Smart Structures and Systems, *Vol. 23, No. 5 (2019) 507-520*
DOI: https://doi.org/10.12989/sss.2019.23.5.507

CNN-based damage identification method of tied-arch bridge using spatial-spectral information

Yuanfeng Duan, Qianyi Chen, Hongmei Zhang*,
Chung Bang Yun, Sikai Wu and Qi Zhu

College of Civil Engineering and Architecture, Zhejiang University, China

(Received January 30, 2019, Revised April 8, 2019, Accepted April 15, 2019)

Abstract. In the structural health monitoring field, damage detection has been commonly carried out based on the structural model and the engineering features related to the model. However, the extracted features are often subjected to various errors, which makes the pattern recognition for damage detection still challenging. In this study, an automated damage identification method is presented for hanger cables in a tied-arch bridge using a convolutional neural network (CNN). Raw measurement data for Fourier amplitude spectra (FAS) of acceleration responses are used without a complex data pre-processing for modal identification. A CNN is a kind of deep neural network that typically consists of convolution, pooling, and fully-connected layers. A numerical simulation study was performed for multiple damage detection in the hangers using ambient wind vibration data on the bridge deck. The results show that the current CNN using FAS data performs better under various damage states than the CNN using time-history data and the traditional neural network using FAS. Robustness of the present CNN has been proven under various observational noise levels and wind speeds.

Keywords: multiple damage identification for hangers; tied-arch bridge; convolutional neural network; deep learning; Fourier amplitude spectra; ambient wind vibration data

1. Introduction

Cables are major load carrying members of a cable supported bridge, which are vulnerable to material degradation and structural damage due to severe environmental and operational loads during their long life span. The structural health monitoring (SHM) system for a cable supported bridge generally includes sensors on the cables as well as on the bridge deck to ensure the structural integrity, public safety, and smooth traffic operation. The cable monitoring has focused on the cable tension, which is an effective indicator for the integrity of the cable and the bridge. Vibration methods have been commonly used, where the tension force can be estimated based on the natural frequencies of the cable obtained from the acceleration response (Cho *et al*. 2010a, b, Jang *et al.* 2010, Kim *et al*. 2013, Chen *et al*. 2016, Ding *et al*. 2016). However, those methods frequently suffer from large estimation errors. Recently, elasto-magnetic sensors (Yim *et al.* 2013, Cho *et al.* 2013) and elasto-magneto-electric sensors (Duan *et al*. 2011, 2012, 2016, Zhang *et al.* 2014, 2018) were developed and successfully applied to the cable tension measurements with relatively high accuracy. However, those methods require special sensor systems with high cost. A guided wave method using magneto-strictive sensors was recently presented and proven to be effective for damage localization in cables (Zhang *et al*. 2018), but its performance was not satisfactory for identification of damage severities.

In this study, a method for identification of multiple damages in bridge hangers is presented, which can be easily integrated into a conventional global monitoring system of the bridge structure. Acceleration responses at the cable anchorage points along the deck are used, so that no special sensor systems are needed. A convolutional neural network (CNN) technique is employed for automated operation using raw measurement data without complex and time-consuming procedures for extracting of engineering features related to the damage, such as the modal properties (Yun and Bahng 2000, Sekhar 2004, Jin *et al*. 2014). Fourier amplitude spectra (FAS) of the acceleration responses were used as an input to the CNN, because the modal properties are much more apparent in the FAS data than in the time-history data.

The CNN is a kind of machine learning (ML) methods, which have been actively developed for pattern recognition and feature extraction in recent decades (Hubel and Wiesel 1968, LeCun *et al*. 1989, 2015, Erhan *et al*. 2009, Scherer *et al.* 2010). Early researches on the application of ML in SHM focused on the ML algorithms like neural network (NN), sparse coding, and autoencoder. Many works on the ML algorithms for the SHM demonstrated their impressive ability of feature extraction for the damage identification. Yun and Bahng (2000) and Lee *et al*. (2005) presented NN-based approaches to damage identification for beam and frame structures using the modal properties. Min *et al*. (2012) used the NN for damage identification at bolted joints using mechanical impedance signals. Guo *et al*. (2014) employed a sparse coding to enhance the

*Corresponding author, Associate Professor
E-mail: zhanghongmei@zju.edu.cn

ISSN: 1738-1584 (Print), 1738-1991 (Online)

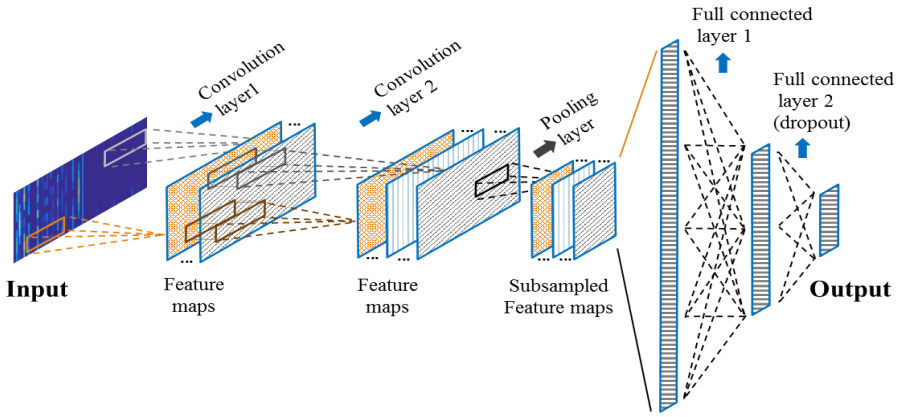


Fig. 1 An example of the CNN architecture

classification performance of NN for condition assessment of a bridge structure using acceleration data. Pathirage *et al.* (2018) presented an autoencoder based framework for structural damage identification using the modal data. Arangio and Bontempi (2015) introduced Bayesian neural networks for damage identification of a cable-stayed bridge from acceleration responses. With the enhancement of the computer technologies in hardware and software, deep learning (DL) algorithms become hot subjects in ML. Deep learning is a class of ML algorithms which typically consist of multiple layers of nonlinear processing units for feature extraction and transformation (Deng and Yu 2014). The CNN is the most famous DL algorithm for the outstanding performance in the computer vision. It generally includes convolution, pooling, and fully-connected layers. Abdeljaber *et al.* (2017) applied the CNN to damage identification in a steel frame. Cha *et al.* (2017) presented a CNN-based algorithm for crack detection on a surface of concrete structure. Kim *et al.* (2018) also proposed a CNN-based methodology for identifying concrete cracks from crack-like noisy patterns. Lin *et al.* (2017) proposed a CNN to estimate the damage locations and severities in a beam structure using acceleration time-history responses.

A numerical simulation study was carried out for multiple damage identification in hangers using a CNN with ambient wind vibration data. Bridge responses were generated using the simulated wind forces on the bridge under the daily wind condition, and used as the training and testing samples for the CNN. The vector form intrinsic finite-element (VFIFE) method (Ting *et al.* 2004, Duan *et al.* 2014, 2018, 2019, Yuan *et al.* 2018) was used to speed up the response calculations in generating numerous samples for different wind speeds and damage states. The CNN analyses were carried out using TensorFlow machine learning frame work (Abadi *et al.* 2015). The performance of the current CNN using FAS data was evaluated in comparison with the other methods, such as a CNN using time-history data and a traditional NN using FAS for various damage states. Robustness of the present CNN were investigated under various observational noise levels and wind speeds.

2. Theory of convolutional neural networks

Convolutional neural networks (CNNs) are biologically-inspired variants of neural networks (NNs) for pattern recognition (Liang *et al.* 2018). It was inspired by the animal visual cortex which is sensitive to small sub-regions of the visual field. To cover the whole visual field, the sub-regions are tiled and the cells in the cortex operate as local filters (Hubel and Wiesel 1968). Inspired by the visual nervous system, Fukushima (1980) proposed an improved neural network model, which is considered the foundation of the present CNN. With the rapid improvement of computer performance in recent decades, various kinds of deep CNN have been proposed with multiple hidden layers for many different pattern recognition problems.

The CNN architecture used in this study is shown in Fig. 1, and is designed for identification of multiple damages in a tied-arch bridge. The present CNN consists of two convolution layers, a pooling layer, and two fully-connected layers. The training procedure has basically two steps: the forward propagation (FP) for calculating the loss and the back propagation (BP) for computing the gradients and updating the parameters in the network.

2.1 Architecture of the CNN

2.1.1 Convolution layers

In the CNN, the convolution layer is for operations that compute the convolutions of the input and a kernel. A convolution layer consists of input (the feature maps in the previous layer), convolution kernels, bias, an activation function, and output (the feature map extracted by the kernel). 2-D kernels are used in this study because the input for the example analyses is a 9*1024 matrix representing the frequency responses (Fourier amplitude spectra: FAS) at 9 locations on the bridge deck (as shown in Section 3, later).

The convolution operation in the CNN is defined as

$$S(i,j)=\sum_{m}\sum_{n} I(i+m,\, j+n)\,\mathrm{K}(m,n) \tag{1}$$

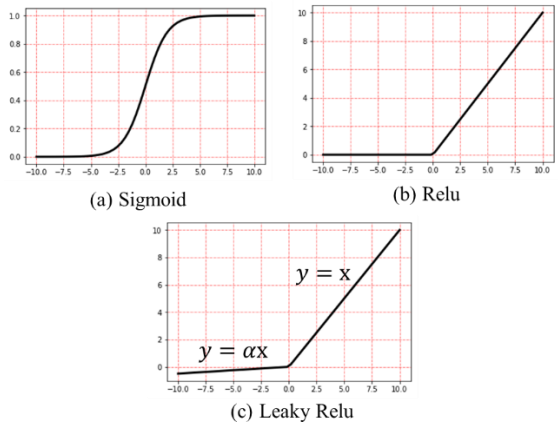


Fig. 2 Activation functions

where I is the input matrix and K is the kernel. A same kernel is applied along the length and width of an input to construct a feature map, whereas different kernels are used for different feature maps. Biases are added to the results of the convolution operation before the activation function, and zero padding is applied to preserve the information at the edges of the input matrix. The reasons for using the same kernel is twofold. On the one hand, in a matrix data like an image, values in a local group are often highly correlated, forming distinctive local motifs that are easily detected. On the other hand, some local image statistics are invariant to locations, which leads to the idea of using same filter (kernel) throughout an input matrix to detect the invariant feature (LeCun *et al*. 2015). The parameters (weights and biases) of kernels are updated through BP operations in the CNN.

2.1.2 Pooling layer

The pooling layer in the CNN down-samples the output of a convolution layer (Erhan *et al*. 2009, Hubel and Wiesel 1968, Scherer *et al*. 2010). The function of this layer is to semantically merge similar features so that small shifts can be withstood, noises can be reduced, the dimension of the representation can be reduced (Krizhevsky *et al*. 2012), and thereby effectively prevent overfitting. The most commonly used pooling in the CNN is the max pooling, which picks the maximum value in the region it covers. Max pooling is used in this paper and the size of the pooling region is taken as 1*4.

2.1.3 Fully-connected layers

The last few layers in the CNN are usually taken as fully-connected layers. Like the conventional NN, the receptive field for those layers covers every neuron in the next layer. First, the 3-dimensional features after the convolution and pooling operations are arranged into a vector. Then a fully-connected NN with a hidden layer is applied to map the features to the label space with the true values. Nonlinear activation functions are introduced to the neurons in each layer and the parameters, including synaptic weights and biases, are updated by the BP operation.

2.2 Activation functions, loss function, and Mini-batch

2.2.1 Activation functions

Activation functions are employed to the neurons in each layer to introduce nonlinear mapping. However, if the gradients of the activation functions in each layer are between 0 and 1, the gradients computed by the BP in the first few layers become extremely small. This phenomenon is called 'Vanishing Gradient', which may cause difficulty in training a deep NN with many layers as in the CNN. The most frequently used activation functions to solve this problem are Relu and Leaky Relu, whose slopes are always 1 for positive input and zero or a small positive value (α) for negative input as in Fig. 2. In this paper, Leaky Relu is chosen as the activation function for all the hidden layers, whereas Sigmoid is used for the last output layer. The value of α is a hyperparameter to be pre-determined by a parametric study.

2.2.2 Loss function

A loss function is used to evaluate the prediction error in the last output layer as

$$L = \frac{1}{N}\sum_i L_i = \frac{1}{N}\sum_i \left[\frac{1}{n}\sum_{j=1}^{n}(f_{ij} - y_{ij})^2\right] \quad (2)$$

where L_i is the loss for the ith sample, which is taken as a mean square error (MSE). N is the number of training samples in a batch. y_{ij} is the jth output representing the jth element-level damage index in the ith training sample in a batch, f_{ij} is the true value in the label, and n is the number of the elements for damage identification. The loss function is used in the training and validating processes.

2.2.3 Back propagation, mini-batch, and parameters

Back propagation is a way to compute the gradients in the CNN operations for updating the parameters. In this study, the mini-batch gradient descent algorithm (Dekel *et al*. 2012) is used, where the loss is computed for each mini-batch instead of all the training samples, so that the computational efficiencies can be improved drastically. Batch normalization (Ioffe and Szegedy 2015) and drop-out (Srivastava *et al*. 2014) are used to prevent overfitting in this study, as described in Appendix I. Parameters to be updated include weights and bias in the convolutional and fully-connected layers, and two parameters (γ, β) in the batch normalization. The initial values of the weights are

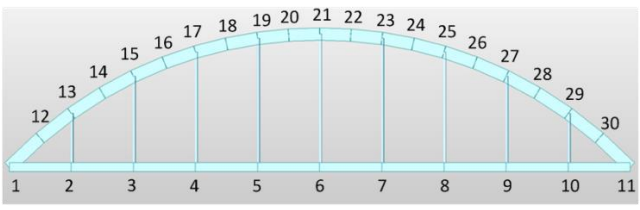


Fig. 3 A tied-arch bridge model

Table 1 Material properties and geometric parameter

Component	Arch rib	Main girder	Hanger
Material	C50	C40	1860 strand
Elastic modulus (Pa)	$3.45*10^{10}$	$3.25*10^{10}$	$1.95*10^{11}$
Poisson's ratio	0.2	0.2	0.3
Mass density (kg/m^3)	2549	2549	8005
Sectional area (m^2)	1.2	6	0.00567
Section moment of inertia (m^4)	0.2488	0.625	-
Section Height (m): H	1.4	1	-
Section Width (m): B	1.2	8	-

chosen using the He initial (He *et al*. 2015). Hyperparameters to be pre-determined as fixed values include the number of layers, the sizes of the convolution and pooling kernels, the number of neurons in each fully-connected layer, mini-batch size, learning rate, and retaining probability for drop-out.

3. Numerical example

3.1 Modeling of the arch bridge

A tied-arch bridge with 9 vertical hangers is considered in this numerical simulation study. The main span is 70m and the rise of arch is 15 m. A 2-D structural model was built using the vector form intrinsic finite-element (VFIFE) method as in Fig. 3, which was developed and verified in various example analyses for dynamic interaction problems (Duan *et al*. 2018), progressive collapse (Duan *et al*. 2014), and crack propagation (Duan *et al*. 2017). The material properties and geometric parameters are listed in Table 1. In this VFIFE model, the arch and girder elements behave as beams, whereas the hangers behave as cables with tensions due to the static load and the dynamic response. The bridge model consists of 30 beam elements and 9 cable elements.

3.2 Simulation of wind forces

To simulate the realistic bridge vibration, typical daily wind forces are considered in this study. Vertical acceleration responses of the bridge are computed for the lift force. The lift wind forces are calculated along the bridge as (Simiu and Scanlan 1996)

$$F_L(x,t) = -\frac{1}{2}\rho\bar{U}^2 B\left[2C_L(\alpha)\chi_L\frac{u(\mathrm{x,t})}{\bar{U}} + (C_L'(\alpha)+\frac{H}{B}\mathrm{C}_D)\chi_L\frac{w(\mathrm{x,t})}{\bar{U}}\right] \quad (3)$$

where $\bar{U}$ denotes the mean horizontal wind speed; $u(\mathrm{x,t})$ and $w(\mathrm{x,t})$ denotes the simulated fluctuating wind velocity components in the lateral and vertical directions; H denotes the height and B denotes the width of the girder (or arch); χ_L denotes aerodynamic admittances; C_L, C_D, and $C_L'(\alpha)$ denote the static wind coefficients. In this study, χ_L, C_L, C_D, and $C_L'(\alpha)$ are taken as 1.0, -0.320, 0.943 and 0.129 (Wang 2018). The procedure for the wind velocity simulation is summarized in Appendix II.

Daily wind speeds between 5 and 10 m/s were mainly considered in this study. For training sets, three cases of mean wind speed $\bar{U}$ are considered: 5, 8 and 10 m/s. For each mean wind speed, 50 different sets of wind forces were computed at 18 nodes on the bridge (anchorage points of 9 cables on the deck and arch) considering the spatial correlation of the wind velocity fluctuations.

Fig. 4 shows example time-histories of the simulated vertical wind velocity at different nodes on the bridge deck. The time-histories of Node 2 and 3 (which are 7 m apart) look similar, and they are quite different from the one at Node 10 (which is 56 m from Node 2), due to the spatial correlation of wind velocities. For testing, 150 sets of wind forces were newly generated for 3 wind speeds considered in the training sets, and another 150 sets were simulated for 3 other wind speeds of 3, 7 and 15 m/s.

3.3 Simulation of bridge response

3.3.1 Time-history of acceleration responses

The vertical acceleration responses of the bridge shown in Fig. 3 were computed using the VFIFE method. The time step for the VFIFE was taken as $5\times10^4 s$, to ensure the stability in the response computation, whereas the responses to be used as training samples were down-sampled to 100 Hz. The modal properties of this VFIFE analysis were obtained from the computed responses using the frequency domain decomposition (FDD), because the global stiffness matrix was not constructed in this method. The modal properties were found to be almost identical to those of a FE analysis using Midas (2017) as in Table 2 and Fig. 5. The time duration for each response is 20.48 seconds with 2048 data points. Acceleration response data at Nodes 2-10 on the main girder were used for CNN-based damage identification.

3.3.2 Fourier amplitude spectra of the responses

As the change in the modal properties reveals the degradation in the structure more clearly, time-history responses were converted into the frequency domain. With a time-history data of 20.48 s, the frequency resolution of the fast Fourier transform (FFT) results was 0.049 Hz.

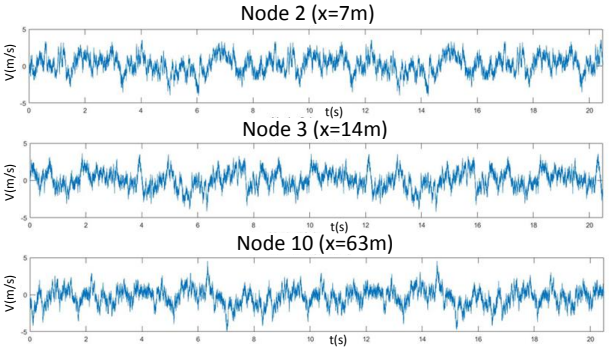


Fig. 4 Examples of vertical wind velocities at 3 nodes on the deck ($\bar{U}$=7m/s)

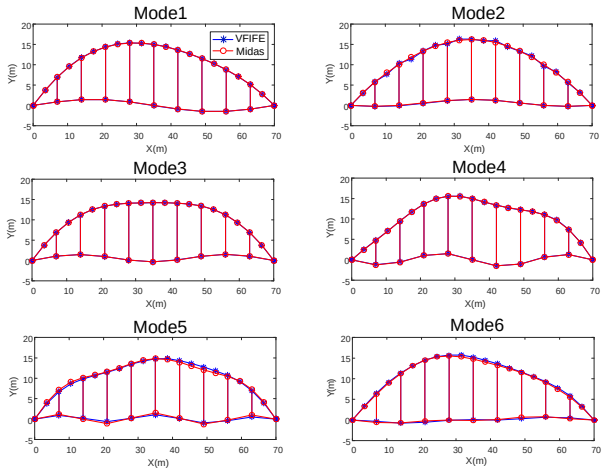


Fig. 5 Mode shapes of the-arch bridge

Table 2 Natural frequencies (cycle/sec) and MAC values

Mode numbers	Frequencies by VFIFE	Frequencies by Midas	Relative Errors	MAC
1	1.465	1.469	0.27%	1.000
2	2.542	2.531	0.43%	1.000
3	4.237	4.223	0.33%	1.000
4	5.912	5.893	0.32%	1.000
5	9.259	9.186	0.79%	0.963
6	9.308	9.295	0.14%	0.919

Notes: MAC indicates the modal assurance criteria

To reduce the leakage of the spectrum and eliminate the effect of random noises, the moving Hanning windows w(t) shown in Eq. (6), were applied with a window (T) of 5.12 s and a time shift of 1.28 s before the FFT (Chen and Mei 2010), and the results of the FAS were averaged.

$$w(t)=\begin{cases}\dfrac{1-\cos(2\pi t/T)}{2}, & 0\le t\le T\\ 0, & t<0 \text{ or } t>T\end{cases} \tag{4}$$

The FAS at 9 nodes were used for the input matrix to the CNN, whose size was 9*1024. Min-max normalization (Eq. (7)) was carried out on each matrix to keep the FAS amplitudes at the same level, which benefitted the CNN training, as

$$\hat{f}_{mn}=\frac{F_m(\omega_n)-\min\limits_{m,n}\{F_m(\omega_n)\}}{\max\limits_{m,n}\{F_m(\omega_n)\}-\min\limits_{m,n}\{F_m(\omega_n)\}} \tag{5}$$

where $F_m(\omega_n)$ are the FAS for the response at the *m*th node and ω_n; $\hat{f}_{mn}$ is the normalized value of $F_m(\omega_n)$; and $\max\limits_{m,n}\{F_m(\omega_n)\}$ and $\min\limits_{m,n}\{F_m(\omega_n)\}$ are the maximum and minimum values across two indices of m and n. The flow chart of the sample generation is shown in Fig. 6.

3.3.3 Simulation of different damage

Damage in a hanger is represented by a decrease in the cross-sectional area. Accordingly, the elemental-level damage index is defined as the ratio of the decrease in the cross-section. In the simulation, element-level damage indices for 9 hangers were taken to be in the range of 0.0 to 0.5, while the maximum number of damaged hangers was limited to 3.

3.4 CNN architecture

The CNN models used in this study were built with TensorFlow, which is an open-source software library developed by Google (Abadi *et al.* 2015). A numerical simulation study was conducted using a high-performance cluster at the Institute of Transportation Engineering, Zhejiang University. The cluster contained five servers, including one management node with 128-Gb of memory and four computing nodes with 24-Gb of memory. The performance of each node was boosted by hyper-threading. We used Python 2.7 and TensorFlow 1.8.0 in this study. Multiple damages that occurred in different hangers of the tied-arch bridge model were identified.

The vertical acceleration responses (FAS) at 9 nodes were arranged as an input matrix to the CNN for each damage case. To train the CNN, the labels in each damage case were assigned with the damage values in the last output layer. Each label is a vector with size 9 representing the damage indices (0 – 0.5). The loss function at the last output layer is the mean square error (MSE: Eq. (4)), which is most widely used in regression problems.

For training, damage indices were taken as one of 0, 0.1, 0.2, 0.3, 0.4, and 0.5. Assuming damages occurred at a maximum of 3 hangers, 11450 damage cases are generated including all possible combinations of element damage indices. 10000 randomly selected cases were used for training, and the rest were used for validation. Training was carried out for a mini-batch consisting of 128 sample cases selected randomly from the 10000 cases at each iteration. Table 3 shows the parameters for the training process. The size of each convolution kernel was chosen as 3*7, considering the dimension of the input matrix of 9*1024. Similarly, the size of the kernel for the pooling layer was taken as 1*4.

4. Numerical results and discussions

4.1 Training and testing results

The loss was calculated in terms of MSE (Eq. (4)) for 1450 sets of validation samples during training, and BP was conducted once in each iteration with a mini-batch. The loss for validation was found to be as small as 1.86×10^{-4} after training as shown in Fig. 7. It took about 510 min. for training with 10000 iterations by the computer cluster at Zhejiang University described in Section 3.4.

Table 3 CNN parameters for training process

Layers	Input size	Activation functions	Output size	Kernel size:	Remarks
Convolution Layer 1	[9,1024,1]* [9,2048,1]**	Leaky relu (α=0.005)	[9,1024,32]* [9,2048,32]**	32*[3,7]	Padding: same Strides:[1,1,1,1]
Convolution Layer 2	[9,1024,32]* [9,2048,32]**	Leaky relu (α=0.005)	[9,1024,64]* [9,2048,64]**	64*[3,7]	Padding: same Strides:[1,1,1,1]
Pooling Layer 1	[9,1024,64]* [9,2048,64]**	Leaky relu (α=0.005)	[9,256,64]* [9,512,64]**	[1,4]	Max pooling Padding: same
Full-connected Layer 1	[9*256*64]* [9*512*64]**	Leaky relu (α=0.005)	[1024]	-	-
Full-connected Layer 2	[1024]	Sigmoid	[9]	-	Retaining probability: 0.5

Notes: * for input matrix with FAS response; ** for input matrix with time-history response; and mini-batch size = 128, learning rate = 1*10-5, optimizer = Adam, and max iterations = 10000

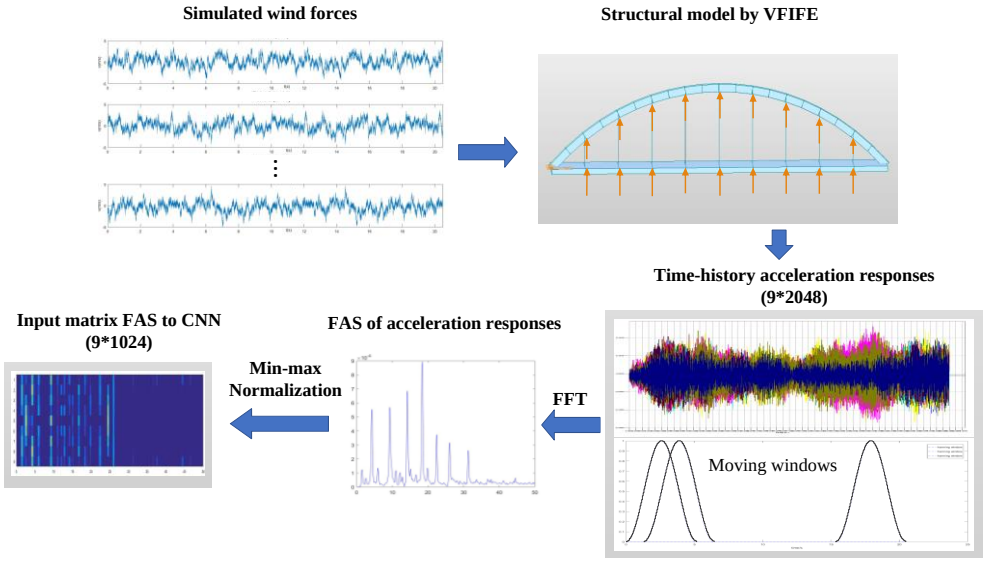


Fig. 6 Procedure for generating samples

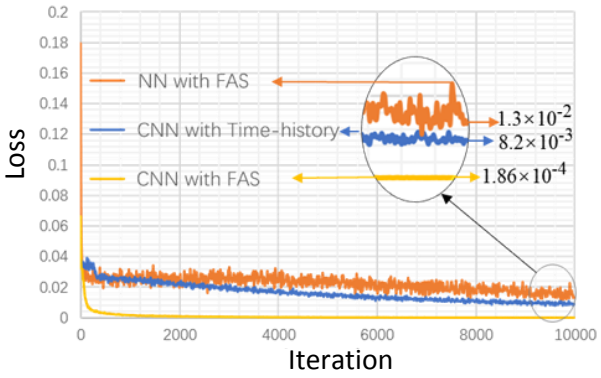


Fig. 7 Loss vs. iteration for CNNs and NN

Testing was primarily carried out for two wind force data sets with different mean wind speeds. Testing Set 1 included the newly generated wind forces but at the same mean wind speeds $\bar{U}$ used in the training set. Testing Set 2 included wind forces at different wind speeds: 3, 7 and 15 m/s. The damage indices for the testing samples were taken differently from those for training examples as 0, 0.05, 0.15, 0.25, 0.35, and 0.45. For comparative evaluation of the current CNN, the damage conditions were classified into 3 damage states: minor, moderate, and severe. The maximum damage index of hangers for minor damage were in the range 0-0.1. The maximum damage index in the moderate and severe damage states were taken as 0.1-0.3 and 0.3-0.5 respectively. For instance, if the label for one sample is [0.1,0,0,0,0,0.3,0,0,0.2], the damage state for this sample is moderate damage as the maximum damage index is 0.3, which is in the range 0.1-0.3. Each testing set consisted 17500 sample cases, including 2500 for the minor damage state and 7500 each for the other two damage states. Example results of the predicted damage indices are shown for various damage states in Fig. 8, which indicates excellent damage identification performance of the presented CNN.

Table 4 Average relative errors in the predictions for 3 classes of damage states

Damage States		Relative Errors								
		Testing Set 1 ($\overline{U}$ = 5, 8, or 10 m/s)			Testing Set 2 ($\overline{U}$ = 3, 7, or 15 m/s)			Testing Set 3 ($\overline{U}$ = 25 or 45 m/s)		
		without noise	With 10% noise	With 20% noise	without noise	With 10% noise	With 20% noise	without noise	With 10% noise	With 20% noise
Minor Damage	Ave	0.014	0.013	0.030	0.015	0.013	0.031	0.012	0.010	0.024
	Max	0.047	0.052	0.087	0.049	0.055	0.093	0.037	0.030	0.078
	SD	0.006	0.006	0.013	0.006	0.006	0.014	0.005	0.005	0.012
Moderate Damage	Ave	0.015	0.016	0.053	0.015	0.016	0.055	0.015	0.016	0.049
	Max	0.044	0.051	0.151	0.047	0.050	0.148	0.038	0.040	0.117
	SD	0.005	0.005	0.017	0.005	0.005	0.018	0.005	0.005	0.016
Severe Damage	Ave	0.013	0.014	0.037	0.014	0.015	0.039	0.014	0.015	0.033
	Max	0.056	0.068	0.141	0.056	0.069	0.142	0.048	0.049	0.106
	SD	0.005	0.006	0.016	0.005	0.006	0.018	0.005	0.006	0.014
Average errors		0.014	0.015	0.043	0.014	0.015	0.045	0.014	0.015	0.039
Average SD		0.005	0.006	0.018	0.005	0.006	0.020	0.005	0.006	0.018

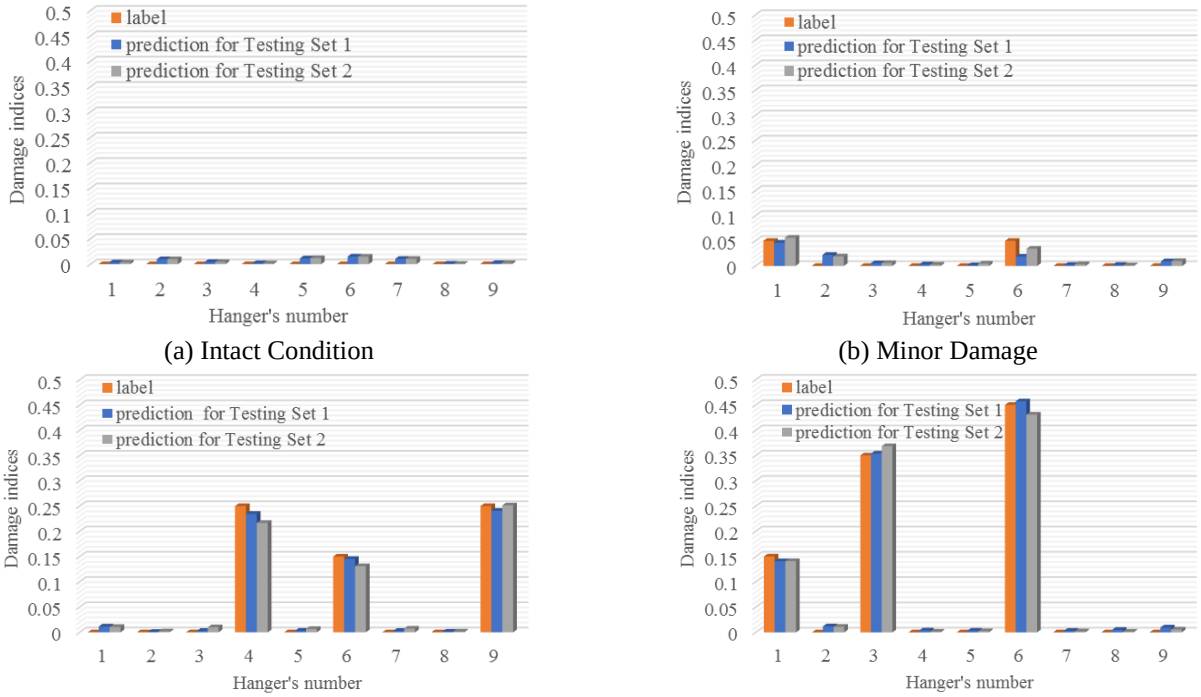
Notes: Ave and Max indicate the average and maximum errors; SD indicates the standard deviation; noise levels are the same in the training and testing cases; and training are using the FAS data at $\overline{U}$ =5, 8, and 10 m/s

(a) Intact Condition (b) Minor Damage

(c) Moderate Damage (d) Severe Damage

Fig. 8 Examples of predicted damage indices for various damage states (without observation noise)

The average prediction error results are summarized for various damage states and testing sets in Table 4. They are the average, maximum, and coefficient of variation of the error for a large number of testing samples considered in each damage state. The prediction errors for the *i*th testing sample are defined in terms of the root mean square (RMS)



(a) Intact Condition (b) Minor Damage

(c) Moderate Damage (d) Severe Damage

Fig. 9 Examples of predicted damage indices with 10% observation noise in RMS

of the relative values as

$$Error_i = \sqrt{\sum_{j=1}^{9}(f_{ij} - y_{ij})^2 / \sum_{j=1}^{9}(1 - f_{ij})^2} \qquad (6)$$

where y_{ij} is the predicted damage index for the jth hanger for the ith sample, and f_{ij} is the corresponding label (true) value. In Eq. (8), $Error_i$ is taken as the relative error of the safety index (1-y_{ij}) rather than the damage index (y_{ij}), because f_{ij} is zero for the cases without damages. The accuracy of the predicted damage indices is consistently excellent for various damage states. The results for Testing Sets 1 and 2 in Table 4 show that the average relative prediction errors are less than 0.02, the coefficient variations are less than 0.43, and the maximum error is less than 0.06, if the observation noises are not considered. The above results indicate excellent performance of the current CNN with FAS data for multiple damage identification.

Table 5 Traditional neural network parameters for training process

Layers	Size	Activation functions
Input	[9216]	Sigmoid
Hidden layer 1	[18432]	Sigmoid
Hidden layer 2	[1024]	Sigmoid
Output	[9]	-

Notes: Mini-batch size =128, retaining probability for hidden layer 2 = 0.75, learning rate =1*10^{-5}, optimizer = Adam, and max iteration =10000

4.2 Comparison with other ML-based methods

Two different neural network methods are considered for damage identification for the purpose of comparison. One is a CNN with input of time-history responses, and the other is a traditional NN with input of FASs.

4.2.1 CNN with time-history input

Acceleration response time histories were used as the input to the CNN. The size of each input was 9*2048, which is two times larger than the FAS data. The architecture of the CNN model basically remained unchanged except for the double size of the first fully-connected layer. The converging processes of the loss functions with iterations are also shown in Fig. 7. After training with 10000 iterations, the loss stays around 0.0082, which is almost 50 times larger than the previous value (1.86×10^{-4}) using FAS, which shows better efficiency of the proposed method using the FAS data. Besides, Table 6 shows the prediction error of CNN with time-history input for Testing Set 2. When using time-history input, both the average error and standard variance increase compared with the performance of CNN with FAS input. When 10% noises are introduced, CNN with FAS still have the least meanerror and standard deviation. The results may be caused by the fact that FAS data show the dynamic modal properties more clearly than the time-history data, so the changes in the properties related to damages can be identified more effectively.

Table 6 Average relative errors in the predictions for 3 ML-based Methods

Damage States		Relative Errors for testing Set 2 ($\overline{U}$ = 3, 7, or 15 m/s)					
		Without noise			With 10% noise		
		CNN with FAS	CNN with time-history	NN with FAS	CNN with FAS	CNN with time-history	NN with FAS
Minor Damage	Ave	0.015	0.078	0.083	0.013	0.082	0.089
	Max	0.049	0.271	0.153	0.055	0.276	0.167
	SD	0.006	0.062	0.011	0.006	0.063	0.017
Moderate Damage	Ave	0.015	0.089	0.090	0.016	0.094	0.084
	Max	0.047	0.248	0.168	0.050	0.252	0.137
	SD	0.005	0.042	0.018	0.005	0.044	0.015
Severe Damage	Ave	0.014	0.128	0.155	0.015	0.145	0.136
	Max	0.056	0.280	0.269	0.069	0.302	0.272
	SD	0.005	0.048	0.004	0.006	0.050	0.040
Average errors		0.014	0.104	0.117	0.015	0.114	0.107
Average SD		0.005	0.052	0.044	0.006	0.056	0.038

4.2.2 Neural network with FAS

The traditional neural network (NN) is used with the input of FAS data for damage identification. The NN consists of 2 hidden layers with 18432 and 1024 neurons, respectively. The parameters of the NN are shown in Table 5. The same training, validation and testing sets are used as in the CNN. After the same number (10000) of iterations, the loss of validation set reached 0.013 (Fig. 7), which is about 80 times larger than the result of the CNN. After 50000 iterations, the loss reached 0.0012, which is still 7 times larger than the loss by the CNN, indicating the slow convergence and inefficient performance of the traditional NN for identification of multiple damages compared with the CNN. Table 6 also indicates that the average value and standard deviation of the prediction error of CNN with FAS is much smaller than the other two methods.

4.3 Robustness to observation noises and wind speeds

4.3.1 Observation noise effect to CNN

White Gaussian noises are introduced to the acceleration responses during the numerical simulation. The signal-to-noise ratios (SNRs) were taken as 20 and 14db, which are equivalent to the RMS levels of 10% and 20%, respectively. The procedures for generating the training, validation, and testing samples are the same as those in the noise-free condition. The CNN architecture remains unchanged, and it is also trained with 10000 iterations. Example cases of the predicted damages shown in Fig. 9 indicate that the damage locations and severities have been identified excellently under the observation noises of 10% at the RMS level. As listed in Table 4, the average relative errors under 10% and 20% noises are less than 0.02 and 0.05, respectively. They are slightly larger than the errors without noises, but still at an acceptable level for the purpose of damage identification and quantification. The results indicate that noises do hinder the CNN learning process, but the identification accuracy remains acceptable.

4.3.2 Prediction with response data under severer wind

The CNN trained with the daily wind condition was tested for response data under severer conditions with mean wind speeds of 25 m/s and 45 m/s. The prediction errors are also listed as Testing Set 3 in Table 4. The average prediction errors for 3500 sample cases were found to be at almost the same level as those of Testing Sets 1 and 2 under the daily wind speeds. This is owing to the monotonically decaying shapes of the wind spectra (Fig. 11) in the daily wind as well as in the severer wind in the range over 1 Hz, where the structural natural frequencies are located. The wind forces are very much wide-banded, similar to white noise. Furthermore, min-max normalization is applied to the FAS of response. So, the overall amplitudes' changes of acceleration responses are normalized. The results indicate that the CNN trained under the daily wind condition can be effectively utilized for damage identification under severer wind conditions.

4.3.3 Noise injection learning

Noise injection learning is a scheme of training with noisy input data to reduce the noise effect in the testing stage (Matsuoka 1992, Yun and Bahng 2000). The prediction errors for 6 cases with different noise conditions in the training and testing data sets are summarized in Table 7, which shows considerable improvement in prediction error, particularly the robustness to the different noise levels between the training and testing data.

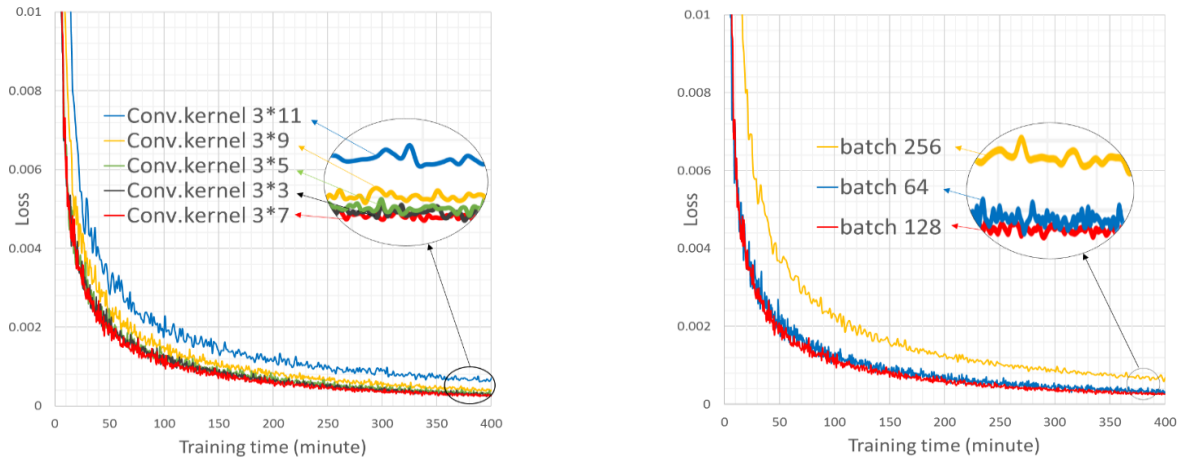
4.4 Selection of hyperparameters

For deep neural networks like CNN, the selection of the hyperparameters is one of the biggest challenges (Miikkulainen *et al.* 2017). In CNN, the sizes of the convolution kernel and mini-batch are essential hyper-

Table 7 Results of noise-injection learning

Damage states		Relative Errors					
		Training without noise			Training with 10% noise		
		Testing Without noise	Testing With10% noise	Testing with 20% noise	Testing Without noise	Testing with 10% noise	Testing with 20% noise
Minor Damage	Ave	0.014	0.016	0.065	0.013	0.013	0.060
	Max	0.047	0.056	0.126	0.046	0.055	0.134
	SD	0.006	0.007	0.018	0.006	0.006	0.020
Moderate Damage	Ave	0.015	0.017	0.084	0.015	0.016	0.087
	Max	0.044	0.061	0.180	0.041	0.050	0.168
	SD	0.005	0.006	0.023	0.005	0.005	0.021
Severe Damage	Ave	0.013	0.015	0.062	0.014	0.015	0.061
	Max	0.056	0.076	0.176	0.069	0.069	0.155
	SD	0.005	0.006	0.027	0.006	0.006	0.023
Average error		0.014	0.016	0.072	0.014	0.015	0.071
Average SD		0.005	0.006	0.026	0.005	0.006	0.025

Notes: Mean wind speed is one of 5, 8, and 10 m/s in training and 3, 7, and 15 m/s in testing



(a) Training with same batch size of 128 (b) Training with same kernel size of 3*7

Fig. 10 Loss vs. iteration for different convolution kernel and batch sizes

parameters to be selected before training.There is not rigorous procedure for their selection. With increased kernel sizes, the feature extraction capability may be improved. However, a large number of the parameters need to be determined during the training and the computational complexity increases drastically. When the batch size is enlarged, it may be easier for the loss function to converge to the minimum. However, the computing time increases because more samples are to be processed in the batch. Based on the results of the parametric study shown in Fig. 10, the convolution kernel size is taken as 3*7 and the mini-batch size as 128.

5. Conclusions

A method using the convolutional neural network (CNN) is proposed for structural damage identification on hangers in a tied-arch bridge structure. The spatial and spectral information of the acceleration responses on the deck are used as the input to the CNN, which consists of Fourier amplitude spectra (FAS) of the wind-induced responses at multiple locations on the bridge deck. No special sensors are needed for local monitoring of the cables, such as elasto-magneto-electric sensors and guided wave sensors. No extensive pre-processing such as modal identification is required, and the size of the input matrix to the CNN can be

reduced by using FAS instead of the time-history data. Hierarchical processes are presented for the construction of the CNN model and the localization and quantification of multiple damages. Discussions are presented on the hyperparameter optimization related to the overfitting problem and the convergence and accuracy of damage identification. Performance of the proposed CNN is investigated through a numerical simulation study under daily wind conditions. Comparative studies are carried out for various damage states, wind speeds, and observation noise. The results of the present study are summarized as follows:

- The overall performance of the present CNN is excellent for various damage states. The mean values of the relative prediction errors are less than 0.02 for the testing sample cases without the observation noises,

which indicates the excellent generalization performance of the CNN. In addition, the maximum relative errors are found to be less than 0.06, showing the consistently high prediction accuracy of this method.

- The CNN performance is very robust against the observation noise. For samples with 10% and 20% noises, average relative errors are less than 0.02 and 0.06 respectively, which are slightly larger but acceptable for damage localization and quantification.

Noise injection learning is proven to be effective in improving the robustness to the difference of the observation noise level in the training and testing data.

- The proposed CNN trained under the daily wind condition was found to be very robust to severer wind conditions. The average and maximum values of the prediction error remained almost the same.
- Compared with the traditional NN with FAS and the CNN with time-history data, the present CNN with FAS learns faster and achieves a smaller error within the same training time, owing to the better capability of the FAS data in representing important features for damage identification such as modal properties.

Future research and applications are suggested in the following areas: experimental validation, application to real cable-supported structures, and other kinds of cable damage such as wire breakage and corrosion. Besides, changes in the environmental condition such as temperature and traffic loads may cause significant effects on the modal properties of the bridge. Further research needs to be carried out to develop proper methodologies to deal with those effects on the damage identification.

Acknowledgments

This research work was supported by the National Natural Science Foundation of China (U1709216, 51578419, 51522811, 51478429, and 90915008), the National Key R&D Program of China (2017YFC0806100), and the Fundamental Research Funds for the Central Universities (2015XZZX004-28).

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E. and Zheng, X. (2015), TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems.

Abdeljaber, O., Avci, O., Kiranyaz, S., Gabbouj, M. and Inman, D. J. (2017), "Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks", *J, Sound Vib.*, **388**, 154-170. https://doi.org/10.1016/j.jsv.2016.10.043.

Arangio, S. and Bontempi, F. (2015), "Structural health monitoring of a cable-stayed bridge with Bayesian neural networks", *Struct. Infrastruct. Eng.*, **11**(4), 575-587. https://doi.org/10.1080/15732479.2014.951867.

Cao, Y., Xiang, H. and Zhou, Y. (2000), "Simulation of stochastic wind velocity field on long-span bridges", *China Civil Eng. J.*, **126**(1), 1-6. https://doi.org/10.1061/(ASCE)0733-9399(2000)126:1(1).

Cha, Y.J., Choi, W. and Büyüköztürk, O. (2017), "Deep learning-based crack damage detection using convolutional neural networks", *Comput.-Aided Civil Infrastruct. Eng.*, **32**(5), 361-378. https://doi.org/10.1111/mice.12263.

Chen, C., Wu, W., Liu, C. and Lai, G. (2016), "Damage detection of a cable-stayed bridge based on the variation of stay cable forces eliminating environmental temperature effects", *Smart Struct. Syst.*, **17**(6), 859-880. http://dx.doi.org/10.12989/sss.2016.17.6.859.

Chen, K.F. and Mei, S.L. (2010), "Composite interpolated fast fourier transform with the hanning window", *IEEE T. Instrument.Measurement*, **59**(6), 1571-1579. DOI: 10.1109/TIM.2009.2027772.

Cho, S., Jo, H., Jang, S., Park, J., Jung, H.J., Yun, C.B., Spencer, B. F. and Seo, J.W. (2010), "Structural health monitoring of a cable-stayed bridge using wireless smart sensor technology: data analyses", *Smart Struct. Syst.*, **6**(5-6), 461-480. http://dx.doi.org/10.12989/sss.2010.6.5_6.461.

Cho, S., Lynch, J.P., Lee, J. and Yun, C.B. (2010), "Development of an automated wireless tension force estimation system for cable-stayed bridges", *J. Intel. Mat. Syst. Str.*, **21**(3), 361-376. https://doi.org/10.1177/1045389X09350719.

Cho, S., Yim, J., Shin, S.W., Jung, H., Yun, C.B. and Wang, M.L. (2013), "Comparative field study of cable tension measurement for a cable-stayed bridge", *J. Bridge Eng.*, **18**(8), 748-757. https://doi.org/10.1061/(ASCE)BE.1943-5592.0000421.

Dekel, O., Ran, G.B., Shamir, O. and Xiao, L. (2012), "Optimal distributed online prediction using mini-batches", *J. Mach. Learn. Res.*, **13**(1), 165-202.

Deng, L. and Yu, D. (2014), Deep learning: methods and applications. Foundations and Trends® in Signal Processing, **7**(3-4), 197-387.

Deodatis, G. (1996), "Simulation of ergodic multivariate stochastic processes", *J. Eng.Mech.*, **122**(8), 778-787. https://doi.org/10.1061/(ASCE)0733-9399(1996)122:8(778).

Ding, Y., An, Y. and Wang, C. (2016), "Field monitoring of the train-induced hanger vibration in a high-speed railway steel arch bridge", *Smart Struct. Syst.*, **17**(6), 1107-1127. http://dx.doi.org/10.12989/sss.2016.17.6.1107.

Duan, Y.F., He, K., Zhang, H., Ting, E.C., Wang, C.Y., Chen, S.K., and Wang, R.Z. (2014), "Entire-process simulation of earthquake-induced collapse of a mockup cable-stayed bridge by Vector Form Intrinsic Finite Element (VFIFE) method", *Adv. Struct. Eng.*, **17**(3), 347-360. https://doi.org/10.1260/1369-4332.17.3.347.

Duan, Y., Tao, J., Zhang, H., Wang, S. and Yun, C. (2019), "Real‐time hybrid simulation based on vector form intrinsic finite element and field programmable gate array", *Struct. Control Health Monit.*, **26**(1), e2277. https://doi.org/10.1002/stc.2277.

Duan, Y.F., Wang, S.M., Wang, R.Z., Wang, C., Shih, J.Y. and Yun, C.B. (2018), "Vector form intrinsic finite-element analysis for train and bridge dynamic interaction", *J. Bridge Eng.*, **23**(1), 04017126. https://doi.org/10.1061/(ASCE)BE.1943-5592.0001171.

Duan, Y.F., Wang, S.M., Wang, R.Z., Wang, C.Y. and Ting, E.C. (2017), "Vector form intrinsic finite element based approach to simulate crack propagation", *J. Mechanics*, **33**(6), 1-16. https://doi.org/10.1017/jmech.2017.85.

Duan, Y., Wang, S. and Yau, J. (2018), "Vector form intrinsic finite element method for analysis of train–bridge interaction problems considering the coach-coupler effect", *Int. J. Struct. Stab. Dynam.*, 1950014. https://doi.org/10.1142/S0219455419500147.

Duan, Y.F., Zhang, R., Dong, C.Z., Luo, Y.Z., Or, S.W., Zhao, Y., and Fan, K.Q. (2016), "Development of Elasto-Magneto-Electric (EME) sensor for in-service cable force monitoring", *Int. J. Struct. Stab. Dynam.*, **16**(4), S68-S78. https://doi.org/10.1142/S0219455416400162.

Duan, Y.F., Zhang, R., Zhao, Y., Or, S.W., Fan, K.Q. and Tang, Z. F. (2011), "Smart elasto-magneto-electric (EME) sensors for stress monitoring of steel structures in railway infrastructures", *J. Zhejiang Univ. –Sci. A*, **12**(12), 895-901.

Duan, Y.F., Zhang, R., Zhao, Y., Or, S.W., Fan, K.Q. and Tang, Z. F. (2012), "Steel stress monitoring sensor based on elasto-magnetic effect and using magneto-electric laminated composite", *J. Appl. Phys.*, **111**(7), 68. https://doi.org/10.1063/1.3679420.

Erhan, D., Bengio, Y., Courville, A. and Vincent, P. (2009), "Visualizing higher-layer features of a deep network", *University of Montreal*, **1341**(3), 1.

Fukushima, K. (1980), "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position", *Biol. Cybern.*, **36**(4), 193-202.

Guo, J., Xie, X., Bie, R. and Sun, L. (2014), "Structural health monitoring by using a sparse coding-based deep learning algorithm with wireless sensor networks", *Personal & Ubiquitous Computing*, **18**(8), 1977-1987.

He, K., Zhang, X., Ren, S. and Sun, J. (2015), "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification", *International conference on computer vision*, 1026-1034.

Hubel, D.H. and Wiesel, T.N. (1968), "Receptive fields and functional architecture of monkey striate cortex", *J. Physiology*, **195**(1), 215-243. https://doi.org/10.1113/jphysiol.1968.sp008455.

Ioffe, S. and Szegedy, C. (2015), "Batch normalization: accelerating deep network training by reducing internal covariate shift", *International conference on machine learning*, 448-456.

Jang, S., Jo, H., Cho, S., Mechitov, K., Rice, J.A., Sim, S., Jung, H., Yun, C., Spencer, B.F. and Agha, G.A. (2010), "Structural health monitoring of a cable-stayed bridge using smart sensor technology: deployment and evaluation", *Smart Struct. Syst.*, **6**(5), 439-459. http://dx.doi.org/10.12989/sss.2010.6.5_6.439.

Jin, S., Cho, S., Jung, H., Lee, J. and Yun, C. (2014), "A new multi-objective approach to finite element model updating", *J. Sound Vib.*, **333**(11), 2323-2338. https://doi.org/10.1016/j.jsv.2014.01.015.

Kaimal, J.C., Wyngaard, J.C., Izumi, Y. and Cote, O.R. (1972), "Spectral characteristics of surface‐layer turbulence", *Quarterly J. Roy. Meteorol. Soc.*, **98**(417), 563-589. https://doi.org/10.1002/qj.49709841707.

Kim, H., Ahn, E., Shin, M. and Sim, S. (2018), "Crack and noncrack classification from concrete surface images using machine learning", *Struct. Health Monit.*, 147592171876874. https://doi.org/10.1177/1475921718768747.

Kim, J., Ho, D., Nguyen, K., Hong, D., Shin, S.W., Yun, C.B. and Shinozuka, M. (2013), "System identification of a cable-stayed bridge using vibration responses measured by a wireless sensor network", *Smart Struct. Syst.*, **11**(5), 533-553. http://dx.doi.org/10.12989/sss.2013.11.5.533.

Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2012), *ImageNet Classification with Deep Convolutional Neural Networks.* Paper presented at the neural information processing systems.

LeCun, Y., Bengio, Y. and Hinton, G. (2015), "Deep learning", *Nature*, **521**(7553), 436.

LeCun, Y., Boser, B.E., Denker, J.S., Henderson, D., Howard, R. E., Hubbard, W. and Jackel, L.D. (1989), "Backpropagation applied to handwritten zip code recognition", *Neural Comput.*, **1**(4), 541-551.

Lee, J., Lee, J.W., Yi, J., Yun, C.B. and Jung, H.Y. (2005), "Neural networks-based damage detection for bridges considering errors in baseline finite element models", *J. Sound Vib.*, **280**(3), 555-578. https://doi.org/10.1016/j.jsv.2004.01.003.

Liang, Z.J., Liao, S.B. and Hu, B.Z. (2018), "3D convolutional neural networks for dynamic sign language recognition", *Comput. J.*, **61**(11), 1724-1736. https://doi.org/10.1093/comjnl/bxy049.

Lin, Y., Nie, Z. and Ma, H. (2017), "Structural damage detection with automatic feature-extraction through deep learning", *Comput.-Aided Civil Infrastruct. Eng.*, **32**(12), 1025-1046. https://doi.org/10.1111/mice.12313.

Matsuoka, K. (1992), "Noise injection into inputs in back-propagation learning", *Syst. Man Cybernetics*, **22**(3), 436-440. DOI: 10.1109/21.155944.

Midas (2017), MIDAS Information Technology Co., Ltd., Korea. http://www.MidasUser.com.

Miikkulainen, R., Liang, J.Z., Meyerson, E., Rawal, A., Fink, D., Francon, O., Raju, B., Shahrzad, H., Navruzyan, A. and Duffy, N. (2017), Evolving Deep Neural Networks. *arXiv: Neural and Evolutionary Computing*, 293-312.

Min, J., Park, S., Yun, C.B., Lee, C. and Lee, C. (2012), "Impedance-based structural health monitoring incorporating neural network technique for identification of damage type and severity", *Eng. Struct.*, **39**, 210-220. https://doi.org/10.1016/j.engstruct.2012.01.012.

Pathirage, C.S.N., Li, J., Li, L., Hao, H., Liu, W. and Ni, P. (2018), "Structural damage identification based on autoencoder neural networks and deep learning", *Eng. Struct.*, **172**, 13-28. https://doi.org/10.1016/j.engstruct.2018.05.109.

Scherer, D., Muller, A. and Behnke, S. (2010), *Evaluation of pooling operations in convolutional architectures for object recognition*, Paper presented at the international conference on artificial neural networks.

Sekhar, A.S. (2004), "Crack identification in a rotor system: a model-based approach", *J. Sound Vib.*, **270**(4), 887-902. https://doi.org/10.1016/S0022-460X(03)00637-0.

Simiu, E. and Scanlan, R.H. (1996), Wind effects on structures: Fundamentals and application to design, *Book published by John Willey & Sons Inc*, *605*.

Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. (2014), "Dropout: a simple way to prevent neural networks from overfitting", *J. Mach. Learn. Res.*, **15**(1), 1929-1958.

Ting, E.C., Shih, C. and Wang, Y. (2004), "Fundamentals of a vector form intrinsic finite element: Part I. basic procedure and a plane frame element", *J. Mechanics*, **20**(2), 113-122. https://doi.org/10.1017/S1727719100003336.

Wang, S.M. (2018), *Dynamic analysis of wind-train-rail-long span bridge based on the vector form intrinsic finite element.* Zhejiang University.

Yim, J., Wang, M. L., Shin, S.W., Yun, C.B., Jung, H., Kim, J. and Eem, S. (2013), "Field application of elasto-magnetic stress

sensors for monitoring of cable tension force in cable-stayed bridges", *Smart Struct. Syst.*, **12**(3-4), 465-482. https://doi.org/10.12989/sss.2013.12.3_4.465.

Yuan, X., Chen, C., Duan, Y. and Qian, R. (2018), "Elastoplastic analysis with fine beam model of vector form intrinsic finite element", *Adv. Struct. Eng.*, **21**(3), 365-379. https://doi.org/10.1177/1369433217718984.

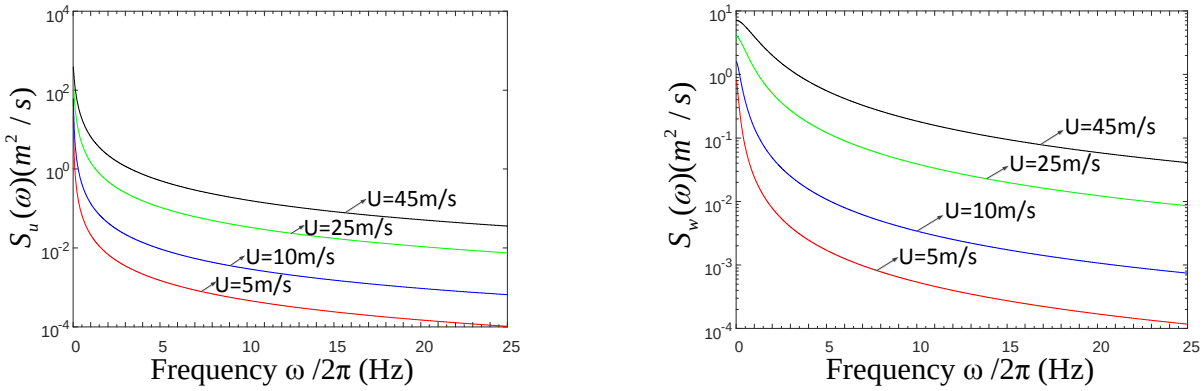
Yun, C.B. and Bahng, E.Y. (2000), "Substructural identification using neural networks", *Comput. Struct.*, **77**(1), 41-52. https://doi.org/10.1016/S0045-7949(99)00199-6.

Zhang, P., Tang, Z., Duan, Y., Yun, C.B. and Lv, F. (2018), "Ultrasonic guided wave approach incorporating SAFE for detecting wire breakage in bridge cable", *Smart Struct. Syst.*, **22**(4), 481-493. https://doi.org/10.12989/sss.2018.22.4.481.

Zhang, R., Duan, Y., Or, S.W. and Zhao, Y. (2014), "Smart elasto-magneto-electric (EME) sensors for stress monitoring of steel cables: design theory and experimental validation", *Sensors*, **14**(8), 13644-13660. https://doi.org/10.3390/s140813644.

Zhang, R., Duan, Y., Zhao, Y. and He, X. (2018), "Temperature compensation of Elasto-Magneto-Electric (EME) sensors in cable force monitoring using BP neural network", *Sensors*, **18**(7), 2176. https://doi.org/10.3390/s18072176.

HJ



(a) Horizontal wind velocity spectra (b) Vertical wind velocity spectra

Fig. 11 Power spectra of wind velocity under three different average wind velocities

Appendix I: Bath normalization and drop-out

Batch normalization (BN) is helpful in accelerating the training and improving the accuracy of both training and testing sets (Ioffe and Szegedy 2015). The BN operational is related to the mini-batch gradient descent. In BN, the input for an activation function is transformed to a standard Gaussian variable as

$$\hat{x}=\frac{x-\mu}{\sqrt{\sigma^2+\varepsilon}} \quad \text{and} \quad y=\gamma\hat{x}+\beta \tag{I-1}$$

where x and y are the input and the corresponding output of the BN operation; μ and σ are the mean and standard deviation for the mini batch; and ε is a small number to prevent the denominator from becoming zero. γ and β are the scale and shift parameters. At first, μ and σ are computed for each mini-batch, then y is obtained. γ and β are to be updated during training. In this study, the BN is applied to the activation functions in the convolution layers.

Dropout is a regularization method whose idea is randomly drop units form the neural networks during training (Srivastava *et al.* 2014). This technique is introduced to reduce the overfitting problem.

Appendix II: Simulation of wind velocities: $u(t)$ and $w(t)$

Time-histories of the horizontal and vertical wind velocity fluctuations at the *j*th point, $u_j(t)$ and $w_j(t)$, can be simulated from the wind spectra $S_u(\omega)$ and $S_w(\omega)$ considering the spatial coherence as (Cao et al. 2000, Deodatis 1996)

$$\begin{aligned} u_j(t)&=\sqrt{2(\Delta\omega)}\sum_{m=1}^{j}\sum_{l=1}^{N}\sqrt{S_u(\omega_{ml})}G_{jm}(\omega_{ml})\cos(\omega_{ml}t+\Phi_{ml}) \\ j&=1,2,...n \\ w_j(t)&=\sqrt{2(\Delta\omega)}\sum_{m=1}^{j}\sum_{l=1}^{N}\sqrt{S_w(\omega_{ml})}G_{jm}(\omega_{ml})\cos(\omega_{ml}t+\Phi_{ml}) \\ j&=1,2,...n \end{aligned} \tag{II-1}$$

$$G(\omega)=\begin{bmatrix} 1 & & & \\ C^1 & \sqrt{1-C^2} & & \\ ... & ... & ... & ... \\ C^{n-1} & C^{n-2}\sqrt{1-C^2} & ... & \sqrt{1-C^2} \end{bmatrix}, C=\exp(-\frac{\lambda\omega\Delta}{2\pi U(z)}) \tag{II-2}$$

$$\omega_{ml}=(l-1)\Delta\omega+\frac{m}{N}\Delta\omega,\ l=1,2,...,N \tag{II-3}$$

where $G_{jm}(\omega_{m\,l})$ is a cross-spectral density matrix representing the spatial coherence of $u(t)$ and $w(t)$ at ω between the *j*th and *m*th points, Δ is the uniform distance between adjacent points; λ is a constant value for the spatial coherence which is usually taken between 7 and 10; Φ_{ml} is a random phase angle with a uniform distribution in $0\text{-}2\pi$; $\Delta\omega$ is the sampling frequency; and N is the number of frequency points.

In this study, $S_u(\omega)$ and $S_w(\omega)$ are taken as Kaimal spectra (Kaimal *et al.* 1972) as

$$\begin{aligned} S_u(\omega)&=\frac{1}{2\pi}\frac{200f}{(1+50f)^{5/3}}\cdot\frac{U_*^2}{\omega/2\pi} \\ S_w(\omega)&=\frac{1}{2\pi}\frac{3.36f}{1+10f^{5/3}}\cdot\frac{U_*^2}{\omega/2\pi} \end{aligned} \tag{II-4}$$

where $f=\frac{\omega z}{2\pi U(z)}$; $U_*=\frac{KU(z)}{\ln(z/z_0)}$; $K=0.4$; $U(z)$ is the mean horizontal wind speed at z; z is the elevation of the deck above the ground; and z_0 is the ground roughness.

The power spectra of horizontal and vertical wind velocity for different average velocities are shown in Fig. 11.