Journal of Civil Structural Health Monitoring (2023) 13:947–959
https://doi.org/10.1007/s13349-023-00690-9

ORIGINAL PAPER

B-CNN: a deep learning method for accelerometer-based fatigue cracks monitoring system

Yanjie Zhu[1,2] · Hidehiko Sekiya[1,3] · Takayuki Okatani[3,4] · Masayuki Tai[5] · Shogo Morichika[1]

Received: 20 August 2022 / Accepted: 21 March 2023 / Published online: 31 March 2023

Abstract
The maintenance of fatigue damage is essential to keep steel bridges safe since the fatigue crack can lead to brittle fracture. However, real-time monitoring of fatigue crack propagation using traditional strain-based sensors is still a critical challenge due to their limited ductility and durability. In this paper, we propose a fatigue crack monitoring system by employing accelerometers due to their portability, easy replaceability, and sustainability. A corresponding blind source separation (BSS) joint convolutional neural network (CNN) classifier, namely B-CNN, is also proposed to identify fatigue crack conditions by interpreting acceleration measurements directly. We introduce the BSS as the first feature conversion layer to identify the pseudo modal properties from the highly sampled acceleration measurements before any further deep learning progress conducted by CNN. Those converted features are the sparse representation of the structural performance, where the crack conditions are hidden. By adopting BSS, the time history data can be sparse expressed, and the subsequent feature learning could be accelerated by employing a shallow CNN configuration. The proposed B-CNN classifier is further utilized for predicting crack conditions from forthcoming measurements. To examine the proposed monitoring solution, an experimental case study on a steel girder is conducted. About 1-month acceleration measurements were collected and investigated in this paper. The results obtained can demonstrate a shallow B-CNN configuration is sufficient to accurately identify cracks condition with high precision, over 93%, and proper prospection on unseen measurements.

Keywords Blind source separation · Fatigue crack monitoring · Convolutional neural network · Acceleration · Structural health monitoring

1 Introduction

The gradual degradation of bridge components, due to randomly cyclic loadings, can lead to the initiation and propagation of fatigue cracks in fatigue-susceptible regions among the bridge. These fatigue cracks can significantly damage the bridge operating performance and even result in brittle fracture. In light of this concern, fatigue cracks monitoring has attracted increasingly attention for bridge health assessment, safety control and maintenance [1, 2].

Conventional inspecting approaches, such as visual inspection and non-destructive testing, can provide accurate assessment, however, several constraints limit their applications for real-time fatigue monitoring. For instance, it is time-consuming and costly to conduct tests on a complex structure and it may interfere traffic during the inspection. In an effort to monitor structural real-time behaviors, structural health monitoring (SHM) has been widely implemented by arranging a network of sensors on the bridge. In terms of

✉ Yanjie Zhu
zhuyanjie2018@163.com

[1] Department of Urban and Civil Engineering, Tokyo City University, 1-28-1 Tamazutsumi, Setagaya-ku, Tokyo 158-8557, Japan

[2] Department of Bridge Engineering, School of Transportation, Southeast University, Nanjing 211189, China

[3] RIKEN Center for Advanced Intelligence Project, 1-4-1 Nihonbashi, Chuo, Tokyo 103-0027, Japan

[4] Graduate School of Information Sciences, Tohoku University, 6-3-09 Aoba, Aramaki-aza Aoba-ku, Sendai 980-8579, Japan

[5] Civil Engineering Program, School of Engineering, Faculty of Engineering, University of the Ryukyus, 1 Senbaru, Nishihara, Okinawa 903-0213, Japan

fatigue crack monitoring, the widespread sensing techniques is strain-related sensors, because the variations in signals can provide a straightforward reflection of crack's conditions. For example, traditional metal foil gages can enable the surface cracks detection [3], which are benefit from high impedance and low cost, however, these transducers are too sensitive to the surface quality and too small to the desired monitoring surface [4]. The other widespread utilized sensors are Fiber Bragg grating, which are immune to electromagnetic interference and can withstand corrosive environments [5–7]. However, the obvious drawback of fiber optic sensors is high cost. Moreover, strain sensors are easily damaged under cracking because of limited ductility, which therefore are impractical for continuous monitoring of cracks [4].

To overcome the aforementioned limitations, we adopt the solution of using mature micro-electro-mechanical (MEMS) accelerometers to replace conventional strain-based sensors at the fatigue-susceptible area to monitor fatigue cracks continuously. Relative attempts can be found in previous research [8–10]. In this paper, we propose a robust deep-learning-based method to interpret acceleration data for crack state identification, because it is still challenging for traditional vibration-based approaches to discover fatigue cracks [11–13]. Efforts are devoted to develop shallow model for crack states identification, considering the difficulty of obtaining various training samples for supervised approaches [14].

Recently developed one-dimension convolutional neural network (1D CNN), proposed by Kiranyaz et al. [15, 16], has provided an alternative powerful tool to discover damage-induced variations from 1D signals [17, 18]. Relative applications on accelerations measurements can be found for structural connection damage detection and structural state identification [19, 20], but rarely for fatigue cracks identification. For example, Abdeljaber et al. proposed to assign a 1D CNN model to each structural joint separately and training correspondingly [21]. For each model, training data are collected when the relevant joint is undamaged and totally damaged. It is evident that a big volume of data is required to prepare the whole training dataset for all joints, which can even be a big challenge for large scale and complex structures. Moreover, it is hard to obtain a dataset for each possible damage location. To overcome this drawback, Abdeljaber et al. further revised the system which requires only two data sessions that are measured when the structure is undamaged, i.e., all joints are in healthy condition, and fully damaged, i.e., all joints are fully damaged [22]. However, the critical issue is that the fully damaged condition is infeasible to be defined for fatigue cracks and impractical to be measured in advance. Zhang et al. also proposed to train a deep 1D CNN model to detect the bolted connection damage [23]. But the measurements were collected under impact hammer test, therefore, more evidence is required to prove the well-trained model could be accurate for structure under ambient excitations. Concerning applications on structural state identification, Lin et al. conducted research on a numerical beam with various damage conditions, simulating as element height reduction [24]. The vibration data were divided into 3-s-long segments as the input for a deep 1D CNN model. Zhang et al. further examined a simplified 1D CNN model for structural state identification [23]. The structural states were manually defined and altered by increasing the structural stiffness or mass, i.e. adding mass block or affix steel plate. The raw vibration data were also collected under specific excitations, i.e., hammer impact excitation to generate impulse load and actuator to generate white noise to excite the specimen.

Apparently, exiting attempts are relying on CNN model to automatically extract features from raw accelerations measurements directly. However, for real-time and continuous monitoring of fatigue cracks, massive data are expected to be recorded. If training the time-series data directly, especially the high-sampled acceleration data in a long-term monitoring manner, it will be time-consuming for CNN model to learning features. Hence, to reduce the difficulty in training process and to simplify the model configurations, we introduce blind source separation (BSS), specifically the second-order blind identification (SOBI) as the first feature conversion layer before any convolutional layers [25, 26]. The entire classifier we proposed is designated as B-CNN classifier. The main contributions of this paper can be highlighted as follows:

- In this paper, efforts are devoted to developing a deep-learning-based solution for a fatigue crack monitoring system using MEMS accelerometers, which can supplant conventional strain sensors at fatigue-susceptible regions, i.e., the welded details of the steel girder, for cracks monitoring in a long-term manner. Because accelerometers are portable, easy maintained, and sustainable under cracking condition.
- To delve the productive information on fatigue cracks from the vibration response, we introduce a BSS layer for sparse feature expression before convolutional layers. The highly sampled input data, regardless of data length, is converted to a group of sparse features in this BSS layer. In other words, the inputs with various lengths or sampling rates can be normalized to a standardized collection of sparse features. Hence, the length of input sources can be adjustable, it could be 2-s long for timely monitoring purposes or 10-min long for periodical evaluation targets, which will depend on the monitoring strategy.
- The B-CNN model is a shallow CNN model to save computation costs and improve its generalization

ability for fatigue crack identification when applied to other bridges. To obtain an optimal configuration of B-CNN classifier, various model hyperparameters are investigated in this paper. Furthermore, we also demonstrate the potential application of the well-trained B-CNN classifier to predict crack condition from unseen measurements.

- The ability of B-CNN classifier for classification and prediction is evaluated on an experimental fatigue test, which was conducted on a steel girder for about 1 month.

This paper is organized as follows. Section 2 first describes the configuration of proposed B-CNN classifier and detailed introduction of BSS and CNN. Section 3 then presents experimental bridge and data preparation procedure. After that, Sect. 4 discusses the B-CNN training, evaluation and hyperparameter investigation results, while Sect. 5 demonstrates the ability of B-CNN classifier for crack states prediction from unseen accelerations. Finally, results are summarized in Sect. 5.

2 Architecture of B-CNN classifier

In this section, the architecture of developed B-CNN classifier is exhibited. As described in Fig. 1, the input accelerations can be divided into two sets. 80% of prepared accelerations are utilized for model training, while the rest 20% accelerations are employed for model evaluation. The training process includes three blocks, namely BSS blocks, convolutional blocks and classification blocks, whose descriptive explanations will be given in the next three sub-sections.

2.1 BSS blocks

In this part, blind source separation (BSS) [26], specifically the second-order blind identification (SOBI) algorithm proposed by McNeill and Zimmerman [25], is employed to establish the sparse expression of the input acceleration data with high sampling rate. SOBI has been approved in many studies to be a sufficient tool to separate components in agreement with modal responses [27, 28]. The procedure of SOBI can be summarized as follows.

(a) The pre-processing procedure includes centering and whiten process. The raw acceleration measurements are first centered and combined with 90° shifted data [25]. The centralized combined data for blind separation can be expressed in Eq. (1).

$$\boldsymbol{X} = \boldsymbol{MS} = \sum_{i=1}^{n} a_i s_i \tag{1}$$

where $\boldsymbol{X}$ is the centralized combined measurements, $\boldsymbol{M}$ is mixing matrix, $\boldsymbol{S}$ is original sources, n is the number of sensors. The subsequent whitening process on $\boldsymbol{X}$ is to reduce the estimated number of parameters in mixing matrix and subsequently reduce the computational complexity. The proof can be found in [25] and previous paper [29]. The whitened data, $\boldsymbol{Z}$, and the whitening matrix, $\boldsymbol{M}_w$, can be obtained for further separation.

(b) Apply joint approximate diagonalization (JAD) to $\boldsymbol{Z}$ to obtain the joint diagonalize, $\boldsymbol{\Psi}$, where $\boldsymbol{\Psi}$ approximately diagonalizes a set of covariance matrices, seeing Eq. (2) [25].

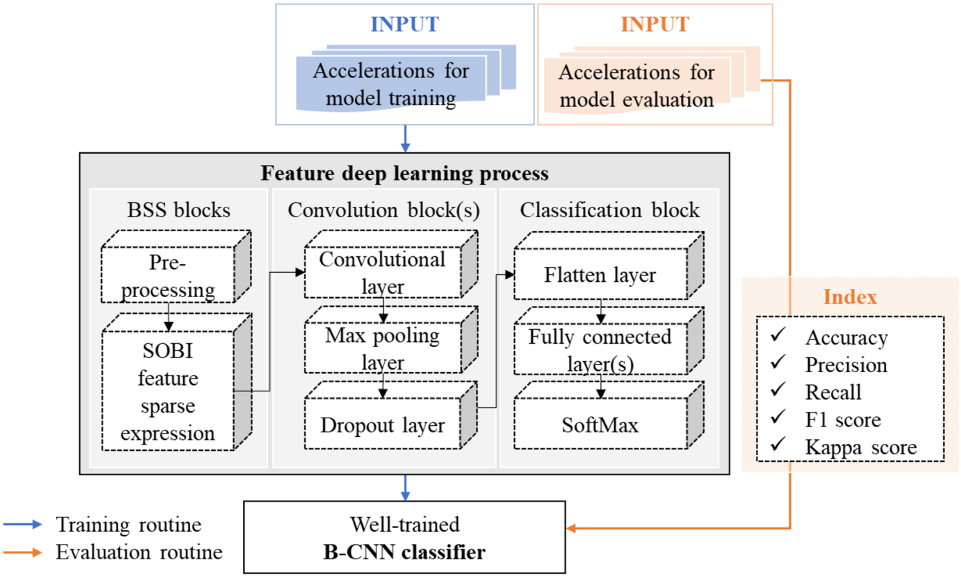


Fig. 1 Schematic of B-CNN method for fatigue crack monitoring

$$\boldsymbol{\Psi}^T R_Z\left(\tau_i\right)\boldsymbol{\Psi} \approx \text{diagonal} \tag{2}$$

where $\tau_i, i = 1, \ldots, j,$ is the time lag, and $R_Z\left(\tau_i\right) = E\{z(t)z(t+\tau_i)\}$ represents time-lagged covariance matrices of whiten sources, $z(t)$. SOBI jointly diagonalizes several time-lagged covariance matrices, because Belouchrani et al. illustrated that considering several time lags can significantly increase the robustness of SOBI [30].

(c) According to $\boldsymbol{\Psi}$ and the whitening matrix $\boldsymbol{M}_w$, the de-mixing matrix can be calculated by $\boldsymbol{W} = \boldsymbol{\Psi}^T\boldsymbol{M}_w$ [25], followed by the estimated sources by $\boldsymbol{S}^* = \boldsymbol{WX}$.

(d) The expected sparse expression includes pseudo mode shape estimates, $\boldsymbol{\Phi}_c$, which is the de-mixing matrix, $\boldsymbol{W}$, and the pseudo modal response estimates, $q(t)$, which is the estimated sources $s^*(t)$ [26, 31, 32]. The pseudo modal response can be further interpreted into pseudo frequencies and damping features by adopting SDOF analysis [25].

The outputs of BSS blocks are the sparse expression extracted from raw accelerations data. The number of sparse features is $n_{sf} = 2 \times n^2$, where n is the number of sensors. Apparently, the sparse features are not affected by the length of raw measurements, which means the length of input data is flexible.

2.2 Convolution and classification blocks

The function of convolution blocks is the feature deep learning, which comprises a 1D convolutional layer, a max pooling layer (subsampling), and a dropout layer. In the convolution layer, the filter is sliding along input vector, where the dot product is computed at each step. In this paper, Rectified Linear Units (ReLU) is selected and applied on the calculated values point-wisely to learn non-linear decision boundaries. After that, the feature maps can be generated as the output of convolutional layer. A max pooling layer is then employed to reduce feature representations. Then, the dropout technique is considered to address the overfitting challenge [33, 34].

In the classification block, the output from the convolution blocks, i.e., feature maps, are then flattened into a vector and passed to the feed-forward and fully connected layers for pattern classification. Finally, the outputs from fully connected layer are passed to the layer of SoftMax which contains the same number of expected classes neurons.

The layers in convolution blocks and classification blocks are stacked and trained together, where the stochastic gradient descent (SGD) is adopted as the optimizer [35]. The updated learning rate decay is employed to faster the learning and convergence speed by adopting large learning rate at the beginning for quick converge and small learning rate during later training procedure for precise adjustments to reach the optimum.

In summary, the configuration of a training process is generally formed by the following hyper-parameters, which may significantly affect the model classification accuracy: number of hidden convolutional and fully connected layers, number of filters or neurons, kernel size in each convolutional layer, pooling layer type and length, activation functions, learning rate decay schedule, and dropout layer configuration.

2.3 Model evaluation index

The trained model is assessed by five indexes, which are model accuracy, macro precision, macro recall, macro F1-value, and Cohen's kappa score [36], seeing Eqs. (3)–(6).

$$Macro\_P = \frac{1}{N}\sum_{i=1}^{N} P_j, \quad \text{whre } P_i = \frac{TP_i}{TP_i + FP_i} \tag{3}$$

$$Macro_R = \frac{1}{N}\sum_{i=1}^{N} R_i, \quad \text{where } R_i = \frac{TP_i}{TP_i + FN_i} \tag{4}$$

$$Macro\_F1 = 2 \times \frac{Macro\_P \times Macro\_R}{Macro\_P + Macro\_R} \tag{5}$$

$$\kappa = \frac{P_0 + P_c}{1 - P_c}, \quad \text{where } P_0 = \sum_{i=1}^{N} P\left(x_{ii}\right), P_c = \sum_{i=1}^{N} P\left(x_{i,\sim}\right)P(x_{\sim,i}) \tag{6}$$

Here, N is the number of classes, TP is true positive, TN means true negative, FP represents false positive, and FN is short for false negative. For Eq. (7), P_0 is accuracy and P_c means the agreement probability, while $P(x_{ii})$ represents the probabilities on the main diagonal of the confusion matrix, and $P\left(x_{i,\sim}\right)P(x_{\sim,i})$ are the columns and rows marginal probabilities, respectively. The higher kappa score means the better performance of classification. Normally, $\kappa > 0.8$ shows good classification results [37].

3 Experimental bridge description and data preparation

The evaluation procedures on proposed B-CNN classifier were carried out on an experimental fatigue test. The test was conducted by Tokyo City University in 2016 on an I-beam steel girder for about 1 month. Figure 2 provides an overview of test girder. A harmonic load with amplitude from 40 to 700 kN and a frequency of 1.55 Hz were applied on two positions, indicated in Fig. 2, where three MEMS

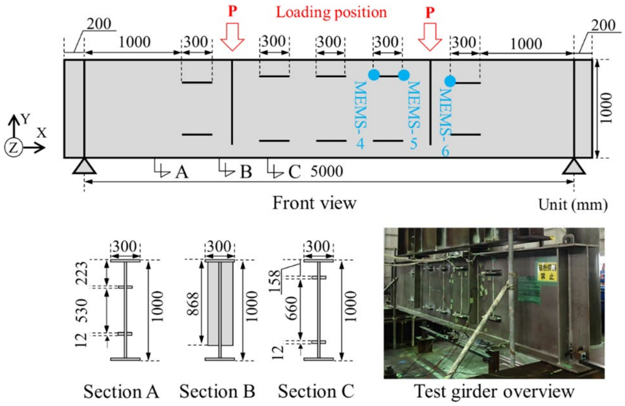


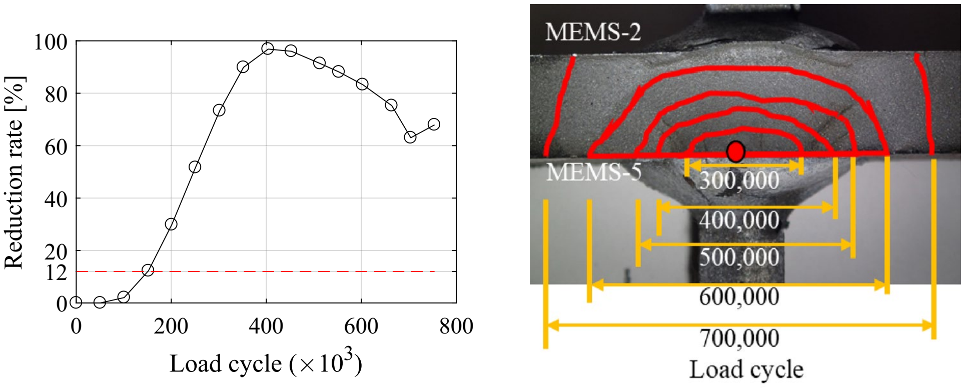
Fig. 2 Test girder overview with loading position and sensor location

accelerometers, MEMS4-6 were placed on the fatigue-susceptible regions, i.e., the welded details.

With cyclic loading repeats, fatigue cracks were initiating and propagating. According to strain monitoring records, the initial cracks first appeared near the MEMS-5 sensor, because the strain range dropped by 12% when the load cycles reach 152,000, as shown in Fig. 3a. After that, cracks were developing from MEMS-5 to its opposite side, where MEMS-2 located, seeing Fig. 3b, and the length and depth of cracks were accumulating consequently. The fatigue cracks propagation records are summarized in Fig. 4.

Three segments of raw measurements from MEMS-5 with various crack conditions are presented in Fig. 5. Apparently, the amplitude of measured accelerations cannot show any evidence of the crack propagation. Moreover, it is also

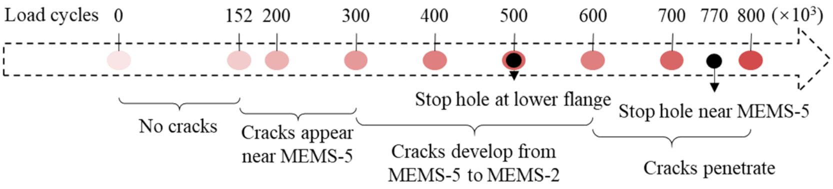
Fig. 3 Fatigue crack propagation with strain monitoring

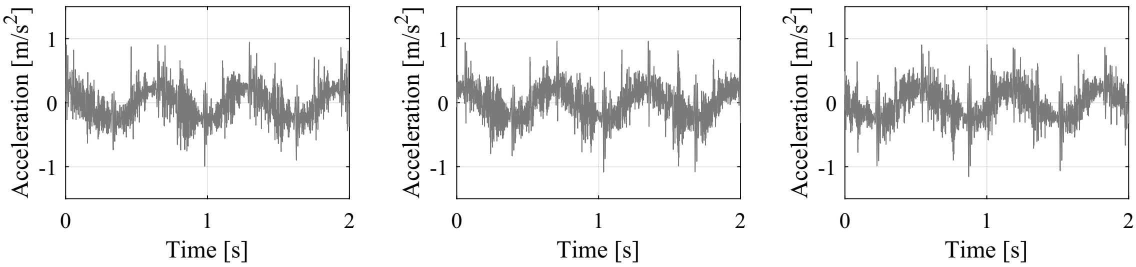


(a) Strain monitoring near MEMS-5

(b) Crack conditions

Fig. 4 Test log of cracks' propagation





(a) Without cracks (50,000 cycles)

(b) Crack initiation (153,000 cycles)

(c) Crack developing (250,000 cycles)

Fig. 5 Raw measurement data from MEMS-5 with various crack conditions

reflected and proved by our acceleration measurements in time domain that fatigue cracks near MEMS-5 cannot result in visible variations in MEMS-4 and MEMS-6. In this paper, the B-CNN classifier is designed to investigate the regional conditions and we believe that it is efficient to reflect cracks condition in this area. Therefore, the subsequent two sections will examine the B-CNN method for fatigue crack identification and prediction.

4 B-CNN classifier for classification

The target of this section is to evaluate the stability and accuracy of the proposed B-CNN classifier for identifying crack conditions at different stages from acceleration measurements.

The detailed flowchart of the proposed B-CNN classifier is presented in Fig. 6. Details are discussed in the succeeding subsections. Moreover, this section also set out to explore intrinsic hyperparameters within the model that may affect the classification results. The results obtained are summarized at the end of this section.

4.1 Data segmentation

To examine the classification ability of the proposed B-CNN model, measurements from three sensors are manually annotated into six groups, from level 0 to level 5, based on the loading cycles and crack propagation. As can be seen from Fig. 7, level 0 indicates the initial development stage of cracks, while levels 1 and 2 represent the developing and accelerating stages. Three-channel raw measurements are then split into 10-s segments. In total 4200 segments are prepared, including 1590 segments for level 0, 450 for level 2, and 540 for the rest of other levels. 80% of total 4200 segments, i.e., 3360 samples, are selected randomly for model training, while the rest 20% is saved for model evaluation.

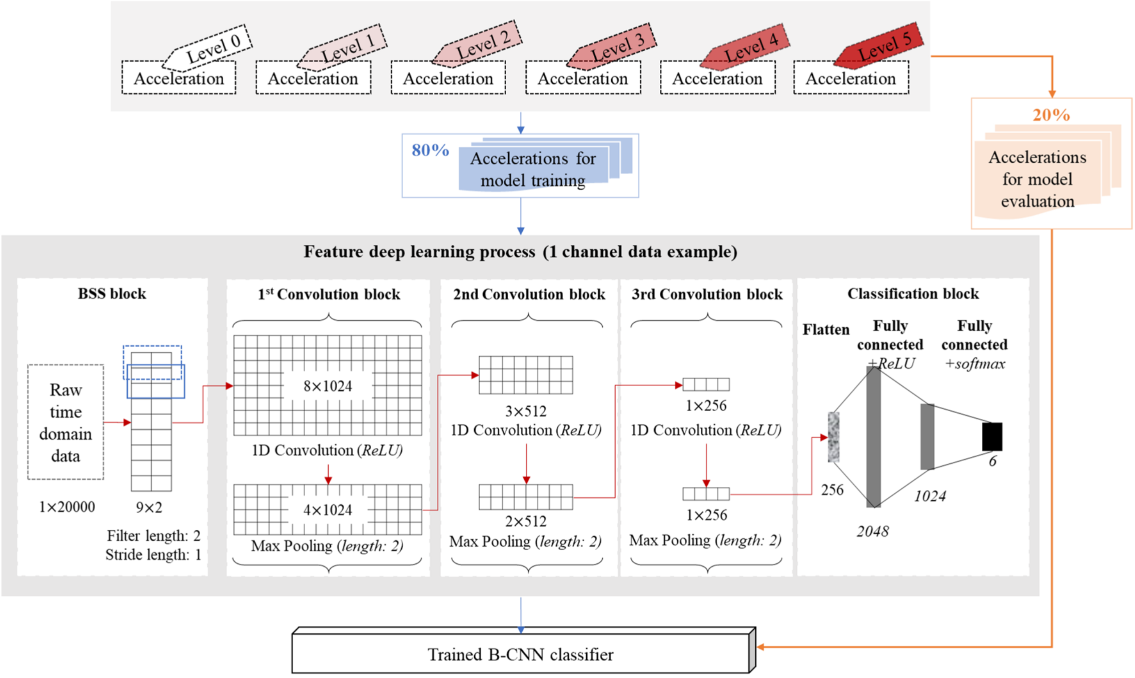
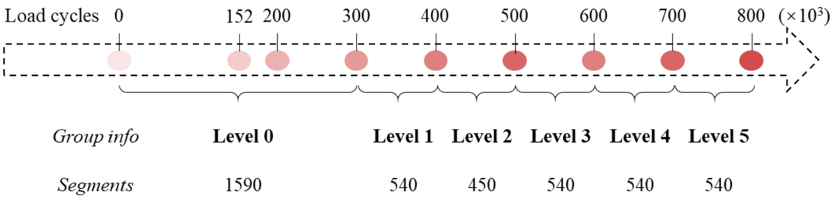


Fig. 6 Flowchart of training the B-CNN classifier for fatigue crack classification

Fig. 7 Data segmentation for crack condition classification



4.2 Feature deep learning process

As illustrated in Fig. 6, the labeled acceleration segments were first sent to BSS blocks to obtain the sparse expression. In this case, each segment is three-channel and 10-s long, thus the number of sparse features is 18.

Those 18 sparse features are then transformed into convolution block for further feature learning. It can be seen from Fig. 6 that input layers is reshaped from 1 × 18 to 9 × 2. Three convolution blocks are composed, including convolution layer and max pooling layer. The activation function is ReLU for all convolution layers. Three dropout layers are following each convolution block with the dropout rate set to 0.2, 0.2, and 0.4, respectively. Three fully connection layers are established with 2048, 1024, and 6 neurons embedded individually, while the ReLU activation function are used for first two fully connected layers, and the SoftMax function are employed in the final layer. The model is set to training 150 epochs, while batch size is 20 and stochastic gradient descent is employed as the optimal function.

Among our proposed model, learning rate is decayed from 0.01 to 0 within the total 150 or 200 epochs. The decay function is followed by Eq. (7)

$$\text{decay} = \left(1 - \frac{epoch}{maxEpochs}\right)^5, \quad a = a_{\text{initial}} * \text{decay} \qquad (7)$$

where α is learning rate, *maxEpochs* means the total number of epochs, and *epoch* represents current epoch number.

During the training process, cross-validator was adopted in this paper, thus 10% training segments were utilized for validation during the model training. The training history is shown in Fig. 8. As can be seen from Fig. 8, the accuracy and loss ratios are converged well between the training and validation process, which means the model can be trained properly. The model is eventually converged after 100 epochs, and the classification accuracy can reach up to 96% during the deep learning process. The feature deep learning process requires 2624 s to generated a trained CNN model. As aforementioned, 840 samples have been saved for model evaluation. Figure 9 indicates the confusion matrix of evaluation results and Table 1 lists detailed assessment criteria. The B-CNN model can identify crack conditions with 93% accuracy, macro precision at 0.91, and macro recall is 0.91, macro F1 measure is 0.91, while kappa scores reaches 0.90.

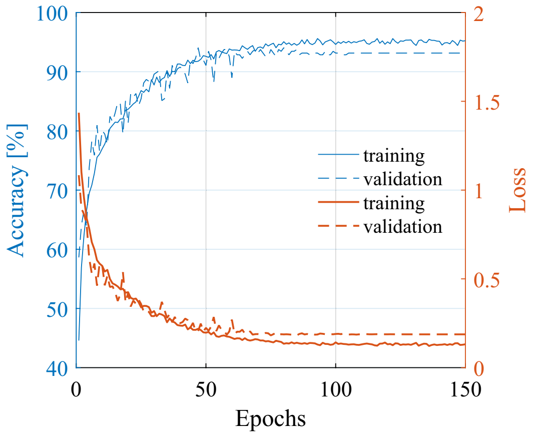
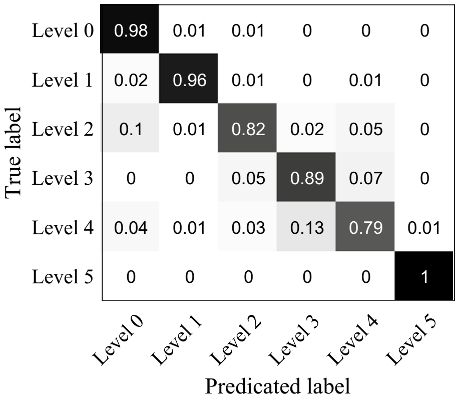


Fig. 8 Model training history

In this feature deep learning model, the classification ability is seriously affected by a number of hyperparameters. According to exist research outcomes, there is no exact guidelines for optimal parameters configuration [35, 38]. Hence, the rest part of this section will explore the optimal network architecture via trial-and-error analysis and investigate the impact of several hyperparameters, including model depth, the number of kernels, batch size, kernel size, and dropout rate.

4.2.1 Model depth

According to [16], recent research works have observed that shallow 1D CNN architectures are capable of learning features from 1D signals. Therefore, we examine the model depth up to three convolutional blocks. The following Fig. 10 presents six examples with different CNN configurations, for each layer conditions. The difference among six attempts under same layer condition is the various filters number and dropout rate to avoid overfittings.



	Level 0	Level 1	Level 2	Level 3	Level 4	Level 5
Level 0	0.98	0.01	0.01	0	0	0
Level 1	0.02	0.96	0.01	0	0.01	0
Level 2	0.1	0.01	0.82	0.02	0.05	0
Level 3	0	0	0.05	0.89	0.07	0
Level 4	0.04	0.01	0.03	0.13	0.79	0.01
Level 5	0	0	0	0	0	1

Fig. 9 Model evaluation results

Table 1 Assessment of the trained model for crack condition classification

Index	Accuracy	Macro precision	Macro recall	Macro F1	Kappa
Value	93%	0.91	0.91	0.91	0.90

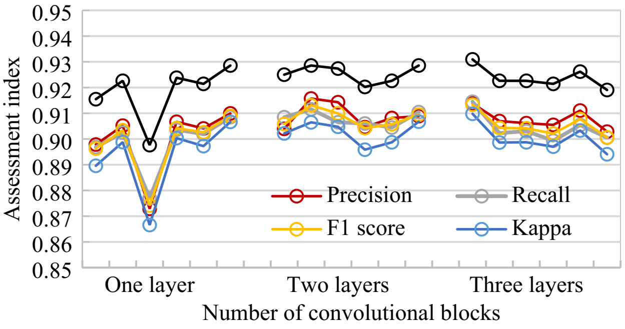


Fig. 10 Dependency between modal performance and model depth

As shown in Fig. 10, marginal improvements can be demonstrated when the model is getting deeper. For models with only one convolutional block, the highest model accuracy is 93% while the lowest is 90%. When increasing the convolutional block to two or three, the model accuracy can be improved and between 92 and 93%. Same tendency can also be identified in model macro-precision, macro-recall, macro-F1 measure, and Kappa index.

4.2.2 Filter number

To get further investigation on the dependency between filter number and model performance, a group of evaluations are conducted on a CNN model with only one convolutional block. The number of filters examined in this paper are 16, 32, 64, 128, 256, 512, 1024, and 2048. Results are compared in Fig. 11.

It is notable from Fig. 11a that augment of filters' number has a positive effect on the accuracy during training process. However, there is a trade-off between accuracy and overfitting, since the valid accuracy is lower than the train accuracy when the filter number is over 256. As illustrated from Fig. 11b, when the filter number varies from 512 to 2048, the improvement is negligible in terms of accuracy, macro precision, macro recall, macro F1 score, and kappa scores. The precise values can be found in Table 2. The accumulated training time is also listed in Table 2. The maximum consuming training time occurs when 64 filters is chosen, which costs 3467 s. However, the dependency between training time consumption and the number of convolutional filters is not distinct.

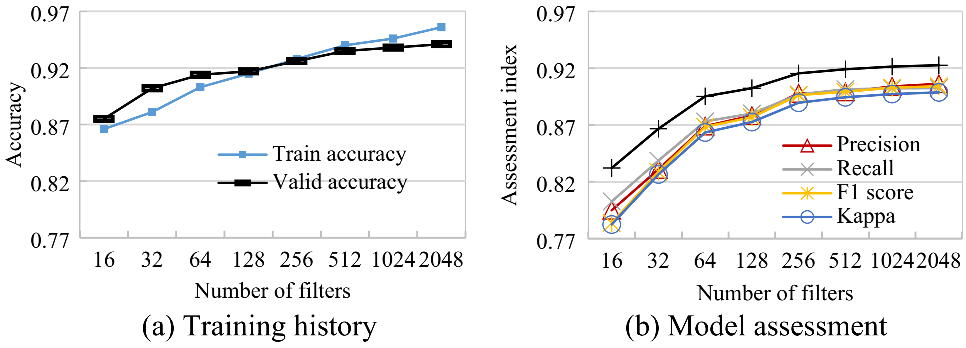
Generally speaking, there is a clear trend of increasing filter size can increase the model accuracy. More specifically, smaller amount of kernels cannot learn sufficient representations but require abundant computation cost, while, on the contrary, considerably larger amount of kernels can result in critical overfitting issues.

4.2.3 Batch size

We further examine the influence of batch size on model classification accuracy on the condition of filter size 2048, as shown in Fig. 12. When the batch size is over 20, the model accuracy and other criteria are all drop down. In our paper, considering the model performance and time consuming, the batch size between 10 and 20 are utilized.

4.2.4 Filter size

It can be demonstrated from Fig. 13 that the network is not very sensitive to this parameter. The accuracy does not drop significantly while changing the filter size.



(a) Training history

(b) Model assessment

Fig. 11 Dependency between the model performance and number of convolutional filters

Table 2 Model train time of various filters numbers

Filter number	16	32	64	128	256	512	1024	2048
Train time (s)	1107	1302	3467	1579	1203	1502	1279	713

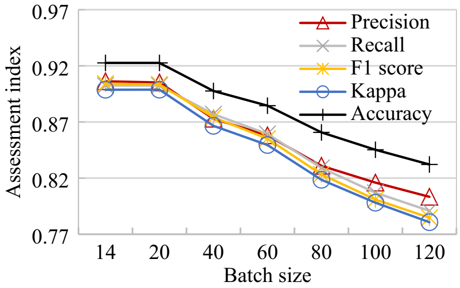


Fig. 12 Dependency between model performance and batch size

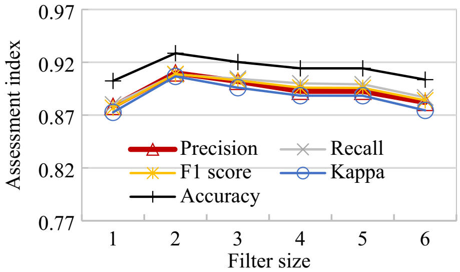


Fig. 13 Dependency between model performance and filter size

4.2.5 Dropout rate

Another crucial parameter of CNN is the dropout rate. According to Fig. 14, the smaller the dropout rate the higher model accuracy, but evidential model overfitting followed. In other words, dropout rate is another valid tool to avoid the overfitting issues. Therefore, its values between 0.3 and 0.4 turned to be the most efficient in this task.

4.3 Discussion and summary

From above experimental investigation, we can see that the B-CNN model can recognize crack conditions with 93% accuracy. Moreover, the model has high robustness since the classification accuracy is stable when modulating hyper-parameters. To be specific, the investigation results suggest that a shallow CNN model is sufficient since marginal improvements can be observed when the model is getting deeper. The number of filters should be considered not less than 256 because a smaller number of kernels cannot learn sufficient representations. The batch size is suggested between 10 and 20, while the dropout layer with 0.1–0.4 values can be adopted to avoid overfitting conditions. At last, this model is not sensitive to the filter size due to small dimension of input segment.

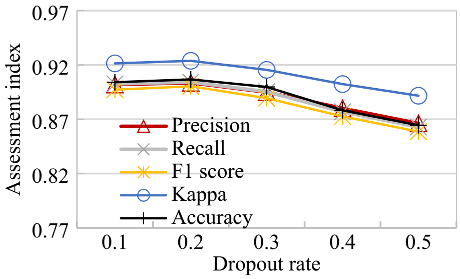


Fig. 14 Dependency between model performance and dropout rates

In this case study, the feature deep learning process are carried out using a PC with MATLAB 2016b and NVIDIA GeForce GTX 1080 Ti.

5 B-CNN classifier for prediction

The previous section has discussed and proved the stability of the proposed B-CNN classifier for crack recognition during cracks' propagation stage. In this section, the prediction procedure based on the well-trained B-CNN classifier will be investigated. The schematic prediction procedure is elaborated in Fig. 15.

The acceleration measurements when the loading cycle is lower than 300,000 times are utilized for B-CNN classifier training, while future records when load cycles are over 300,000 will be predicted by the pre-trained classifier. Details will be discussed in the forthcoming subsections.

5.1 Data segmentation

Figure 16 below indicates the data segmentation for this case study. According to crack condition monitoring records that were previously shown in Fig. 4a, measurements when loading cycle is lower than 300,000 can be classified into three groups as displayed in Fig. 16. First group data are designated as 'Health', when the structure was in healthy condition without observed cracks. The second group named 'Crack L-I', when cracks initiated, while the third group is 'Crack L-II' when cracks developed and are more serious

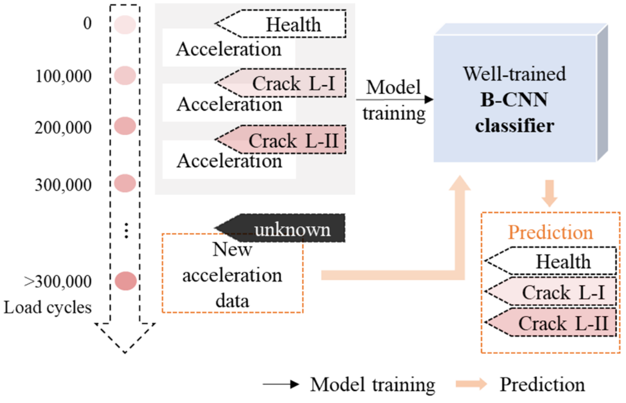


Fig. 15 Schematic of B-CNN model for fatigue crack prediction

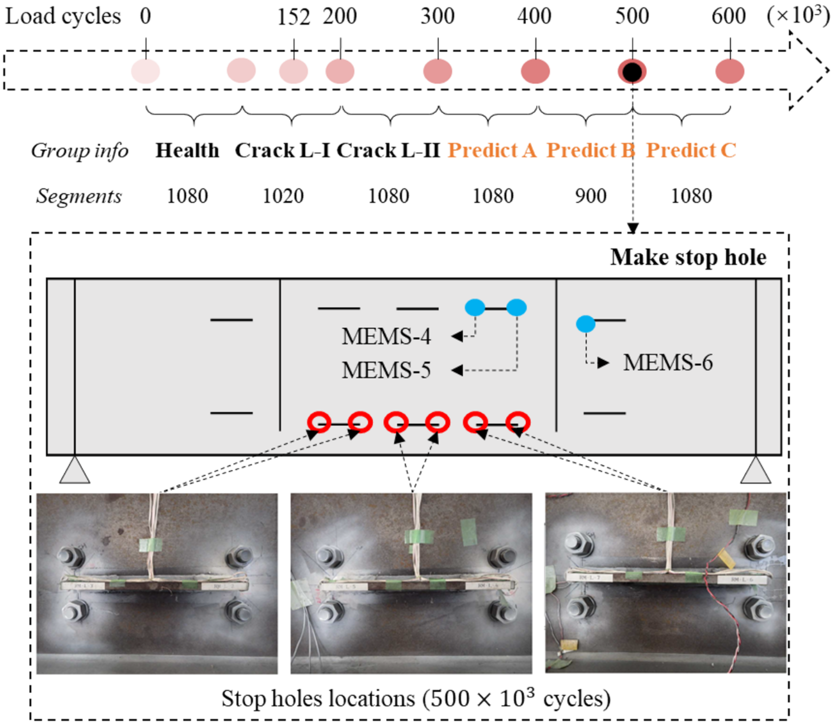

Load cycles 0 152 200 300 400 500 600 (×10³)
Group info Health Crack L-I Crack L-II Predict A Predict B Predict C
Segments 1080 1020 1080 1080 900 1080
Make stop hole
MEMS-4
MEMS-5
MEMS-6
Stop holes locations (500 × 10³ cycles)

Fig. 16 Data segmentation for classifier training and prediction

than ‘Crack L-I’. Another three groups of unseen acceleration records, designated as ‘Predict A’, ‘Predict B’, and ‘Predict C’ when the loading cycle is over 300,000, are utilized to examine the prediction performance of the trained classifier. As shown in Fig. 16, stop holes were tightened by bolts after 500,000 load cycles at the lower edge of the web. Hence, the influence due to stop holes reinforce should be reflected in MEMS-4, MEMS-5, and MEMS-6, since the distributed monitoring network is monitoring the regional vibration features. However, the variance in raw measurements, seeing Fig. 17, is not apparent. Hence, we evaluate the ability of proposed B-CNN method for structural performance prediction in subsequent subsections. All raw acceleration data are split into 5-s segments and Fig. 16 lists the specific number of segments for all six groups.

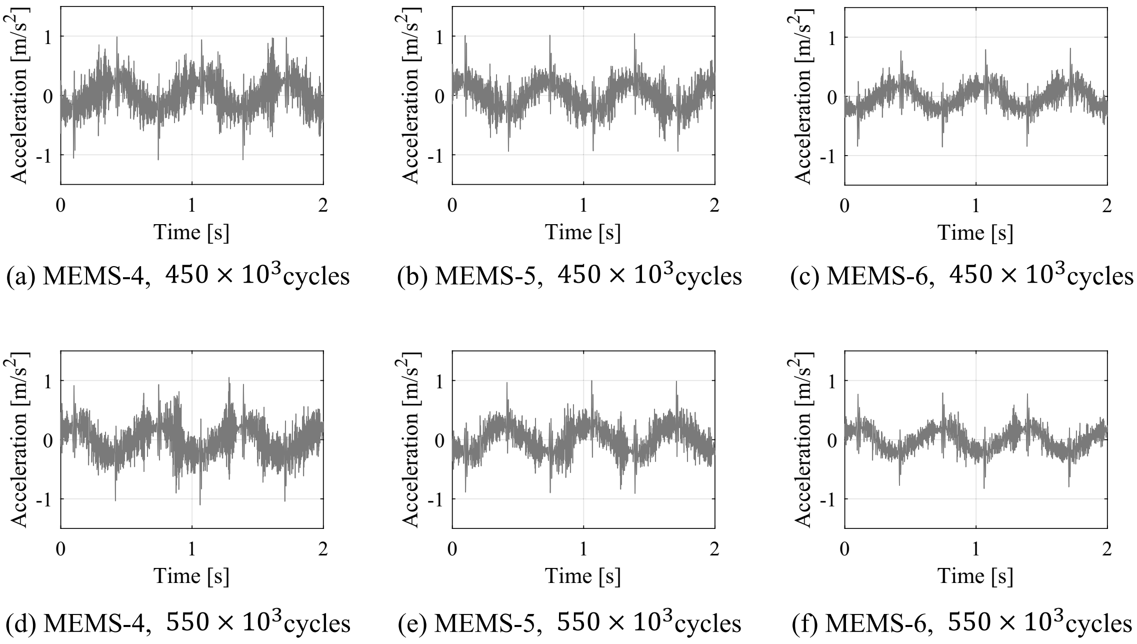
5.2 Classifier training

The classifier training process is same as the processes described in preceding Sect. 4. 80% of segments from ‘Health’, ‘Crack L-I’ and ‘Crack L-II’ are randomly selected for training while the remainder 20% are used for classifier evaluation. The assessment results are summarized in Table 3. Apparently, the B-CNN classifier can identify labelled data with 91% accuracy, while the precision, recall, and F1 measure are all 0.91, and the Kappa score is 0.87.

5.3 Prediction

Three groups of data, termed as ‘Predict A’, ‘Predict B’, and ‘Predict C’, are employed to assess the prediction performance of the pre-trained B-CNN classifier. ‘Predict A’ contains samples collected when the load cycle is between 300,000 and 400,000 times, while ‘Predict B’ covers samples collected when the load cycle is between 400,000 and 500,000 times. Therefore, the data from ‘Predict B’ is expected to indicate a more serious cracking condition when compared with ‘Predict A’. After that, the ‘Predict C’ contains samples collected after the stop holes were introduced, which means the cracking condition was repaired. We then utilize pre-trained B-CNN to identify the status of these three datasets, whose results are listed in Table 4.

As can be seen from Table 4, 98% of segments from ‘Predict A’ are identified in crack conditions either ‘Crack L-I’ or ‘Crack L-II’, and only 2% of segments are identified as ‘Health’ conditions. The same trend can be observed for ‘Predict B’ and ‘Predict C’ as well. As previously described in Fig. 16, the label ‘Crack L-I’ represents the initiated cracks initiated, while the label ‘Crack L-II’ means the cracks developed, which is more serious than ‘Crack L-I’. According to Table 4, the dataset of ‘Predict B’ has a higher chance under serious cracking conditions, since 72% of samples have been classified into ‘Crack L-II’ while 62% of samples of ‘Predict A’ are predicted as ‘Crack L-II’.



(a) MEMS-4, 450×10^3cycles (b) MEMS-5, 450×10^3cycles (c) MEMS-6, 450×10^3cycles

(d) MEMS-4, 550×10^3cycles (e) MEMS-5, 550×10^3cycles (f) MEMS-6, 550×10^3cycles

Fig. 17 Raw measurement data from MEMS 5 before stop holes installed

Table 3 Assessment of trained model for crack condition prediction

Index	Accuracy	Macro precision	Macro recall	Macro F1	Kappa
Value	91%	0.91	0.91	0.91	0.87

Table 4 Prediction results

Dataset	Samples	'Health' (%)	'Crack L-I' (%)	'Crack L-II' (%)
'Predict A'	1080	2	36	62
'Predict B'	900	2	26	72
'Predict C'	1080	2	58	40

In terms of 'Predict C', the data are collected after stop holes were introduced, as indicated in Fig. 16. The results show that 98% of 1080 segments are identified with unhealthy conditions. But more segments are identified as mild damaged conditions ('Crack L-I'), i.e., increasing from 26 to 58% when comparing with prediction outcomes of group 'Predict B', which can reflect that those stop holes might alleviate the crack severity. Therefore, we believe the B-CNN model can predict the improved tendency of those stop holes for preventing cracks developed. Moreover, we believe if the sensors are installed at all joints, the accuracy of prediction results could be increased.

6 Concluding remarks

In this paper, we employ portable and mature MEMS accelerometers to monitor fatigue crack propagations. A corresponding B-CNN classifier has also been developed to interpret the massive acceleration data. The performance of proposed B-CNN classifier for cracks condition classification and prediction are evaluated using an experimental fatigue test performed on a steel girder. The results can demonstrate the feasibility of considered B-CNN classifier in the following aspects.

- First of all, the proposed B-CNN classifier has the benefits of adaptive input data arrangement. Instead of directly commencing model training on high-sampled acceleration data, raw time series are first sparsely expressed through the first BSS block before convolutional blocks. Benefit from this, the input sources are flexible in length and the B-CNN classifier is suitable for both long-term and in-time monitoring requirements.
- To obtain an optimal B-CNN classifier to identify cracks in various propagation stage, multi-hyperparameters within the 1D CNN framework have been discussed in this paper based on the trial-and-error analysis. Evaluation results attest the B-CNN classifier has high robustness. To be specific, it implies that shallow layers are sufficient enough since the input data have been sparsely expressed in the BSS block. The overfitting

issues due to a deeper architecture or more filters can be solved by adding a dropout layer ranging from 0.1 to 0.4 after each convolutional block. Generally, a larger filter number should pair with a large dropout rate. In terms of batch size and filter size, we recommend batch size not exceeds 20 and the influence of filter size is not considerable in this case.

- To predict the crack condition from the unseen measurements, we also trained the B-CNN classifier using three groups of measurements, which represents the healthy condition without cracks, the initial cracking stage, and the propagation stage, respectively. The forthcoming acceleration measurements are evaluated by this classifier and predicted results show the correct tendency as expected.

Limitations of the proposed B-CNN exist since new training samples are necessary when applying to other bridges. Moreover, to improve the prediction accuracy, the B-CNN classifier is suggested to be updated to encompass the latest condition of fatigue cracks. Therefore, future research work will devote to enhancing the generalization ability of B-CNN by reducing the size of training samples.

Acknowledgements This study was carried out as cooperative research with the Tokyo Metropolitan Expressway Co., Ltd., Shutoko Engineering Co., Ltd., and the Highway Technology Research Center. The acceleration measurements were supported by Kyowa Electronic Instruments. This work is partial supported by the National Natural Science Foundation of China (Grant No. 52108118).

Data availability Some or all data, models, or code generated or used during the study are proprietary or confidential in nature and may only be provided with restrictions. These data can be provided by the second author, if all partners agree with the provision. These data can only be used for research, and publishing results based on those data should be approved by the second author in advance.

References

1. Quqa S, Martakis P, Movsessian A et al (2022) Two-step approach for fatigue crack detection in steel bridges using convolutional neural networks. J Civ Struct Heal Monit 12:127–140. https://doi.org/10.1007/s13349-021-00537-1
2. Brownjohn JMW, De Stefano A, Helmut YX, Emin WA (2011) Vibration-based monitoring of civil infrastructure: challenges and successes. J Civ Struct Heal Monit. https://doi.org/10.1007/s13349-011-0009-5
3. Zhang Y (2006) In situ fatigue crack detection using piezoelectric paint sensor. J Intell Mater Syst Struct 17:843–852. https://doi.org/10.1177/1045389X06059957
4. Kong X, Li J, Collins W et al (2017) A large-area strain sensing technology for monitoring fatigue cracks in steel bridges. Smart Mater Struct 26:085024. https://doi.org/10.1088/1361-665X/aa75ef
5. Kesavan K, Ravisankar K, Parivallal S, Sreeshylam P (2005) Applications of fiber optic sensors for structural health monitoring. Smart Struct Syst 1:355–368. https://doi.org/10.12989/sss.2005.1.4.355
6. Glisic B, Inaudi D (2012) Development of method for in-service crack detection based on distributed fiber optic sensors. Struct Heal Monit An Int J 11:161–171. https://doi.org/10.1177/1475921711414233
7. Wang L, Han J, Song Y (2014) Fatigue performance monitoring of full-scale PPC beams by using the FBG sensors. Smart Struct Syst 13:943–957. https://doi.org/10.12989/sss.2014.13.6.943
8. Nguyen VK, Olatunbosun O (2007) A proposed method for fatigue crack detection and monitoring using the breathing crack phenomenon and wavelet analysis. J Mech Mater Struct 2:399–420. https://doi.org/10.2140/jomms.2007.2.399
9. Blunt DM, Keller JA (2006) Detection of a fatigue crack in a UH-60A planet gear carrier using vibration analysis. Mech Syst Signal Process 20:2095–2111. https://doi.org/10.1016/j.ymssp.2006.05.010
10. Yan G, De Stefano A, Matta E, Feng R (2013) A novel approach to detecting breathing-fatigue cracks based on dynamic characteristics. J Sound Vib 332:407–422. https://doi.org/10.1016/j.jsv.2012.09.008
11. Salawu OS (1997) Detection of structural damage through changes in frequency: a review. Eng Struct 19:718–723. https://doi.org/10.1016/S0141-0296(96)00149-6
12. Zhang T, Biswal S, Wang Y (2020) SHMnet: condition assessment of bolted connection with beyond human-level performance. Struct Heal Monit 19:1188–1201. https://doi.org/10.1177/1475921719881237
13. Koto Y, Konishi T, Sekiya H, Miki C (2019) Monitoring local damage due to fatigue in plate girder bridge. J Sound Vib 438:238–250. https://doi.org/10.1016/j.jsv.2018.09.009
14. Wang Z, Cha Y-J (2021) Unsupervised deep learning approach using a deep auto-encoder with a one-class support vector machine to detect damage. Struct Heal Monit 20:406–425. https://doi.org/10.1177/1475921720934051
15. Kiranyaz S, Ince T, Gabbouj M (2016) Real-time patient-specific ECG classification by 1-D convolutional neural networks. IEEE Trans Biomed Eng 63:664–675. https://doi.org/10.1109/TBME.2015.2468589
16. Kiranyaz S, Avci O, Abdeljaber O et al (2019) 1D convolutional neural networks and applications: a survey. Mech Syst Signal Process 151:107398
17. Ye XW, Jin T, Yun CB (2019) A review on deep learning-based structural health monitoring of civil infrastructures. Smart Struct Syst 24:567–585. https://doi.org/10.12989/sss.2019.24.5.567
18. Sun L, Shang Z, Xia Y et al (2020) Review of bridge structural health monitoring aided by big data and artificial intelligence: from condition assessment to damage detection. J Struct Eng 146:04020073. https://doi.org/10.1061/(ASCE)ST.1943-541X.0002535
19. Tang Z, Chen Z, Bao Y, Li H (2019) Convolutional neural network-based data anomaly detection method using multiple information for structural health monitoring. Struct Control Heal Monit 26:1–22. https://doi.org/10.1002/stc.2296
20. Duan Y, Chen Q, Zhang H et al (2019) CNN-based damage identification method of tied-arch bridge using spatial-spectral information. Smart Struct Syst 23:507–520. https://doi.org/10.12989/sss.2019.23.5.507
21. Abdeljaber O, Avci O, Kiranyaz S et al (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
22. Abdeljaber O, Avci O, Kiranyaz MS et al (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

23. Zhang Y, Miyamori Y, Mikami S, Saito T (2019) Vibration-based structural state identification by a 1-dimensional convolutional neural network. Comput Civ Infrastruct Eng 34:822–839. https://doi.org/10.1111/mice.12447
24. Lin YZ, Nie ZH, Ma HW (2017) Structural damage detection with automatic feature-extraction through deep learning. Comput Civ Infrastruct Eng 32:1025–1046. https://doi.org/10.1111/mice.12313
25. McNeill SI, Zimmerman DC (2008) A framework for blind modal identification using joint approximate diagonalization. Mech Syst Signal Process 22:1526–1548. https://doi.org/10.1016/j.ymssp.2008.01.010
26. McNeill S (2012) An analytic formulation for blind modal identification. J Vib Control 18:2111–2121. https://doi.org/10.1177/1077546311429146
27. Poncelet F, Kerschen G, Golinval JC, Verhelst D (2007) Output-only modal analysis using blind source separation techniques. Mech Syst Signal Process 21:2335–2358. https://doi.org/10.1016/j.ymssp.2006.12.005
28. Antoni J, Chauhan S (2013) A study and extension of second-order blind source separation to operational modal analysis. J Sound Vib 332:1079–1106. https://doi.org/10.1016/j.jsv.2012.09.016
29. Zhu Y, Ni Y-Q, Jin H et al (2019) A temperature-driven MPCA method for structural anomaly detection. Eng Struct 190:447–458. https://doi.org/10.1016/j.engstruct.2019.04.004
30. Belouchrani A, Abed-Meraim K, Cardoso JF, Moulines E (1997) A blind source separation technique using second-order statistics. IEEE Trans Signal Process 45:434–444. https://doi.org/10.1109/78.554307
31. Kerschen G, Poncelet F, Golinval J-C (2007) Physical interpretation of independent component analysis in structural dynamics. Mech Syst Signal Process 21:1561–1575. https://doi.org/10.1016/j.ymssp.2006.07.009
32. Mahvash A, Lakis AA (2016) Independent component analysis as applied to vibration source separation and fault diagnosis. J Vib Control 22:1682–1692. https://doi.org/10.1177/1077546314544349
33. Olson DH, Camblor MA, Villaescusa LA, Kuehl GH (2004) Light hydrocarbon sorption properties of pure silica Si-CHA and ITQ-3 and high silica ZSM-58. Microporous Mesoporous Mater 67:27–33. https://doi.org/10.1016/j.micromeso.2003.09.025
34. Srivastava N, Hinton G, Krizhevsky A et al (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15:1929–1958
35. Cha Y-J, Choi W, Büyüköztürk O (2017) Deep learning-based crack damage detection using convolutional neural networks. Comput Civ Infrastruct Eng 32:361–378. https://doi.org/10.1111/mice.12263
36. Cohen J (1960) A coefficient of agreement for nominal scales. Educ Psychol Meas 20:37–46. https://doi.org/10.1177/001316446002000104
37. de la Torre J, Puig D, Valls A (2018) Weighted kappa loss function for multi-class classification of ordinal data in deep learning. Pattern Recognit Lett 105:144–154. https://doi.org/10.1016/j.patrec.2017.05.018
38. Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press, London

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.