

# Damage localization and quantification of a truss bridge using PCA and convolutional neural network

Jiajia Hao <sup>1a</sup>, Xinqun Zhu <sup>\*1</sup>, Yang Yu <sup>1b</sup>, Chunwei Zhang <sup>2c</sup> and Jianchun Li <sup>1c</sup>

<sup>1</sup> School of Civil and Environmental Engineering, Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW, Australia

<sup>2</sup> Multidisciplinary Center for Infrastructure Engineering, Shenyang University of Technology, Shenyang 110870, China

(Received March 28, 2022, Revised July 22, 2022, Accepted September 2, 2022)

**Abstract.** Deep learning algorithms for Structural Health Monitoring (SHM) have been attracting the interest of researchers and engineers. These algorithms commonly used loss functions and evaluation indices like the mean square error (MSE) which were not originally designed for SHM problems. An updated loss function which was specifically constructed for deep-learning-based structural damage detection problems has been proposed in this study. By tuning the coefficients of the loss function, the weights for damage localization and quantification can be adapted to the real situation and the deep learning network can avoid unnecessary iterations on damage localization and focus on the damage severity identification. To prove efficiency of the proposed method, structural damage detection using convolutional neural networks (CNNs) was conducted on a truss bridge model. Results showed that the validation curve with the updated loss function converged faster than the traditional MSE. Data augmentation was conducted to improve the anti-noise ability of the proposed method. For reducing the training time, the normalized modal strain energy change (NMSEC) was extracted, and the principal component analysis (PCA) was adopted for dimension reduction. The results showed that the training time was reduced by 90% and the damage identification accuracy could also have a slight increase. Furthermore, the effect of different modes and elements on the training dataset was also analyzed. The proposed method could greatly improve the performance for structural damage detection on both the training time and detection accuracy.

**Keywords:** convolutional neural network (CNN); damage detection; normalized modal strain energy change; principal component analysis (PCA)

## 1. Introduction

During the long service period, bridge infrastructures continuously suffer from both environmental influences such as eroding effects from the ground or the sea, and operational effects including the traffic (e.g., traffic-induced vibrations and traffic mass effects), temperature, and wind (Kim *et al.* 2018). Because of these effects, structural integrity is threatened all the time. Thus, it is of vital importance to monitor the health conditions of those major bridges, especially those newly constructed and aging bridges. Moreover, urgent damage detection tasks are often needed for those bridges which suffered from large-scale natural disasters like earthquakes and hurricanes. Huge losses on human life and property would have been avoided if effective monitoring systems had been set up on these bridges.

Damage detection has always been one main research interest in the structural health monitoring domain. The

vibration-based damage detection (VBDD), aiming at detecting the existence, location, and severity of structural damage, has attracted massive research interest in civil (Xu *et al.* 2017), mechanical (Aswal *et al.* 2021), and aircraft structures (Santos *et al.* 2016). The basic idea of VBDD is that structural damage induces changes in mechanical properties such as stiffness and mass. These mechanical property changes could be further indicated by the variations of vibration properties (e.g., natural frequencies and modal shapes). By analyzing the damage features extracted from the vibration data, structural damage can be detected, localized, and quantified. Compared to the traditional visual inspection and non-destructive testing (NDT) methods such as ultrasonic, acoustic and piezoelectric active sensing methods, the VBDD are superior in the following two aspects (Zhou *et al.* 2021), 1) they are suitable for even complex structures because they evaluate the overall structural condition rather than examining only the potential damaged local area like NDT methods; 2) Prices of the sensing system are quite acceptable.

Generally, the implementation of VBDD involves two procedures: feature extraction and feature discrimination. The former extracts damage sensitive features from the original measured responses while the latter discriminates features from damaged to healthy quantitatively. Various

\*Corresponding author, Ph.D., Professor,  
E-mail: Xinqun.Zhu@uts.edu.au

<sup>a</sup> Ph.D. Student

<sup>b</sup> Ph.D.

<sup>c</sup> Professor

damage features have been developed in the last three decades, including natural frequencies (Kordestani *et al.* 2021), mode shapes (Lee *et al.* 2021) and their derivatives such as modal strain energy (Talebpour *et al.* 2020), mode shape curvature (Cao *et al.* 2014), the flexibility curvature (Nick and Aziminejad 2021) and the energy of acceleration signal (Kordestani *et al.* 2018). Compared to natural frequencies, mode shapes provide spatial information and are less sensitive to environmental changes. Hence, in this paper, mode shapes and one of its derivatives, namely the modal strain energy, are used as damage features.

The VBDD can be classified into the model-based (or termed as parametric) and data-driven methods (or non-parametric). The model-based methods aim to establish a computational model, e.g., finite element model, by updating model parameters with real measurements (Sun *et al.* 2020). Despite the high accuracy they often achieve, they require prior knowledge and associated assumptions about the real structure which are sometimes not accessible or accurate enough. Unlike the model-based method, the data-driven methods employ statistical models to directly interpret the vibration data into structural patterns without any prior knowledge. Recently, advances in the Artificial Intelligence (AI) technique revolutionized the data-driven methods by allowing large data sets processing with high computation speed. This improvement has made the data-driven method as one of the most attractive and promising approaches in the SHM domain.

During the last two decades, research on machine learning based structural damage detection have been extensively conducted (Sharma and Sen 2021). Basically, the machine learning based method is a process of mapping the monitoring data into different structural patterns. Depending on the availability of vibration data in damaged conditions, they could be divided into supervised and unsupervised. To be more specific, supervised learning requires labeled target values to indicate the real structural conditions while unsupervised ones do not. Nevertheless, both methods have been proven to be effective on structural damage detection. Additionally, some research combines multiple machine learning algorithms to achieve better detection performances. For example, as vibration signals are high dimensional, dimension reduction algorithms like Principal Components Analysis (PCA) are often performed to transfer the original feature vectors to low dimensional yet informative feature components (Islam and Kim 2019). It turns out this dimension reduction operation often brings a significant increase in the computational efficiency at the cost of an ignorable decrease in detection accuracy.

The deep learning (DL) technique, a sub-branch of machine learning, has been successfully applied to computer vision, natural language processing, medical diagnosis, online advertising, literary translation, and autopilot. The DL establishes deep neural networks (DNNs) with multiple layers to learn features of the data with multiple inherent patterns. DNNs characterise their automatic feature extraction and extraordinary ability to deal with big data sets. Great advantages of introducing DNNs into the SHM domain include 1) expert intervention of feature extraction in traditional SHM will be no longer

necessary; 2) real-time monitoring is expected as big data sets from large scale structures could be effectively and rapidly processed.

Among various DNNs, convolutional neural networks (CNNs) are considered to be state-of-art due to their high accuracy and computational efficiency in object detection and classification. A CNN-based approach was first used to perform vibration-based damage detection on a steel frame (Abdeljaber *et al.* 2017). Since then, CNN has gained increasing attention on structural damage detection and more related research was conducted. Abdeljaber *et al.* (2018) verified the efficiency of the vibration-based CNN on damage quantification using acceleration data from a benchmark study. Teng *et al.* (2020) improved the classification accuracy and convergence speed by training the CNN with multiple parameters. The aforementioned CNNs are trained with structural responses. On the other hand, some CNNs are the direct adaptation of computer vision as they are trained with real structural images, e.g., crack images (Cha *et al.* 2017). Commonly used CNNs for image processing are the regional CNNs (RCNNs) and faster RCNNs, combinations of CNNs and region selection techniques. An example of RCNN by Huynh *et al.* (2019) has verified the effectiveness of RCNNs in detecting and cropping bolts on a box girder ridge connection.

Those CNNs have their advantages and challenges on damage detection. Using structural response directly as input is more promising to extract the features for structural damage detection. For the CNNs using images as input, one biggest challenge is that they rely heavily on image quality (Yu *et al.* 2022). Consequently, detection performance could be affected by lighting conditions, shooting angles, and shooting distances. Generally, based on the authors' knowledge, main considerations and concerns about CNN-based damage detection include architecture optimization, the sufficiency of training data, robust ability, real-time application, and uncertainty effect on the measured data.

In this paper, a CNN based structural damage detection is developed for a truss bridge using dynamic responses. CNNs were trained using the mode shapes and normalized modal strain energy change (NMSEC) respectively. NMSEC features were conducted with PCA dimension reduction before inputted into the CNN. We intend to overcome the common challenges of the CNNs using dynamic responses like high computational cost, poor robust ability and make corresponding suggestions. The main contribution of this study is an updated mean square error (UMSE) as the loss function and performance indices of damage localization accuracy (*DLA*) and mean absolute error of damage degrees (*MAEDD*) for damage detection problems. The UMSE loss function achieved faster convergence speed and higher accuracy on damage quantification than that by the commonly used *MSE*. We also compared performances of different *UMSE* weighting factor combinations and made suggestions to cater for different needs. The *DLA*, *MAEDD* and *MSE* were employed for *CNN* performance evaluation. Compared to testing *MSE*, the *DLA* and *MAEDD* were proved to be a more straightforward way of evaluating performances for damage detection problems. Also, the CNN features were

visualized using t-distributed stochastic neighbor embedding (t-SNE) transformation to demonstrate the automatic feature extraction ability of the CNN. Significantly, implementing PCA on the NMSEC enables the training time reduced by 90%. Thus, this NMSEC-PCA based method would be more suitable for structural damage detection problems in practice, especially for large scale structures.

## 2. Methodology

### 2.1 The convolutional neural network (CNN) architecture for damage detection

The CNN consists of multiple convolutional layers, activation layers, pooling layers, fully connected layers, and output layers. The brief descriptions of these layers are as below.

- Input layer. The input layer could be vectors, matrices, images, or any other kinds of data. For damage detection problems, mode shapes, acceleration signals and images of the target structures are commonly used inputs.
- Convolutional layer. The convolutional layer, a key operation of the CNN, is a process of weights learning of kernels, with various kernels sliding over the input features. This process could also be simplified as  $\text{Output}=\text{kernel}\otimes\text{Input}$ , where  $\otimes$  represents the convolution operation. The depth of kernels is the same as the input but the width and height are smaller. Each kernel produces an output feature and then all kernel features are stacked together and considered as the input of the next layer.
- Activation layer. The convolutional layer is usually followed by a nonlinear activation layer to enable a nonlinear mapping and efficient training. Commonly used activations include Sigmoid, Tanh, and more recently, Rectified linear unit (Relu). Relu is used in this paper since it's faster and does not have the gradient vanishing effect (Wang *et al.* 2020).
- Pooling layers. Pooling is a process of compressing feature dimensions (width and height) by taking the maximum or average value in each input patch. The two most commonly used pooling layers are Max-pooling and Average-pooling. It is worth noting that pooling is not a learning process but only a size reduction operation. It could reduce the number of parameters to learn and improve training speed.
- Fully connected (FC) layers. The fully connected layers are the same as in artificial neural networks. At this point, the outputs are usually flattened into 1D dimensional vectors before going through the fully connected layers.
- Output layer. The output layer constrains the network's output through the number of units and activation function. The unit number determines the output classes while the activation constrains the output value. For example, the sigmoid activation converts arbitrary values into the  $[0,1]$  interval, and

the Relu activation zeros out negative values. In this paper, the CNN output layer ended with a sigmoid activation to achieve damage quantification.

- Mini-batch. Training an entire huge dataset at one time is slow and sometimes infeasible because of memory constraints. In such cases, the datasets could be split into small batches for calculating error and updating coefficients of the model. Each fixed size of samples is called a mini-batch and the fixed number of samples in each mini-batch is called the batch size. For each iteration, a small subset of the whole dataset was processed. The mini-batch technique could improve computational efficiency as could allow reaching the global minimum quickly in the cost function.
- Dropout. Drop out is one of the most commonly used techniques for reducing overfitting. When applying a dropout layer, the dropping out rate is required, representing the fraction of features that are zeros out, and this rate is usually set between 0.2-0.5. For a typical CNN architecture, as shown in Fig. 4, repeated tests are conducted to modify the model parameters. During the training, the overfitting problem might occur. When the model is overfitted to the training data, the validation accuracy degrades with the training process going. Reducing the network's size and regularization techniques are effective ways to avoid overfitting (Yu *et al.* 2019).

To achieve better performances, hyper-parameters such as the learning rate, the unit number of each layer and dropping out rate could be optimised. These hyper parameters are usually decided based on empirical results or parameter studies.

### 2.2 Loss function and damage evaluation indices

The loss function is also called the objective function. It measures the error between the predicted value and the true value. The commonly used MSE loss function is as follows.

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \quad (1)$$

where  $Y$  and  $\hat{Y}$  are the true and predicted values of the  $i^{\text{th}}$  sample among the total  $n$  samples respectively. Another commonly used loss function is the mean absolute error ( $MAE$ ) of the difference between the predictions and the targets, as shown in Eq. (2).

$$MAE = \frac{1}{n} \sum_{i=1}^n |Y_i - \hat{Y}_i| \quad (2)$$

More often, the  $MAE$  is used as a training metric during the training process while  $MSE$  is used for testing the CNN performances. In this paper, an updated  $MSE$  loss function  $UMSE$ , a training metric  $MAED$  and two testing performance evaluation indices  $DLA$  and  $MAEDD$  were specifically developed for damage detection using a popular framework Pytorch.

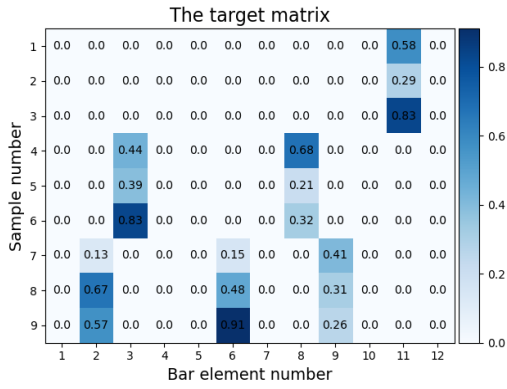


Fig. 1 The example target matrix

### 2.2.1 The UMSE loss function

For the damage detection problem, the training label of CNN is a matrix of which all locations are zeros except for some locations with specified damage degrees. Hence these locations in the training matrix could be divided into two categories: one called the intact region, the other called the damaged region. Fig. 1 illustrates the intact regions (white) and damaged regions (blue) using an example target matrix, where there are nine samples in total and three damage scenarios, three samples for each scenario. A loss function based on this location division could be designed as follows.

$$UMSE = \alpha \times MSEI + \beta \times MSED \quad (3)$$

where  $MSEI$  and  $MSED$  are the mean square error of the wrong detection in the intact region and the failure detection in the damaged region respectively.  $\alpha$  and  $\beta$  are the weighting factors for  $MSEI$  and  $MSED$  respectively. They could be calculated by the following equations.

$$MSEI = \frac{1}{m} \sum_{i=1}^m (Y_i - \hat{Y}_i)^2 \quad i = 1, 2 \dots m \quad (4)$$

$$MSED = \frac{1}{h} \sum_{i=1}^h (Y_i - \hat{Y}_i)^2 \quad i = 1, 2 \dots h \quad (5)$$

where  $m$  and  $h$  are the number of intact and damaged locations in the label. It is worth noting that the sum of  $\alpha$  and  $\beta$  is equal to one as they represent weighting factors for the intact and damaged regions which are the two components that make up the matrix of the training label. For example, after certain damage localization accuracy is reached, an assignment of  $\beta = 0.8$  allows the network to focus on the iterations on damage degrees rather than wasting computation on damage locations. Moreover, an updated loss function with  $\alpha = \beta = 0.5$  is just the same as the  $MSE$  loss function except that they differ only in scale. The proposed loss function including the prior knowledge of the damage region, such the key substructure, and this is useful for real large-scale truss bridges.

### 2.2.2 MAED training metric

The  $MAED$ , the Mean absolute error ( $MAE$ ) of failure

detection in the damaged region, was proposed and used as a training metric. Suppose there are  $m$  damage scenarios, each with  $N_m$  training samples and for each sample  $i$ , there are  $k$  damaged elements, the  $MAED$  is defined as follows.

$$MAED = \frac{1}{N_1 + 2N_2 \dots + kN_m} \left( \sum_{i=1}^{N_1} |T_{i1} - P_{i1}| + \sum_{i=1}^{N_2} \sum_{j=1}^2 |T_{ij} - P_{ij}| + \dots + \sum_{i=1}^{N_m} \sum_{j=1}^k |T_{ij} - P_{ij}| \right) \quad (6)$$

where  $T_{ij}$  and  $P_{ij}$  stand for the true and predicted damage degree of the  $j^{th}$  damaged member for the  $i^{th}$  sample; Basically,  $MAED$  is the mean absolute difference of the true and predicted matrix in the damaged region. While the  $MAE$  evaluates the prediction matrix as a whole region, the  $MAED$  evaluates only the damage degrees in the damage region. Thus, the  $MAED$  is a more straightforward way of validating the model prediction accuracy on damage degrees.

### 2.2.3 Damage localization and quantification indices: DLA and MAEDD

It is of vital importance to fully evaluate the performance of the deep learning network on damage identification. The commonly used  $MSE$  and  $MAE$  indices do tell the damage detection accuracy as a whole but fail to indicate the accuracy of damage localization and quantification separately. To solve this problem, two indices, damage localization accuracy ( $DLA$ ) and  $MAE$  of damage degrees ( $MAEDD$ ) were proposed. The  $DLA$  is defined as follows

$$DLA = \frac{R}{TN} \quad (7)$$

where  $R$  represents the number of samples of which the CNN outputs the correct locations of damaged members, while  $TN$  is the total number of testing samples. The definition of  $MAEDD$  is defined as follows.

$$MAEDD = \frac{1}{m} \left( \frac{1}{N_1} \sum_{i=1}^{N_1} |T_i - P_i| + \frac{1}{2N_2} \sum_{i=1}^{N_2} \sum_{j=1}^2 |T_{ij} - P_{ij}| + \dots + \frac{1}{kN_m} \sum_{i=1}^{N_m} \sum_{j=1}^k |T_{ij} - P_{ij}| \right) \quad (8)$$

Different from the testing  $MSE$  and  $MAE$ , the  $MAEDD$  evaluates only the damage degree prediction accuracy of damaged members. The smaller the  $MAEDD$ , the higher damage quantification accuracy the network achieves. To avoid the incorrectly identified members have impacts on both  $DLA$  and  $MAEDD$ , the calculation of  $MAEDD$

accounts only the correctly identified bars. With *DLA* and *MAEDD*, the overall performance of CNN on damage detection could be fully evaluated.

### 2.3 The mode shape based and modal strain energy based damage index

This section introduces the commonly used features for structural damage detection, namely, mode shape and modal strain energy based features. To get structural mode shapes, modal analysis is conducted. For free vibration of the undamped structure, this could be represented by Paz and Kim (2019)

$$(K - \omega_i^2 M)\varphi_i = 0, i = 1 \dots \dots ndf \quad (9)$$

where  $K$  and  $M$  are the stiffness matrix and mass matrix respectively.  $\omega_i$  is the  $i^{th}$  natural frequency and  $\varphi_i$  is the modal shape vector corresponding to the  $i^{th}$  frequency. In this paper, the first six modes of the designed damaged scenarios were extracted using the finite element software Ansys 19.0.

The modal strain energy of the  $e^{th}$  structural element in mode  $i$  is defined as Wang and Xu (2019)

$$mse_i^e = \frac{1}{2} \varphi_i^{eT} K^e \varphi_i^e, \quad (10)$$

$$i = 1, \dots, ndf, e = 1, \dots, nte$$

where  $\varphi_i^e$  is the nodal displacement of the  $e^{th}$  element in mode  $i$ , namely the mode shape.  $K^e$  is the stiffness matrix of the  $e^{th}$  element. The total modal strain energy of the mode  $i$  could be calculated as

$$mse_i = \sum_{e=1}^{nte} mse_i^e \quad (11)$$

$$i = 1, \dots, ndf, e = 1, \dots, nte$$

By Eq. (11), the modal strain energy of  $e^{th}$  element in  $i^{th}$  mode is normalized by the modal energy summation of the  $e^{th}$  element

$$nmse_i^e = \frac{mse_i^e}{mse_i} \quad (12)$$

$$i = 1, \dots, ndf, e = 1, \dots, nte$$

Structural damage, defined by the reduction of the elemental stiffness, leads to changes of the  $nmse_i^e$ . Thus, an index called normalized modal strain energy change (NMSEC) is defined based on the difference of the  $nmse_i^e$  before and after damage, as expressed in Eq. (13).

$$nmsec = (nmse_i^e)^d - (nmse_i^e)^h, \quad (13)$$

$$e = 1, 2, \dots, nte$$

where  $(nmse_i^e)^d$  and  $(nmse_i^e)^h$  denote the normalized modal strain energy in damaged and healthy scenarios.

Another commonly used modal strain energy based index for damage detection is *MSEBI*, defined as follows (Lee et al. 2021).

$$MSEBI^e = \max \left[ 0, \frac{(nmse_i^e)^d - (nmse_i^e)^h}{(nmse_i^e)^h} \right] \quad (14)$$

In this paper, the original NMSEC datasets were used for CNN training rather than *MSEBI*. The main reason is that the NMSEC does not have the max operation, meaning that negative values are also taken into consideration. Also, the NMSEC matrix is filled with values in the range of [-1,1] which are inherently suitable for the training since no normalization will be needed in such a case. We firstly conducted the principal component analysis (PCA) on the original NMSEC data and then fed the produced datasets into the CNN.

### 2.4 Effects by element and mode based on principal component analysis (PCA)

In this paper, the effects by element and mode were also investigated. This started with dimension reduction of the original NMSEC dataset with PCA. After that, the selected *PCs* could be expressed by the linear combination of the original variables. In this paper, the original NMSEC input is 960 dimensional (160 elements  $\times$  6 modes). Suppose 36 *PCs* were selected (this result could be further seen in Section 3.4), these selected *PCs* could be expressed as follows.

$$\begin{bmatrix} PC_1 & PC_2 & \dots & PC_{36} \end{bmatrix} = \begin{bmatrix} X_1 & X_2 & \dots & X_{960} \end{bmatrix} \begin{bmatrix} w_{11} & w_{1,2} & \dots & w_{1,36} \\ w_{2,1} & w_{22} & \dots & w_{2,36} \\ \vdots & \vdots & \vdots & \vdots \\ w_{960,1} & w_{960,2} & \dots & w_{960,36} \end{bmatrix} \quad (15)$$

where  $PC_j$  and  $X_i$  is the  $j^{th}$  principal component and the  $i^{th}$  original variable. The  $w_{ij}$  is the coefficient of the  $i^{th}$  original variable in the  $j^{th}$  principal component, namely the loadings. This equation could be simplified as

$$P = XW \quad (16)$$

where  $P$ ,  $X$  and  $W$  denote the principal component matrix, original variable vector, and the loadings. The importance of original variables could be measured by multiplying the principal component matrix  $P$  by the explained variance ratio vector, as written in

$$PR^T = XWR^T \quad (17)$$

$$\begin{aligned}
 PR^T &= [X_1 \ X_2 \ \dots \ X_{960}] \begin{bmatrix} w_{11} & w_{1,2} & \dots & w_{1,36} \\ w_{2,1} & w_{22} & \dots & w_{2,36} \\ \vdots & \vdots & \vdots & \vdots \\ w_{960,1} & w_{960,2} & \dots & w_{960,36} \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_{36} \end{bmatrix} \\
 &= [X_1 \ X_2 \ \dots \ X_{960}] \begin{bmatrix} w_{11}r_1 + w_{1,2}r_2 + \dots + w_{1,36}r_{36} \\ w_{2,1}r_1 + w_{22}r_2 + \dots + w_{2,36}r_{36} \\ \vdots \\ w_{960,1}r_1 + w_{960,2}r_2 + \dots + w_{960,36}r_{36} \end{bmatrix}
 \end{aligned} \tag{18}$$

where  $R = (r_1, r_2, \dots, r_{36})$  and  $r_j$  stands for the explained variance ratio of the  $j^{th}$  principal component ( $PC_j$ ). Eq. (18) could also be written in the matrix form.

$$XWR^T = c_1X_1 + c_2X_1 + \dots + c_{960}X_{960} \tag{19}$$

where  $c_i = w_{i1}r_1 + w_{i2}r_2 + \dots + w_{i,36}r_{36}$ ,  $i = 1, 2, \dots, 960$ .  $c_i$ , the coefficient for  $X_i$ , represents the importance of  $X_i$  for the whole dataset.

To evaluate the effect of the element, for the element  $k$ , the coefficients of the six modes could be added together and then normalized with respect to the total coefficients' summation. This could be considered as the coefficient for the bar element  $k$ , as written in

$$\begin{aligned}
 E_k &= \frac{\sum_{m=1}^6 |c_{160(m-1)+k}|}{\sum_1^{960} |c_n|} \\
 m &= 1, 2 \dots 6, \quad k = 1, 2, \dots, 160, \quad n = 1, 2, \dots, 960
 \end{aligned} \tag{20}$$

Accordingly, the effect by mode could also be evaluated according to the following equation.

$$\begin{aligned}
 M_i &= \frac{\sum_{k=160(i-1)}^{k=160i} |c_k|}{\sum_1^{960} |c_n|} \\
 i &= 1, 2 \dots 6, \quad k = 160(i - 1), \quad n = 1, 2, \dots, 960
 \end{aligned} \tag{21}$$

where  $M_i$  represents the effects of mode  $i$ . The normalized  $M_i$  were used for mode effect evaluations. The effect

evaluation on bar elements and modes would potentially work as a guidance for representative information selection for damage detection and thus would greatly reduce the computational cost.

### 2.5 Flowchart of this paper

For clarity, Fig. 2 illustrates the procedures of damage detection on a truss bridge model using the proposed loss function and indices. Firstly, the mode shapes under three damaged scenarios were extracted from a FE model of a truss bridge. These mode shapes were trained in CNN with the proposed UMSE as a loss function. After the training, *DLA* and *MAEDD* were used for performance evaluation on damage detection. Moreover, the modal strain energy was extracted and corresponding NMSEC were calculated and then conducted with PCA. The effects by element and mode were analyzed based on theories in Section 2.4. Finally, dimension-reduced data (The PCs, namely the NMSEC-PCA data) were inputted into the CNN for damage detection.

## 3. Numerical simulations

### 3.1 Numerical modelling

Fig. 3 shows the finite element model of a truss bridge constructed in Ansys workbench 19.0. The bridge model consists of 56 nodes and 160 beam elements. It is with

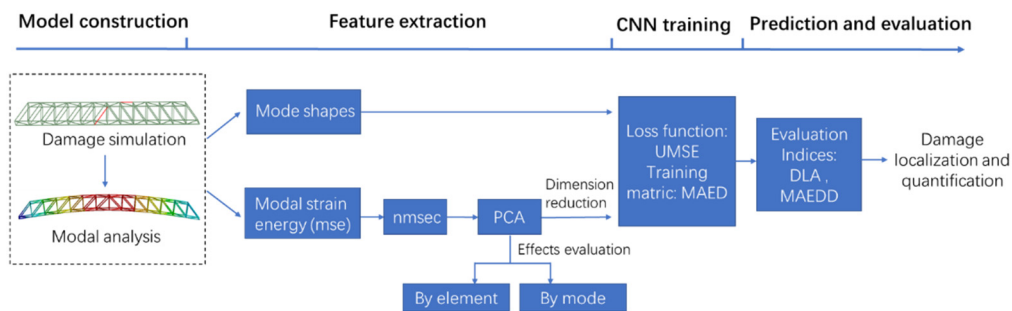


Fig. 2 Flowchart of the investigations

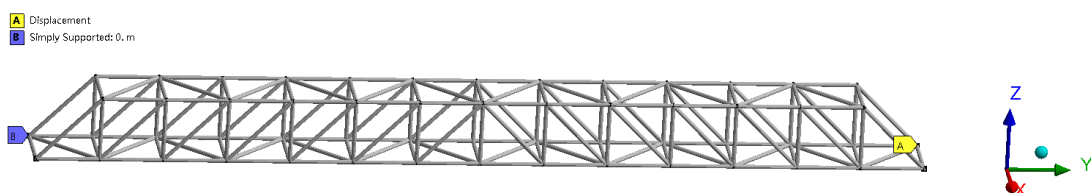
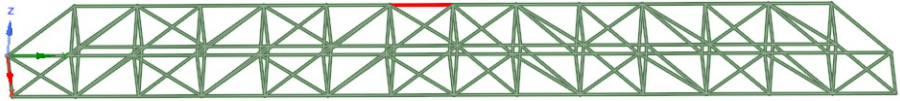
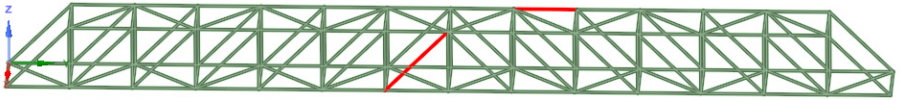
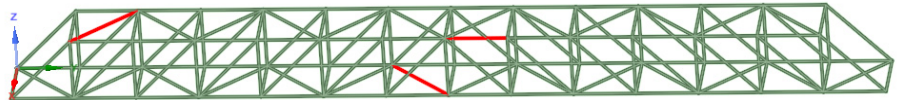


Fig. 3 Finite element model of the truss bridge

Table 1 Scenarios for damage detection

Scenarios	Damaged elements	
	No.	Location
Single damage	129	
Double damage	82,96	
Triple damage	13,89 100	

\*Elements in red are damaged

length of 8.4 m, a width of 0.6 m, and a height of 0.6 m. The left end of the model is pin supported. The right-end are constrained in X, Z directions and free in the longitudinal direction Y. The bridge model is a frame structure with members simulated by BEAM188 element with circular tube section, giving 336 (56 × 6) degrees-of-freedom (DOFs). The Young’s modulus and density are 2.0E11N/m<sup>2</sup> and 7800 kg/m<sup>3</sup> respectively. The designed damage scenarios consist of single damage, double damage, and triple damage. Details of the damaged element number and locations are given in Table 1. The damage is simulated by elastic modulus reduction of bar elements. For instance,

10% damage of a certain bar element means a 10% reduction of the elastic modulus on that bar.

### 3.2 Implementation details of CNN

Modal analysis was conducted on the finite element model of the truss bridge. The first six mode shapes of the designed damaged scenarios were extracted and taken as the original input of CNN. Fig. 4 shows the data preparation and the architecture of CNN. As can be seen, the original data were nodal displacement data, namely the mode shapes, at six DOFs in six modes and they were flattened to

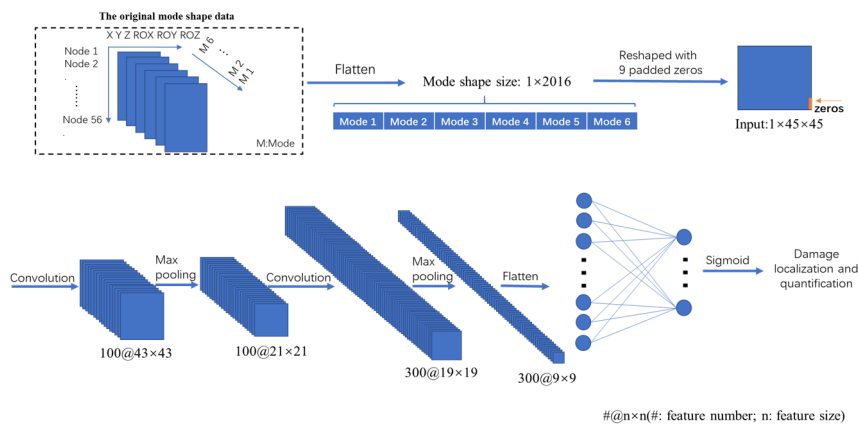


Fig. 4 The data preparation and the architecture of the CNN

Table 2 Mode shape based CNN architecture

Layer	Layer type	Kernel size	No. of kernels/neurons	Stride	Activation	Output shape
1	Input	-	-	-	-	45×45
2	Convolution	3×3	100	1	ReLU	43×43
3	Max pooling	2×2		2	-	21×21
4	Convolution	3×3	300	1	ReLU	19×19
5	Max pooling	2×2		2	-	9×9
6	FC	-	300	-	-	-
7	Output	-	160	-	Sigmoid	-

Table 3 NMSEC-PCA based CNN architecture

Layer	Layer type	Kernel size	No. of kernels/neurons	Stride	Activation	Output shape
1	Input	-	-	-	-	6×6
2	Convolution	2×2	100	1	ReLU	5×5
3	Max pooling	2×2		2	-	2×2
4	Convolution	2×2	300	1	ReLU	1×1
5	FC	-	300	-	-	-
6	Output	-	160	-	Sigmoid	-
7	Input	-	-	-	-	6×6

one-dimensional (1D) vector of shape  $1 \times 2016$  ( $56 \text{ nodes} \times 6 \text{ DOFs} \times 6 \text{ modes}$ ). To apply 2D convolution, they were reshaped to a square matrix of  $1 \times 45 \times 45$  with 9 padded zeros. Before being fed into CNN, these mode shapes were rescaled to a range of  $[-1, 1]$  using min-max normalization.

To get enough training samples, for each scenario, 2000 randomly generated damage degrees from 0 to 1 were generated and corresponding mode shapes were extracted and used for training. In total, the dataset concluded 6000 samples from three scenarios, each with 2000 samples. The dataset division was 70% (4200 samples) for training, 15% (900 samples) for validation, and 15% (900 samples) for testing.

Table 2 presents the architecture of the CNN with mode shapes as input. It contained two convolutional layers with kernel size 3, each followed by a ReLU activation and max-pooling layer with a kernel of size  $2 \times 2$ . After the convolution, damage features were flattened into vectors and went through multiple fully connected (FC) layers. The first FC layer contained 300 neurons while the output layer contained 160 neurons as there are 160 potential damaged bar elements. Since CNN aimed to quantify damage, the output layer was a regression layer with the sigmoid activation. The proposed *UMSE* (Eq. (3)) was used as the loss function and rmsprop as the optimizer. The learning rate was 0.0002. The proposed *MAED* (Eq. (6)) worked as a training metric and *DLA* (Eq. (7)) and *MAEDD* (Eq. (8)) were used for the performance evaluation.

### 3.3 Data augmentation

Data augmentation is a commonly used technique for mitigating overfitting in computer vision. With data augmentation, more training samples could be generated from the existing training samples. In this study, 900 extra training samples were generated by adding 5% Gaussian white noise to the original datasets as shown in Eq. (22), 300 for each damage scenario.

$$\bar{h} = h(1 + Nl * R) \quad (22)$$

where  $h$  is the original data, and  $\bar{h}$  is the data with the noise.  $R$  follows the normal distribution of  $N(0,1)$ .  $Nl$  is the noise level and here is 5%. By feeding the network with these extra samples, the learning ability and anti-noise ability are expected to improve.

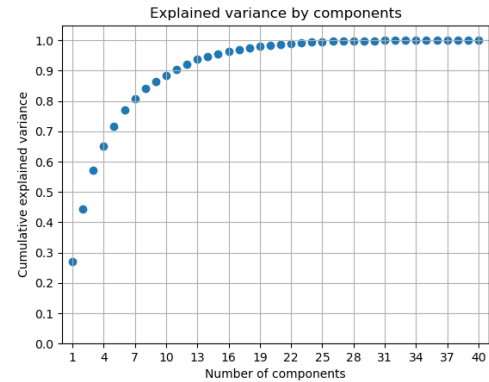


Fig. 5 Explained variance by components

### 3.4 Modal strain energy based PCA

The NMSEC training data was conducted with PCA for dimensionality reduction. The explained variance by the first forty components was shown in Fig. 5. The result showed that the first 10 components contained approximately 90% of the variance, while around 30 components described close to 100% of the variance. Here 36 components, retaining 99.98% of the variance, were selected. With the PCA, the dimension of the train data matrix was reduced from the original 960 ( $160 \text{ elements} \times 6 \text{ modes}$ ) to 36. The transformed matrix was then reshaped to  $6 \times 6$  and inputted into CNN. The architecture of NMSEC-PCA based CNN is given in Table 3. The proposed *UMSE* and *MAED* were used as the loss function and training metric. *DLA* and *MAEDD* were used for performance evaluation.

## 4. Results and discussions

### 4.1 Architecture selection

Network architecture selection is important to achieve high performance. In this section, three architectures with a different number of convolutional layers were compared and the commonly used *MSE* was used as a loss function. Fig. 6 shows the validation loss comparison of these architectures. Among all, one convolutional layer CNN with 100 kernels (CNN1) converges the fastest, while the network with two convolutional layers (CNN2) achieves the best loss performance. After training, for each CNN, 900



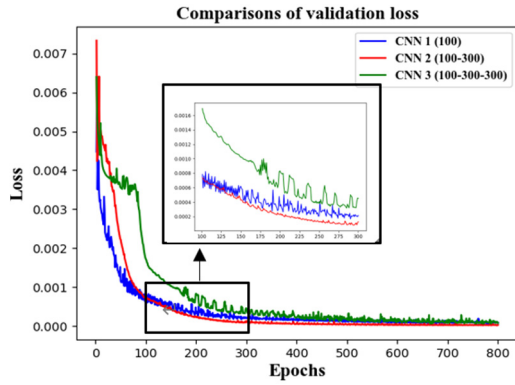


Fig. 6 Validation loss of different architectures

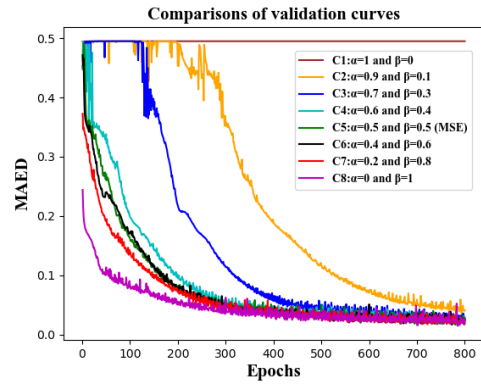


Fig. 7 Comparisons of validation curves relating to different weighting factor combination

Table 4 Comparisons of architectures

Architectures	Test MSE	DLA	MAEDD
100	6.3165e-05	96.33%	0.038
<b>100-300</b>	<b>2.1903e-05</b>	<b>99.66%</b>	<b>0.026</b>
100-300-300	6.3046e-05	98.00%	0.037

samples were tested and a prediction matrix could be obtained. Table 4 shows the prediction results when evaluated with three statistical indices. It can be seen that CNN2 outperforms other architectures in terms of testing MSE (2.1903e-05), DLA (99.66%), and MAEDD (0.026). Overall, CNN with two convolutional layers achieved the best performance and was thus used for the rest of this paper.

#### 4.2 Effects of the loss function

To study the effects of weighting factors  $\alpha$  and  $\beta$  on CNN performance, a parameter study on the updated loss function was conducted. The validation curves of the updated MSE (UMSE) with selective weighting parameter combinations were compared, as presented in Fig. 7. It can be seen that the updated MSE with  $\alpha = 0$  and  $\beta = 1$  (C8) converged the fastest among all, followed by  $\alpha = 0.2$  and  $\beta = 0.8$  (C7). Moreover, with enough epochs of 800, most MAED curves converged to the same level. Interestingly, for cases where  $\beta < 0.4$ , namely C1-C3, the MAED curves remained stable around 0.5 at the beginning. For C1, the

MAED even ended at around 0.5. This is because the CNN is trapped into a local optimum of a prediction matrix filled mostly with values close to zero. In such a case, the values in the damaged regions were also near zero, leading to a poor damage localization accuracy.

Testing performances of these combinations (C1-C8) were also compared in terms of testing MSE, DLA, and MAEDD as shown in Table 5. It can be seen that from C2 to C7, as  $\beta$  grows, the DLA remained stable at around 99% while both testing MSE and MAEDD saw improvements, test MSE from 4.466 e-5 to 1.307e-5 and MAEDD from 0.0359 to 0.0170. Among all combinations, C7 ( $\alpha = 0.2$  and  $\beta = 0.8$ ) performed the best, achieving low test MSE (1.307e-05) and MAEDD (0.0170).

When  $\alpha = 1$ , C1 reached a random DLA of 33% as the prediction matrix was filled mostly with near-zero numbers. As expected, in C8 where  $\beta = 1$ , the CNN achieved the best MAEDD but the worst DLA (0.78%). This means that it failed to locate damage in 893 out of a total 900 testing samples. The main reason is that the UMSE of C8 considered only the values in the damaged regions, leading to a prediction matrix where intact regions filled mostly with values around 0.5.

Based on the above analysis,  $\beta$  relates to how close the damage degrees are predicted, namely the MAEDD, while  $\alpha$  relates to the damage localization accuracy (DLA). In other words, with high  $\beta$ , the updated MSE would usually achieve good damage quantification performance, while with a high  $\alpha$ , a better damage localization accuracy could be expected.

Table 5 CNN testing results evaluation of the updated MSE

Combination No.	Loss function of CNN	Testing MSE	DLA (%)	MAEDD
C1	$\alpha = 1$ and $\beta = 0$	0.0041	33.33	0.517
C2	$\alpha = 0.9$ and $\beta = 0.1$	4.466e-05	98.33	0.0359
C3	$\alpha = 0.7$ and $\beta = 0.3$	1.920e-05	99.78	0.0202
C4	$\alpha = 0.6$ and $\beta = 0.4$	2.741e-05	97.67	0.0265
C5	$\alpha = 0.5$ and $\beta = 0.5$ (MSE)	2.221e-05	99.00	0.0268
C6	$\alpha = 0.4$ and $\beta = 0.6$	2.355e-05	98.67	0.0236
C7	$\alpha = 0.2$ and $\beta = 0.8$	1.307e-05	98.67	0.0170
C8	$\alpha = 0$ and $\beta = 1.0$	0.2558	0.78	0.0086

It is worth noting that, in practice,  $\alpha$  and  $\beta$  should be rigorously assigned according to situations. Generally,  $\beta$  in the range of [0.3,0.8] is recommended to get a higher CNN performance as well as avoiding local optimum. For the rest of this paper, the updated MSE with  $\alpha = 0.2$  and  $\beta = 0.8$  was used as it achieved not only a faster convergence speed but also the best testing MSE, MAEDD, and a relatively higher DLA.

Merits of the proposed UMSE loss function are as follows: 1). It is with high flexibility as the users could weigh the damage localization and damage quantification by assigning different weighting factor values. 2). The updated MSE loss function ( $\beta > 0.5$ ) avoids unnecessary iterations on the intact region but focuses on only the

predictions of damage degrees in the damaged region, thus achieving a higher damage quantification accuracy. 3). The practical engineering projects usually consist of thousands of bar elements, resulting in a large CNN training label matrix. The customized UMSE would greatly reduce the epoch number, iteration time and enhance the damage detection accuracy of the network.

### 4.3 Feature visualization

To demonstrate the automatic feature extraction ability of the proposed CNN, the t-distributed Stochastic Neighbor Embedding (t-SNE) was used for feature visualization. The t-SNE was proposed by Maaten and Hinton (2008) for dimension reduction of high-dimensional data for visualization. The t-SNE firstly converts the points similarities to joint probabilities and then takes the Kullback-Leibler divergences between joint probabilities of the low-dimensional space and high dimensional data as cost function (Lin *et al.* 2017). The result of t-SNE varies due to the different initializations.

In this study, the testing datasets (900 samples, three scenarios each with 300 samples) were inputted into the CNN for visualization. It is worth noting that as the original outputs of the convolutional layers were four dimensional array (samples, channel, width, height), they were reshaped to two dimensional array before conducting the t-SNE transformation. Four output features were visualized as shown in the Fig. 8, including the features of the original testing data, features after two convolutional layers and features after fully connected layer. It can be seen that the original testing data were dispersed into several clusters (Fig. 8(a)) while after the two convolution layers, fewer clusters were seen. In Fig. 8(d), most points from the same

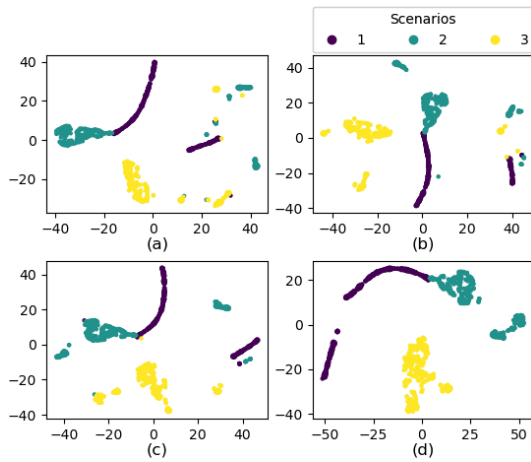


Fig. 8 CNN feature visualization with t-SNE: (a) the input; (b) after first Conv layer; (c) after the second Conv layer; (d) after the FC layer

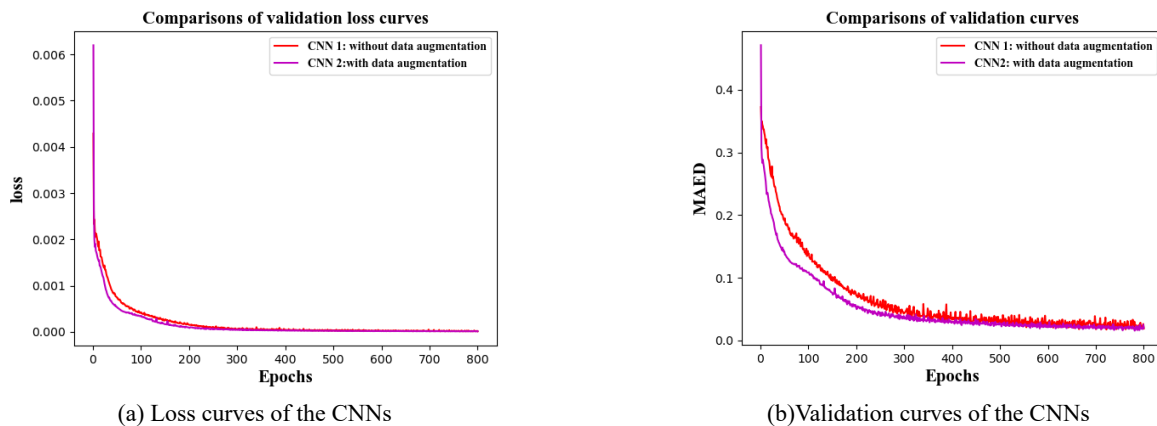


Fig. 9 Comparisons of the CNN with and without data augmentation

Table 6 CNN performance evaluation using noise-free and 5% noise testing samples

Test samples	Testing MSE		DLA		MAEDD	
	CNN 1	CNN 2	CNN 1	CNN 2	CNN1	CNN 2
Noise-free	1.3067e-5	1.9706e-05	98.67%	98.88%	0.0170	0.0186
3% noise	0.0003	0.0002	89.22%	92.33%	0.0850	0.0649
5% noise	0.0006	0.0004	81.00%	88.11%	0.134	0.0941

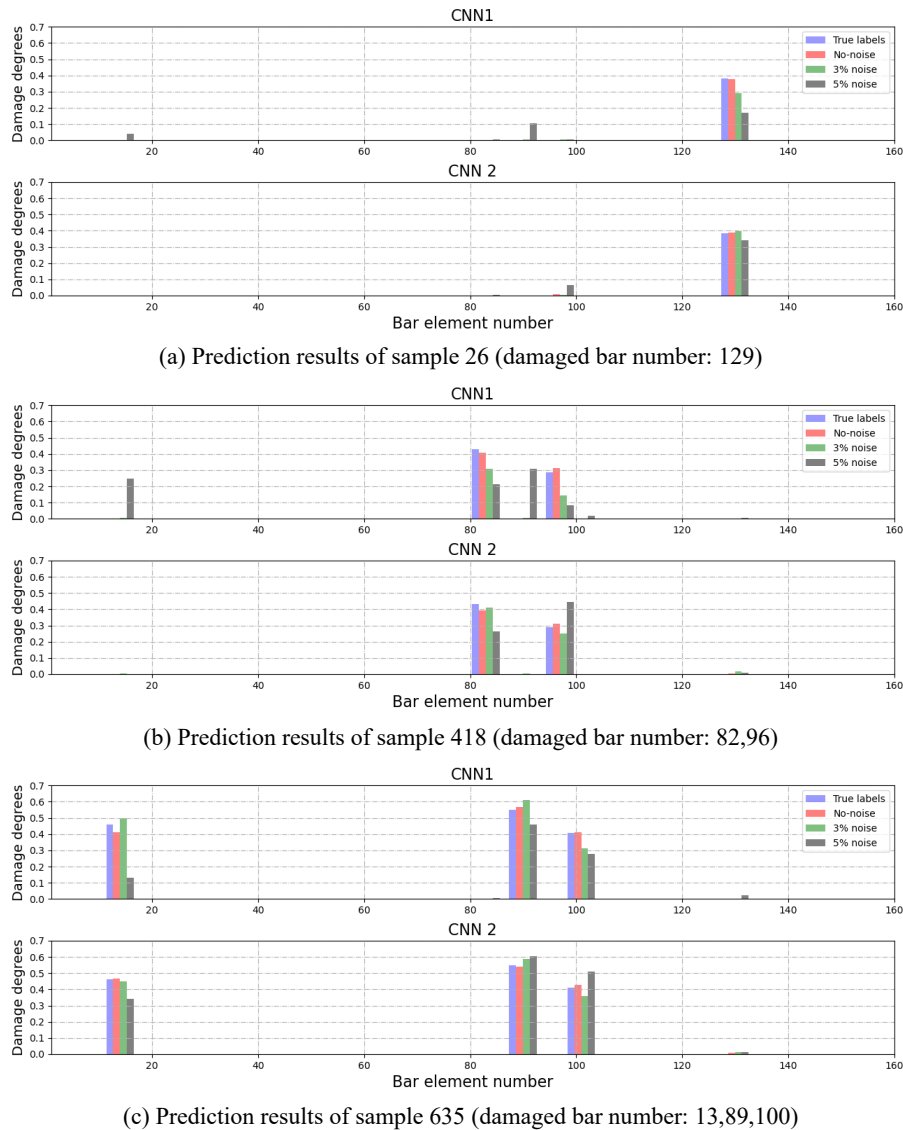


Fig. 10 Prediction results of selective samples using CNN1 and CNN2

scenarios gather in the same cluster, indicating good feature extraction ability of the proposed CNN.

#### 4.4 Effects of data augmentation

For comparison, two Convnets, CNN1 and CNN2, were trained, CNN2 with data augmentation while CNN1 without. Implementation details of data augmentation could be seen in Section 3.3. After the training, the original 900 testing samples were added with 3% and 5% noise and were tested respectively. Fig. 9 shows the validation loss and metric (*MAED*) of the Convnets. It can be seen that the CNN2 converges slightly fast in the first 200 epochs. A positive effect of data augmentation on anti-noise ability could be seen in Table 6 where the prediction results of noise-free 3% and 5% noise polluted samples are evaluated in terms of testing *MSE*, *DLA*, and *MAEDD*. When tested with noise-free samples, both CNN1 and CNN2 achieved high damage detection accuracy, with the testing *MSE* from  $1.30e$  to  $5-1.97e-5$ , *DLA* at around 98%, and *MAEDD* at around 0.018. However, when tested with noise polluted

samples, both showed noticeably declined performances in all three indices. Despite of the declines, *CNN2* achieved higher damage localization and quantitation accuracy than *CNN1* for both 3% and 5% noise polluted data.

To further prove the effectiveness of data augmentation, prediction results of three randomly selected samples (Samples 26, 418 and 635) were presented in Fig. 10. It can be seen that 1) when tested with non-noise testing data, both CNN1 and CNN2 achieved high detection accuracy on damage localization and quantification in all samples; 2) for the polluted data (3% and 5% noise), CNN2 outperformed CNN1 on damage localization and quantification. Particularly, in Fig. 10(b), for the 5% noise polluted sample, CNN1 failed to detect the damaged bar 96 and showed false alarms on other elements while CNN2 successfully identified the damaged bars with acceptable prediction accuracy on damage degrees. Overall, although Convnets trained by the original mode shape datasets could achieve high damage detection accuracy, they may exhibit slightly poor robustness to the noise. In such cases, data augmentation is a way to mitigate this problem.

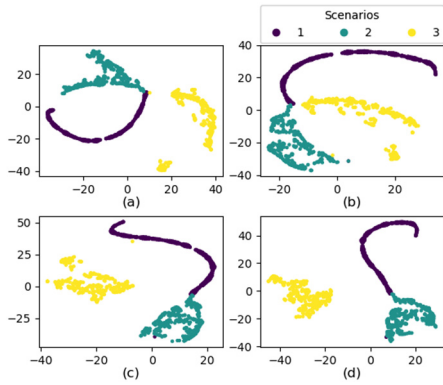


Fig. 11 NMSEC-PCA based CNN feature visualization with t-SNE: (a) the input; (b) after first Conv layer; (c) after the second Conv layer; (d) after the FC layer

4.5 The NMSEC-PCA based CNN

According to Section 3.4, the PCA dimensionality reduction process produced a NMSEC-PCA based input as a 36 dimensional matrix. It was then reshaped to  $1 \times 6 \times 6$  and fed into CNN. After the training, the features of the CNN using testing samples were visualized using t-SNE method, as presented in Fig. 11. Fig. 12 presents the validation curve comparisons between the mode shape

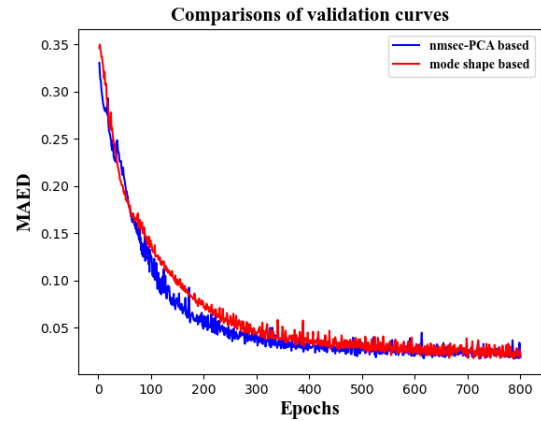


Fig. 12 Validation curve comparisons between mode shapes based and NMSEC-PCA based index

Table 7 Comparisons of the two methods

Methods	Mode shape based	NMSEC-PCA based
Input shape	$4200 \times 1 \times 45 \times 45$	$4200 \times 1 \times 6 \times 6$
Training time	Around 10 minutes	Within a minute
Testing MSE	1.3067e-05	3.9385e-5
Testing DLA	98.67%	99.22%
Testing MAEDD	0.0170	0.0154

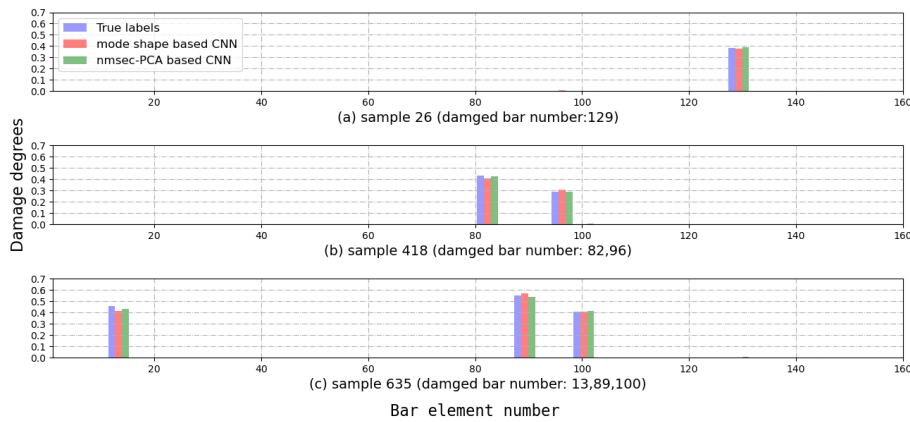


Fig. 13 Prediction results of selective samples using mode shape based CNN and NMSEC-PCA based CNN

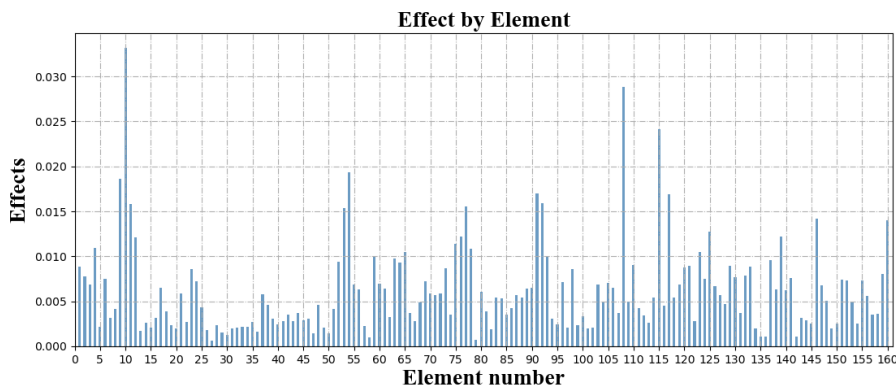


Fig. 14 The effect by bar element

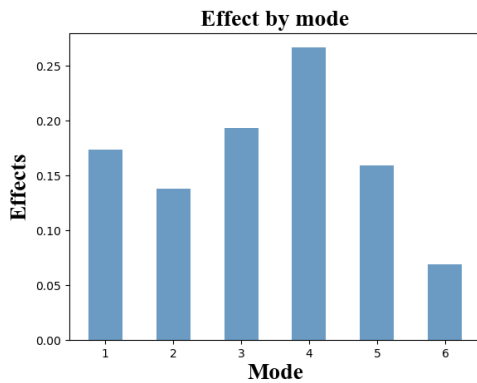


Fig. 15 The effect by mode

based method and the NMSEC-PCA based method. Accordingly, Table 7 compares these two methods in terms of input shape, training time, and testing indices. Testing results on selected samples (Samples 26,418 and 635) of these two CNNs were also compared in Fig. 13. Overall, it can be seen that these two methods achieved almost the same convergence rate and testing performance. In terms of testing *DLA* and *MAEDD*, the NMSEC-PCA based CNN achieved slightly higher accuracy. However, the training of NMSEC-PCA based method is ten times faster than that of the mode shape based one. This advantage would enable the NMSEC-PCA based method to have more potential for real application on the damage detection problem of large-scale structures.

#### 4.6 Effects by element and mode

From the obtained PCs, effects by element and mode were calculated according to Eqs. (20)-(22) and results are shown in Figs. 14-15 respectively. According to Fig. 14, Bar elements 10,108 and 115 account for the first three most valuable pieces of modal strain energy information. Fig. 15 shows that Mode 4 accounts for the most information of the whole dataset while Mode 6 contributes the least information, thus Mode 6 could be removed. The further analysis could help estimate the sensitivity of elements and modes on the modal strain energy.

## 5. Conclusions

This paper proposed a novel updated loss function *UMSE* and damage detection evaluation indices for damage detection problem and the efficiency of the proposed method was demonstrated using a truss bridge model. PCA was conducted on the NMSEC for dimension reduction. The NMSEC-PCA based data were trained by CNN and damage detection accuracy were compared to the mode shape based CNN. The following conclusions could be drawn.

- CNN with two convolutional layers were selected as it achieved the best damage detection accuracy. The t-SNE were employed to visualize the output of each network layer to demonstrate the automatic feature extraction of the proposed CNN.

- The proposed *UMSE* loss function outperforms the commonly used *MSE* as it could greatly reduce the computational cost, achieve faster convergence speed and higher detection accuracy, and is highly flexible for practical engineering structures applications as it could be customized according to needs. Based on this research, a combination of  $\alpha=0.2$  and  $\beta=0.8$  in the *UMSE* is suggested for damage quantification problems.
- For a clear comparison, we used both testing *MSE* and proposed indices (*DLA* and *MAEDD*) for evaluation. A near-zero testing *MSE* cannot indicate an overall high detection accuracy since they are inherently near zeros in this study. In reality, even a slight change in testing *MSE* could mean a great difference in the damage accuracy. An example of this could be seen in Table 6. When the two CNNs were tested by 5% noise polluted samples, there is only a slight difference in the testing *MSE* (CNN1:0.0006 and CNN2:0.0004) while notable differences were seen in *DLA* and *MAEDD*. Therefore specifically developed indices are needed.
- When tested with noise polluted data, the CNN with data augmentation achieved better detection accuracy than the one without. In practice, the mode shape based CNN may exhibit a slightly poor robust ability. In such a case, data augmentation, that is generating samples by adding noise, could be a way to mitigate the noise effect.
- Compared to mode shape input, the NMSEC-PCA as input is computationally cheaper yet equally effective. This reduction on computation cost would allow CNN more suitable for real structural application especially for large scale and complex structures.
- Effects by element and mode were analyzed based on NMSEC-PCA index. The result could potentially provide guidance on selecting valuable components from the original modal strain energy training data for the future research.

The experimental study will be conducted to further verify the proposed method next step.

## Acknowledgments

This research is financially supported by the Ministry of Science and Technology of China (Grant No. 2019YFE0112400).

## References

- Abdeljaber, O., Avci, O., Kiranyaz, M.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>
- Abdeljaber, O., Avci, O., Kiranyaz, M.S., Boashash, B., Sodano, H., and Inman, D.J. (2018), "1-D CNNs for structural damage detection: Verification on a structural health monitoring

- benchmark data”, *Neurocomputing*, **275**, 1308-1317.  
<https://doi.org/10.1016/j.neucom.2017.09.069>
- Aswal, N., Sen, S. and Mevel, L. (2021), “Estimation of local failure in tensegrity using Interacting Particle-Ensemble Kalman Filter”, *Mech. Syst. Signal Process.*, **160**, 107824.  
<https://doi.org/10.1016/j.ymsp.2021.107824>
- Cao, M., Radziński, M., Xu, W. and Ostachowicz, W. (2014), “Identification of multiple damage in beams based on robust curvature mode shapes”, *Mech. Syst. Signal Process.*, **46**(2), 468-480. <https://doi.org/10.1016/j.ymsp.2014.01.004>
- Cha, Y.-J., Choi, C. and Büyükköztürk, O. (2017), “Deep learning-based crack damage detection using convolutional neural networks”, *Comput.-Aided Civil Infrastr. Eng.*, **32**(5), 361-378.  
<https://doi.org/10.1111/mice.12263>
- Huynh, T.-C., Park, J.-H., Jung, H.-J. and Kim, J.-T. (2019), “Quasi-autonomous bolt-loosening detection method using vision-based deep learning and image processing”, *Automat. Constr.*, **105**, 102844.  
<https://doi.org/10.1016/j.autcon.2019.102844>
- Islam, M.M. and Kim, J.-H. (2019), “Vision-based autonomous crack detection of concrete structures using a fully convolutional encoder-decoder network”, *Sensors*, **19**(19), 4251. <https://doi.org/10.3390/s19194251>
- Kim, C.-W., Zhang, Y., Wang, Z.R., Oshima, Y. and Morita, T. (2018), “Long-term bridge health monitoring and performance assessment based on a Bayesian approach”, *Struct. Infrastr. Eng.*, **14**(7), 883-894.  
<https://doi.org/10.1080/15732479.2018.1436572>
- Kordestani, H., Xiang, Y.-Q. and Ye, X.-W. (2018), “Output-only damage detection of steel beam using moving average filter”, *Shock and Vibration*, **2018**, 1-13.  
<https://doi.org/10.1155/2018/2067680>
- Kordestani, H., Zhang, C.W., Masri, S.F. and Shadabfar, M. (2021), “An empirical time-domain trend line-based bridge signal decomposing algorithm using Savitzky-Golay filter”, *Struct. Control Health Monitor.*, **28**(7), e2750.  
<https://doi.org/10.1002/stc.2750>
- Lee, S., Park, S., Kim, T., Lieu, Q.X. and Lee, J. (2021), “Damage quantification in truss structures by limited sensor-based surrogate model”, *Appl. Acoust.*, **172**, 107547.  
<https://doi.org/10.1016/j.apacoust.2020.107547>
- Lin, Y.Z., Ni, Z.H. and Ma, H.W. (2017), “Structural damage detection with automatic feature-extraction through deep learning”, *Comput.-Aided Civil Infrastr. Eng.*, **32**(12), 1025-1046. <https://doi.org/10.1111/mice.12313>
- Maaten, L. and Hinton, G. (2008), “Visualizing data using t-SNE”, *J. Mach. Learn. Res.*, **9**(86), 2579-2605.
- Nick, H. and Aziminejad, A. (2021), “Vibration-based damage identification in steel girder bridges using artificial neural network under noisy conditions”, *J. Nondestr. Eval.*, **40**(1).  
<https://doi.org/10.1007/s10921-020-00744-8>
- Paz, M. and Kim, Y.H. (2019), *Structural Dynamics: Theory and Computation*, Cham: Springer International Publishing.  
<https://doi.org/10.1007/978-3-319-94743-3>
- Santos, F.L.M. dos, Peeters, B., Van der Auweraer, H., Góes, L.C.S. and Desmet, W. (2016), “Vibration-based damage detection for a composite helicopter main rotor blade”, *Case Stud. Mech. Syst. Signal Process.*, **3**, 22-27.  
<https://doi.org/10.1016/j.csmssp.2016.01.001>
- Sharma, S. and Sen, S. (2021), “Bridge damage detection in presence of varying temperature using two-step neural network approach”, *J. Bridge Eng.*, **26**(6).  
[https://doi.org/10.1061/\(ASCE\)BE.1943-5592.0001708](https://doi.org/10.1061/(ASCE)BE.1943-5592.0001708)
- Spencer, B.F., Hoskere, V. and Narazaki, Y. (2019), “Advances in computer vision-based civil infrastructure inspection and monitoring”, *Eng.*, **5**(2), 199-222.  
<https://doi.org/10.1016/j.eng.2018.11.030>
- Sun, L.M., Shang, Z.Q., Xia, Y., Bhowmick, S. and Nagarajaiah, S. (2020), “Review of bridge structural health monitoring aided by big data and artificial intelligence: From condition assessment to damage detection”, *J. Struct. Eng.*, **146**(5), 04020073.  
[https://doi.org/10.1061/\(ASCE\)ST.1943-541X.0002535](https://doi.org/10.1061/(ASCE)ST.1943-541X.0002535)
- Talebpour, M.H., Goudarzi, Y. and Sharifnezhad, M. (2020), “Clustering elements of truss structures for damage identification by CBO”, *Periodica Polytechnica Civil Engineering*, October. <https://doi.org/10.3311/PPci.16636>
- Teng, S., Chen, G.F., Gong, P.P., Liu, G. and Cui, F.S. (2020), “Structural damage detection using convolutional neural networks combining strain energy and dynamic response”, *Meccanica*, **55**(4), 945-959.  
<https://doi.org/10.1007/s11012-019-01052-w>
- Wang, S.Q. and Xu, M.Q. (2019), “Modal strain energy-based structural damage identification: a review and comparative study”, *Struct. Eng. Int.*, **29**(2), 234-248.  
<https://doi.org/10.1080/10168664.2018.1507607>
- Wang, R.H., Chencho, An, S., Li, J., Li, L., Hao, H. and Liu, W.Q. (2020), “Deep residual network framework for structural health monitoring”, *Struct. Health Monitor.*, **20**(4), 147592172091837.  
<https://doi.org/10.1177/1475921720918378>
- Xu, J., Hao, J.J., Li, H.N., Luo, M.Z., Guo, W. and Li, W.J. (2017), “Experimental damage identification of a model reticulated shell”, *Appl. Sci.*, **7**(4), 362.  
<https://doi.org/10.3390/app7040362>
- Yu, Y., Wang, C.Y., Gu, X.Y. and Li, J.C. (2019), “A novel deep learning-based method for damage identification of smart building structures”, *Struct. Health Monitor.*, **18**(1), 143-163.  
<https://doi.org/10.1177/1475921718804132>
- Yu, Y., Rashidi, M., Samali, B., Mohammadi, M., Nguyen, T.N. and Zhou, X.X. (2022), “Crack detection of concrete structures using deep convolutional neural networks optimized by enhanced chicken swarm algorithm”, *Struct. Health Monitor.*, p. 14759217211053546.  
<https://doi.org/10.1177/14759217211053546>
- Zhou, Z., Wegner, L.D. and Sparling, B.F. (2021), “Data quality indicators for vibration-based damage detection and localization”, *Eng. Struct.*, **230**, 111703.  
<https://doi.org/10.1016/j.engstruct.2020.111703>