct.* 2014, 69, 183–193.
9. Dutta, A.; Talukdar, S. Damage detection in bridges using accurate modal parameters. *Finite Elem. Anal. Des.* 2004, 40, 287–304.
10. Sung, S.H.; Koo, K.Y.; Jung, H.J. Modal flexibility-based damage detection of cantilever beam-type structures using baseline modification. *J. Sound Vib.* 2014, 333, 4123–4138.
11. Cha, Y.; Buyukozturk, O. Structural Damage Detection Using Modal Strain Energy and Hybrid Multiobjective Optimization. *Comput.-Aided Civ. Infrastruct. Eng.* 2015, 30, 347–358.
12. Hou, Z.; Noori, M.; Amand, R.S. Wavelet-Based Approach for Structural Damage Detection. *J. Eng. Mech.* 2000, 126, 677–683.
13. Li, Z.; Park, H.S.; Adeli, H. New method for modal identification of super high-rise building structures using discretized synchrosqueezed wavelet and Hilbert transforms. *Struct. Des. Tall Spec. Build.* 2017, 26, e1312.
14. Tibaduiza, D.A.; Torres-Arredondo, M.A.; Mujica, L.E.; Rodellar, J.; Fritzen, C.P. A study of two unsupervised data driven statistical methodologies for detecting and classifying damages in structural health monitoring. *Mech. Syst. Signal Process.* 2013, 41, 467–484.
15. Ghiasi, R.; Ghasemi, M.R. Optimization-based method for structural damage detection with consideration of uncertainties- a comparative study. *Smart Struct. Syst.* 2018, 22, 561–574.
16. Shangong; Chen; Bin; Lin; Xuesong; Han; Xiaohu; Liang. Automated inspection of engineering ceramic grinding surface damage based on image recognition. *Int. J. Adv. Manuf. Technol.* 2013, 66, 431–443
17. Screen efficiency comparisons of decision tree and neural network algorithms in machine learning assisted drug design. *Sci. China 2019*, 62, 110–118.
18. Lee, J.J.; Lee, J.W.; Yi, J.H.; Yun, C.B.; Jung, H.Y. Neural networks-based damage detection for bridges considering errors in baseline finite element models. *J. Sound Vib.* 2005, 280, 555–578.
19. Hakim, S.J.S.; Razak, H.A. Structural damage detection of steel bridge girder using artificial neural networks and finite element models. *Steel Compos. Struct.* 2013, 14, 367–377.
20. Zang, C.; Imregun, M. Structural damage detection using artificial neural networks and measured frf data reduced via principal component projection. J. Sound Vib. 2001, 242, 813–827.
21. Yao, X. Evolutionary artificial neural networks. Int. J. Neural Syst. 1993, 4, 203–222.
22. Yao, W.S. The Researching Overview of Evolutionary Neural Networks. Comput. Sci. 2004, 31, 125–129.
23. 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, 1–13, doi:10.3233/ICA-170551.
24. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv 2014, arXiv:1409.1556.
25. Cha, Y.; Choi, W.; Büyükoztürk, Ö. Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks. Comput.-Aided Civ. Infrastruct. Eng. 2017, 32, 361–378.
26. Lin, Y.Z.; Nie, Z.H.; Ma, H.W. Structural Damage Detection with Automatic Feature extraction through Deep Learning. Comput.-Aided Civ. Infrastruct. Eng. 2017, 32, 1–22.
27. Abdeljaber, O.; Avci, O.; Kiranyaz, 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.
28. Zhang, Y.; Miyamori, Y.; Mikami, S.; Saito, T. Vibration-based structural state identification by a 1-dimensional convolutional neural network. Comput.-Aided Civ. Infrastruct. Eng. 2019, 34, 1–18.
29. Teng, S.; Chen, G.; Gong, P.; Liu, G.; Cui, F. Structural damage detection using convolutional neural networks combining strain energy and dynamic response. Meccanica 2019, 55, 945–959, doi:10.1007/s11012-019-01052-w.
30. Scherer, D.; Müller, A.; Behnke, S. Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition. In Proceedings of the International Conference on Artificial Neural Networks, Thessaloniki, Greece, 15–18 September 2010.
31. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In Proceedings of the International Conference on Neural Information Processing Systems, Lake Tahoe, Nevada, 3–6 December 2012; pp. 1097–1105.
32. Kingma, D.; Ba, J. Adam: A Method for Stochastic Optimization. arXiv 2014, arXiv:1412.6980.
33. Garcia-Macías, E.; Ubertini, F. MOVA/MOSS: Two integrated software solutions for comprehensive Structural Health Monitoring of structures. Mech. Syst. Signal Process. 2020, 143, 106830, doi:10.1016/j.ymssp.2020.106830.
34. Kita, A.; Cavallaghi, N.; Ubertini, F. Temperature effects on static and dynamic behavior of Consoli Palace in Gubbio, Italy. Mech. Syst. Signal Process. 2019, 120, 180–202.

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).