

Structural Damage Detection with Modal Strain Energy Using Convolution Neural Network



Gongfa Chen and Shuai Teng

Abstract Based on the excellent feature extraction and non-linear learning ability of a convolutional neural network (CNN), a structural damage detection method is proposed in this paper. When the structure is damaged, the changes of its modal parameters reflect the damage information of the structure. A simply supported beam was used and structural damage was introduced at different locations. The finite element method was used to simulate the free vibration of the beam and obtain the first-order modal strain energy for various damage scenarios. The obtained modal parameters and the damage information were used as the training samples of the neural network. A CNN was designed to detect damage (both location and level), which detected damage location with 100% accuracy and damage level with 5% relative error. Compared with a traditional Back Propagation (BP) neural network, the CNN had more advantages than the BP neural network in detecting damage location, and it was more economical in computational costs, the uptime of the CNN was about 5%–40% that of the BP neural network. It is found the CNN has excellent performance in detection of both damage locations and levels, the detection effect exceeds BP neural network, and it is more economical in computational cost than a BP neural network as it uses convolutional operation.

Keywords Structural damage detection · Convolutional neural networks · Modal strain energy · Feature extraction · Simply supported beam

G. Chen (✉) · S. Teng
School of Civil and Transportation Engineering, Guangdong University of Technology,
Guangzhou 510006, China
e-mail: gongfa.chen@gdut.edu.cn

S. Teng
e-mail: 1114ai0805@sina.com

© Springer Nature Singapore Pte Ltd. 2021
C. M. Wang et al. (eds.), *EASEC16*, Lecture Notes in Civil Engineering 101,
https://doi.org/10.1007/978-981-15-8079-6_121

1289

1 Introduction

Structural damage detection is an essential approach to prevent sudden collapse of structures and avoid casualties and heavy economic losses. A series of damage detection methods have been proposed [1, 6, 11, 18, 25], the principle of vibration-based methods is that the modal parameters (e.g., natural frequencies and modal shapes) of a structure vary with the changes of the structural physical parameters such as the stiffness and mass. By collecting the vibration excitation and response data of a structure, the modal parameters are obtained, and then the potential damage of the structure is detected by analyzing the change of its modal parameters. Frequency-based structural damage detection method has been used in damage detection of composite structures [2, 3]. Modal-based methods show that local damage causes irregularity of mode shapes [21] which is evident for relatively large damage [17]. Nevertheless, the changes of natural frequencies and mode shapes are unable to detect small damage [30]. As the modal strain energy (MSE) is related to the second-order derivatives of mode shapes for beam-like structures, it is much more sensitive to the damage than natural frequencies and mode shapes. The MSE has been used to successfully locate structural damage and quantify the damage level [4, 10, 16, 20, 22] for simple structures. Structural damage can cause changes in many mechanical parameters, normally, a single damage index is generally impossible to reflect all types of damage of the real structures. Thus it is essential to develop a comprehensive damage detection method, such as artificial neural networks (ANNs) [8], which is able to integrate multiple damage features into a detection method. An ANN is similar to a human brain and has excellent non-linear learning ability. Combination of ANNs and traditional damage indicators for obtaining structural damage information may advance the damage detection technology.

It has been demonstrated that an ANN is able to locate and quantify structural damage owing to its powerful data fitting as well as pattern recognition capability [7, 9]. It has achieved promising results [19, 24]. The traditional ANN (e.g., BP neural network) has its inherent shortcomings, such as low convergence rate, time-consuming, and over-fitting of data [27, 28], etc. When the traditional ANN is used to detect damage location, its training time is very long for high-dimension input data. To overcome the limitations of ANNs, the convolutional neural network (CNN), with convolution layers and pooling layers, has been developed to extract the features of the image [29] and been proved successful. It has more powerful feature learning ability and feature expression ability [12] than traditional ANNs. At present, the CNN has been widely used in license plate detection, face recognition and other fields [13, 23, 26]. The CNN has also been applied to SHM, such as crack detection [5, 14] and damage feature extraction from low-order vibration signals [15]. The application of CNNs to SHM provides a new intelligent method for structural damage detection. It is expected that CNNs can be applied to predict the locations and levels of damage in a structure.

In this paper, a CNN is proposed to detect damage of a simply supported beam; the modal strain energy is used as the input data of the CNN as it can reflect the damage information of structures, and comparisons are made between the performance of the CNN and traditional BP neural networks.

2 Methods

The CNN was trained with the modal strain energy of various damage scenarios and then used to predict new damage scenarios. In this paper, the collected data were arranged into a two-dimensional matrix as the CNN input data. The abnormality of the modal strain energy was extracted by the CNN to predict the damage location and damage level.

2.1 Numerical Calculation and Sample

The beam model used in this paper had a length of 9 m and a rectangular cross section of 0.3 m × 0.2 m. The structure was divided into 36 elements as numbered in Fig. 1.

The Young’s modulus, Poisson’s ratio and density of the steel beam model were 211 GPa, 0.288, and 7800 kg/m³ respectively. Structural damage was simulated by reducing the Young’s modulus of the concerned element. Totally 18 damage levels for each element were simulated, which were from 5% to 90% with an increment of 5%. In-house python scripts were used to prepare the training samples for the CNN.

Firstly, the detection of damage location is studied. The validation set was based on the cases with 45% damage in an element and the intact case (totally 37 samples), and the testing sets included the cases with either 10% or 60% damage in an element (72 samples).The training set included the following five groups, each group included 37 samples, i.e., 36 scenarios with damage in only one element plus the intact case.

Dataset (A): only 15% damage in one element for each damage scenario plus the intact case;

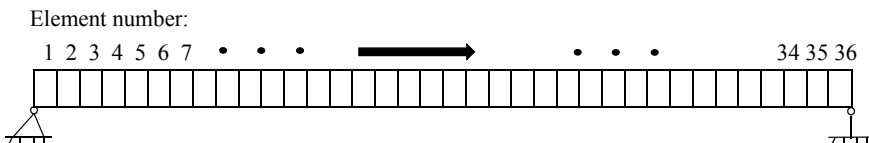


Fig. 1 Simply supported beam model with 36 elements

Dataset (B): only 30% damage in one element for each damage scenario plus the intact case;

Dataset (C): only 75% damage in one element for each damage scenario plus the intact case;

Dataset (D): only 90% damage in one element for each damage scenario plus the intact case.

Then, the detection of damage level is studied. The validation set was based on the data of 45% damage in an element and the intact case (37 samples), the samples for 60% damage in an element (36 samples) were used to test the CNN fitting effect, the training set included the following five groups.

Dataset (A): Every element was simulated with 15% damage levels, thus there were 36 damage scenarios, plus 1 intact case, and thus there were totally 37 samples.

Dataset (B): Every element was simulated with 75% damage levels, thus there were 36 damage scenarios, plus 1 intact case, and thus there were totally 37 samples.

Dataset (C): Every element was simulated with 15%, 30%, 75% and 90% damage levels, thus there were 144 damage scenarios, plus 1 intact case, and thus there were totally 145 samples.

Dataset (D): Every element was simulated with 16 damage levels from 5% to 90% with an increment of 5%, thus there were $36 \times 16 = 576$ damage scenarios, plus 1 intact case, and thus there were totally 577 samples.

2.2 Convolutional Neural Network

The CNN was designed and trained using the Deep Learning Toolbox in MATLAB (MathWorks Inc, Natick, MA, US). The network included an input layer, 2 convolution layers, 1 pooling layer, an activation layer, a fully connected layer and output layer (classification layer or regression layer); for classification problems, a softmax layer was added after the fully connected layer. The explanations for the activation function and convolution and pooling processes were seen in the appendix A. In this paper, a CNN was designed for damage detection. The network architecture and structural parameters of the CNN were shown in Fig. 2 and Table 1.

2.3 Input and Output of Network

The CNN input data, i.e., the modal strain energy of each element, was collected for each damage scenario, and a matrix of 6×6 was constructed as the input. This paper used a classification method to detect the damage location, which was set to

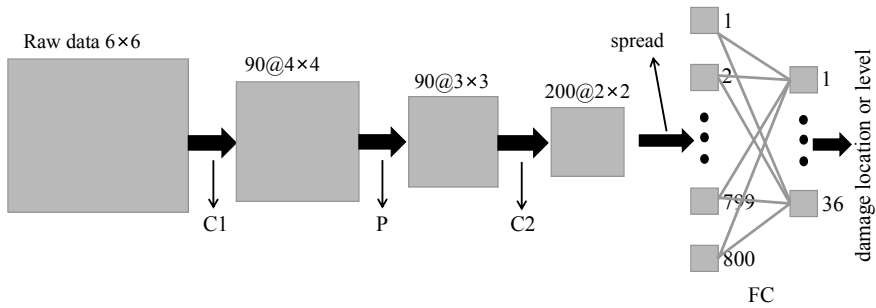


Fig. 2 CNN framework. C1, C2: Convolution layers; P: Pooling layer; FC: Fully Connected layer

Table 1 Structural parameters of the CNN

| Layer | Type | Kernel num. | Kernel size | Stride | Pad | Activation |
|-------|------------------------------|-------------|--------------|--------|------|------------|
| 1 | Input | None | None | None | None | None |
| 2 | Convolution (C1) | 90 | 3×3 | [1, 1] | 0 | Leaky ReLU |
| 3 | Max pooling (P) | None | 2×2 | [1, 1] | 0 | None |
| 4 | Convolution (C2) | 200 | 2×2 | [1, 1] | 0 | Leaky ReLU |
| 5 | FC | None | None | None | None | None |
| 6 | Classification or regression | None | None | None | None | None |

different categories, i.e. the intact condition was set to 0, the damage on the element 1 set to 1, the damage on the element 2 set to 2, and so on.

For detection damage level, by replacing the softmax layer and classification layer with a regression layer of the CNN, the classification problem was transformed into a regression problem. The network output was set as a vector consisting of 36 elements, i.e. the 15% damage in Element 1 was set as [0.15, 0, 0... 0, 0, 0], 30% damage of Element 1 was set to [0.3, 0, 0, 0... 0, 0, 0], and so on; the intact case was defined to be a vector of 36 zeros.

3 Results

3.1 Detection of Single Damage Location

The CNN was trained with the 4 training sets described in Sect. 2.1 separately. The detection results were shown in Table 2.

Table 2 Prediction results by CNN

| Training sets | Testing sets | |
|---------------|--------------|------------|
| | Damage 10% | Damage 60% |
| Dataset (A) | 100% | 100% |
| Dataset (B) | 66.6% | 100% |
| Dataset (C) | 58.3% | 100% |
| Dataset (D) | 61.1% | 100% |

According to the detection effect of the four different Datasets, no matter which dataset was used for training, the prediction accuracy was 100% for the testing set with the damage level of 60%. For the testing set with the damage level of 10%, the prediction accuracy reached 100% only for the CNN trained by Dataset (A).

In order to compare the detection effect between the CNN and a traditional BP neural network, Dataset (A) was inputted into BP neural networks, and the prediction results were shown in Table 3.

Table 3 showed that the detection effect of the BP neural networks was worse than that of the CNN, and the detection effect of BP neural networks did not change significantly with the node number of the hidden layer. The best detection effect was 77.8%.

Table 3 Prediction results by BP neural networks

| Network type | Nodes of hidden layer (n) | Iterations | Termination reason | Accuracy | |
|--------------|---------------------------|------------|--------------------|------------|------------|
| | | | | Damage 10% | Damage 60% |
| BP | 18 | 8 | Validation checks | 72.2% | 72.2% |
| BP | 24 | 8 | Validation checks | 77.8% | 63.9% |
| BP | 30 | 8 | Validation checks | 80.6% | 69.4% |
| BP | 36 | 8 | Validation checks | 77.8% | 77.8% |
| BP | 42 | 8 | Validation checks | 77.8% | 75.0% |
| BP | 48 | 8 | Validation checks | 75.0% | 75.0% |
| BP | 54 | 8 | Validation checks | 77.8% | 72.2% |

Note: Validation checks: The error of continuous 6 iterations no longer changed

3.2 Detection of Damage Level

The testing sets were inputted into the trained network as described in Sect. 2.1. The detection results were shown in Table 4, the relative error decreases with the increase of training samples, and the minimum relative error of the damage level was about 5%.

The relative error was used to evaluate the detection effect of damage level, and the formula was as follows:

$$R_{Error} = \frac{|y_p - y_t|}{y_t} \times 100\% \quad (1)$$

where y_p and y_t were the predicted value and target value of a testing sample. The relative error for n testing samples was:

$$R_Error = \frac{1}{n} \sum_1^n \left(\frac{|y_p - y_t|}{y_t} \right) \times 100\% \quad (2)$$

The Dataset (D) were inputted into the BP neural networks. After training, the detection effect was obtained and shown in Table 5.

As shown in Table 5, with the increase of the nodes of the hidden layer, the relative error of the BP neural networks decreased gradually, and the error was lower than that of the CNN, but the training time increased significantly. The fitting effect of the BP neural networks for damage level was better than that of the CNN. Table 6 showed the uptimes of the CNN and the BP neural networks which had the most comparative relative errors to that of the CNN.

When the relative errors were similar, the uptime of the CNN was about 5%–40% that of the BP neural networks, the iteration time of the CNN was only about 1% that of the BP neural networks. For the smallest relative error (1%) of the BP neural network in Table 5, the relative error of CNN was only 5%, but its uptime was only 2% that of the BP neural network.

Table 4 Prediction result of different Datasets

| Dataset | Sample num. | Damage location | Damage level (relative error) |
|---------|-------------|-----------------|-------------------------------|
| (A) | 37 | 100% | 75.69% |
| (B) | 37 | 100% | 24.50% |
| (C) | 145 | 100% | 23.30% |
| (D) | 577 | 100% | 4.77% |

Table 5 Detection effect of the BP neural networks

| Network type | Nodes of hidden layers | Iterations | Uptime | Termination reason | Relative error (Damage level) |
|--------------|------------------------|------------|--------------|--------------------|-------------------------------|
| BP | 18 | 15 | 44 s | Validation checks | 46% |
| BP | 24 | 19 | 1 min 39 s | Validation checks | 31% |
| BP | 30 | 15 | 2 min 00 s | Validation checks | 17% |
| BP | 36 | 54 | 10 min 24 s | Validation checks | 6% |
| BP | 42 | 306 | 78 min 58 s | Validation checks | 2% |
| BP | 48 | 361 | 122 min 32 s | Validation checks | 1% |
| BP | 54 | 452 | 194 min 29 s | Validation checks | 1% |

Table 6 Comparisons of uptime between CNN and BP neural networks

| Network type | Nodes of hidden layers | Iterations | Uptime | Termination reason | Relative error (Damage level) |
|--------------|------------------------|------------|-------------|--------------------|-------------------------------|
| BP | 36 | 54 | 10 min 24 s | Validation checks | 6% |
| BP | 42 | 306 | 78 min 58 s | Validation checks | 2% |
| CNN | | 2000 | 4 min 21 s | Max epoch | 5% |

4 Discussions

The comparisons of damage locations detected between the CNNs and BP neural networks showed that the CNNs had better classification ability than BP neural networks to extract damage location features. The CNNs used partially connected network structure to extract the main damage features, ignoring trivial information. While the BP neural networks adopted a fully connected network structure, each element in the raw data had an impact on the results.

It can be seen from Sect. 3.2 that the detection effect (damage level) of the BP neural networks was slightly better than that of the CNN. When the BP neural network had 36 and 42 nodes in the hidden layer, the relative errors were 6% and 2%, respectively, and the CNN were 5%. The uptime was 10 min 24 s, 78 min 58 s, 4 min 21 s, respectively. Thus, when the relative error was similar, the uptime of CNN was 5%–40% that of the BP neural networks.

In summary, the BP neural networks had excellent regression fitting effects but consumed a lot of computing resources. Because of its partial connection, the CNN lost some information and its fitting effect was sacrificed in the training process, but the calculation speed was faster than that of the BP neural network.

5 Conclusion

Based on the above discussions, the following conclusions can be drawn:

- 1) The CNN achieved excellent detection results for structural damage location and level.
- 2) The CNN had more advantages than the BP neural networks in detecting damage location.
- 3) The CNN was more economical in computational costs than a BP neural network.

According to the results of this paper, the combination of the CNN and the modal strain energy as a new damage detection method has great potential in structural damage detection.

References

1. Bayissa WL, Haritos N, Thelandersson S (2008) Vibration-based structural damage identification using wavelet transform. *Mech Syst Signal Process* 22(5):1194–1215
2. Cawley P, Adams RD (1979) The location of defects in structures from measurements of natural frequencies. *J Strain Anal Eng Des* 14(2):49–57
3. Cawley P, Adams RD (1979) A vibration technique for non-destructive testing of fibre composite structures. *J Compos Mater* 13(2):161–175
4. Cha Y-J, Buyukozturk O (2015) Structural damage detection using modal strain energy and hybrid multiobjective optimization. *Comput-Aided Civil Infrastruct Eng* 30(5):347–358
5. Cha Y-J, Choi W, Büyüköztürk O (2017) Deep learning-based crack damage detection using convolutional neural networks. *Comput-Aided Civil Infrastruct Eng* 32(5):361–378
6. Chen Y, Hou XB, Li W, Zhang XH (2016) Applications of different criteria in structural damage identification based on natural frequency and static displacement. *SCI CHINA Technol Sci* 59(11):1746
7. Geng X, Lu S (2018) Research on FBG-Based CFRP structural damage identification using BP neural network. *Photonic Sensors* 8(2):1–8
8. Guresen E, Kayakutlu G (2011) Definition of artificial neural networks with comparison to other networks. *Procedia Comput Sci* 3:426–433
9. Hadi S, Saptarshi D (2018) Structural damage identification using image-based pattern recognition on event-based binary data generated from self-powered sensor networks. *Struct Control Health Monitor* 25(1):e2135
10. Hu H, Wu C (2009) Development of scanning damage index for the damage detection of plate structures using modal strain energy method. *Mech Syst Signal Process* 23(2):274–287
11. Kim H, Melhem H (2004) Damage detection of structures by wavelet analysis. *Eng Struct* 26(3):347–362

12. Krizhevsky A, Hinton G (2009) Learning multiple layers of features from tiny images. Technical report, University of Toronto
13. Lawrence S, Giles CL, Tsoi AC, Back AD (1997) Face recognition: a convolutional neural-network approach. *IEEE Trans Neural Networks* 8(1):98–113
14. Li R, Yuan Y, Wei Z, Yuan Y (2018) Unified vision-based methodology for simultaneous concrete defect detection and geolocalization. *Comput-Aided Civil Infrastruct Eng*
15. Lin YZ, Nie ZH, Ma HW (2017) Structural Damage Detection with Automatic Feature extraction through Deep Learning. *Comput-Aided Civil Infrastruct Eng* (6)
16. Pal J, Banerjee S (2015) A combined modal strain energy and particle swarm optimization for health monitoring of structures. *J Civil Struct Health Monitor* 5(4):1–11
17. Pandey AK, Biswas M, Samman MM (1991) Damage detection from changes in curvature mode shapes. *J Sound Vibration* 145(2):321–332
18. Qiao P, Lu K (2007) Curvature Mode Shape-Based Damage Detection in Composite Laminated Plates. *Compos Struct* 80(3):409–428
19. Seyedpoor SM (2012) A two stage method for structural damage detection using a modal strain energy based index and particle swarm optimization. *Int J Non-Linear Mech* 47(1):1–8
20. Shen MHH, Grady JE (1992) Free vibrations of delaminated beams. *AIAA J* 30(5):1361–1370
21. Shi ZY, Law SS, Zhang LM (2000) Structural damage localization from modal strain energy change. *J Eng Mech* 218(5):1216–1223
22. Tensmeyer C, Saunders D, Martinez T (2017) Convolutional Neural Networks for Font Classification. In Paper presented at the International Conference on Document Analysis and Recognition
23. Xu H, Humar JM (2010) Damage detection in a girder bridge by artificial neural network technique. *Comput Aided Civil Infrastruct Eng* 21(6):450–464
24. Yan YJ, Cheng L, Wu ZY, Yam LH (2007) Development in vibration-based structural damage detection technique. *Mech Syst Signal Process* 21(5):2198–2211
25. Yao D, Zhu W, Chen Y, Zhang L (2018) Chinese license plate character recognition based on convolution neural network. In: Chinese Automation Congress
26. Yao WS (2004) The Researching Overview of Evolutionary Neural Networks. *Computer Science*
27. Yao X (1993) Evolutionary artificial neural networks. *Int J Neural Syst* 4(03):203–222
28. Yuan ZW, Zhang J (2016) Feature extraction and image retrieval based on AlexNet. In: Eighth International Conference on Digital Image Processing
29. Zou Y, Tong L, Steven GP (2017) vibration-based model-dependent damage (delamination) identification and health monitoring for composite structures — a review. *J Sound Vib* 10 (1):165–193