

Doctoral Thesis

**VIBRATION-BASED STRUCTURAL STATE
IDENTIFICATION BY USING DEEP LEARNING
METHOD**

ディープラーニングを用いた振動データによる構造物の
状態識別

2019 September

ZHANG YOUQI

Abstract

The deterioration of the aging infrastructures has become a global problem in recent decades, which threatens the public safety. To solve the above problem, researches on structural health monitoring (SHM) and structural damage detection (SDD) have been carried out all over the world. Researchers have made large amounts of efforts in the development of vibration-based SDD methods based on the theories of dynamics and signal processing.

In this thesis, an attempt on SHM has been made on a ballasts concrete railway bridge. By performing a series of vibration experiments, obvious variations on modal parameters have been found. The variations on the modal parameters show difficulties of SHM, which need to be overcome. The difficulties can be summarized as follows: (1) the effects of environmental variation and other uncertainties to the structural dynamic behavior, (2) low efficiency of using the large amounts of monitoring data.

In recent years, the rapid development of Deep Learning technology shows obvious advantages in many fields, such as object detection, medical science, and so on. Firstly, it is a pure data-driven method. By using Deep Learning technology, functions can be generated to link the input data to the results automatically, with no need of any domain knowledge. Secondly, large amounts of data can be used efficiently in the process of training a network.

Therefore, in this thesis, a vibration-based structural state identification method by using 1-D convolutional neural networks (CNNs) has been proposed. The proposed method aims to overcome the difficulties as introduced in the second paragraph by adapting the Deep Learning technology in the civil engineering field to solve the SHM problems.

By using the proposed 1-D CNNs, functions linking the raw vibration data and structural states can be established. Those 1-D CNNs are developed to identify tiny local structural changes, and are validated on actual structures. Databases of structural vibration response are established based on a T-shaped steel beam (in lab), a short steel girder bridge (in test field), and a long steel girder bridge (in service) to validate the performance of the proposed 1-D CNN. The complexities of data in above 3 databases increase progressively. The raw acceleration data are not pre-processed and are directly used as training and validation data. The well-trained CNNs almost perfectly identify the locations of the small local structural changes, demonstrating the high sensitivity of the proposed CNN to tiny changes in actual structures.

The capacity of determining the boundary between data in different structural states is also shown clearly.

Subsequently, to explore the mechanism of the proposed 1-D CNN, the convolutional kernels and outputs of the convolutional and max pooling layers are visualized and analyzed. The effectiveness of the CNN is also proved by visualizing the variation of data structure in each layer of the CNN by the T-SNE method.

Furthermore, to examine the capacity of identify untrained structural changes of the proposed CNN, robustness tests of the CNN models to locations of structural changes and temperature effect are carried out. The results show low capacity of the classification CNN model to identify local structural changes in untrained locations and temperature environment. Fortunately, the robustness to the temperature effect can be easily improved by expanding the training data acquired in diverse temperature conditions.

Finally, to improve the expression capacity of location and the robustness of the CNN to locations of structural changes, a regression CNN model is proposed with the updated encoding of the label and the output layer of the CNN. Comparing to the classification CNN models, higher robustness is obtained in the regression CNN model. Moreover, a deep network with multi-convolution blocks and multi-task outputs is proposed to further improve the robustness of identifying local structural changes in untrained locations. The results show obvious increase in the accuracies of the network to identify untrained local structural changes.

Overall, the expected contributions will be two-folds. For academy, the results of the study demonstrate the feasibility and rationality of using Deep Learning technology to solve SHM and SDD problems in civil engineering field. The potential of developing new Deep-Learning-based SHM and SDD schemes are also shown. For the society, the proposed research will boost the development of technology that guarantees human's daily safety.

Keywords

structural state identification,
deep learning,
convolutional neural network,
vibration experiment,
robustness test

Acronyms

| | |
|------------|------------------------------|
| ANN | artificial neural network |
| CNN | convolutional neural network |
| FC | fully connected |
| SHM | structural health monitoring |
| SDD | structural damage detection |

TABLE OF CONTENTS

| | |
|---|-----------|
| Chapter 1: Introduction | 1 |
| 1.1 Background | 1 |
| 1.2 Deep Learning in Civil Engineering | 3 |
| 1.3 Conventional Vibration-Based SDD Methodologies | 3 |
| 1.4 Deep-Learning-Based SDD Methodologies | 4 |
| 1.5 Purposes of the Thesis | 5 |
| Chapter 2: Investigations on the Variations of Modal Parameters | 7 |
| 2.1 Introduction | 7 |
| 2.2 Bridge Description | 7 |
| 2.3 Outline of the Experiments | 10 |
| 2.4 Results | 12 |
| 2.5 Conclusions | 20 |
| Chapter 3: Basic Concept of Deep Learning and Related Algorithms | 21 |
| 3.1 Introduction of Artificial Neural Network | 21 |
| 3.2 Layers of the Network..... | 22 |
| 3.2.1 1-D convolutional layer | 22 |
| 3.2.2 Batch normalization | 23 |
| 3.2.3 Max pooling layer..... | 23 |
| 3.2.4 Dropout | 24 |
| 3.2.5 Fully connected layer..... | 24 |
| 3.2.6 Softmax output layer..... | 25 |
| 3.3 Mechanisms of Learning | 25 |

| | |
|--|-----------|
| 3.4 Loss Functions | 26 |
| 3.4.1 Categorical cross entropy | 26 |
| 3.4.2 Mean squared error | 26 |
| 3.5 Optimizers | 27 |
| 3.5.1 Stochastic gradient descent | 27 |
| 3.5.2 Adam | 27 |
| 3.6 Deep Learning for Structural State Identification | 28 |
| 3.7 Introduction of Support Vector Machine | 28 |
| 3.8 Introduction of T-Distributed Stochastic Neighbor Embedding | 32 |
| | |
| Chapter 4: Structural State Identification by using a Classification 1-D Convolutional Neural Network | 34 |
| 4.1 Introduction | 34 |
| 4.2 T-shaped Beam Experiment | 36 |
| 4.2.1 Vibration experiment for data generation | 36 |
| 4.2.2 CNN configuration | 39 |
| 4.2.3 Results | 41 |
| 4.3 Short Steel Girder Bridge Experiment | 43 |
| 4.3.1 Vibration experiment for data generation | 43 |
| 4.3.2 CNN configuration | 46 |
| 4.3.3 Results | 47 |
| 4.4 Long Steel Girder Bridge Experiment | 49 |
| 4.4.1 Vibration experiment for data generation | 49 |
| 4.4.2 CNN configuration | 52 |
| 4.4.3 Results | 54 |
| 4.5 Conclusions | 58 |

| | |
|---|-----------|
| Chapter 5: Visualization and Analyses of CNN Model | 60 |
| 5.1 Introduction | 60 |
| 5.2 Convolutional Kernel Visualization | 60 |
| 5.3 Visualization of the Hidden-Layer Outputs | 64 |
| 5.3.1 CNN for the T-shaped beam database | 65 |
| 5.3.2 CNN for the short steel girder bridge database | 69 |
| 5.3.3 CNN for the long steel girder bridge database | 73 |
| 5.4 Variation of Data Structure in Each Layer | 78 |
| 5.4.1 Data structure visualization of the CNN for the T-shaped beam | 78 |
| 5.4.2 Data structure visualization of the CNN for the short steel girder bridge | 83 |
| 5.4.3 Data structure visualization of the CNN for the long steel girder bridge | 86 |
| 5.5 Conclusions | 90 |
| Chapter 6: Robustness Tests of the Classification CNN Model..... | 91 |
| 6.1 Introduction | 91 |
| 6.2 Tests of the Robustness to Locations of Structural Changes | 91 |
| 6.2.1 Vibration experiment for test data generation | 92 |
| 6.2.2 Robustness tests (Round 1) | 93 |
| 6.2.3 Robustness tests (Round 2) with expanded training data..... | 96 |
| 6.3 Tests of the Robustness to Temperature Effect | 102 |
| 6.3.1 Vibration experiment for test data generation | 102 |
| 6.3.2 Results | 103 |
| 6.3.3 Solution | 107 |
| 6.4 Conclusions | 110 |

| | |
|---|------------|
| Chapter 7: Updates of the CNN Model | 112 |
| 7.1 Introduction | 112 |
| 7.2 Regression CNN model | 112 |
| 7.2.1 Details of the regression CNN model..... | 112 |
| 7.2.2 Validation set up | 114 |
| 7.2.3 Results..... | 116 |
| 7.3 Deep Network with Multi-Convolution Blocks and Multi-Task Outputs | 119 |
| 7.3.1 Motivation | 119 |
| 7.3.2 Network details | 121 |
| 7.3.3 Validation set up | 127 |
| 7.3.4 Results..... | 127 |
| 7.4 Conclusions | 130 |
| Chapter 8: Conclusions and Future Works | 131 |
| 8.1 Conclusions | 131 |
| 8.2 Future Works..... | 134 |
| Acknowledgement | 136 |
| List of Publications | 138 |
| References | 140 |

Chapter 1

Introduction

1.1 Background

Many bridges were constructed in the post-World War 2 period, when construction technology was developing. As these bridges age, their service function is being degraded by cracking, fatigue, corrosion, and other structural damages caused by long-time service and geological hazards (AASHTO, 2008; MLIT, 2018), posing a major threat to public safety.

The White Paper On Land, Infrastructure, Transport and Tourism In Japan 2017 reports the detailed severity of the deterioration of the infrastructures in Japan: *“The amount of infrastructure that is 50 years or older is expected to increase at an accelerating pace in the near future. The rapid deterioration of the infrastructure that comprises Japanese land is a major problem that requires society-wide efforts to solve.”* (MLIT, 2018).

Figs. 1.1 and 1.2 show two example of the deterioration of bridges. The bridge in Fig. 1.1 is located in Kitami, Hokkaido, Japan. There are some loosen bolts on the bridge which reduce the local stiffness. Fig. 1.2 shows an operating bridge in Vienna, Virginia, USA. Concrete on the bridge has fallen off, and steel bars have been exposed. The severe deterioration makes local residents feel quite anxiety when crossing the bridge.



Fig. 1.1 Deteriorated bridge with loosen bolts (Kitami, Hokkaido, Japan)



Fig. 1.2 Severe deteriorated bridge (Vienna, Virginia, USA)

Even today, the safety and service function of bridges is often confirmed by visual inspection. However, the effectiveness of visual inspection is compromised by limited human resources, delayed discovery of damages, and huge time and budget costs. Automatically estimating the structural health conditions using smart techniques is therefore indispensable, which is also consistent with the trend of national policy of Japan that: *“development and introduction of monitoring technologies that provide an efficient insight into the conditions of social infrastructures”* (MLIT, 2018).

Thus, monitoring the health condition of old bridges is critically important to reduce the risk of public safety. Researchers have proposed various structural health monitoring (SHM) and structural damage detection (SDD) approaches to understand the structural states. Examples are the vision-based methods of Khuc and Catbas (2017), Barile et al. (2016), and Zaurin and Catbas (2010), and the vibration-based methods of Siringoringo and Fujino (2018), Wattana and Nishio (2017), Chang and Kim (2016), Farahani and Penumadu (2016), Döhler et al. (2014), Oshima et al. (2013), Kumar et al. (2012), and Rice et al. (2010).

1.2 Deep Learning in Civil Engineering

With the advent of the Deep Learning revolution (LeCun et al., 2015) and the increasing size of data, the advantages of data-driven approaches such as image classification (Krizhevsky et al., 2012; Szegedy et al., 2015; Lin et al., 2015; Koziarski and Cyganek, 2017), speech recognition (Graves et al., 2013; Graves and Jaitly, 2014), and natural language processing (Goldberg, 2017, Luong et al., 2015) have become increasingly evident. In data-driven methods, the machine learns from the data, and establishes an estimation model of the output status. Above advantage makes deep learning feasible to be used in the civil engineering field. Deep learning has been applied to civil engineering problems such as estimating concrete compressive strength (Rafiei et al., 2017), reliability analysis of transportation networks (Nabian and Meidani, 2018), and estimating the sales prices of new housing (Rafiei and Adeli, 2016). When implemented through a neural network, deep learning can automatically compute a set of functions in multiple inter-connected layers, linking the input data and the expected outputs in a holistic framework. Meanwhile, the data features are extracted automatically without requiring manual effort.

By adopting the deep learning paradigm, vision-based SDD methods have entered a new era in the civil engineering field. Several researchers have proposed crack and corrosion identification based on image processing techniques with convolutional neural networks (CNNs) (LeCun et al., 1998; LeCun and Bengio, 2003). Such techniques identify, localize and display the cracks and corrosions with very high accuracy (Cha et al., 2017; Cha et al., 2017; Tong et al., 2017; Xue and Li, 2018; Zhang et al., 2017a; Gao and Mosalam, 2018). However, vision-based SDD methods have an obvious limitation: they cannot detect invisible structural damages. In actual civil structures, the interior of the girder, slab, and many other parts cannot be easily inspected. Thus, developing a SDD method that identifies invisible structural damages is highly demanded.

1.3 Conventional Vibration-Based SDD Methodologies

Vibration-based SDD methods can potentially identify invisible structural damages, because (in theory at least) any local structural damage will change the mass and stiffness distributions of the structure, causing variations of the natural frequencies and mode shapes (Cawley and Adams, 1979; Pandey et al., 1991; Adeli and Jiang, 2008). To develop vibration-based SDD

methods, one must accurately identify the vibration parameters. Classical system identification methods have been reviewed in previous papers (Sirca and Adeli, 2012; Reynders, 2012). The literature is also replete with recent developments on system identification (Oh et al., 2017; Huang and Beck, 2018; Yao et al., 2018; Yin et al. 2017; Kim et al., 2017; Perez-Ramirez et al., 2016; Li et al., 2017; Amezquita-Sanchez et al., 2017). Vibration-based SDD methods with system identification have explored the correlations between the vibration features (natural frequency, mode shape, and their derivatives) and the structural damage information (Chang and Kim, 2016; Spiridonakos and Chatzi, 2014; Reynders et al., 2014; Yan et al., 2005; Ratcliffe, 1997; Pandey et al., 1991; Dutta and Talukdar, 2004; Ndambi et al., 2002). Vibration-based SDD methods are reviewed in Salawu (1997), Cruz and Salgado (2008), and Moughty and Casas (2017).

There are several main challenges when applying SDD methods in practice. First, natural frequencies are easily affected by environmental factors (Hu et al., 2013; Xia et al., 2012; Zhang et al., 2017b). The subtle natural frequency changes induced by low-scale damages are easily smothered by environmental effects. Damage detection in changing environments has been discussed in several papers (Deraemaeker et al., 2008; Spiridonakos and Chatzi, 2014; Reynders et al., 2014; Yan et al., 2005; Döhler et al., 2014). As SDD indicators, natural frequencies lack information of the damage locations. The second challenge is that: a lot of vibration-based SDD methods rely heavily on accurate identification of mode shapes. The measurement results of the mode shapes usually fluctuate in a certain range with the components of error. Authors concern that when the error is large, SDD methods based on mode shapes or its derivations have risks of failing to detect damages. Third, most of the traditional vibration-based SDD methods require manual exploration and determination of the structural damage indicators based on the researchers' knowledge and experience. An experience-based feature extraction process may be biased and risk omitting information that is vital to the structural damage diagnosis. These three limitations highlight the complexity of vibration-based SDD problems.

1.4 Deep Learning and Vibration-Based SDD Methodologies

Deep learning methods can potentially resolve the complicated vibration-based SDD issues, because the deep neural network generates a function that automatically links the raw

acceleration data to the damage information. Features of the acceleration data are extracted automatically, and the acceleration data are more fully exploited than in conventional vibration-based SDD methods. Lin et al. (2017) proposed a classification CNN with six convolutional layers and three maximum pooling layers. The CNN was trained by feeding raw vibration data generated by a finite element beam model. The damage was detected with very high accuracy (94.57%), proving that CNNs are suitable for vibration-based SSD. Abdeljaber et al. (2017) and Avci et al. (2018) identified the structural damages caused by loosened bolts on a steel frame by a CNN-based approach with very good performance. Another CNN-based SDD experiment was conducted on a benchmark frame structure (Abdeljaber et al., 2018), where structural damages were created by removing some braces. The method requests only two measurement data under health and severe damage conditions, reporting a good estimation of the overall structural health condition. One major achievement was the development of an unsupervised learning SDD method that does not require a large amount of label data (Rafiei and Adeli, 2017; Rafiei and Adeli, 2018).

However, the above approaches were either implemented on a computer-simulated beam model or on in-lab structures. Their performances on actual structures remains unknown. The feasibility of detecting small local structural changes on real-world structures by CNN-based methods needs to be validated.

1.5 Purposes of the Thesis

It is unknown that the performance of the CNN-based structural state identification method when applying the method on actual structures with more uncertainties. The robustness of the method to locations of structural changes and temperature is also not clear.

To fulfill the gaps as mentioned in previous paragraph, in this study, firstly, a series of vibration experiments were performed on a concrete railway bridge, investigated the variations of the modal parameters of the bridge, and verified the difficulties of SHM by using the modal parameters. Secondly, a 1-D CNN was proposed and validated the structural state identification performance of the proposed CNN-based method on three real-world structures: (1) a 2.3m-long in-lab T-shaped steel beam; (2) a 6.7m-long steel girder bridge in test field; (3) a 27.3m-long in-service steel girder bridge. The performance measure is the accuracy of structural state identification. Thirdly, for a mechanical interpretation of the classification rules of the proposed

CNN, the author visualizes and discuss the convolutional kernels, outputs of the convolutional layer, and the maximum pooling layer. The effectiveness of the CNN is also proved by visualizing the variation of data structure in each layer of the CNN by the T-SNE method. Fourthly, to examine the capacity of identifying untrained local structural changes of the proposed CNN, two rounds of robustness tests of the CNN model were carried out. Finally, to improve the expression capacity of location and robustness of CNN for structural state identification, a regression CNN model, and a deep network with multi-convolution blocks and multi-task outputs were proposed, and was validated on the databases of the T-shaped steel beam.

There are two further purposes of the study. For one thing, the proposed study aims to boost the development of new Deep-Learning-based SHM and SDD schemes. For another, the proposed study intends to promote the development of technology that guarantees human's daily safety.

Chapter 2

Investigations on the Variations of Modal Parameters

2.1 Introduction

For SHM, it has been widely accepted that the vibration-based method provides the possibility of detecting and locating structural damage (Boller et al., 2009). The shifts of dynamic parameters such as natural frequencies, damping ratios, mode shapes, and other relevant numerical parameters may indicate damage information. However, dynamic parameters are easily affected by environmental factors such as temperature and frost.

In this chapter, a series of vibration experiments are performed on a multi-span pre-stressed concrete simply-supported beam ballasted railway bridge. The purposes of the experiments are as follows:

- (1) To investigate the variations of the modal parameters;
- (2) To verify the difficulties of using the modal parameters for tiny structural change identification.

2.2 Bridge Description

The investigated object is a 5-span ballasted pre-stressed concrete simply supported beam bridge located in Kitami City, Hokkaido, Japan. The construction of the bridge was completed in 1976. The bridge connected Kitami City to Ikeda County. A lateral view of the 3 spans of the bridge is shown in Fig. 2.1 and a general drawing is shown in Fig. 2.2.



Fig. 2.1 Three spans of the bridge

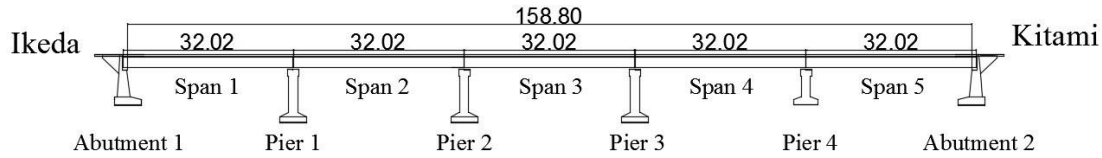


Fig. 2.2 General drawing of the bridge (Unit: m)

The section views of the bridge are shown in Fig. 2.3. On the two sides of each span, the structure was designed as Fig. 2.3 (a) with wide girders, and in the middle of each span the structure was designed as Fig 2.3 (b) with narrow girders. The sections of the girders change at the left and right quarter points of each girder. There are three cross beams on all the quarter points of each span. Walkways are extended of the bridge on the left and right sides. Handrails are also constructed on the edges of the walkways.

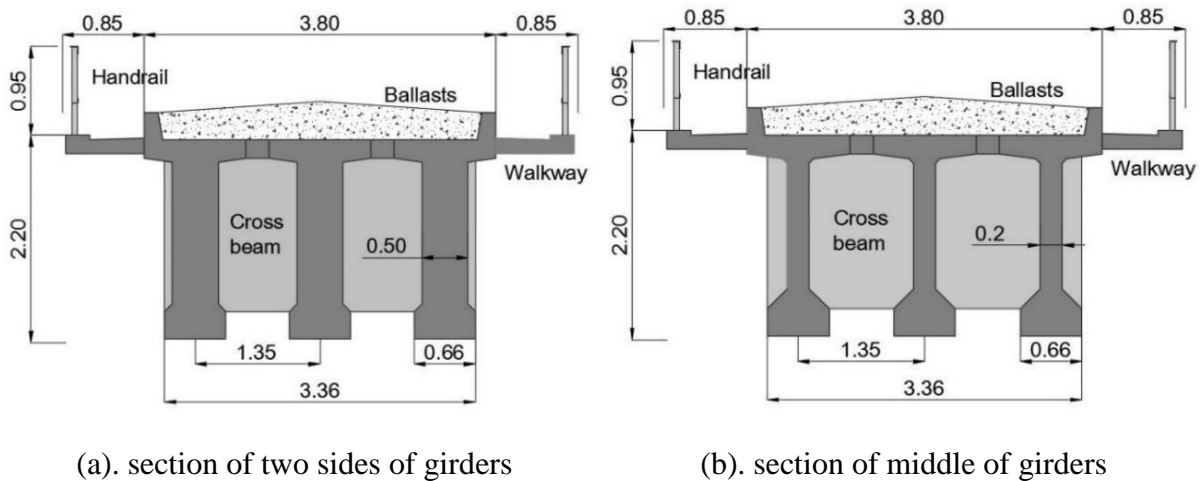


Fig. 2.3 Section views of the bridge (Unit: m)

For the substructure, as shown in Fig. 2.2, the 3 piers on the Ikeda side are higher than the pier on the Kitami side based on the terrain under the bridge. For the boundary conditions, the superstructures are supported at the bottom of the two ends of each girder by line bearing supports, as shown in Fig. 2.4. The length of deck in longitudinal direction is 32.02 m and the span length is 31.3 m.



Fig. 2.4 Line bearing support

Ballasts were distributed on the slab in all the 5 spans, as shown in Fig. 2.5. As the bridge is already out of service, the sleepers and tracks had been removed for several years. In the final stage, before the bridge was totally destroyed, the ballasts, raised concrete lips, walkways, and handrails were removed, as shown in Fig. 2.6. In order to hoist the superstructure of the bridge during destruction, a lot of holes were drilled on the slab, as shown in Fig. 2.7.



Fig. 2.5 Distributed ballasts on the deck



Fig. 2.6 Bridge under destruction



Fig. 2.7 Holes on the slab

2.3 Outline of the Experiments

In order to investigate the variations of the dynamic parameters of the bridge, a series of vibration experiments were carried out during 2 years from November 2015 to December 2017. The time table of the experiments is shown in Table 2.1. In each experiment, the vibration tests were conducted span by span. Note that all the tests were concentrated on only one span. In Table 2.1, the first eight experiments (No. 1 - No. 8) were performed with no sleeper and track

on the bridge, as shown in Fig. 2.5. The last experiment (No. 9) was performed during destruction, with no sleeper, track, ballast, raised concrete lip, walkway, and handrail on the bridge. Some holes were also drilled on the slab for hoisting as shown in Figs. 2.6-2.7.

Table 2.1 Experimental schedule

| <i>Experimental</i> | <i>Date</i> | <i>Experimental</i> | <i>Date</i> |
|---------------------|-------------|---------------------|-------------|
| No. 1 | 2015-11-16 | No. 6 | 2017-2-3 |
| No. 2 | 2016-2-11 | No. 7 | 2017-4-13 |
| No. 3 | 2016-8-18 | No. 8 | 2017-7-25 |
| No. 4 | 2016-10-18 | No. 9 | 2017-12-28 |
| No. 5 | 2016-11-29 | | |

Free damped vibrations of the bridge were measured by the Imote2 wireless acceleration sensor system. The original sensor consists of a SHM-H sensor board (ISHMP, 2009), an Imote2 communication board (Crossbow, 2007) a power supply board, and an antenna, as shown in for Fig. 2.8. In order to add the waterproof function to the sensor, the sensor was updated by installing all the main components in a plastic box, as shown in Fig. 2.9. An additional battery box was also applied to enhance the battery life of the sensor.



Fig. 2.8 Original wireless sensor



Fig. 2.9 Updated wireless sensor

In No. 1, 3, 4 experiments, 10 sensors were installed on or beside the raised concrete lips symmetrically in two rows (5 sensors \times 2 rows) in each span, as shown in Fig. 2.10(a). For efficiency considerations, in the No. 2, and No. 5-9 experiments the number of sensor was reduced to 6 on each span, as shown in Fig. 2.10(b). The excitation method was human jumping and landing owing to its convenience of applying. Symmetric and asymmetric excitations

were conducted on the middle points and the quarter points of the walkways. For each test, the measurement is 30 second with 280 Hz sampling frequency. Free damped vibration of the bridge was recorded and processed.

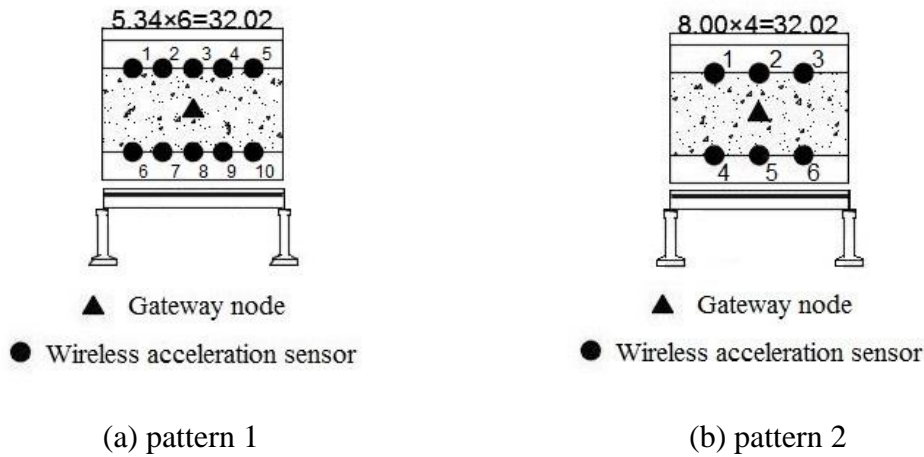
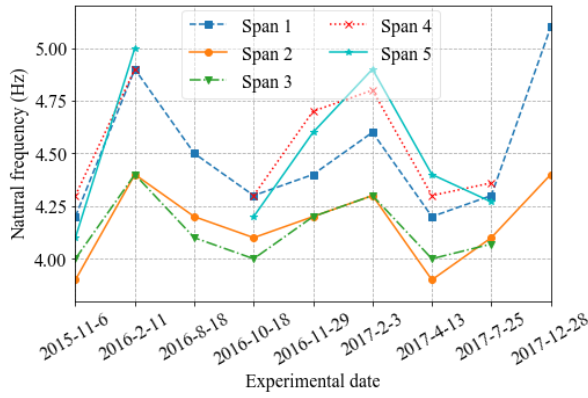


Fig. 2.10 Sensor distribution maps

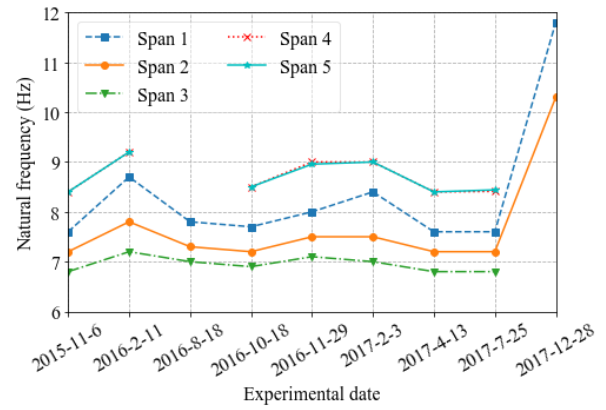
The acceleration data were collected by wireless communication to the computer. As only impulse excitations were applied, the dynamic behavior of the bridge was simple free damped vibration. Thus, the natural frequencies were identified by picking the peaks in the power spectrum. The peak picking method is based on the fact that the frequency response functions reach an extreme value approximately at the natural frequency. The damping ratios were calculated by the half-power bandwidth method. Finally, the mode shapes were estimated by the cross-spectrum method (Liou and Jeng, 1989).

2.4 Results

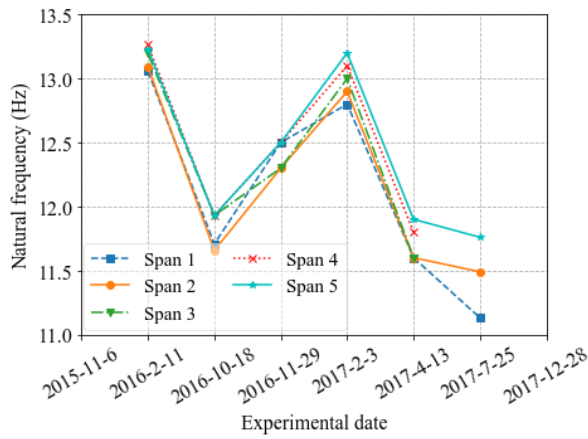
The identified natural frequencies and damping ratios of every span in all the experiments are shown in Figs. 2.11 and 2.12. And the mode shapes of Span 1 in the No. 2 experiment are shown as an example in Fig. 2.13. As only vertical impulse excitations were applied, in total six vertical modes were identified. From lower to higher order, the six modes are the 1st bending mode, the 1st torsional mode, the 2nd torsional mode, the 2nd bending mode, the 3rd torsional mode, and the 3rd bending mode. No horizontal vibration mode was identified.



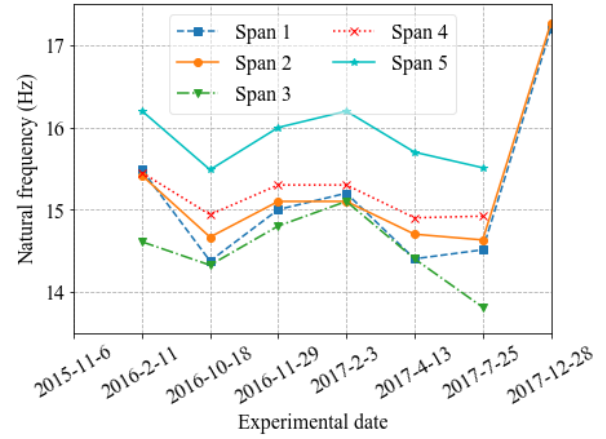
(a) 1st bending mode



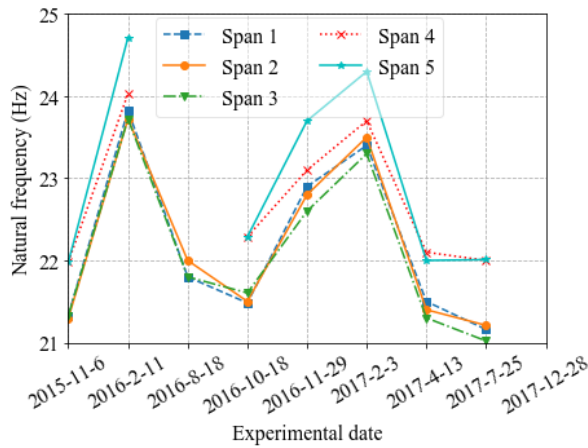
(b) 1st torsional mode



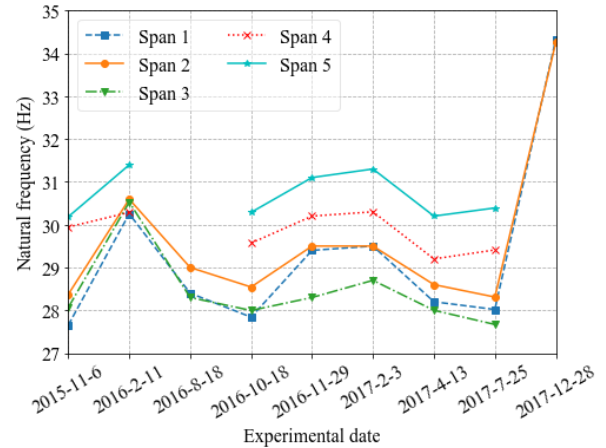
(c) 2nd torsional mode



(d) 2nd bending mode

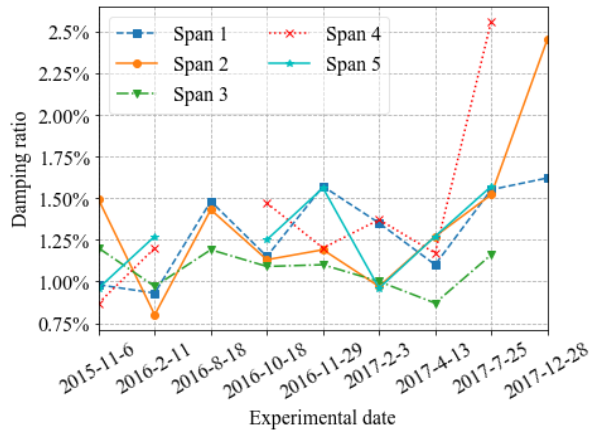


(e) 3rd torsional mode

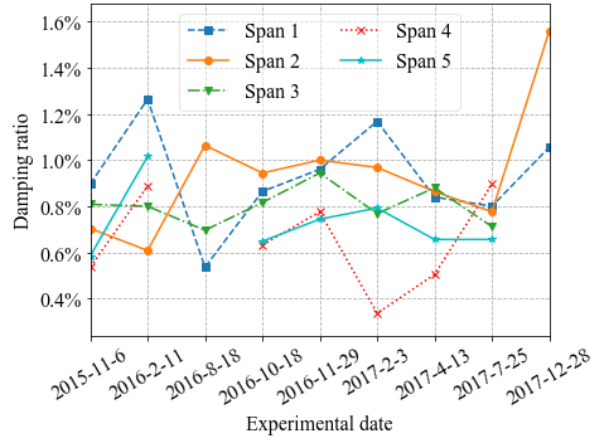


(f) 3rd bending mode

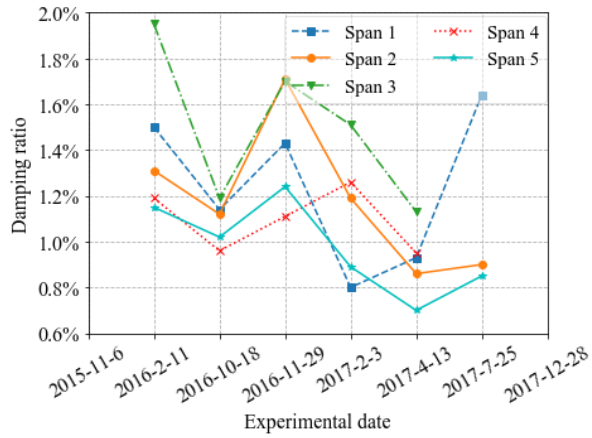
Fig. 2.11 Natural Frequencies



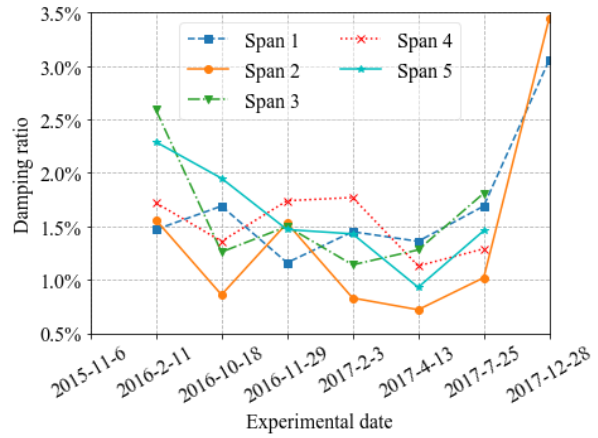
(a) 1st bending mode



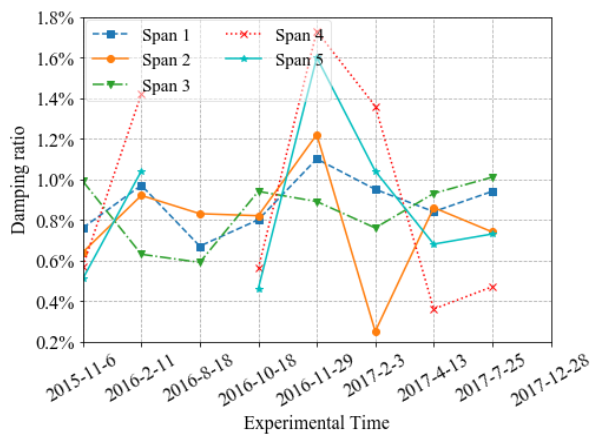
(b) 1st torsional mode



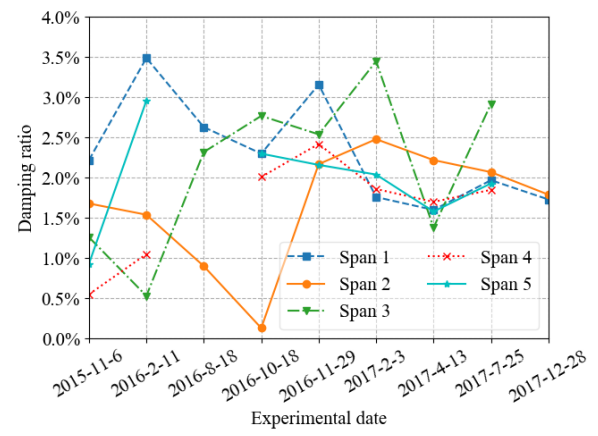
(c) 2nd torsional mode



(d) 2nd bending mode



(e) 3rd torsional mode



(f) 3rd bending mode

Fig. 2.12 Damping ratios

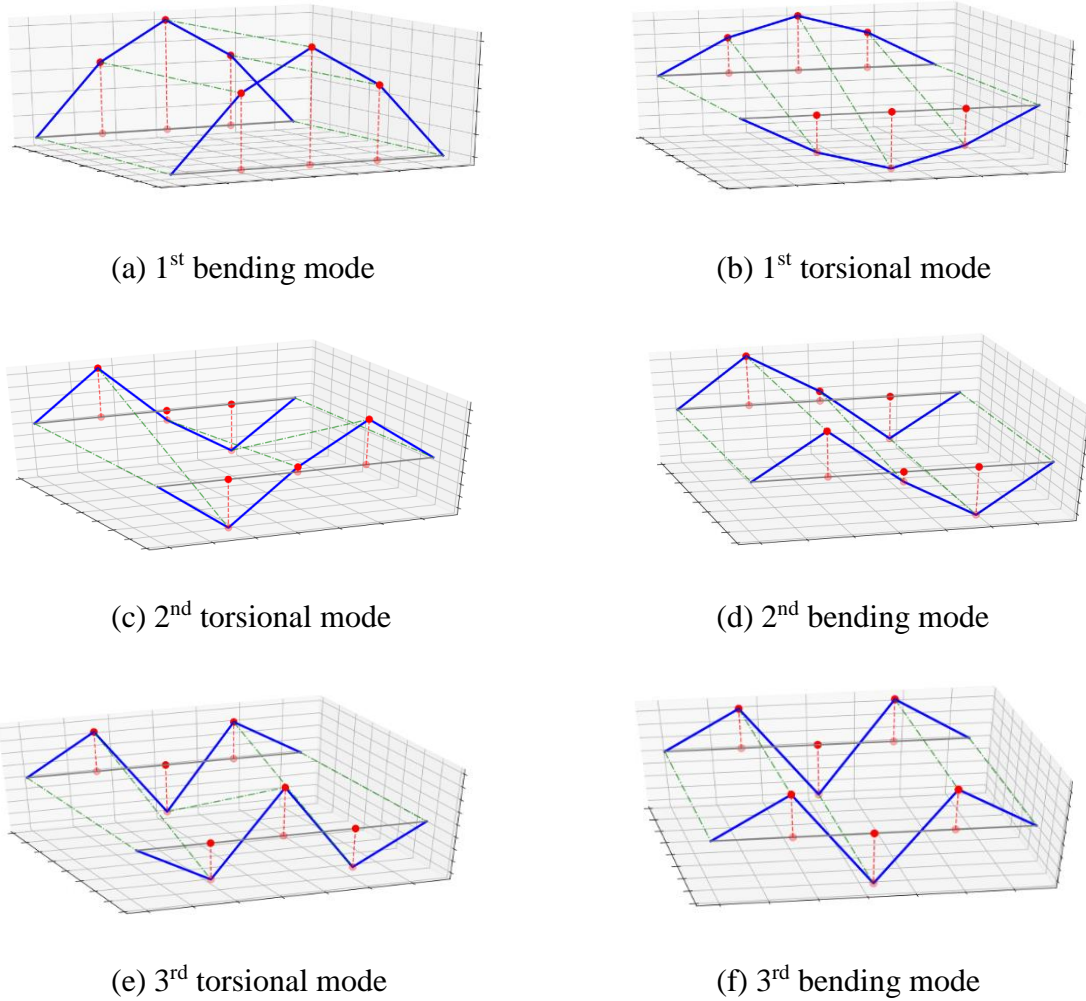


Fig. 2.13 An example of mode shapes

In Fig. 2.11, the natural frequencies in each mode from the No.1 to No.8 experiments show clear seasonal variation tendency that there are 2 local maximum peaks of the natural frequencies in the Februaries of 2016 and 2017. In order to find the reasonable explanation of this phenomenon, a series of investigations were carried out. As Hokkaido is located in the northern hemisphere, the season in February is winter. Normally, the relative high modulus of elasticity of concrete in winter caused by low temperature was considered as the main reason of this phenomenon. However, the temperature gap between the No. 2 (2016-2-11) and No. 5 (2016-11-29) experiments was only 2 °C, as shown in Fig. 2.14. The 2 °C temperature difference could not cause so obvious natural frequency difference. Thus this assumption was excluded. By empirically estimating, the frozen supports could be another possible reason to cause this phenomenon (Alampalli, 1998). However, there was no ice or snow on the supports by visual inspection. The boundary condition of the bridge did not change and it is also not an

acceptable explanation of the phenomenon. In this case, great attention was paid to the frozen ballasts. In this series of experiments, three typical ballast states were obtained, dry, frozen, and wet, as shown in Fig. 2.15.

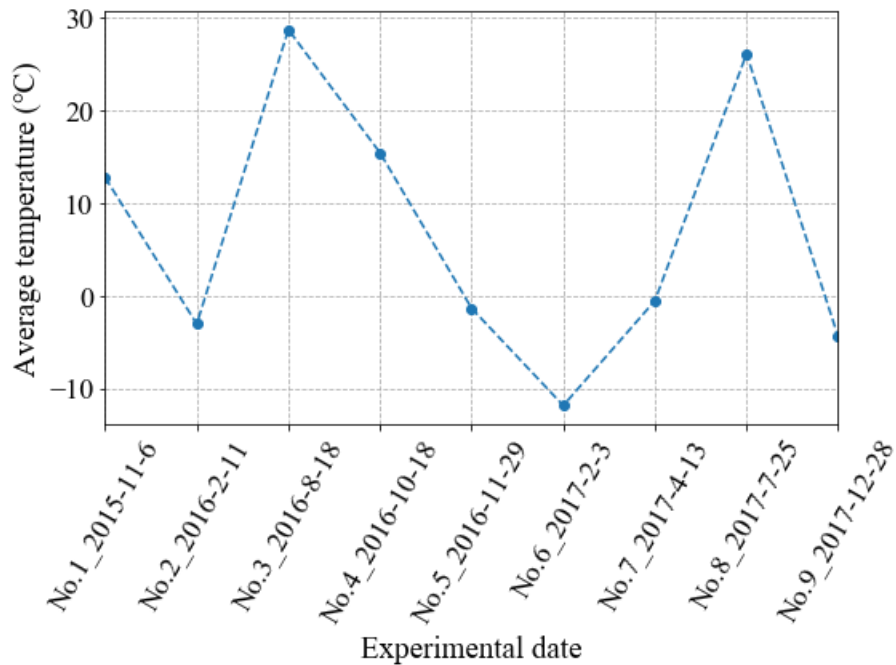


Fig. 2.14 Average temperature of the experimental days

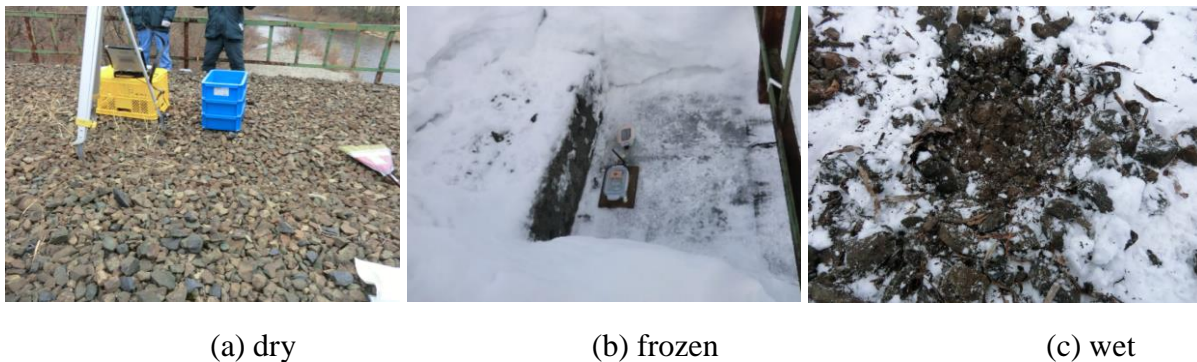


Fig. 2.15 Three ballast states

Firstly, the dry ballast state shown in Fig. 2.15 (a) was presented in the No. 1 experiment. There was no rainfall during the week before the experiment, so the moisture content of the ballasts could be very low. In the No. 1 experiment, the dry ballasts were discrete with low adhesive effect. Secondly, the frozen state of the ballasts shown in Fig. 2.15 (b) appeared in

the No. 2 experiment. The ballasts were totally frozen. Ice filled the gaps between the ballasts and transferred the ballasts into a new composite material. As the special climatic conditions in the winter of Hokkaido, the ballasts had gone through a special temperature variation process. When the daily temperature fluctuated around 0°C , snow on the ballasts would melt into water and seep into the space within the ballasts. Then the temperature dropped to 0°C , the seeped water would freeze. After several loops of that the daily temperature fluctuated around 0°C , the ice which filled in the space within the ballasts would be accumulated. When the ice reached a relatively high amount, the material properties of the frozen ballasts would change obviously. It could be seen as a new composite material. In this hypothesis, ice acts as filler, and ballasts perform as aggregate. Thirdly, the wet ballast state shown in Fig. 2.15 (c) was surveyed in the No. 5 experiment. The ballasts were a little wet, but not frozen. The increase of moisture content of the ballasts would affect the material properties by reducing the friction and increasing the stickiness. However, the property change was too mild comparing to frost.

The natural frequencies of the final experiment (No. 9) was obtained in totally different bridge condition with no ballast, raised concrete lip, walkway, and handrail. Meanwhile, a lot of holes are drilled on the deck. In a word, all the subsidiary components of the structure were removed, and even some structural damages were made on the slab. In these circumstances, the excitation positions of the No. 9 experiment were slightly adjusted to the inside areas for safety reasons. Thus, only 4 modes were identified: the 1st bending mode, the 1st torsional mode, the 2nd bending mode, and the 3rd bending mode, as shown in Fig. 2.11 (a), (b), (d), and (f). The 2nd and 3rd torsional modes were not sufficiently excited. The results show that the natural frequencies in the No. 9 experiment are much higher than the natural frequencies in other experiments. The reason should be that after removing the subsidiary components of the bridge, especially the ballasts, the mass of the bridge was greatly reduced. Lower mass of the structure leads to higher natural frequencies. For the holes on the deck, theoretically, structural damages may decrease the natural frequencies since the stiffness of the bridge is reduced. However, the effect of holes to the natural frequencies is much lower than the effect of removing the subsidiary components of the bridge to the natural frequencies. As a result, the natural frequencies of the bridge during destruction increased obviously even though the slab was drilled.

For damping ratios as shown in Fig 2.12, through a general view, no clear regularity could be found within these data. The damping ratios did not show clear trend as the natural frequencies, and just varied unproportionally with the seasons altering. The complexity and

uncertainty of the ballasts could be possible reasons for the results. The differences between the shapes of ballasts lead to different spaces and interaction mechanisms between the ballasts which would directly affect the moisture content, friction, and stickiness of the ballasts. Furthermore, with the onset of ice in winter, the mechanisms of the ballasts would become more complex. In a word, all the mentioned factors can differ the amount of the absorbed and damped vibration energy, leading to different damping ratios.

For the last experiment (No. 9), the damping ratios of the 3 lower identified modes (the 1st bending mode, the 1st torsional mode, and the 2nd bending mode) increased obviously, which is far beyond the authors' expectation. Before the No. 9 experiment, the damping ratios were expected to decrease, since the ballasts were removed. The ballasts can absorb energy of vibration. Without the ballasts on the slab, the capacity of absorbing energy of the structure is decreased, thus lower damping ratios were expected. One possible reason of the increase of damping ratios is that by removing the subsidiary components of the bridge, the vertical forces on the supports were decreased, leading to lower frictions on the surfaces of the sliding bearings. If there were some motions on the supports, the energy of vibration was consumed in the movement of sliding bearings. However, this assumption needs more study to prove.

For the mode shapes of the bridge, as the sensor distributions in the experiments were not uniformed, the mode shape variation through all the experiment cannot be obtained. However, one phenomenon is found that in any experiment, the mode shapes of any certain span fluctuated in a wide range. Two examples are shown in Figs. 2.16-2.17, corresponding to the amplitudes of the 1st bending mode shapes of Span 2 in the No. 2 and No. 7 experiments. There are 7 tests in the No. 2 experiment and 4 tests in No. 7 experiment. The amplitudes in all channels vary obviously. Here the ratio of variation of the amplitude R is calculated as Equation 2.1. A_i is the amplitudes of Channel i . The variation ratios of amplitudes of the mode shape are summarized in Tables 2.2 - 2.3, corresponding to the results in Figs. 16-17. Tables 2.2 - 2.3 shows that even in the same experiment, the amplitudes of the model shape in different tests vary larger than 20%.

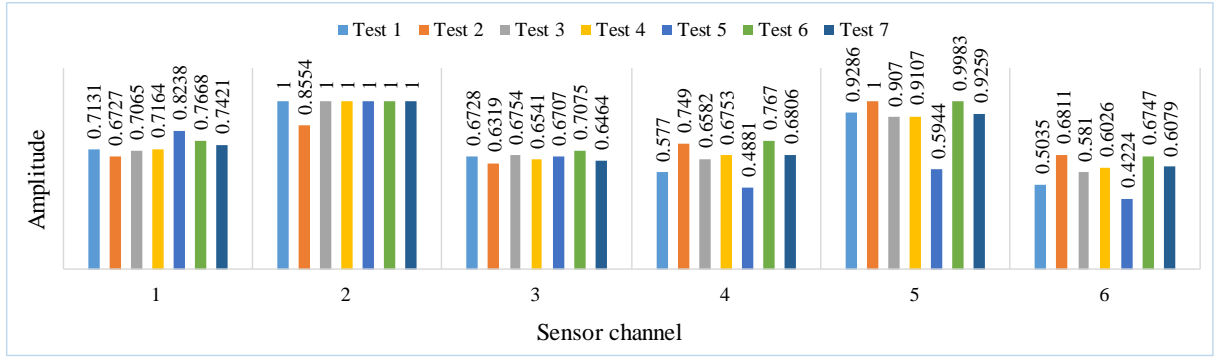


Fig. 2.16 Amplitudes of the 1st bending mode of Span 2 in 7 tests in No. 2 Experiment (2016-2-11)

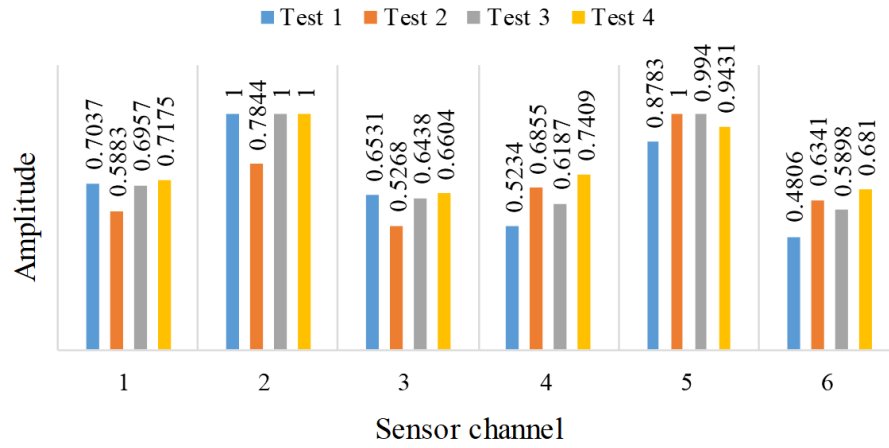


Fig. 2.17 Amplitudes of the 1st bending mode of Span 2 in 4 tests in No. 7 Experiment (2017-4-13)

$$R = [\max(A_i) - \min(A_i)] / \text{mean}(A_i) \quad (2.1)$$

Table 2.2 Variation ratios the amplitudes of the model shape of Span 2 in the No. 2 Experiment

| Channel | Ratio of variation (%) |
|---------|------------------------|
| 1 | 20.57 |
| 2 | 14.76 |
| 3 | 11.35 |
| 4 | 42.49 |
| 5 | 45.32 |
| 6 | 44.46 |
| Average | 29.83 |

Table 2.3 Variation ratios the amplitudes of the model shape of Span 2 in the No. 7

Experiment

| <i>Channel</i> | <i>Ratio of variation (%)</i> |
|----------------|-------------------------------|
| <i>1</i> | 19.10 |
| <i>2</i> | 22.79 |
| <i>3</i> | 21.51 |
| <i>4</i> | 33.87 |
| <i>5</i> | 12.76 |
| <i>6</i> | 33.6 |
| <i>Average</i> | 23.94 |

2.5 Conclusions

Overview the results, all the dynamic parameters (natural frequencies, damping ratios, and mode shapes) are not stable indicators to present the state of structure. The dynamic parameters are easily biased by the environmental changes, measurement errors, or other uncertainties. Since the conventional SDD methods are based on the dynamic parameters, it is difficult to use the dynamic parameters as indicators of small local structural changes. The effect of environmental changes and other uncertainties cannot be overcome easily. Meanwhile, only arbitrary signal processing methods have been used to analyze the vibration data. The vibration data cannot be used effectively. Therefore, a modern approach of designing an indicator for structural state identification was carried out by using Deep Learning in later chapters.

Chapter 3

Basic Concept of Deep Learning and Related Algorithms

3.1 Introduction of Artificial Neural Network

Artificial neural network is a main machine learning method, which is also the quintessential Deep Learning models. The goal of a neural network is to approximate some function f^* , which maps an input x to the expected result y . A neural network defines a mapping $y = f(x; \theta)$ and learns the value of the parameters θ that result in the best function approximation (Goodfellow et al., 2016).

Artificial neural networks are called networks because they are typically represented by composing many different functions together. The model is associated with a directed graph describing how the functions are composed together. For example, there might have three functions $f^{(1)}$, $f^{(2)}$, and $f^{(3)}$ connected in a chain, to form $f(x) = f^{(3)}(f^{(2)}(f^{(1)}(x)))$. These chain structures are the most commonly used structures of neural networks. In this case, $f^{(1)}$ and $f^{(2)}$ are called the first and second layers. The overall length of the chain gives the depth of the model (Goodfellow et al., 2016). In recent years, the chains of functions become longer and longer, such as the VGG network (Simonyan and Zisserman, 2014) with 19 weighted layers, GoogLeNet (Szegedy, Liu, et al., 2014) with 22 weighted layers, and ResNet with even 152 weight layers (He, Zhang, et al., 2016), and so on. If the function chain is plotted as a vertical operation graph, the depth of the graph becomes deeper and deeper in recent years. This is the origin of the name “Deep Learning”.

During neural network training, $f(x)$ is driven to match $f^*(x)$. The training data provides noisy approximate examples of $f^*(x)$ which evaluated at different training data. Each example x is accompanied by a label $y \approx f^*(x)$. The training examples specify directly what the output layer must do at each point x , and it must produce a value that is close to y (Goodfellow et al., 2016).

Generally, the training loop consists of (1) gradient calculation by using back-propagation algorithm (Rumelhart et al., 1986a) and some loss functions, and (2) parameter update by descending the gradients through some optimizers.

3.2 Layers of the Network

3.2.1 1-D convolutional layer

The algorithm of a one-dimensional (1-D) convolutional layer iterates two operations throughout the input array (see Fig. 3.1). To clarify the algorithm for readers, the input array is element-by-element multiplied by the kernel and the products are then summed as shown in Equation 3.1. Second, the summed value plus a bias value is fed into an activation function to obtain the output value. This process moves sequentially along the temporal axis of the input array, as shown in Sub-array 1 and Sub-array 2 of Fig. 3.1.

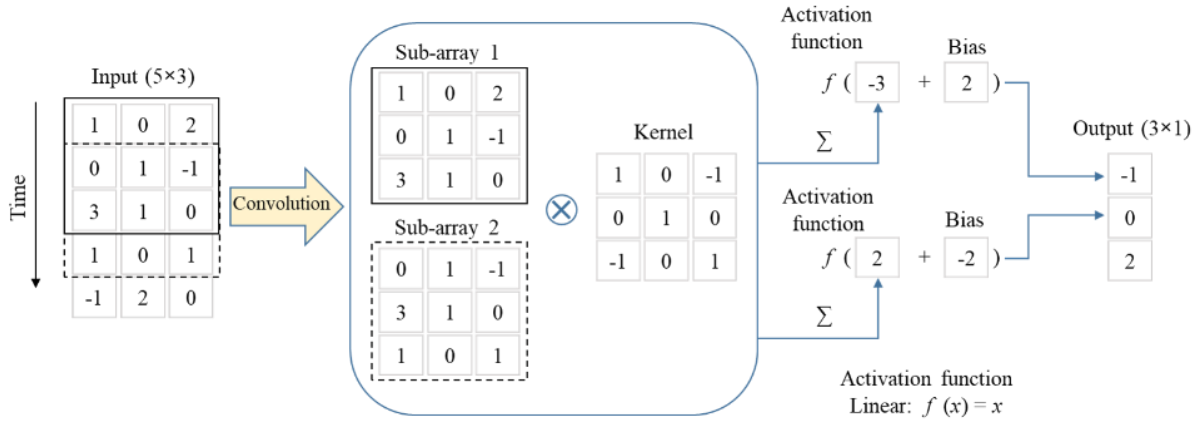


Fig. 3.1 Demonstration of a 1-D convolutional layer

$$f(i) = \sum_{n=1}^{v_k} S(i+n)K(n) \quad (3.1)$$

The kernels are the same width as the input array. To clarify the presentation of the convolutional layer, Fig. 3.1 shows only integer values, but the parameters in the convolutional layer of actual CNN models are real values. The length, number, and sliding size of the kernels are manually assigned. The kernel weights are trainable parameters and are continuously updated by the optimizer during the training process. The convolutional layer has three extraordinary properties that improve a machine learning system: sparse weights, parameter sharing, and equivariant representations (Goodfellow et al., 2016). By virtue of these properties, the convolution highly reduces the memory requirements of the model and improves the statistical efficiency of automatically extracting and learning the local features in the data.

3.2.2 Batch normalization

The training data are learned batch by batch. Therefore, the batch distributions are non-uniformly and unstably distributed and must be fitted by the network parameters in every training iteration, which greatly slows the convergence of the model. To circumvent this problem, the convolutional layer is followed by an adaptive reparametrization method called batch normalization (Ioffe and Szegedy, 2015). The batch normalization algorithm calculates the mean μ_D and variance σ_D^2 of every batch of the training data, then shifts and scales the original data to zero-mean and unity-variance. Finally, the shifted data \hat{x}_i are assigned a weight γ and bias β to increase their expressive power. The calculations are given by Equations 3.2-3.5. The reparametrization in the batch normalization algorithm significantly alleviates the problem of coordinating updates through the layers in the neural network.

$$\mu_D = \frac{1}{m} \sum_{i=1}^m x_i \quad (3.2)$$

$$\sigma_D^2 = \frac{1}{m} \sum_{i=1}^m (x_i - \mu_D)^2 \quad (3.3)$$

$$\hat{x}_i = \frac{x_i - \mu_D}{\sqrt{\mu_D^2 + \epsilon}} \quad (3.4)$$

$$y_i = \gamma \hat{x}_i + \beta \quad (3.5)$$

3.2.3 Max pooling layer

The batch normalization layer in the proposed CNN is followed by a 1-D max pooling layer (Zhou and Chellappa, 1988). The 1-D pooling operation extracts the maximum output within a certain neighborhood along the time-series direction. The neighborhood is defined by a sliding window that strides along the time dimension. An example of 1-D max pooling is shown in Fig. 3.2. After pooling, the representation is approximately invariant to small translations of the input data, meaning that the features in the data are basically unchanged although the data size is reduced. Note that the values in real applications should be real values within the same range of the input, but simple integers are shown for illustrative purposes. The 1-D max pooling

is a dynamic downsampling technique that greatly improves the statistical efficiency and computational speed of the neural network.

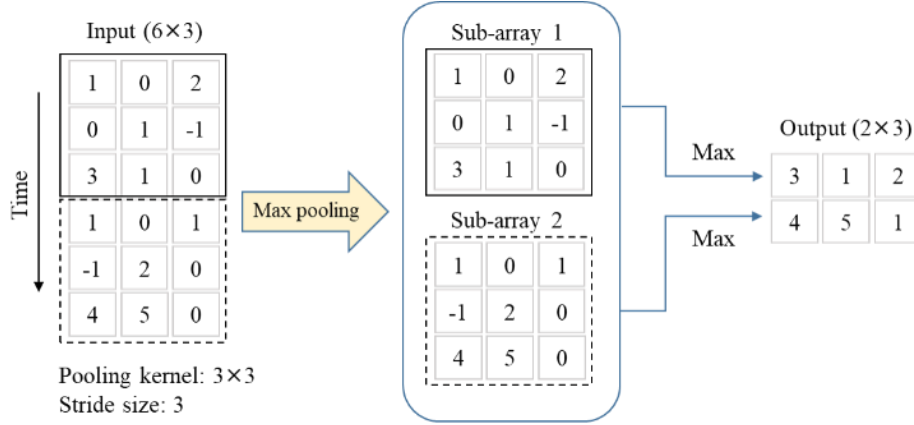


Fig. 3.2 Demonstration of a 1-D max pooling

3.2.4 Dropout

Dropout (Srivastava et al., 2014) is an effective solution to the over-fitting problem. The dropout operation is applied after the flatten layer. Briefly, the dropout operation inactivates some units during training and reactivates them during validation, boosting the performance of the CNN at very low computational cost.

3.2.5 Fully connected layer

The fully connected (FC) layer constitutes part of the classifier in the last layers. The units in the FC layer are linked to all units in the forward layer. As shown in Equation 3.6, the input values u are multiplied by the weights w and the products are summed. A bias b is added to the summed value. Finally, this result is input to an activation function f , which computes the output. The weights w and biases b are trainable variables, and the activation function f is manually assigned. For instance, in this study, the activation function of the FC layer was the rectified linear unit (ReLU) function (Nair and Hinton, 2010), determined by Equation 3.7.

$$y = f(\sum u \times w + b) \quad (3.6)$$

$$y = \max(0, u) \quad (3.7)$$

3.2.6 Softmax output layer

The last layer in the CNN is the softmax output layer, which outputs the structural-state identification result. The calculation is given by Equation 3.8. The probabilities of all predictive candidates are evaluated, and the candidate with the highest possibility is output as the final result.

$$y_i = \frac{\exp(u_i)}{\sum_{i=1}^n \exp(u_i)} \quad (3.8)$$

3.3 Mechanisms of Learning

The process of learning is to solve an optimization problem that minimum the loss L of the network, as shown in Equation 3.9. Here the loss L means the deviation between the label and the network output, which is measured by a loss function. θ means the parameters in the network, such as weights and biases.

$$\theta^* = \arg \min L(\theta) \quad (3.9)$$

The process of learning is implementing gradient descent continuously. Suppose that θ has two variables $[\theta_1, \theta_2]$, and randomly start at $\theta^0 = \begin{bmatrix} \theta_1^0 \\ \theta_2^0 \end{bmatrix}$. The parameters are updated as Equation 3.10.

$$\theta^1 = \begin{bmatrix} \theta_1^1 \\ \theta_2^1 \end{bmatrix} = \begin{bmatrix} \theta_1^0 \\ \theta_2^0 \end{bmatrix} - \eta \begin{bmatrix} \partial L(\theta_1^0) / \partial \theta_1 \\ \partial L(\theta_2^0) / \partial \theta_2 \end{bmatrix} \quad (3.10)$$

Here η is the learning rate, which indicates the ratio of parameter updating. The gradient of θ , $\nabla L(\theta)$, is $\begin{bmatrix} \partial L(\theta_1^0)/\partial \theta_1 \\ \partial L(\theta_2^0)/\partial \theta_2 \end{bmatrix}$. For short, the process of learning can also be summarized as Equation 3.11.

$$\theta^1 = \theta^0 - \eta \nabla L(\theta^0) \quad (3.11)$$

Therefore, the learning process are continuously implemented as Equation 3.12, until the loss L approximately reach the minimum.

$$\theta^{n+1} = \theta^n - \eta \nabla L(\theta^n) \quad (3.12)$$

3.4 Loss Functions

Loss function is used to calculate the deviation between label and network prediction. In this study, two loss functions are applied: categorical cross entropy, and mean squared error.

3.4.1 Categorical cross entropy

The algorithm of categorical cross entropy L is shown in Equation 3.13. Here n is the batch size, y means the label, and a is the network output.

$$L = -\frac{1}{n} \sum_x [y \ln a + (1 - y) \ln(1 - a)] \quad (3.13)$$

3.4.2 Mean squared error

The algorithm of mean squared error L is shown in Equation 3.14. Here n is the batch size, y is the label, a is the network output.

$$L = \frac{1}{n} \sum_{i=1}^n (a_i - y_i)^2 \quad (3.14)$$

3.5 Optimizers

Optimizer is an algorithm which determine the strategy of parameter updating. In this study, two optimizers are utilized: stochastic gradient descent (SGD), and Adam (Kingma and Ba, 2014).

3.5.1 Stochastic gradient descent

SGD is an extension of the gradient descent algorithm (Cauchy, 1847). SGD is the most used optimizer for machine learning or deep learning. The algorithm of SGD estimating the gradient downhill is shown as Equation 3.15. Here x is the input data, θ means the parameters in the network, L is the loss function, η is the learning rate, f is the network, and y is the label.

$$\theta \leftarrow \theta - \eta \times \frac{1}{m} \nabla_{\theta} \sum_i L(f(x_i; \theta), y_i) \quad (3.15)$$

3.5.2 Adam

Adam is an adaptive learning rate optimization algorithm. The algorithm of Adam is as follows. Firstly, learning rate η , decay rates ρ_1 and ρ_2 , and a small constant δ used for numerical stabilization are required to be assigned manually. Secondly, the parameters of the network θ are required. Thirdly, initialize the 1st and 2nd moment variables $s = 0$, $r = 0$, and time step $t = 0$. Fourthly, while stopping criterion not met, calculate as following steps:

- Sample a batch of m data from the training set $[x_1, x_2, \dots, x_m]$ with corresponding labels y_i ;
- Compute gradient: $g \leftarrow \frac{1}{m} \nabla_{\theta} \sum_i L(f(x_i; \theta), y_i)$
- Update biased 1st moment estimate: $s \leftarrow \rho_1 s + (1 - \rho_1)g$
- Update biased 2nd moment estimate: $r \leftarrow \rho_2 r + (1 - \rho_2)g \odot g$
- Correct bias in 1st moment: $\hat{s} \leftarrow \frac{s}{1 - \rho_1^t}$
- Correct bias in 2nd moment: $\hat{r} \leftarrow \frac{r}{1 - \rho_2^t}$

- Update the parameters: $\theta \leftarrow \theta - \eta \frac{\hat{s}}{\sqrt{\hat{P}} + \delta}$

3.6 Deep Learning for Structural State Identification

A neural network learns to identify the structure state by iterating two alternating steps: (1) train the neural network by using vibration data and corresponding label of structural state, and (2) validate the neural network. Different sets of data are required for training and validation accordingly.

In the training step, the network is updated by evaluating and reducing the deviations between the predicted (network output) and actual (labeled) locations of the local structural changes, on a batch-by-batch basis. The uniformity and deviations between the network outputs and labels are called the accuracy and loss, respectively.

The neural network is trained by iteratively feeding the training data, evaluating the loss, and updating the network parameters. The training is complete when the network outputs achieve a high accuracy with relatively low loss and good performance of identifying structural changes.

3.7 Introduction of Support Vector Machine

Support vector machines (SVM) play a dominant role when dealing with the classification tasks before the Deep Learning revolution. SVM was proposed by Cortes and Vapnik in 1995. SVM is a supervised learning algorithm which determines a model to predict the categories of new samples. The concept of SVM is to define a hyperplane ($y = w \cdot x + b$) which can maximize the margin between different categories of data when observing the database in high dimensional space. The determined hyperplane is used for a certain classification task. One example is shown in Fig. 3.3 to demonstrate the basic idea of SVM. The two categories of data can be classified by the hyperplane which maximizes the margin between boundaries of the two categories of data.

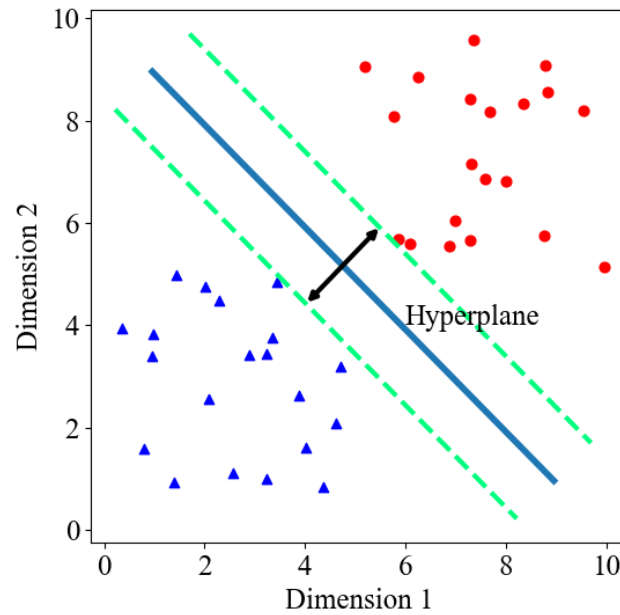
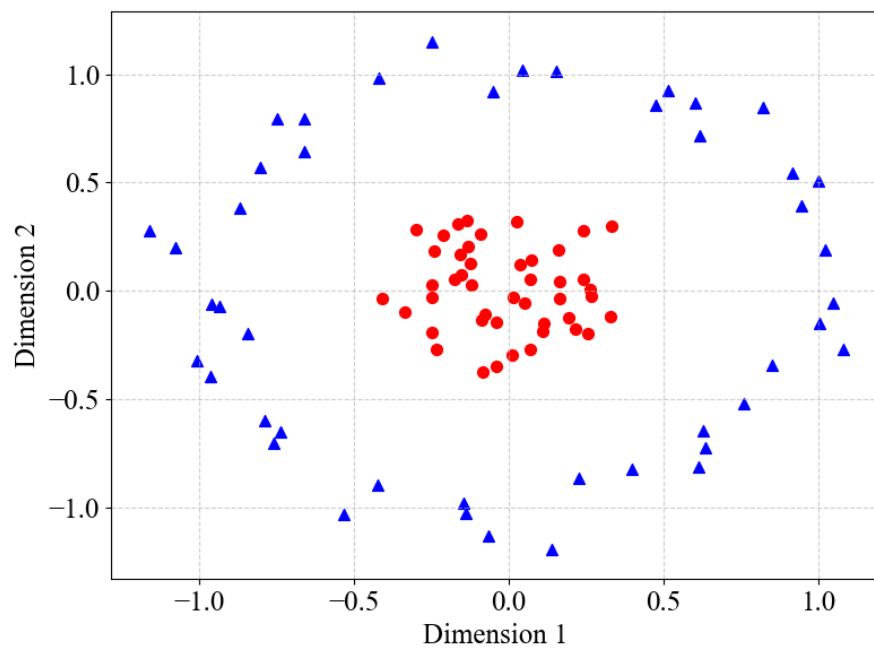
**Fig. 3.3** Concept of SVM

Fig. 3.3 shows an ideal data structure which can be classified easily. However, in actual cases, data structures are usually much more complex, such as the case shown in Fig. 3.4. Even though there is clear gap between the two categories of the data, there is no suitable hyperplane to describe the boundary.

**Fig. 3.4** Complex data structure

To solve above problem, a kernel function is used to increase the dimensions of the data. By observing the data structure in a higher dimensional space, a hyperplane can be determined easily to classify data in different categories which has failed in the original dimensional space. One example is shown in Fig. 3.5, which is the classification result of the data as shown in Fig. 3.4. The raw data are 2-D data. By using the radial basis function (RBF) kernel, the 3rd dimensional coordinate is calculated as Equation 3.16 and expanded to the original 2D coordinates. Here x represents the raw data, x_c is the center of data which is set to 0, and σ is a manually assigned parameters to control the radial range of the kernel function.

$$k(x) = \exp\left(-\frac{\|x-x_c\|^2}{2\sigma^2}\right) \quad (3.16)$$

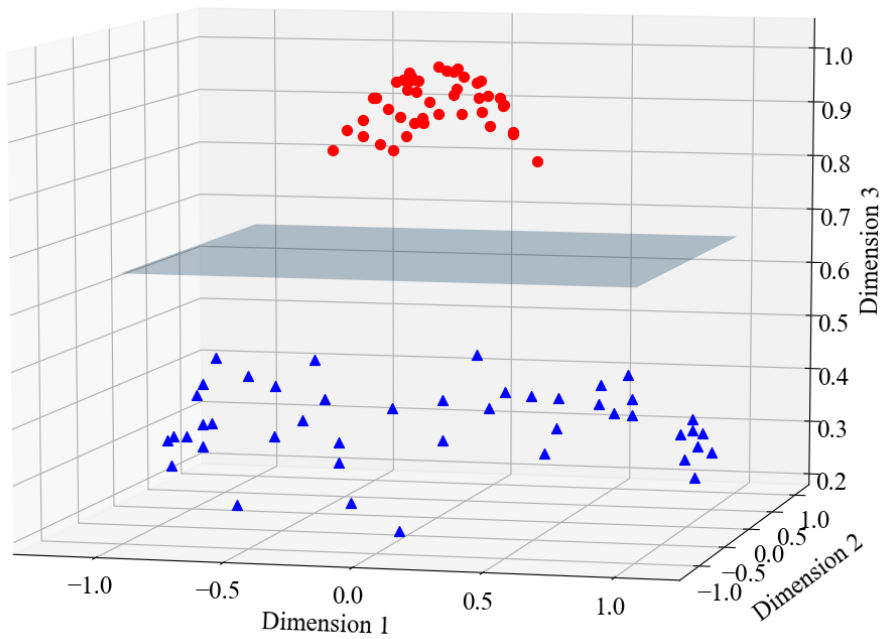


Fig. 3.5 SVM classification result by using RBF kernel

Based on the concept as introduced above, the general optimization objective function of the SVM method can be determined as Equation 3.17. The w and b of the hyperplane are desired which can maximize the margin between different categories of data. Here y_i is label, Φ is kernel function, x_i is data. If we manually define the constrain that $y_i (w \cdot \Phi(x_i) + b) \geq 1$, the optimization objective function can be transformed into Equation 3.18.

$$\operatorname{argmax}_{w,b} \left\{ \frac{1}{\|w\|} \min_i [y_i \cdot (w \cdot \Phi(x_i) + b)] \right\} \quad (3.17)$$

$$\operatorname{argmax}_{w,b} \frac{1}{\|w\|} \quad (3.18)$$

Subsequently, by using Lagrange multiplier, the optimal solution that meets the constrain can be obtained by solving the optimization objective function as shown in Equation 3.19.

$$\max_{\alpha} \min_{w,b} L(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^n \alpha_i (y_i (w \cdot \Phi(x_i) + b) - 1) \quad (3.19)$$

To obtain the minimum values of w and b in Equation 3.19, firstly calculate the partial derivatives of specific parameters as shown in Equation 3.20, thus Equations 3.21 and 3.22 can be obtained.

$$\frac{\partial L}{\partial w} = \frac{\partial L}{\partial b} = 0 \quad (3.20)$$

$$w = \sum_i^n \alpha_i y_i \Phi(x_i) \quad (3.21)$$

$$0 = \sum_{i=1}^n \alpha_i y_i \quad (3.22)$$

Secondly, feed Equations 3.21 and 3.22 into Equations 3.19 to obtain Equation 3.23 which can be transform into Equation 3.24.

$$\max_{\alpha} L(w, b, \alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1, j=1}^n \alpha_i \alpha_j y_i y_j \Phi(x_i) \Phi(x_j) \quad (3.23)$$

$$\max_{\alpha} L(w, b, \alpha) = \min_{\alpha} \left(\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \Phi(x_i) \Phi(x_j) - \sum_{i=1}^n \alpha_i \right) \quad (3.24)$$

With the constrains in Equation and $\alpha_i \geq 0$, by computing $\frac{\partial L}{\partial \alpha_i} = 0$, the optimal α_i can be obtained. Then, feed α , x , and y into Equation 3.21 to calculate w of the hyperplane. Finally, calculating b by feeding y_i , w , and x_i into Equation 3.25 to finish determining of the hyperplane. By using the determined hyperplane, the categories of new data can be predicted.

$$y_i = wx_i + b \quad (3.25)$$

In recent years, SVM are commonly used as references of the accuracy of neural network predictions, since SVM is a mature machine learning method which has achieve very good results in lots of tasks.

3.8 Introduction of T-Distributed Stochastic Neighbor Embedding

T-distributed stochastic neighbor embedding (T-SNE) is a data structure visualization approach proposed by Maaten and Hinton in 2008. The method projects high-dimensional data to a 2-D or 3-D space, and keeps the similar data structure in the low-dimensional space, which means that the data with high correlation keep close in the low-dimensional space.

The method converts the high-dimensional Euclidean distances between data into conditional probabilities. The conditional probability of data in high-dimensional space p_{ij} is defined as Equation 3.26 in the Gaussian distribution, and is set as Equation 3.27 for the reason of convenient computation. The conditional probability q_{ij} in the low-dimensional space is defined as Equation 3.28 in the form of Student t-distribution, and y_i is the coordinate in the low-dimensional space.

$$p_{i|j} = \frac{\exp(-\|x_i - x_j\|^2 / 2\sigma^2)}{\sum_{k \neq l} \exp(-\|x_k - x_l\|^2 / 2\sigma^2)} \quad (3.26)$$

$$p_{ij} = \frac{p_{i|j} + p_{j|i}}{2n} \quad (3.27)$$

$$q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq l} (1 + \|y_k - y_l\|^2)^{-1}} \quad (3.28)$$

The visualization is realized by minimizing the loss between the conditional probabilities of data in the high-dimensional space p_{ij} and the low-dimensional space q_{ij} through the loss function in Equation 3.29. The loss can be reduced by descending the gradient of loss as shown in Equation 3.30. Finally, the process of optimization can be summarized as Equation 3.31. Here Υ is the solution vector of the data structure in the low-dimensional space $\{y_1, y_2, \dots, y_n\}$, t means the number of iteration, η represents learning rate, and α is momentum of parameter updating.

$$C = KL(P \parallel Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}} \quad (3.29)$$

$$\frac{\delta C}{\delta y_i} = 4 \sum_j (p_{ij} - q_{ij})(y_i - y_j) (1 + \|y_i - y_j\|^2)^{-1} \quad (3.30)$$

$$\Upsilon^{(t)} = \Upsilon^{(t-1)} + \eta \frac{\delta C}{\delta \Upsilon} + \alpha(t) (\Upsilon^{(t-1)} - \Upsilon^{(t-2)}) \quad (3.31)$$

Chapter 4

Structural State Identification by using a Classification 1-D Convolutional Neural Network

4.1 Introduction

In this chapter, a classification CNN model is proposed to identify the locations of tiny local structural changes. The architecture of the CNN is shown in Fig. 4.1. The CNN consists of a 1-D convolutional layer, batch normalization layer, a max pooling layer, a flatten layer, a dropout layer, and an output layer.

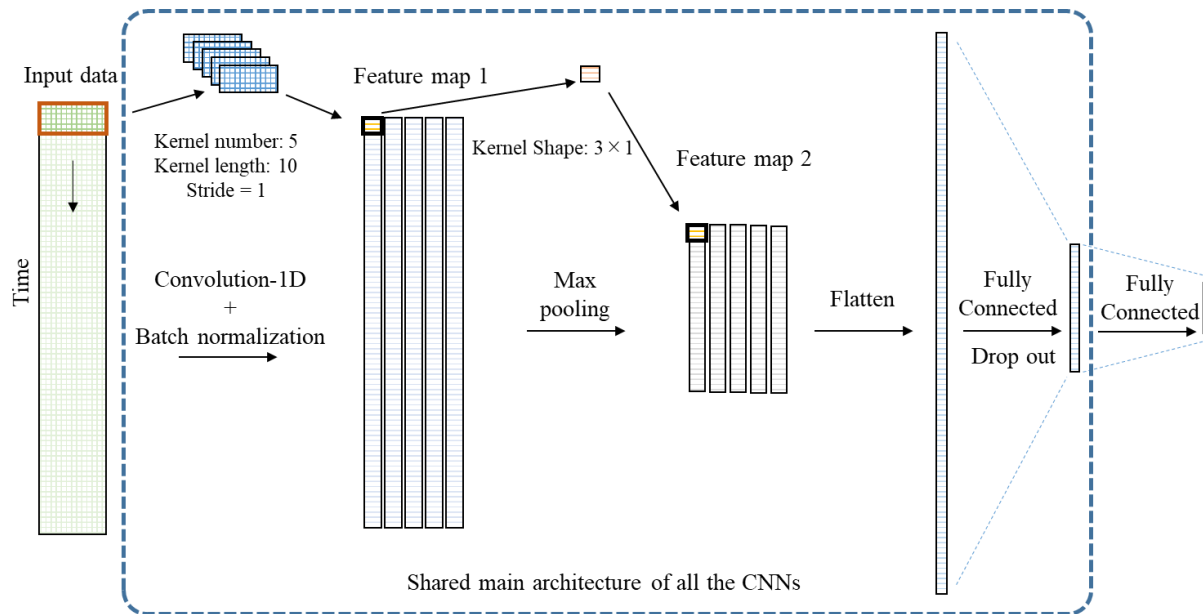


Fig. 4.1 Architecture of the CNN

Vibration experiments were conducted on a steel T-shaped beam (in lab), a short steel girder bridge (in test field), and a long steel girder bridge (in service), establishing three independent databases for training and validating the feasibility of the proposed CNN architecture.

To validate the performance of the proposed method, sufficient acceleration data on both the intact and damaged structural states must be available. Generating the data of damaged structural states is a pivotal problem under current investigation. Releasing a structural component such as a bolt or a brace (Abdeljaber et al. 2017; Avci et al., 2018; Abdeljaber et

al., 2018) is a practical solution, as the damages incurred are easily recovered. However, the artificial vibration data generated by the above damages do not represent the vibration data of local structural changes of bridge girders, which is a precondition to generate and validate a CNN for detecting local structural changes for bridge girders. Fortunately, more realistic data of damaged structural sites can be generated by borrowing the famous sociological concept of *negative population growth* to test the proposed CNN. Common structural damages such as cracks and corrosion, which reduce the local stiffness and mass, can be referred to as positive structural damages. Negative structural damage is then manually generated by slightly increasing the local structural stiffness and mass. By using the vibration data of both the intact structural state and the “negative structural damage” states, whether the CNN-based SDD method detects tiny local stiffness and mass changes on actual structures can be validated.

The sensor system, as shown in Fig. 4.2, is common in all the experiments. The sensor system consists of Ono Sokki NP-3130 accelerometers (sensitivity: $10 \text{ mV/s}^2 \pm 1\text{dB}$), amplifiers, an AD converter (ADC resolution: 16 bits), and an operating computer.



(a) Accelerometer



(b) Data acquisition system

Fig. 4.2 Sensor system

The T-shaped beam and the short steel girder bridge were vertically impacted by a small rubber hammer (weight: 233g). To ensure a diverse database, the hitting force was varied between approximately 100 and 1500 Gal. The hitting period was 2 seconds. To examine more practical excitation methods, the in-service long steel girder bridge was excited by white noise and traffic load.

Afterwards, the performance of the proposed CNN-based structural-state identification method was validated. In each experiment, a 5-fold cross validation was conducted as follows. After shuffling the data order, the database was divided into five equal parts. One of the parts was chosen as the validation set, and the remaining four parts were reserved for training the CNN. The validation was iterated five times, and the accuracies of the five iterations were averaged to give the performance evaluation. Note that in each iteration, the data in the training set and the validation set were never overlapped. To show the good performance of the proposed CNN, the results of CNNs are compared with the results of SVM predictions. The configurations of the SVMs are determined by trial and error. The SVM models were designed with linear kernel, penalty parameter C of 1, and degree polynomial kernel function of 3.

The CNN design was based on the Tensorflow framework (Abadi et al., 2016a; Abadi et al., 2016b) and Keras API (Chollet, 2017) in the Python 3.6 language environment, with an Intel Core i7-6900k CPU and a NVIDIA GTX 1080Ti graphic card to accelerate the training.

4.2 T-shaped Beam Experiment

4.2.1 *Vibration experiment for data generation*

In this first attempt to identify different structural states by a CNN, the experimental subject was a T-shaped steel beam, one of the simplest engineering structures. The beam was 2090 mm long with a 360-mm wide, 10-mm thick flange. The web was 20 mm thick and 390 mm high. The beam was fixed by 8 bolts (4 bolts each on the left and right ends) as shown in Fig. 4.3.

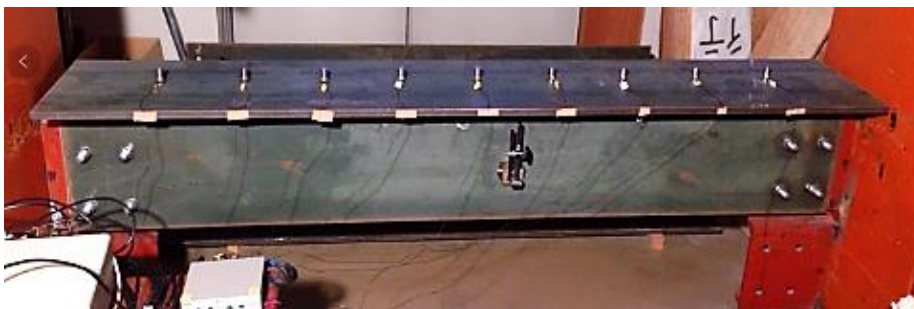


Fig. 4.3 T-shaped steel beam

The layout of the experiment is shown in Fig. 4.4. The vertical vibration was measured by nine accelerometers installed on the surface of the flange. The sampling frequency was 10000

Hz, and the sampling period was set to 1.0 second (a free damped vibration). Thus, each datum was configured as $9 \text{ Ch.} \times 10000$. A magnet weighing 1.37 kg (0.75% structural mass increase) was attached at 8 different positions on the web, providing 8 patterns of the local structural mass changes, and 9 different structural states (State 0: original state with no additional mass, and States 1-8: states with the additional mass). The descriptions of all the structural states are listed in Table 4.1. Each structural state was subjected to approximately 1600 impacts to generate the acceleration data. The data distribution of the database is shown in Table 4.2.

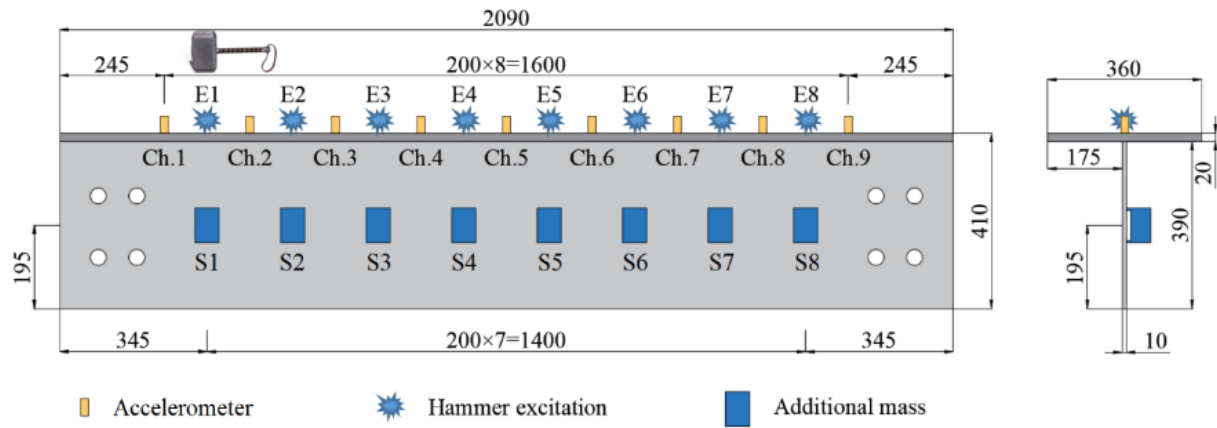


Fig. 4.4 Layout of the T-shaped beam vibration experiment

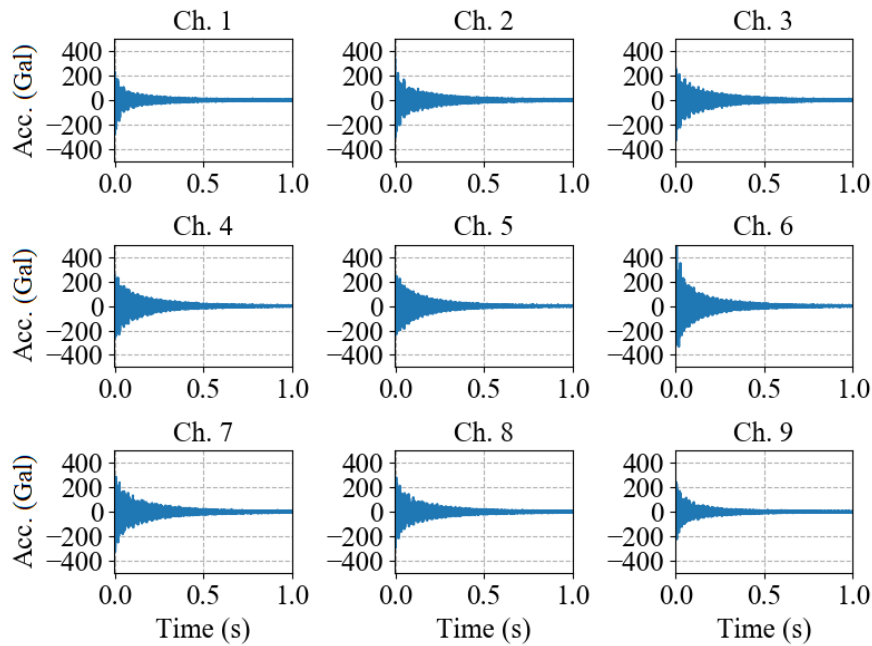
Table 4.1 Description of the structural states of the T-shaped beam vibration experiment

| <i>State</i> | <i>Description</i> |
|--------------|--|
| 0 | Original state, no magnet attached |
| 1 | The magnet is attached on S1, as shown in Fig. 4.4 |
| 2 | The magnet is attached on S2, as shown in Fig. 4.4 |
| 3 | The magnet is attached on S3, as shown in Fig. 4.4 |
| 4 | The magnet is attached on S4, as shown in Fig. 4.4 |
| 5 | The magnet is attached on S5, as shown in Fig. 4.4 |
| 6 | The magnet is attached on S6, as shown in Fig. 4.4 |
| 7 | The magnet is attached on S7, as shown in Fig. 4.4 |
| 8 | The magnet is attached on S8, as shown in Fig. 4.4 |

Table 4.2 Data distribution of the T-shaped beam database

| <i>Category</i> | <i>Amount</i> | <i>Category</i> | <i>Amount</i> |
|-----------------|---------------|-----------------|---------------|
| <i>State 0</i> | 1587 | <i>State 5</i> | 1622 |
| <i>State 1</i> | 1595 | <i>State 6</i> | 1623 |
| <i>State 2</i> | 1595 | <i>State 7</i> | 1619 |
| <i>State 3</i> | 1611 | <i>State 8</i> | 1593 |
| <i>State 4</i> | 1620 | | |
| <i>In total</i> | | | 14465 |

Eight excitation positions (E1-E8) were distributed on the flange surface, as shown in Fig. 4.4. In each data category, the flange was subjected to approximately 200 impacts at each excitation position, generating about 1600 time-free damped vibrations in total. Each free damped vibration of the beam was recorded after a hammer excitation at a different hitting energy. An example of raw vibration data of the T-shaped steel beam data is shown in Fig. 4.5.

**Fig. 4.5** An example of raw vibration data of the T-shaped steel beam database

4.2.2 CNN configuration

The detailed configuration of the applied CNN for structural state identification is shown in Table 4.3, which shares the main architecture shown in Fig. 4.1.

Table 4.3 Configuration of the CNN for the T-shaped beam database

| <i>Layer</i> | <i>Output Shape</i> | <i>Parameter</i> | <i>Activation</i> | <i>Variables</i> |
|----------------------------|---------------------|---|-------------------|------------------|
| <i>Input</i> | 10000×9 | None | None | 0 |
| <i>Convolution 1-D</i> | 9991×5 | Kernel number: 5; Kernel size: 10×9; Stride: 1; Padding: Valid | Linear | 455 |
| <i>Batch normalization</i> | 9991×5 | None | None | 20 |
| <i>Max Pooling 1-D</i> | 3330×5 | Kernel number: 3; Stride: 1 | None | 0 |
| <i>Flatten</i> | 16650 | None | None | 0 |
| <i>Dropout</i> | 16650 | Rate: 0.25 | None | 0 |
| <i>FC</i> | 40 | None | ReLU | 666,040 |
| <i>Output</i> | 9 | None | Softmax | 369 |
| <i>Total parameters</i> | | | | 666,884 |

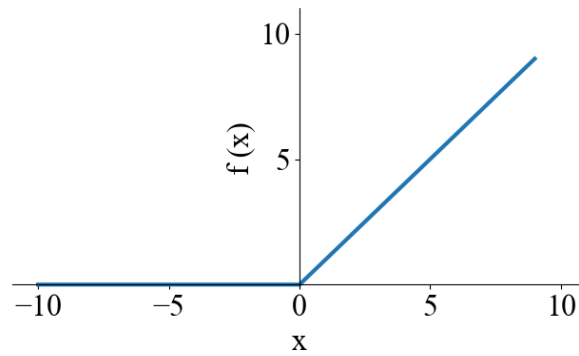
Since a softmax layer is used as the output layer, the corresponding labels is one-hot encoding. One-hot encoding is a vector with only one “1”, and others are “0”. The location of “1” represents the specific data category. The detailed labels of data in each structural state is shown in Table 4.4. For the proposed CNN, the location of “1” in a label indicates the structural state.

Table 4.4 Labels of data in the T-shaped beam database

| | <i>Encoding</i> | | | | | | | | |
|-----------|-----------------|---|---|---|---|---|---|---|---|
| <i>S0</i> | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| <i>S1</i> | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| <i>S2</i> | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| <i>S3</i> | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| <i>S4</i> | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| <i>S5</i> | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| <i>S6</i> | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| <i>S7</i> | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| <i>S8</i> | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

Rectified Linear Unite (ReLU) excitation function is used in the FC layers. The definition of the ReLU excitation function is shown in Equation 4.1, and a figure of ReLU excitation function is shown in Fig. 4.6.

$$f(x) = \max(0, x) \quad (4.1)$$

**Fig. 4.6** ReLU excitation function

All the weights in the network are initialized with Variance Scaling initializer, and all the biases are initialized with zeros.

The loss function of the CNN is categorical cross entropy, as introduced in Subsection 3.4.1.

The CNN has training by the Adam optimizer as introduced in Subsection 3.5.2. The detailed parameters are as follows: learning rate of 0.0001, ρ_1 of 0.9, ρ_2 of 0.999, δ of 1e-8, and decay of 0.

The batch size is 256 in the training process. Thus the network updates 46 times ($14465 \times 0.8 / 256$) in each epoch.

4.2.3 Results

The 5-fold cross-validation results of the T-shaped beam database are shown in Table 4.5. In each fold, the training accuracy was nearly 100.00%, and the validation accuracies ranged between 99.65% and 99.86% (average 99.79%). The detailed validation results of the Fold 1 model are shown in Fig. 4.7 as an example. There are only five wrong predictions in the 2893 validation data. This result confirms that the CNN model is sensitive to tiny local mass changes of the in-lab steel beam structure. The accuracies of the SVM predictions show less than 1% lower than the accuracies of the proposed CNN. The proposed CNN shows no clear advantage in the steel beam experiment, since the structure and the excitation method are both the simplest cases. The merits of the proposed CNN are more obvious in the cases with more complex structure or more complex excitation methods as discussed in Sections 4.3 and 4.4.

Table 4.5 Accuracy of 5-fold cross validation of the T-shaped beam database (%)

| | <i>Training</i> | <i>Validation</i> | <i>Validation (SVM)</i> |
|----------------|-----------------|-------------------|-------------------------|
| <i>Fold 1</i> | 100.00 | 99.83 | 99.27 |
| <i>Fold 2</i> | 99.99 | 99.76 | 99.20 |
| <i>Fold 3</i> | 100.00 | 99.86 | 99.34 |
| <i>Fold 4</i> | 100.00 | 99.65 | 98.86 |
| <i>Fold 5</i> | 100.00 | 99.83 | 99.52 |
| <i>Average</i> | 100.00 | 99.79 | 99.24 |

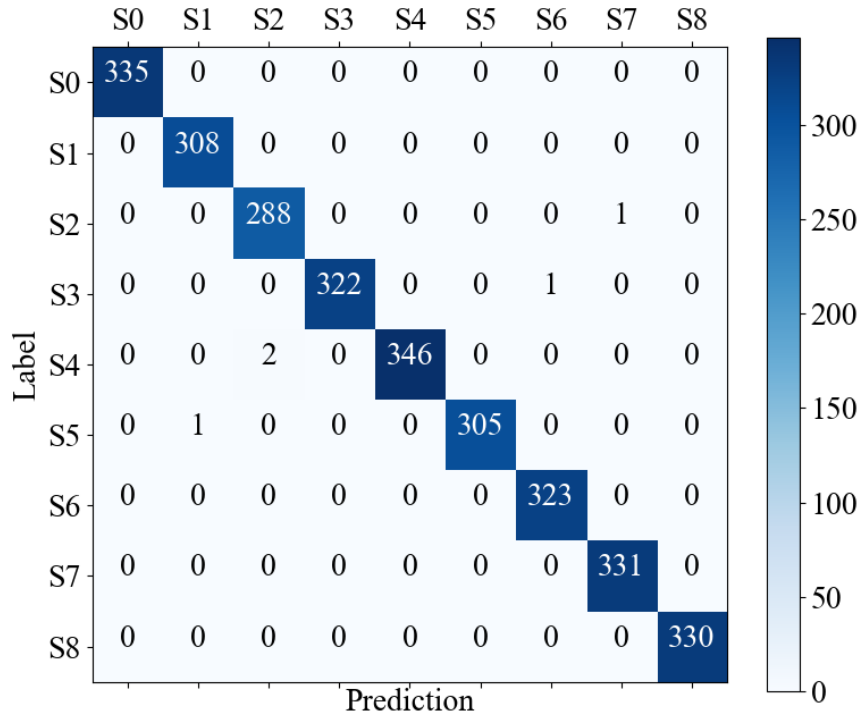


Fig. 4.7 Confusion matrix of the validation results of the Fold 1 model of the T-shaped steel beam

The CNN was trained at a very high convergence speed. The training history of Fold 1 is presented as an example in Fig. 4.8. The accuracies of the training and validation sets in each epoch are plotted. The accuracies of both sets reached 95% after Epoch 2. During the first several epochs, the validation accuracies were higher than the training accuracies, because the training accuracy was calculated by counting the number of correctly predicted training data throughout 46 iterations (number of training data / batch size = $((14465 \times 0.8) / 256)$ on a batch-by-batch basis in each epoch, whereas the validation accuracy was obtained after each epoch. That is, in each epoch, the training accuracy was calculated batch-by-batch in all 46 time updates of the CNN, whereas the validation accuracy was computed after the 46th time update.

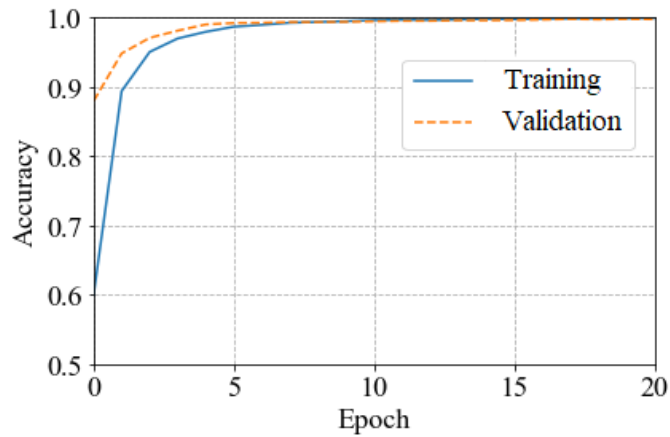


Fig. 4.8 Training procedure of the CNN for the T-shape beam database (Fold 1)

4.3 Short Steel Girder Bridge Experiment

4.3.1 *Vibration experiment for data generation*

To validate whether the proposed CNN can identify local stiffness and mass changes in an actual civil structure, the vibration experiment was repeated on a steel girder railway bridge constructed in 1925, as shown in Fig. 4.9. The sleepers and tracks on the bridge have been removed, and the bridge has been transported to the test field of the authors' institute. The bridge is 6.45 m long, 1.994 m wide, and 0.775 m high.



Fig. 4.9 Short steel girder bridge

The experimental layout is shown in Fig. 4.10. The vertical vibrations were measured by 15 accelerometers installed on the surface of the upper flange of the main girder. The sampling

frequency and period were set to 10000 Hz and 0.6 second (a free damped vibration), respectively. Thus, each datum was configured as 15 Ch. \times 6000.

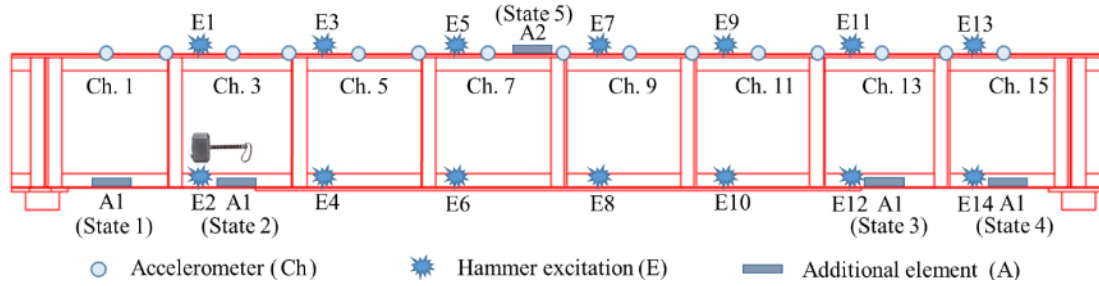


Fig. 4.10 Layout of the short steel girder bridge vibration experiment

In this experiment, the local structural state was changed in two ways. First, a steel plate on the lower flange was affixed to the bridge by two clamps, as shown in Fig. 4.11(a). The steel plate was 18 cm long, 8 cm wide and 1.2 cm thick, and weighed 5.11 kg (0.21% structural mass increase). The fixed steel plate slightly increased both the local stiffness and mass of the bridge. In the second way, a mass block (3.21 kg, 0.13% structural mass increase) was attached to the surface of the upper flange, as displayed in Fig. 4.11(b). The mass block was an actuator with its excitation function disengaged. The actuator is a convenient additional mass because it affixes to the surface by its magnet.



(a) steel plate



(b) mass block

Fig. 4.11 Additional elements in the short steel girder bridge experiment

In total there are six structural states in this experiment: an initial state with no additional element installed (State 0), four structural states created by fixing the steel plate at four different

positions on the two ends of the lower flange (States 1-4), and one structural state generated by attaching the mass block to the middle of the upper flange (State 5). The detailed descriptions of the structural states are summarized in Table 4.6.

Table 4.6 Descriptions of the structural states of the short steel girder bridge vibration experiment

| <i>State</i> | <i>Description</i> |
|--------------|---|
| 0 | Original state, no additional element attached |
| 1 | The steel plate is fixed on the left side of the lower flange, as shown in Fig. 4.10, changing local stiffness and mass |
| 2 | The steel plate is fixed on the second left side of the lower flange, as shown in Fig. 4.10, changing local stiffness and mass |
| 3 | The steel plate is fixed on the second right side of the lower flange, as shown in Fig. 4.10, changing local stiffness and mass |
| 4 | The steel plate is fixed on the right side of the lower flange, as shown in Fig. 4.10, changing local stiffness and mass |
| 5 | The mass block is attached on the middle of the upper flange, as shown in Fig. 4.10, changing local mass |

Moreover, there are 14 excitation positions on the bridge: seven positions on the upper flange (E1, E3, E5, ..., E13), and another seven on the lower flange (E2, E4, E6, ..., E14), as shown in Fig. 4.10. After applying an impulse load by striking the bridge with a hammer, the acceleration data of the 0.6-s free damped vibration data were collected. The data distribution is shown in Table 4.7. The data of State 0 were collected first. Owing to time limitations, the numbers of vibration test in States 1-5 were reduced, so the data distribution was somewhat unbalanced. In total, 8595 data are stored in this database.

Table 4.7 Data distribution of the short steel girder bridge database

| <i>Category</i> | <i>Amount</i> | <i>Category</i> | <i>Amount</i> |
|-----------------|---------------|-----------------|---------------|
| <i>State 0</i> | 2073 | <i>State 3</i> | 1362 |
| <i>State 1</i> | 1279 | <i>State 4</i> | 1286 |
| <i>State 2</i> | 1414 | <i>State 5</i> | 1181 |
| <i>In total</i> | | | 8595 |

4.3.2 CNN configuration

The detailed configuration of the applied CNN for structural state identification is shown in Table 4.8. The CNN shares the main architecture with the CNN for the T-shaped beam in Section 4.2, as display in Fig. 4.1. Since the amounts of accelerometers and structural states are different, the data shapes vary when data flowing through layers.

Table 4.8 Configuration of the CNN for the short steel girder bridge database

| <i>Layer</i> | <i>Output Shape</i> | <i>Parameter</i> | <i>Activation</i> | <i>Variables</i> |
|----------------------------|---------------------|---|-------------------|------------------|
| <i>Input</i> | 6000×15 | None | None | 0 |
| <i>Convolution 1-D</i> | 5991×5 | Kernel number: 5; Kernel size: 10×9; Stride: 1; Padding: Valid | Linear | 755 |
| <i>Batch normalization</i> | 5991×5 | None | None | 20 |
| <i>Max Pooling 1-D</i> | 1997×5 | Kernel number: 3; Stride: 1 | None | 0 |
| <i>Flatten</i> | 9985 | None | None | 0 |
| <i>Dropout</i> | 9985 | Rate: 0.25 | None | 0 |
| <i>FC</i> | 40 | None | ReLU | 399,440 |
| <i>Output</i> | 6 | None | Softmax | 246 |
| <i>Total parameters</i> | | | | 400,461 |

All the weights in the network are initialized with Variance Scaling initializer, and all the biases are initialized with zeros. The encoding of labels is also one-hot. The detailed labels of all the structural states are shown in Table. 4.9.

Table 4.9 Labels of data in the short steel girder bridge vibration database

| <i>State</i> | | <i>Encoding</i> | | | | |
|--------------|---|-----------------|---|---|---|---|
| <i>0</i> | 1 | 0 | 0 | 0 | 0 | 0 |
| <i>1</i> | 0 | 1 | 0 | 0 | 0 | 0 |
| <i>2</i> | 0 | 0 | 1 | 0 | 0 | 0 |
| <i>3</i> | 0 | 0 | 0 | 1 | 0 | 0 |
| <i>4</i> | 0 | 0 | 0 | 0 | 1 | 0 |
| <i>5</i> | 0 | 0 | 0 | 0 | 0 | 1 |

The loss function of the CNN is categorical cross entropy, as introduced in Subsection 3.4.1.

The CNN has training by the Adam optimizer as introduced in Subsection 3.5.2. The detailed parameters are as follows: learning rate of 0.0001, ρ_1 of 0.9, ρ_2 of 0.999, δ of 1e-8, and decay of 0.

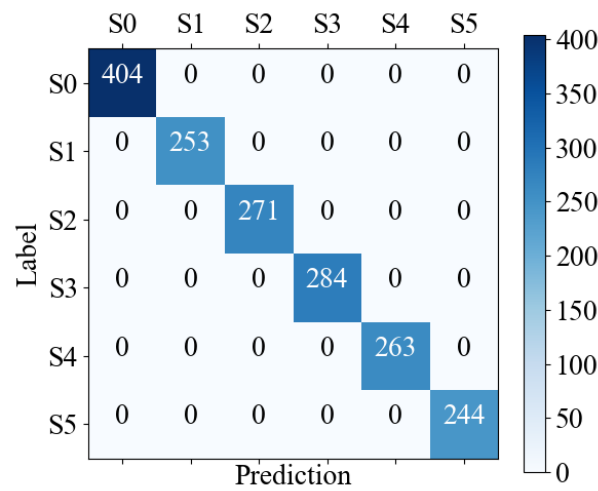
The batch size is 256 in the training process. Thus the network updates 27 times ($8595 \times 0.8 / 256$) in each epoch.

4.3.3 Results

The accuracy of the proposed CNN model was also confirmed in a cross-validation of the short steel bridge vibration experiment (see Table 4.10). The training and validation accuracies obtained in the five cross-validations ranged from 99.96% to 100.00% and from 98.89% to 100.00%, respectively, averaging 99.98% and 99.36% respectively. The detailed validation results of the Fold 1 model are shown in Fig. 4.12 as an example. There is only no wrong prediction of the 1719 validation data.

Table 4.10 Accuracy of 5-fold cross validation of the short steel girder bridge database (%)

| | <i>Training</i> | <i>Validation</i> | <i>Validation (SVM)</i> |
|----------------|-----------------|-------------------|--------------------------|
| <i>Fold 1</i> | 99.97 | 100.00 | 97.67 |
| <i>Fold 2</i> | 99.96 | 99.24 | 97.79 |
| <i>Fold 3</i> | 100.00 | 98.89 | 97.50 |
| <i>Fold 4</i> | 99.96 | 99.07 | 97.50 |
| <i>Fold 5</i> | 99.99 | 99.59 | 97.73 |
| <i>Average</i> | 99.98 | 99.36 | 97.64 |

**Fig. 4.11** Confusion matrix of the validation results of the Fold 1 model of the short steel girder bridge

The high accuracy of the 5-fold cross-validation confirms the high sensitivity of the CNN model to tiny local mass and stiffness changes, even though the ratios of the local mass changes were only 0.21% in States 1-4 and 0.13% in State 5. The validation accuracy was slightly lower than that of the T-shaped beam, probably because the steel bridge has a complex structure and the data were acquired in an outdoor environment.

The accuracies of the SVM predictions in each fold or the overall result are about 2% lower than the accuracies of the proposed CNN. As the structure and environment become more complex, the advantages of the CNN begin to emerge.

The training converged quickly despite the unbalanced database. As an example, Fig. 4.13 plots the training and validation accuracies in Fold 1. The accuracies of both the training and validation datasets exceeded 90% after five epochs of training, and 99% after 20 epochs of

training. The fast convergence of the CNN model also proves the sensitivity of the CNN model to the small local stiffness and mass changes of an actual outdoor structure.

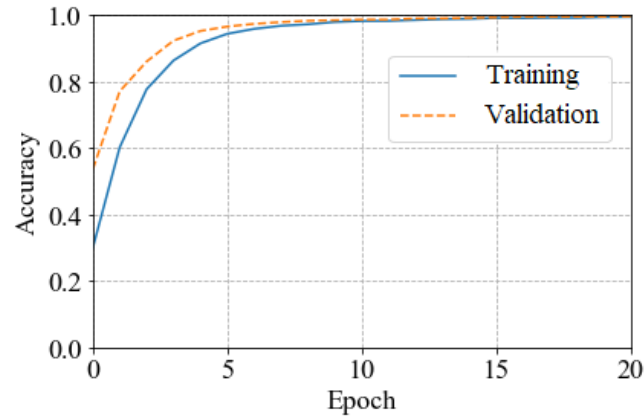


Fig. 4.13 Training procedure of the CNN for the short steel girder bridge database (Fold 1)

4.4 Long Steel Girder Bridge Experiment

4.4.1 *Vibration experiment for data generation*

To validate whether the proposed CNN is sensitive to small local mass changes in actual in-service bridge, the vibration experiment was examined on a steel girder bridge in Kitami, Hokkaido, Japan, as shown in Fig. 4.14. The bridge is 27.3 m long and 13.8 m wide, and has 6 main steel girders. The general drawing of the bridge is shown in Fig. 4.15.



Fig. 4.14 Long steel girder bridge

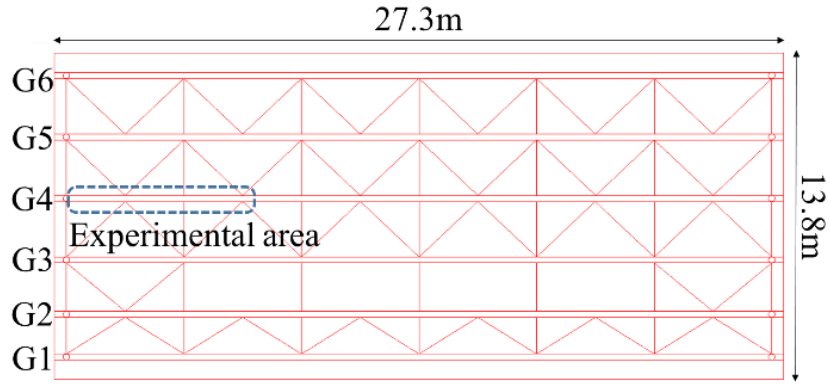


Fig. 4.15 General drawing of the long steel girder bridge

The experimental layout is shown in Fig. 4.16. The vibration experiment was performed on the left quarter of the Girder 4. The vertical vibrations were measured by 15 accelerometers installed on the upper surface of the lower flange close to the web, as shown in Figs. 4.16 - 4.17. The sampling frequency was set to 10000 Hz.

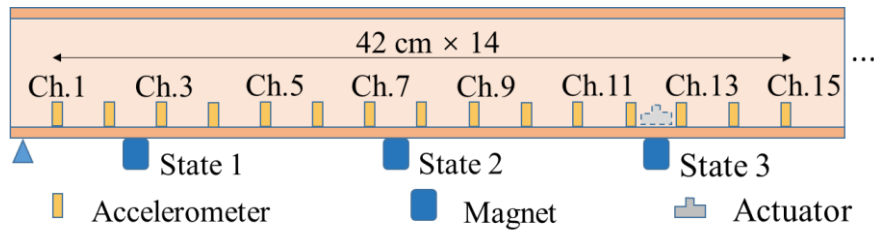


Fig. 4.16 Layout of the long steel girder bridge vibration experiment

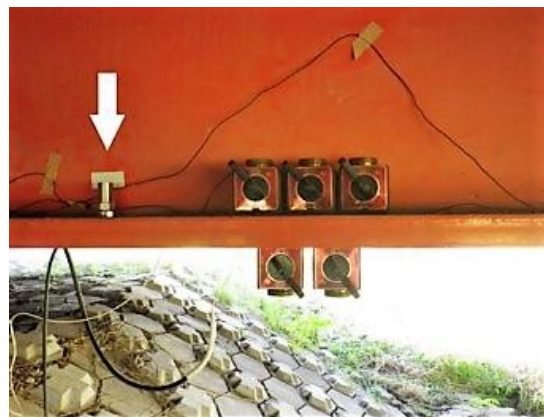


Fig. 4.17 Accelerometer and magnets

To validate a more practical excitation method rather than hammer impact excitation, an actuator was installed on the lower flange, as shown in Fig. 4.18, generating white noise to excite the girder. As the traffic had not been restricted, the acquired vibration data has the component affected by the traffic load.



Fig. 4.18 Actuator

Five magnets weighing 4.72 kg (approximately 0.0015% structural mass increase) was attached at 3 different positions on the lower flange, as shown in Figs. 4.16-4.17, providing 3 patterns of the local structural mass changes, and 4 different structural states (State 0: original state with no additional mass, and States 1-3: states with the additional mass). The detailed descriptions of the structural states are summarized in Table 4.11.

Table 4.11 Description of the structural states in the long steel girder bridge experiment

| <i>State</i> | <i>Description</i> |
|--------------|--|
| 0 | Original state, no magnet attached |
| 1 | Magnets are attached on left side the lower flange, as shown in Fig. 4.16 |
| 2 | Magnets are attached on second left side the lower flange, as shown in Fig. 4.16 |
| 3 | Magnets are attached on third left side the lower flange, as shown in Fig. 4.16 |

In each structural state, after engaging the actuator, vibration of the girder was measured for 40 minutes. Afterwards, the vibration data was split to 1200 pieces of 2s-long data. Therefore,

a database, which consists of 4800 data (4 structural states \times 1200 data), was established. The detailed data distribution is shown in Table 4.12. Meanwhile, a datum in State 0 is shown as an example in Fig. 4.19.

Table 4.12 Data distribution of the long steel girder bridge database

| <i>Category</i> | <i>Amount</i> | <i>Category</i> | <i>Amount</i> |
|-----------------|---------------|-----------------|---------------|
| <i>State 0</i> | 1200 | <i>State 2</i> | 1200 |
| <i>State 1</i> | 1200 | <i>State 3</i> | 1200 |
| <i>In total</i> | 4800 | | |

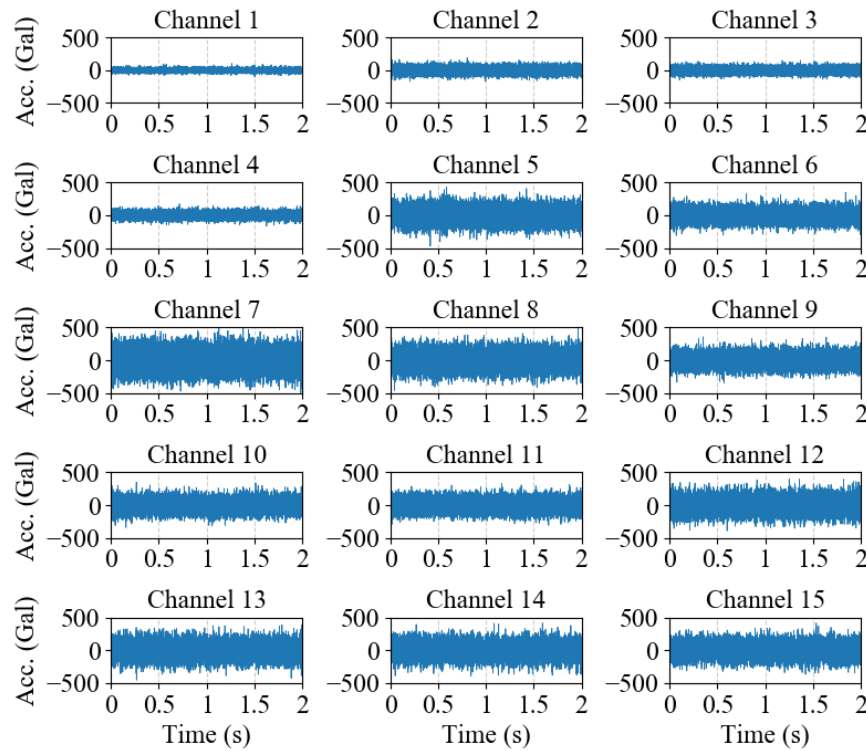


Fig. 4.19 An example of vibration data of the long steel girder bridge database

4.4.2 CNN configuration

The detailed construction of the applied CNN for structural state identification is shown in Table 4.13. The CNN shares the main architecture with the CNNs for the T-shaped beam in Section 4.2 and the short steel girder bridge in Section 4.3, as displayed in Fig. 4.1, even though the amounts of accelerometers, shapes of input data, shapes of output data in each layer, and amounts structural states are different in each case.

Table 4.13 Configuration of the CNN for the long steel girder bridge database

| <i>Layer</i> | <i>Output Shape</i> | <i>Parameter</i> | <i>Activation</i> | <i>Variables</i> |
|----------------------------|---------------------|---|-------------------|------------------|
| <i>Input</i> | 20000×15 | None | None | 0 |
| <i>Convolution 1-D</i> | 19991×5 | Kernel number: 5; Kernel size: 10×9; Stride: 1; Padding: Valid | Linear | 755 |
| <i>Batch normalization</i> | 19991×5 | None | None | 20 |
| <i>Max Pooling 1-D</i> | 6663×5 | Kernel Stride: 3; | None | 0 |
| <i>Flatten</i> | 33315 | None | None | 0 |
| <i>Dropout</i> | 33315 | Rate: 0.25 | None | 0 |
| <i>FC</i> | 40 | None | ReLU | 1,332,640 |
| <i>Output</i> | 4 | None | Softmax | 164 |
| <i>Total parameters</i> | 1,333,569 | | | |

The labels are also one-hot encoding. The detailed labels of all the structural states are shown in Table. 4.14. The location of “1” in the vector indicates the structural state.

Table 4.14. Labels of data in the long steel girder bridge database

| <i>State</i> | <i>Encoding</i> | | | | |
|--------------|-----------------|---|---|---|---|
| <i>0</i> | 1 | 0 | 0 | 0 | 0 |
| <i>1</i> | 0 | 1 | 0 | 0 | 0 |
| <i>2</i> | 0 | 0 | 1 | 0 | 0 |
| <i>3</i> | 0 | 0 | 0 | 0 | 1 |

All the weights in the network are initialized with Variance Scaling initializer, and all the biases are initialized with zeros.

The loss function of the CNN is categorical cross entropy, as introduced in Subsection 3.4.1.

The CNN has training by the Adam optimizer as introduced in Subsection 3.5.2. The detailed parameters are as follows: learning rate of 0.0001, ρ_1 of 0.9, ρ_2 of 0.999, δ of 1e-8, and decay of 0.

The batch size is 256 in the training process. Thus the network updates 15 times ($4800 \times 0.8 / 256$) in each epoch.

4.4.3 Results

Table 4.15 shows the result of cross validation of the long steel girder bridge database. The training and validation accuracies obtained in the five cross-validations were all 100.00% and ranged from 96.15% to 98.12%, respectively, averaging 100.00% and 97.23% respectively. The detailed validation results of the Fold 1 model are shown in Fig. 4.20 as an example. There are only 18 wrong predictions in the 960 validation data. Comparing to the results in Subsections 4.2.3 and 4.3.3, the accuracy decreases a little. Considering the complexity of the structure and loads, the accuracy is still very high. The high accuracy confirms the high sensitivity of the CNN model to tiny local structural changes in the actual operating bridge, even though the bridge has complex structure, and excited by white noise and traffic load.

Table 4.15 Accuracy of 5-fold cross validation of the long steel girder bridge database (%)

| | <i>Training</i> | <i>Validation</i> | <i>Validation (SVM)</i> |
|----------------|-----------------|-------------------|-------------------------|
| <i>Fold 1</i> | 100.00 | 98.12 | 23.85 |
| <i>Fold 2</i> | 100.00 | 98.12 | 29.79 |
| <i>Fold 3</i> | 100.00 | 96.35 | 26.35 |
| <i>Fold 4</i> | 100.00 | 97.40 | 31.35 |
| <i>Fold 5</i> | 100.00 | 96.15 | 28.33 |
| <i>Average</i> | 100.00 | 97.23 | 27.93 |

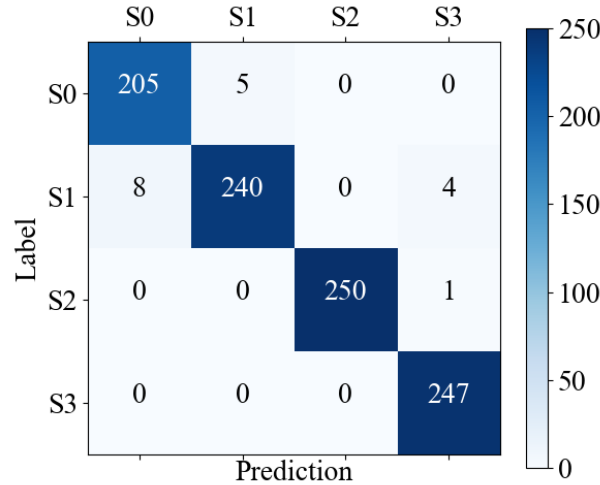


Fig. 4.20 Confusion matrix of the validation results of the Fold 1 model of the long steel girder bridge

The SVM predictions only show 27.93% accuracy in the overall result, as shown in Table 4.15. Comparing to the results of the T-shaped beam and the short steel girder bridge as discussed in Subsections 4.2.3 and 4.3.3, the accuracy decreased about 70% suddenly. Firstly, changing the excitation method to white noise and traffic load is considered as the main reason of the accuracy decrease, since the complexity of the data are highly increased. In previous two cases introduced in Subsections 4.2.3 and 4.3.3, the excitation method is hammer impact, and the data are regular free damped vibration. In this case, random data are obtained owing to the white noise excitation. Secondly, in previous two cases, the shapes of data are 10000×9 and 6000×15 respectively. In this case, the shape of each data is increased to 20000×15 , and the amount of sample in each data is 3.3 times of the previous two cases. The expanding of data shape is also an important reason, because enlarging the number of dimensions of data will highly increase the difficulty of SVM to define the hyperplane between different categories of data.

To better understand the reason of the low accuracies of the SVM results in this case, the data structures of the databases of the T-shaped beam, the short steel girder bridge, and the long steel girder bridge are visualized by the T-SNE method (introduced in Section 3.8) in Figs. 4.21 - 4.23. Comparing the complexity of the three data structures, the data structures of the T-shaped beam and the short steel girder bridge show many small clusters of a certain category of data, but the data structure of the long steel girder shows no accurate cluster with an incomprehensible data distribution. The database of the long steel girder bridge shows more

complex data structure than databases of the T-shaped beam and the short steel girder bridge, which causes low accuracies of the SVM results of the long steel girder bridge database.

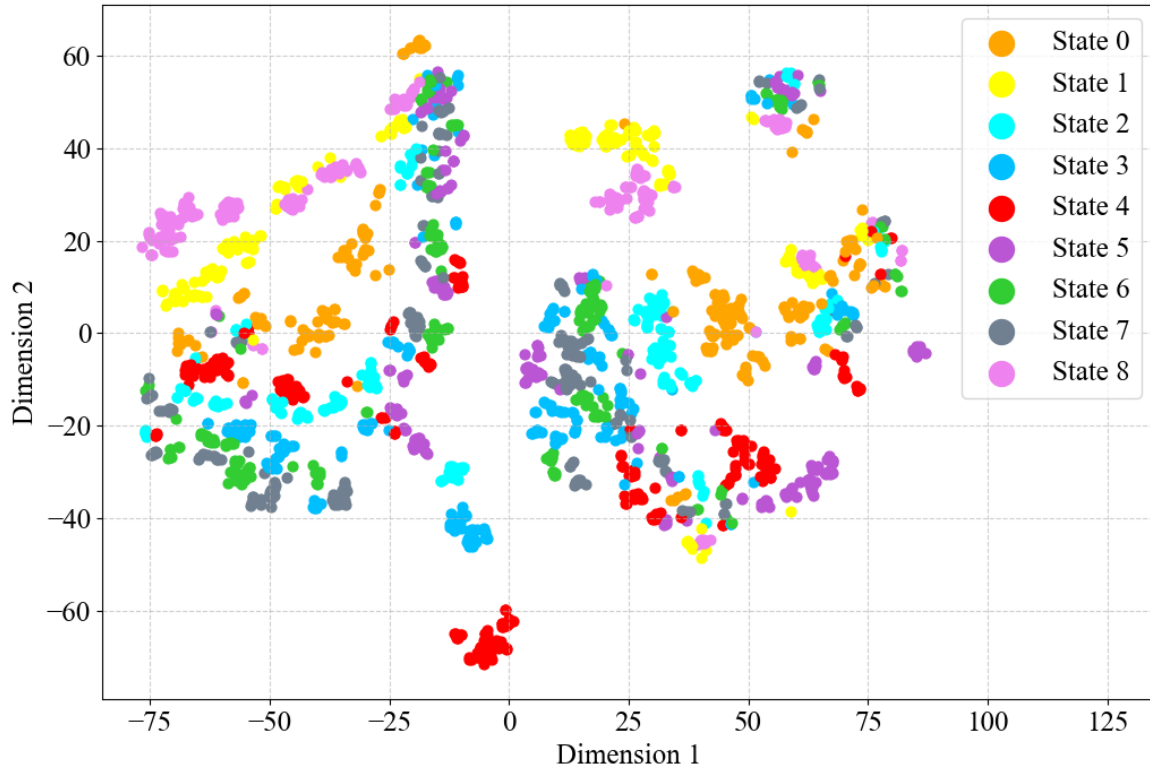


Fig. 4.21 Data structure of the T-shaped steel beam database

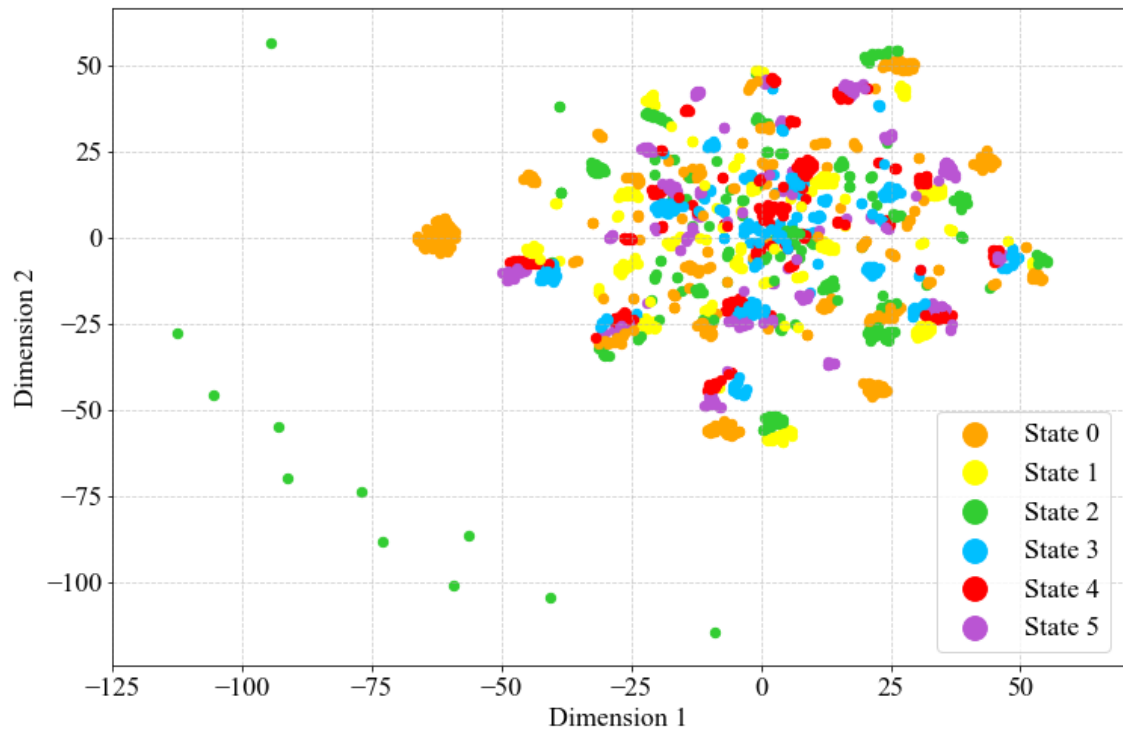


Fig. 4.22 Data structure of the short steel girder bridge database

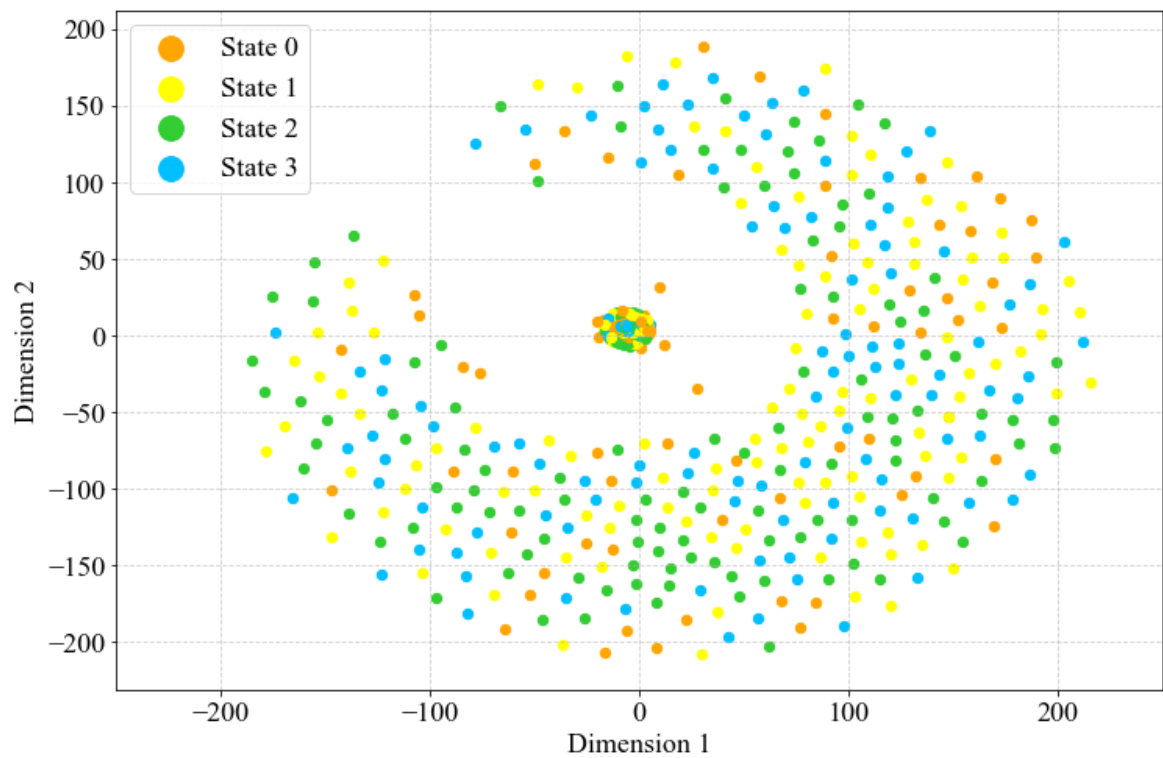


Fig. 4.23 Data structure of the long steel girder bridge database

The reason of that why the CNN obtained very high accuracies and the SVM obtained very low accuracies as shown in Table 4.51 could be explained by the fundamental concepts or algorithms of the two methods. Generally, CNN processes a high-dimensional data through a series of operations to output a low-dimensional prediction. The data dimension flows from a high dimension to a low dimension, which continuously decreases the complexity of data. As a comparison, the SVM method increases the dimension of data to define the hyperplane between each category of data, which increases the complexity of data. Therefore, when dealing with the classification tasks with high-dimensional of data, the CNN method could show better performance than the SVM method.

Fig. 4.24 shows the training procedure of Fold 2 as an example. The training procedure is smooth and fast even though the data were acquired from an actual in-service bridge excited by white noise and vehicle load. The fast convergence also proves the sensitivity of the proposed simple CNN to the small local structural changes of an operating bridge.

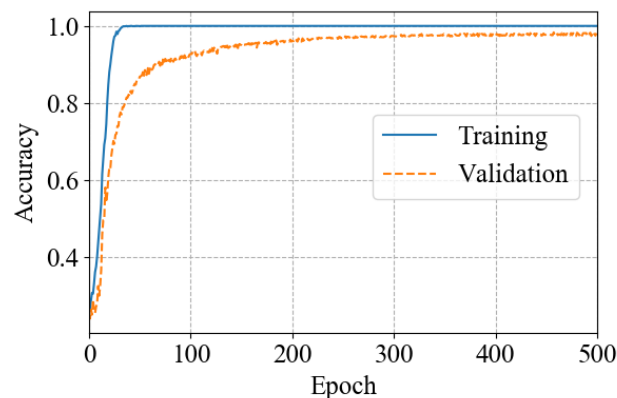


Fig. 4.24 Training procedure of the CNN for the long steel bridge database (Fold 2)

4.5 Conclusions

The proposed CNN shows very high accuracies of identifying tiny local structural changes in the T-shaped steel beam, the short steel girder bridge, and the long steel girder bridge. The CNNs abstract the features of vibration data automatically. The sensitivity of the proposed CNN to the tiny local structural changes of actual structures has been proved. The results of the long steel girder bridge experiment also show good inclusiveness for different excitation

methods of the proposed CNN, since the vibration data acquired on the long steel girder bridge was excited by white noise with the component of traffic load.

The limitation of the proposed CNN is that only structural changes in trained categories are validated with data acquired in relative stable temperature conditions. The robustness of the CNN to location of structural change and temperature effect is unknown. Thus, the robustness of the CNN to location of structural change and temperature effect is examined and discussed in Chapter 6.

Chapter 5

Visualization and Analyses of the CNN Model

5.1 Introduction

Chapter 4 confirmed that the CNNs accurately identify tiny local structural changes in the T-shaped beam, the short steel girder bridge, and the long steel girder bridge. These results of CNN performance are very encouraging. However, as structural damage detection is a highly responsible task, the classification rules of the CNN that link the abstract CNN algorithm with the classical dynamics must be understood. To this end, this section visualizes and discusses the parameters and data flow processes of the CNN, which will help to elucidate the physical mechanism of structural state identification by the CNN-based method.

5.2 Convolutional Kernel Visualization

To understand the features in the training data learned by the convolutional layer, many researchers visualize the convolutional kernels as heatmaps. In image-related tasks, the kernels usually exhibit some oriented strips or blobs (Krizhevsky et al., 2012), which are considered as the features learned from the images. Therefore, the author generated heatmaps of convolutional kernels sized (10×9) , (10×15) , and (10×15) in the proposed CNNs of the T-shaped beam, the short steel girder bridge, and the long steel girder bridge databases, respectively. The results are shown in Figs. 5.1-5.3, respectively. The color intensities of the kernels indicate the weight values (the darker the color, the heavier the kernel weight). Unlike the CNN kernels in image-related tasks, the kernels in the vibration-based structural-state identification show no strip or blob. Whereas the pixels in images are strongly correlated with the 2-D surrounding pixels, the vibration data were strongly correlated only with their neighboring data in the 1-D temporal dimension. In Figs. 5.1-5.3, the high and low values in each kernel alternate along the temporal dimension, implying vibrational characteristics. Thus, attention is focused on the 1-D features of the kernels along the temporal dimension.

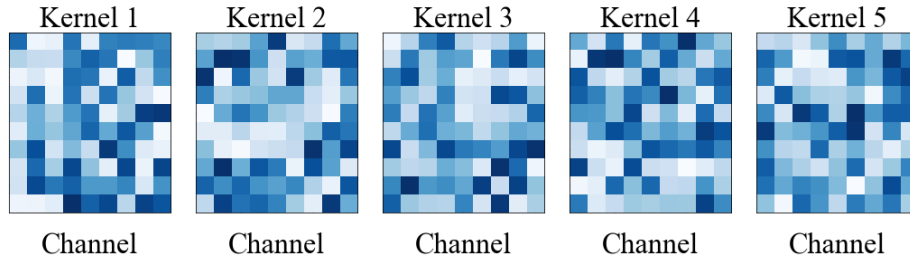


Fig. 5.1 Kernels for the T-shaped beam database

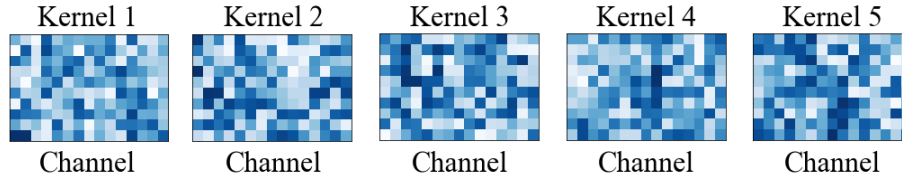


Fig. 5.2 Kernels for the short steel girder bridge database

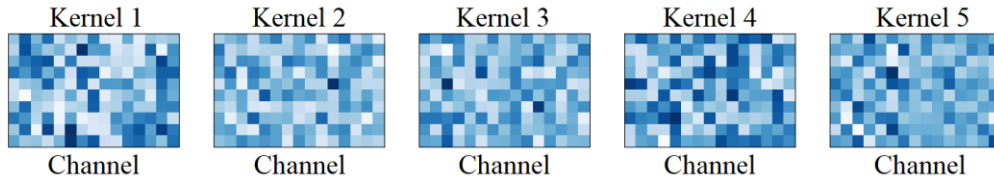


Fig. 5.3 Kernels for the long steel girder bridge database

As an example, Fig. 5.4 plots the 1-D window functions in each channel of the five convolutional kernels for the T-shaped beam database. Clearly, the convolutional kernels learned the vibration features in the training data corresponding to each channel of the raw acceleration data. In all channels of the five kernels, the learned vibration features were approximately constant in amplitude (~ 0.2), but obviously differed in their vibration frequencies. For example, the approximate number of periods in the limited length of the kernel was 3.5 in Channel 6 of Kernel 1, but 2 in Channel 8 of Kernel 1. The convolutional kernels operate as dynamic filters in each channel of the acceleration data.

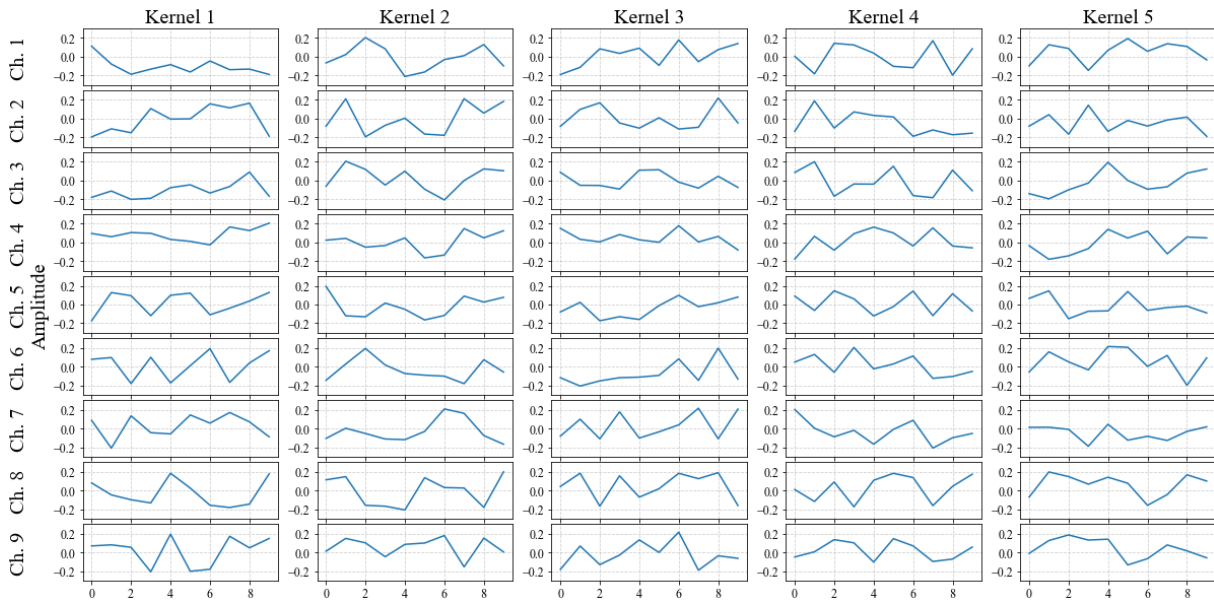


Fig. 5.4 Visualization of kernels for the T-shape beam database in the view of window function by channel

The convolutional kernels of the CNNs for the short steel girder bridge database and the long steel girder database are also visualized as 1-D functions in Figs. 5.5 and 5.6 respectively. Even though the specimens are quite different, the convolutional kernels of the two bridges show same characteristics to the convolutional kernels for the T-shaped beam database as shown in Fig. 5.4. The amplitude of each channel are all in an approximate range of ± 0.2 with different vibration frequencies.

As shown in Figs. 5.4-5.6, the configurations of the convolutional kernels largely differ from those of traditional filters. One benefit of filtering through convolutional layers is that (unlike traditional filters) the parameters of each filter need not to be manually designed. Instead, the filter parameters are trained by the CNN, rendering them more compatible with the specific data structure. Meanwhile, the kernel parameters are considered as features of the data corresponding to the structural-state identification problem, which defies a physical explanation.

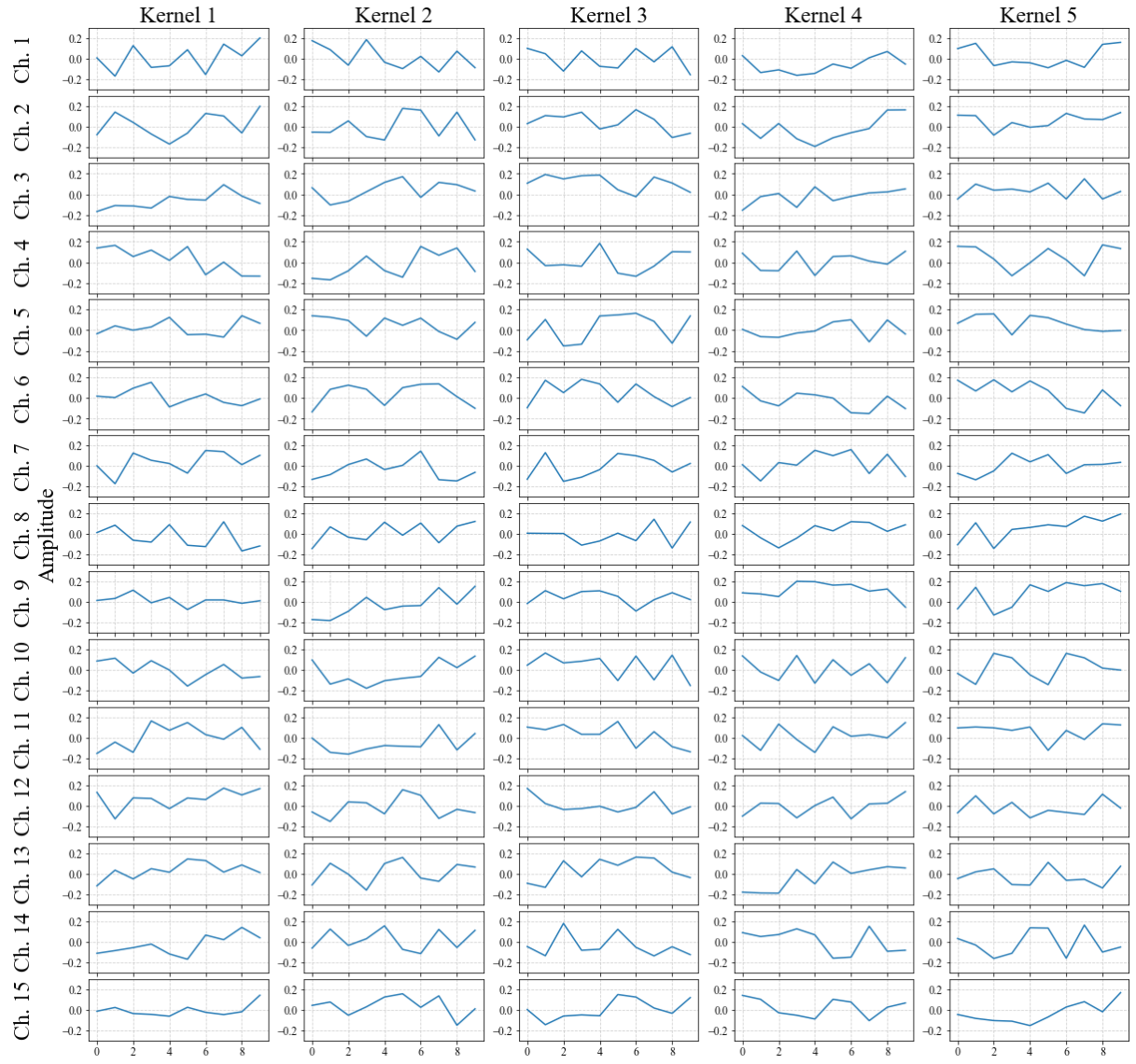


Fig. 5.5 Visualization of kernels for the short steel girder bridge database in the view of window function by channel

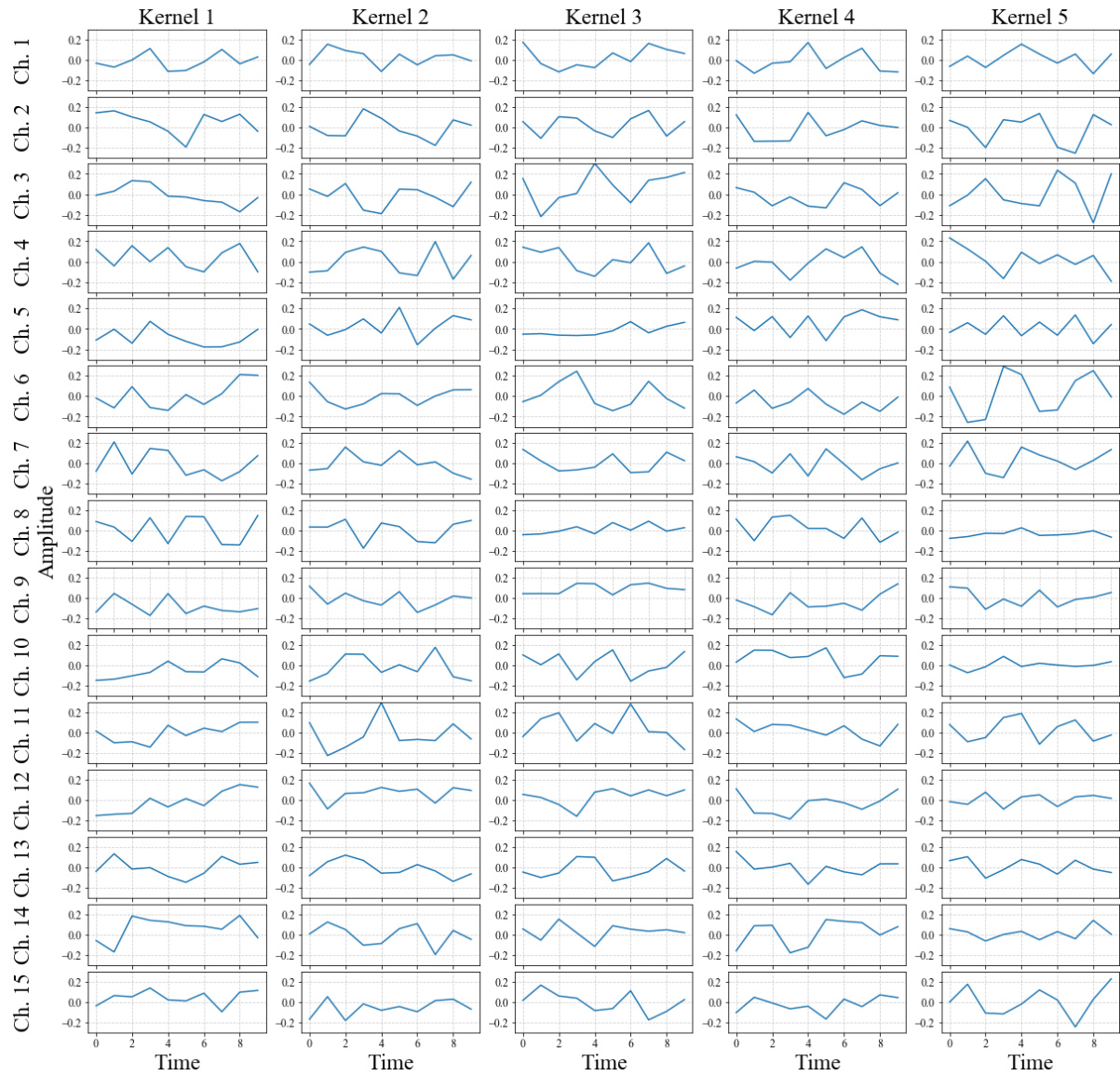


Fig. 5.6 Visualization of kernels for the long steel girder bridge database in the view of window function by channel

5.3 Visualization of the Hidden Layer Outputs

To better understand the identification mechanism of the CNN, this section visualizes the outputs of the convolutional layer and the max pooling layer. The outputs of both layers are presented in the time-domain and frequency-domain views.

5.3.1 CNN for the T-shaped beam database

To investigate the process of data flow through the CNN, many data were traced sequentially through the CNN layers, from input to output. As an example, the author demonstrates the data flow of a datum in State 1 of the T-shaped beam database. Figs. 5.7 and 5.8 show the input data in the time and frequency domains, respectively. Free damped vibration is clarified in Fig. 5.7 and the obvious peaks at 184 Hz, 425 Hz, 630 Hz, and 680 Hz in Fig. 5.8, indicating the modes of the vibration. The corresponding mode shapes are shown Fig. 5.9. As no accelerometer was set on the flange, no mode shape of flange could be obtained. Therefore, for easy understanding, numerical analysis results are also presented to show the mode shapes of the flange.

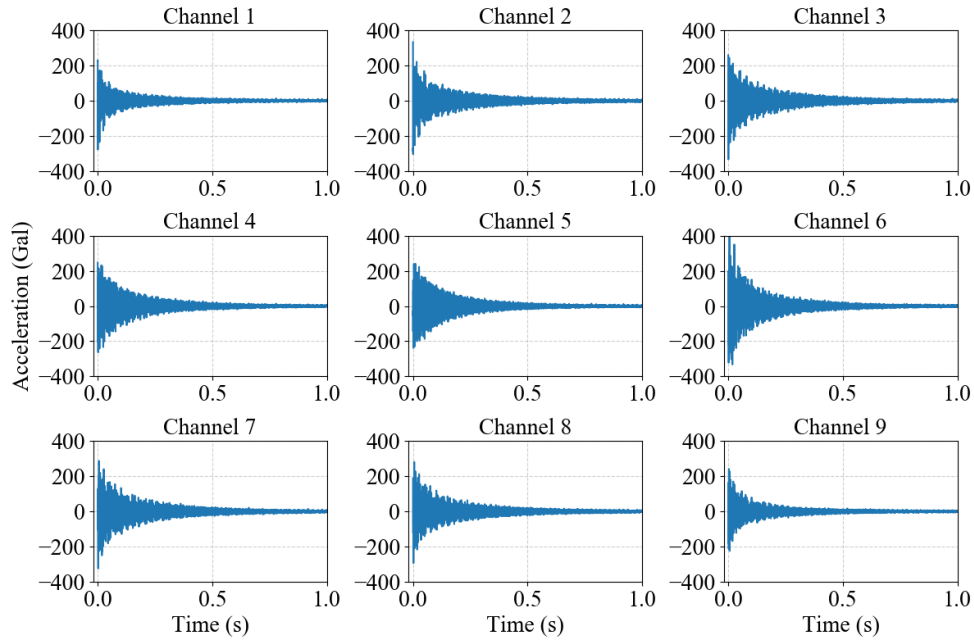


Fig. 5.7 Input data in time domain (example in the T-shaped steel beam database)

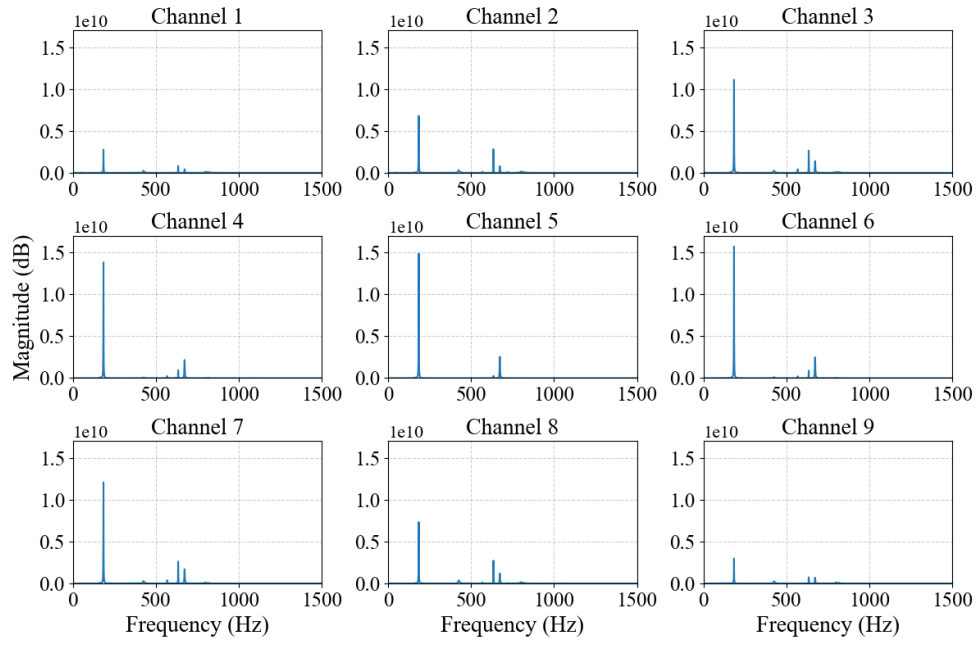


Fig. 5.8 Input data in frequency domain (example of the T-shaped steel beam database)

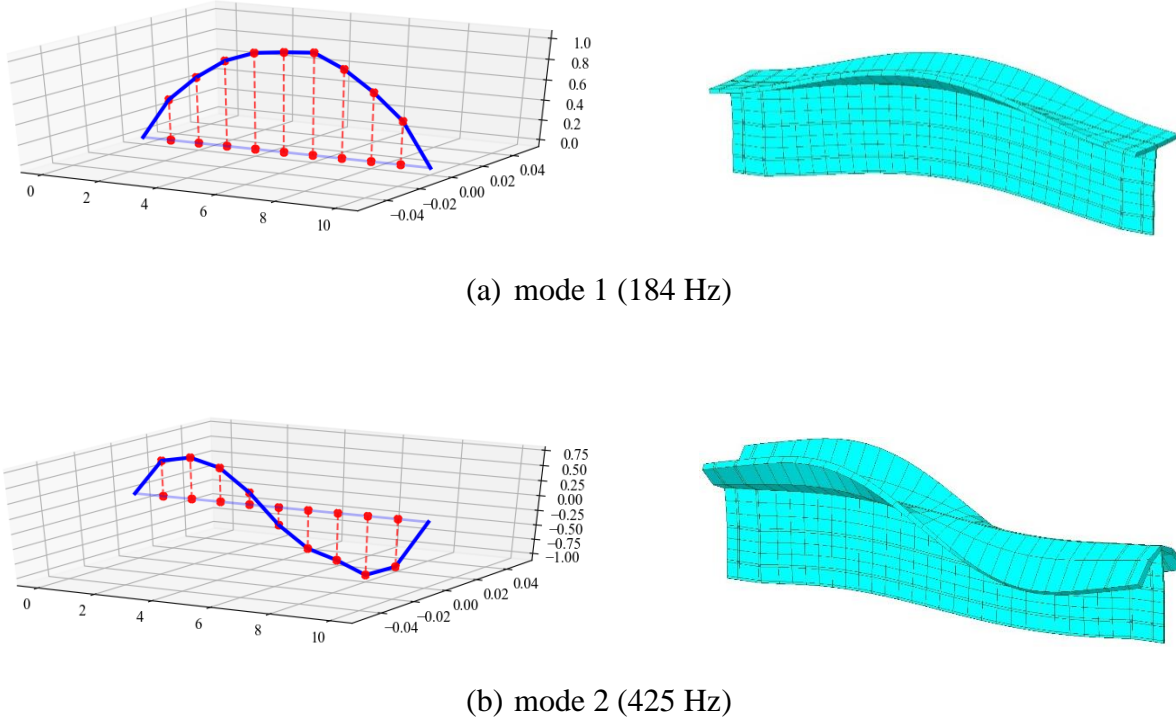


Fig.5.9 Analytical and numerical mode shapes

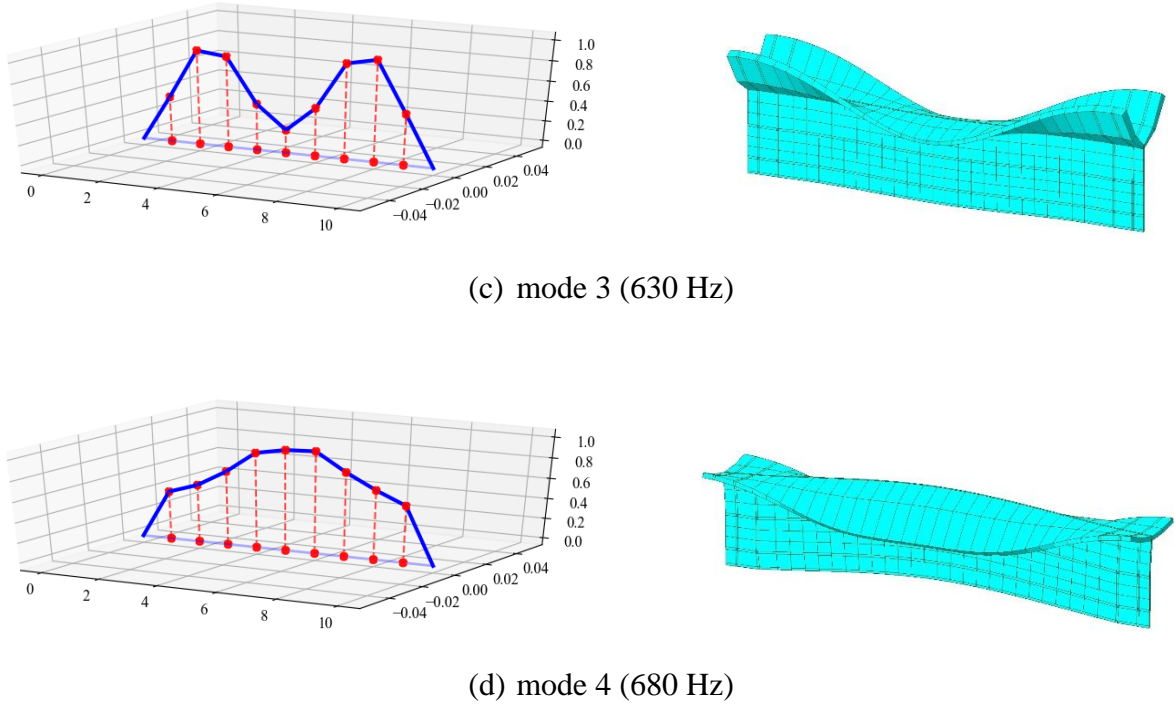


Fig.5.9 Analytical and numerical mode shapes (continue)

Figs. 5.10 and 5.11 show the output of the convolutional layer in the time and frequency domains, respectively. In Fig. 5.10, the envelope of each feature map is a free damped vibrational waveform, implying that the convolutional layer maintains the features of the input data. In Fig. 5.11, the peaks in the power spectral density (PSD) of the convolutional layer output correspond to those of Fig. 5.8 (i.e., 184 Hz, 425 Hz, 630 Hz, and 680 Hz), but the relative amplitudes of the modal peaks are adjusted. In particular, the relative amplitude of the first mode is decreased in Fig. 5.11, while the relative amplitudes of the higher modes are increased. This indicates the presentation of more modes in the output of the convolutional layer. Thus, the convolutional layer works as a band-pass filter in this case.

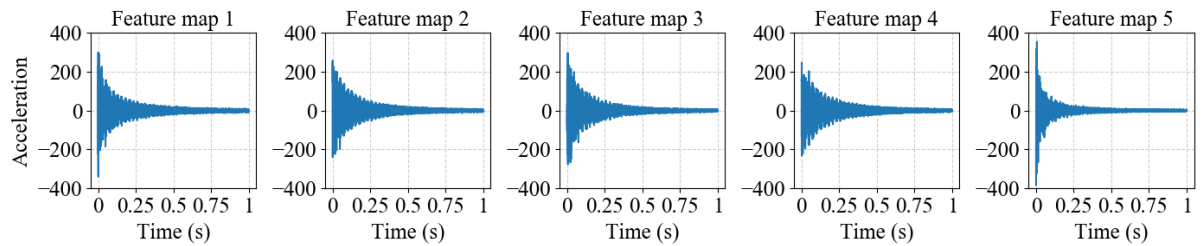


Fig. 5.10 Output of the convolutional layer (example of the T-shaped steel beam database)

As the convolution change the amount of channel of data from 9 to 5, the output of convolutional layers only maintained of mode frequencies and loss the information of mode shapes.

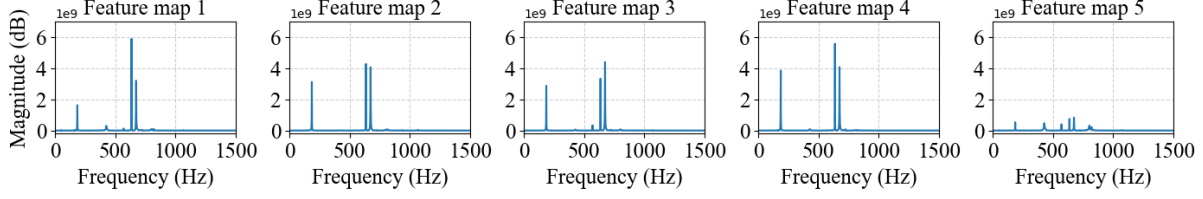


Fig. 5.11 Output of the convolutional layer in frequency domain (example of the T-shaped steel beam database)

Figs. 5.12 and 5.13 show the output of the max pooling layer in the time and frequency domains, respectively. The kernel length of the max pooling was 3, so the mean dynamic sampling of the max pooling layer was set to 3333.33 Hz. The max pooling algorithm filtered many of the acceleration data in the negative range (Fig. 5.12), shrinking the lower parts of the waveforms. However, the PSD of the max pooling layer output (Fig. 5.13) was almost identical in shape to the PSD of the convolutional layer output, indicating that the max pooling operation extracted and maintained the features while reducing the data size.

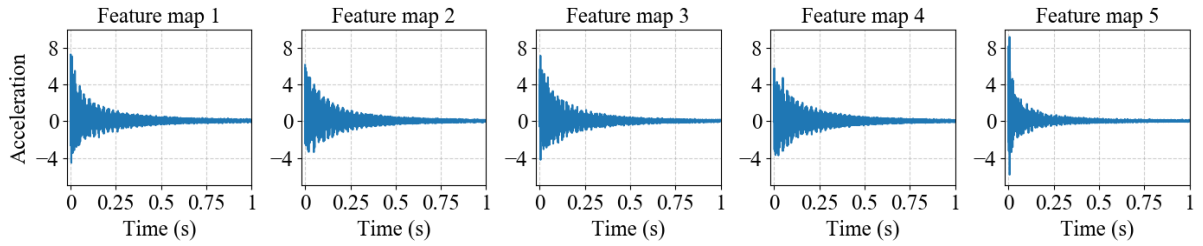


Fig. 5.12 Output of the max pooling layer (example of the T-shaped steel beam database)

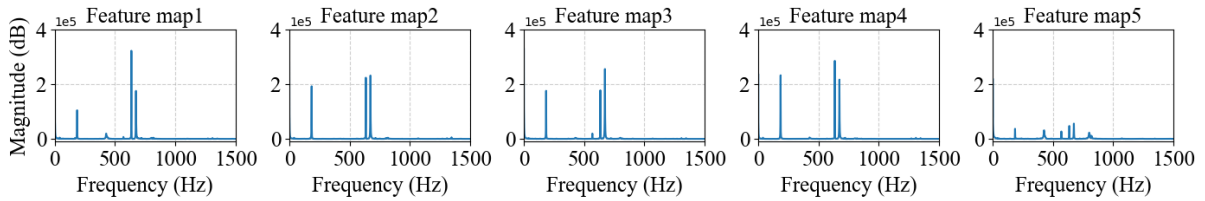


Fig. 5.13 Output of the max pooling layer in frequency domain (example of the T-shaped steel beam database)

Overview the PSD figures of the convolutional layer output and the max pooling layer output as shown in Figs 5.11 and 5.13, since the number of channels is reduced from 9 to 5 in the convolution operation, the information of mode shapes of structure is lost. However, the information of natural frequencies is well maintained by the convolution and max pooling. The reason is that the input data is free damped vibration accelerometer data, whose dynamic characteristics be easily abstracted.

5.3.2 CNN for the short steel girder bridge database

By implementing the same process as introduced in Subsection 5.3.1, the input data, convolutional layer output, and max pooling layer output of the CNN for the short steel girder bridge database are visualized substantially to investigate the process of data flowing.

Firstly, one of input data is chosen randomly for demonstration. The input data is shown in time domain and frequency domain in Figs. 5.14 and 5.15 respectively. As the specimen becomes more complex as shown in Fig. 4.9, the free damped vibration waveforms in Figs 5.14 are not as ideal as the waveforms of the T-shaped beam as shown in Fig. 5.7. There are distinct protrusions or missing parts in the envelopes of the waveforms.

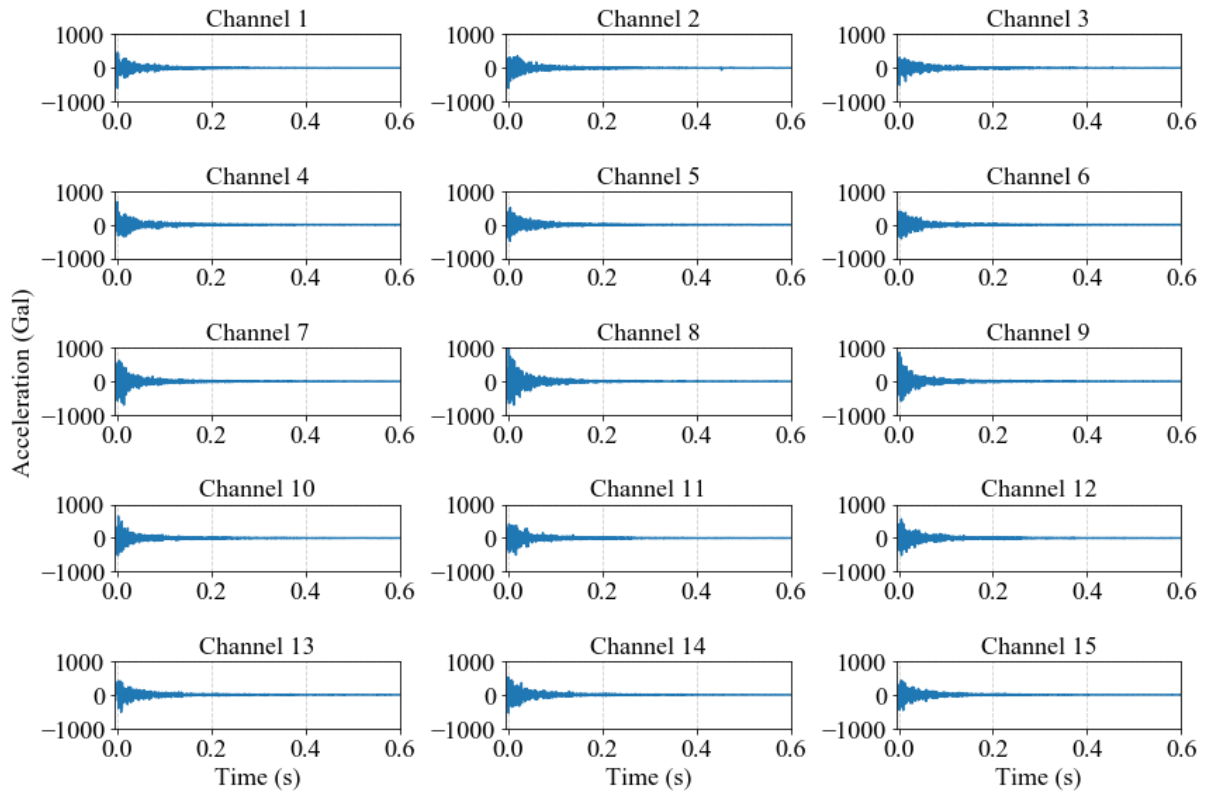


Fig. 5.14 Input data in time domain (example of the short steel girder bridge database)

The PSD figure of each channel of the input data is shown in Fig. 5.15 to identify the vibration modes. Overview the PSD of the 15 channels, there are obvious peaks on 234 Hz, 400 Hz, 500 Hz, 770 Hz, and 1030 Hz. The corresponding mode shapes are also presented in Fig. 5.16. All the identified modes are bending modes.

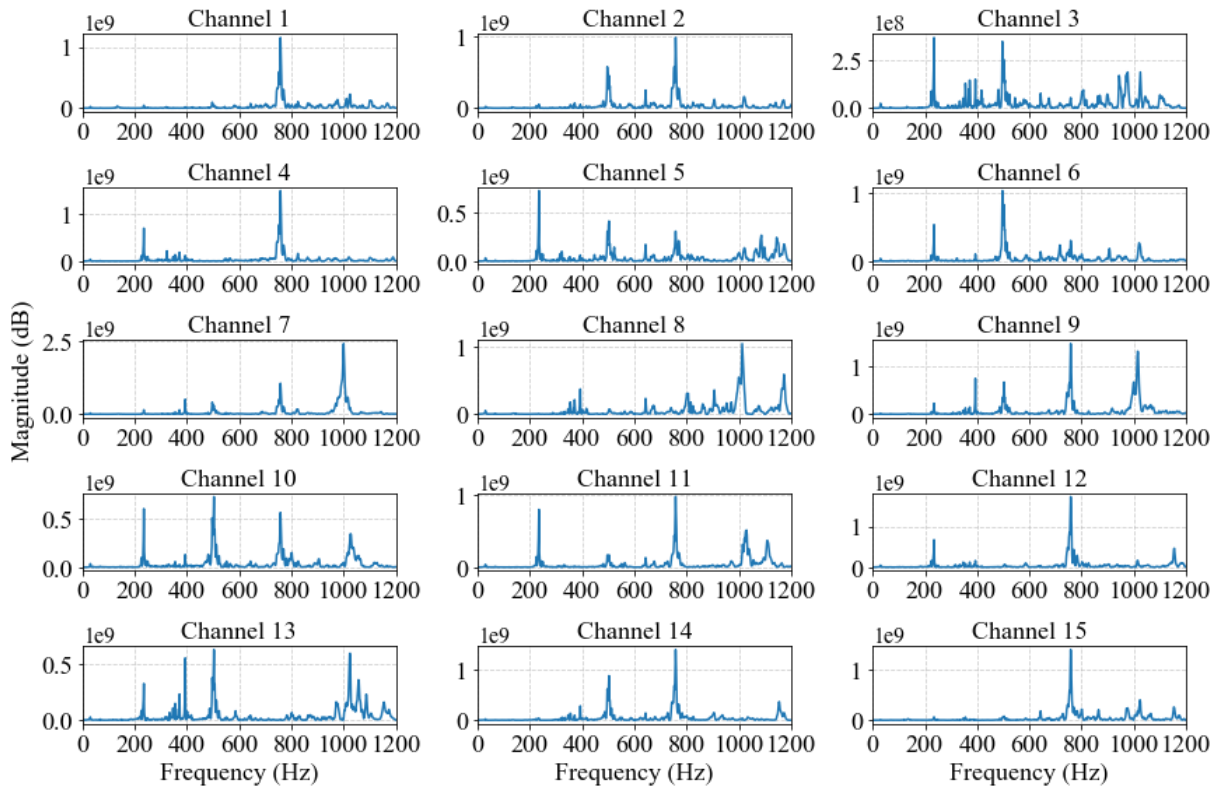


Fig. 5.15 Input data in frequency domain (example of the short steel girder bridge database)

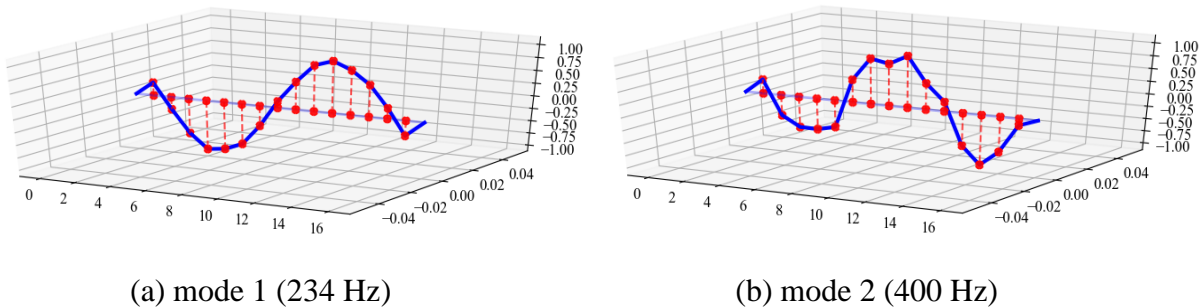
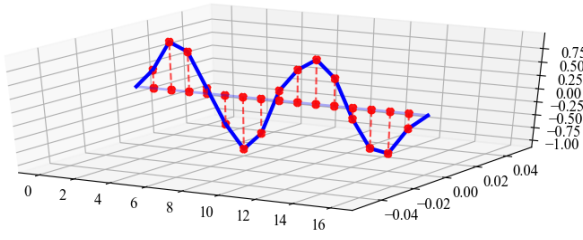
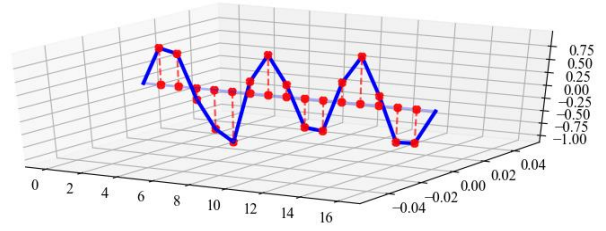


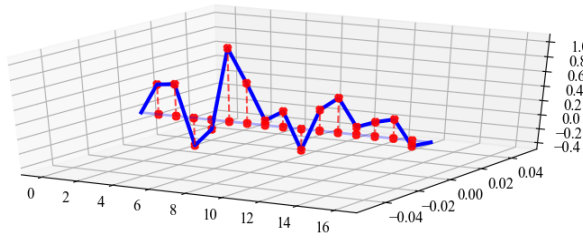
Fig. 5.16 Identified mode shapes



(c) mode 3 (500 Hz)



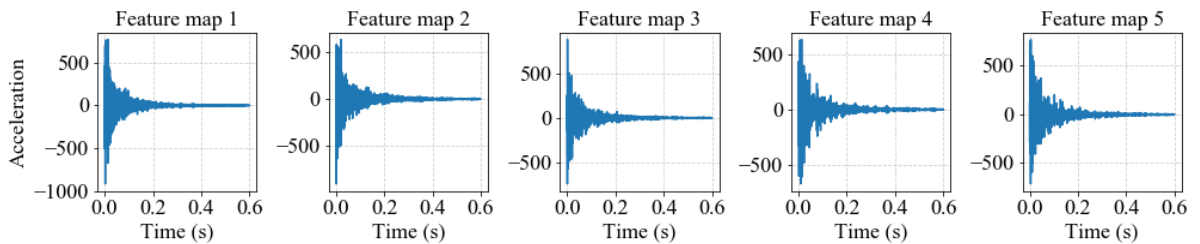
(d) mode 4 (770 Hz)



(e) mode 5 (1030 Hz)

Fig. 5.16 Identified mode shapes (continue)

Secondly, the convolutional layer output is visualized in time domain and frequency domain in Figs 5.17 and 5.18 respectively. Same characteristics to the convolutional layer output of the T-shaped beam case as introduced in Subsection 5.3.1 are presented. The waveforms of the convolutional layer output remain the envelopes of free damped vibration in 5.17. Meanwhile, the features of mode frequencies on 234 Hz, 400 Hz, 500 Hz, 770 Hz, and 1030 Hz are well maintained with adjusted amplitudes. Since the convolution operation reduces the amount of data channel from 15 to 5, the convolution operation can only mountain the features of mode frequencies and losses the information of mode shapes.

**Fig. 5.17** Output of the convolutional layer (example of the short steel girder bridge database)

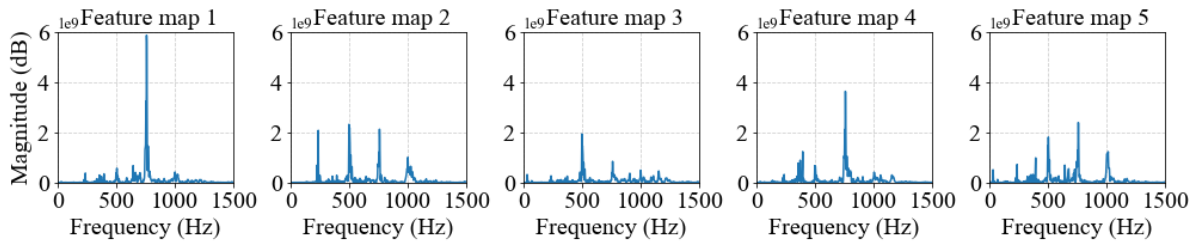


Fig. 5.18 Output of the convolutional layer in frequency domain (example of the short steel girder bridge database)

Thirdly, the output of the max pooling layer is visualized in time domain and frequency domain respectively. The characteristics of the max pooling layer output are also same to the T-shaped beam case as introduced in Subsection 5.3.1. The waveforms of the max pooling output remain the envelopes of free damped vibration with shrunk negative component of the waveforms. Meanwhile, the features of mode frequencies on 234 Hz, 400 Hz, 500 Hz, 770 Hz, and 1030 Hz are well maintained.

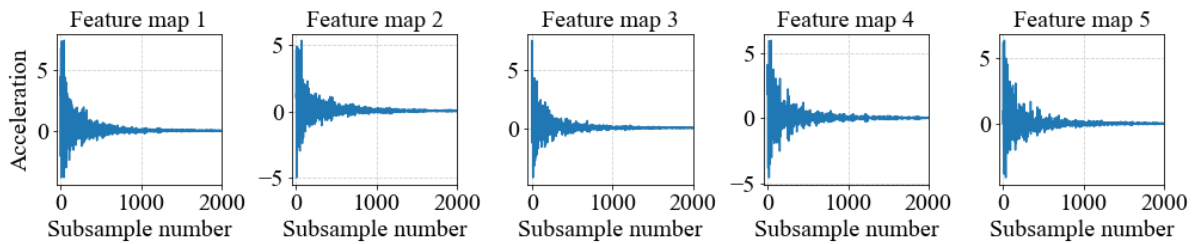


Fig. 5.19 Output of the max pooling layer (example of the short steel girder bridge database)

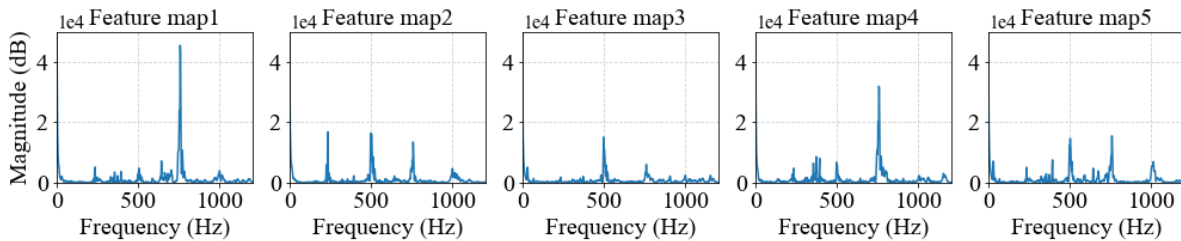


Fig. 5.20 Output of the max pooling layer in frequency domain (example of the short steel girder bridge database)

The visualizations of the layer outputs of the short steel girder bridge case show same characteristics to the case of the T-shaped beam, since both database consist only free damped vibration data samples, which contains much features of the structure.

5.3.3 CNN for the long steel girder bridge database

Data in the long steel girder bridge flowing in the CNN is also traced and visualized. Since the raw data is random vibration data excited by white noise, the raw data shows obvious differences to the cases of the T-shaped beam and the short steel girder bridge databases.

Fig. 5.21 shows an input datum, and the corresponding PSD is shown in Fig. 5.22. Since the data is random vibration, the response of the bridge is very complicated. Unlike the cases of the T-shaped beam and the short steel girder bridge in Figs. 5.8 and 5.15, there is no common peaks in the PSD of all the channel in Fig. 5.22. In each channel of the PSD figure, the amplitude of each frequency component are distinct, which represents the features of random excitation.

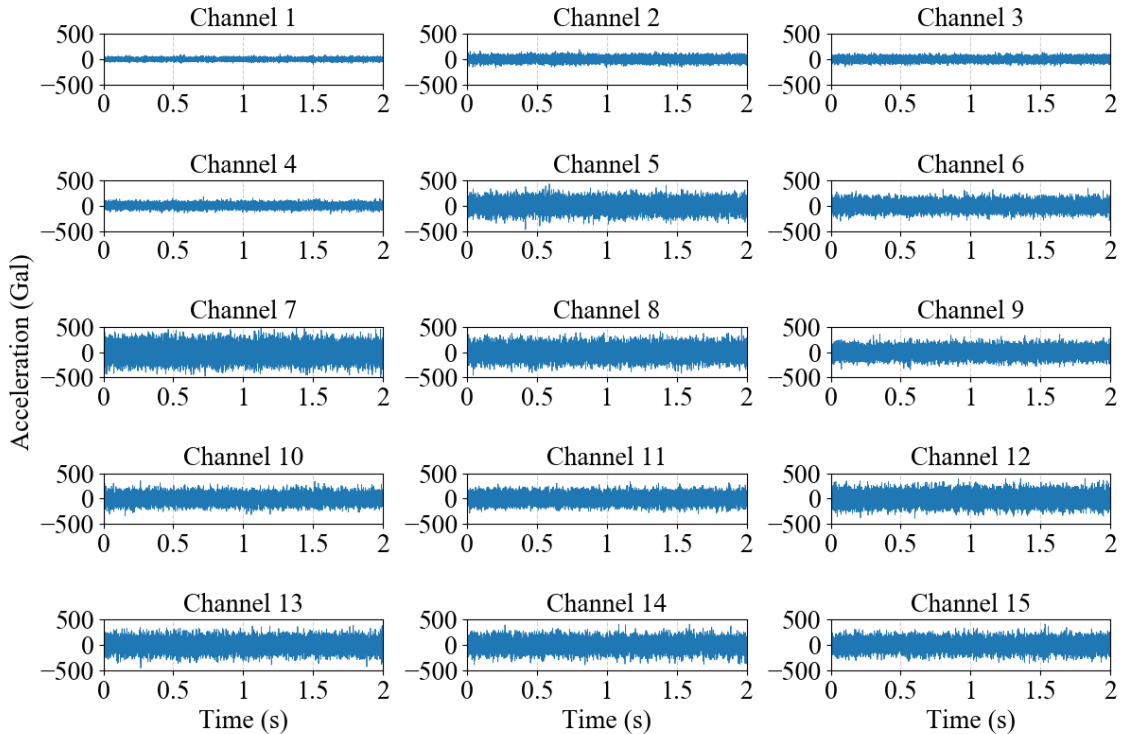
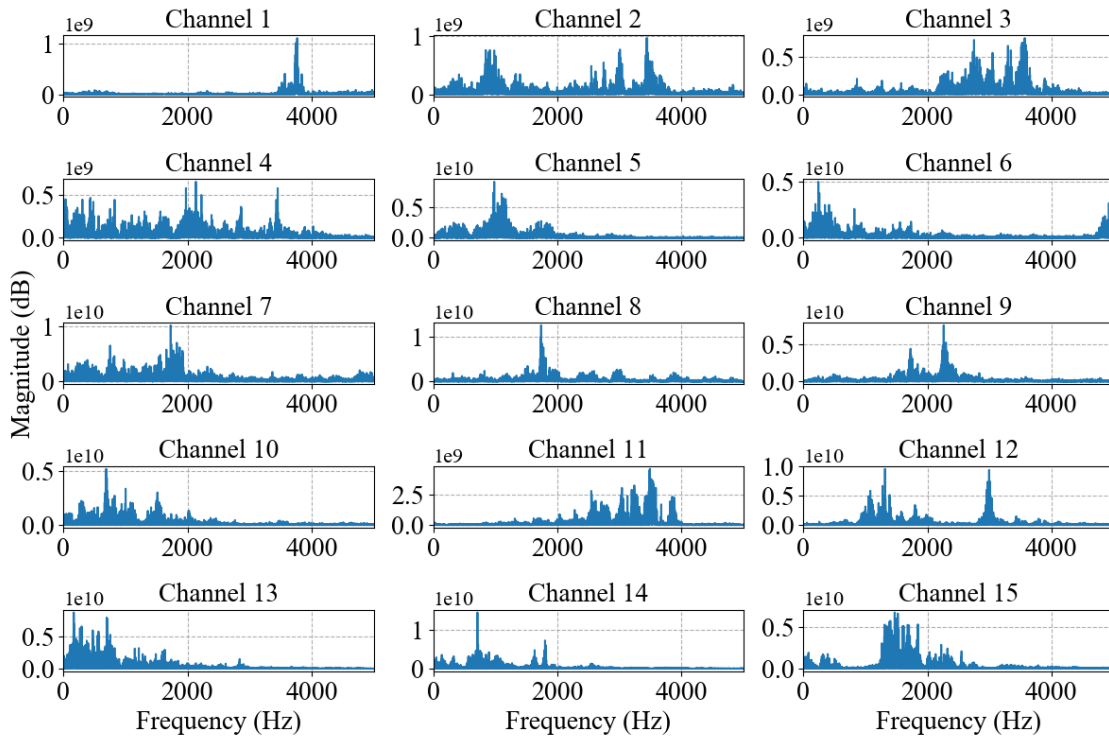
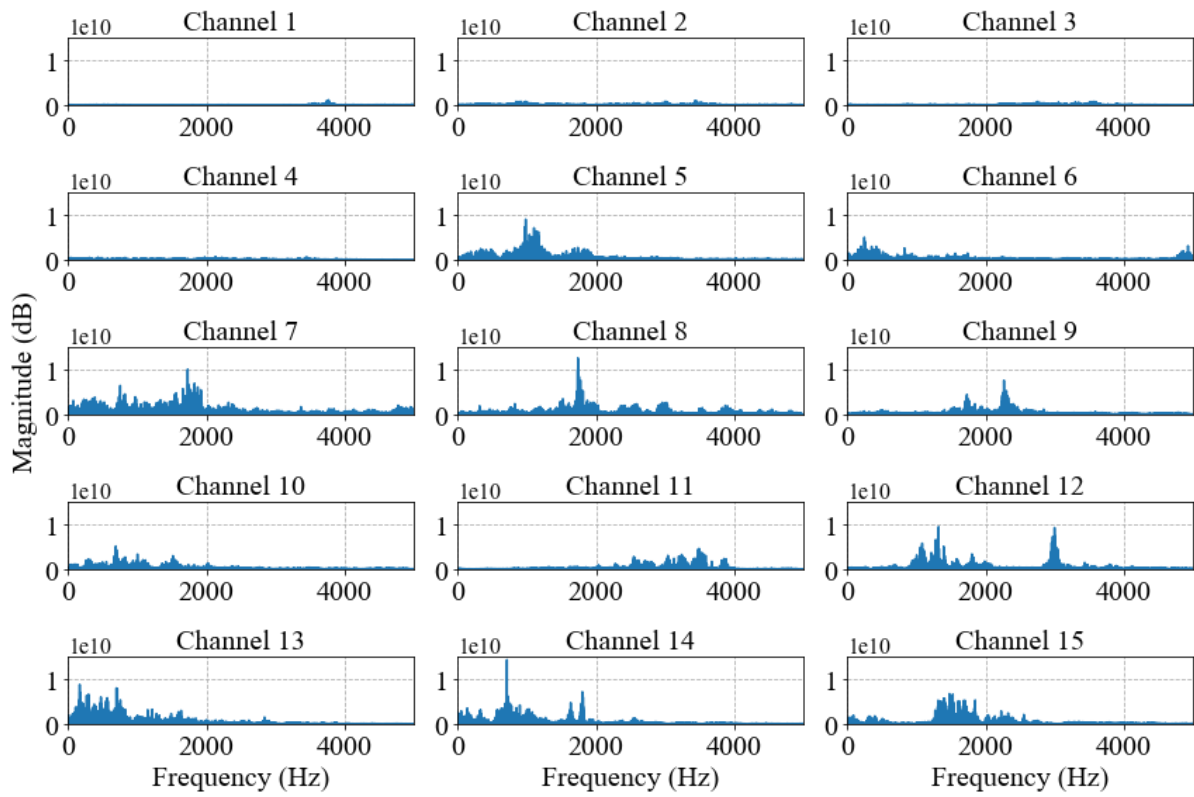


Fig. 5.21 Input data in time domain (example of the long steel girder bridge database)



(a) individual range of y axis for easy reading



(b) uniformed range of y axis for easy comparison

Fig. 5.22 Input data in frequency domain (example of the long steel girder bridge database)

Subsequently, the output of convolutional layer is visualized in time domain and frequency domain in Figs 5.23 and 5.24. The data maintains the envelopes of waveforms of random vibration in Fig. 5.23. However, the PSD plots in Fig 5.24 cannot maintain the information of frequency component of the input data very well. For example, there is an obvious peak near 3000 Hz in Channels 12 in Fig. 5.22, but in Fig. 5.24 the amplitude of the peak near 3000 Hz is very small. The phenomenon means that when using random vibration data to train a CNN, the convolution operation will discard some information. One possible explanation could be that CNN must find the most effective information from the random vibration data during training, thus CNN pays more attention to those features that have a great impact on the prediction results, and ignores the features that have little impact on the prediction results. To make easy comparison to the PSD plots of the max pooling layer output, the x axis of Fig. 5.24 is limited in the range from 0 Hz to 1500 Hz as shown in Fig. 5.25.

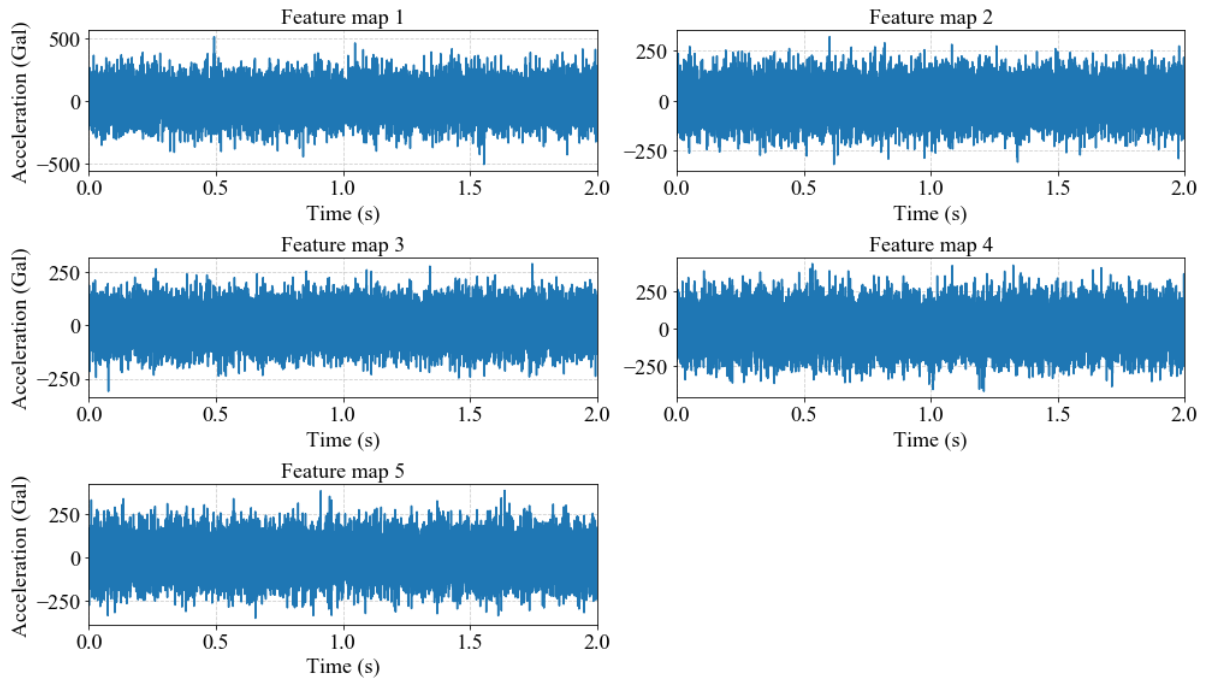


Fig. 5.23 convolutional layer output in time domain (example of the long steel girder bridge database)

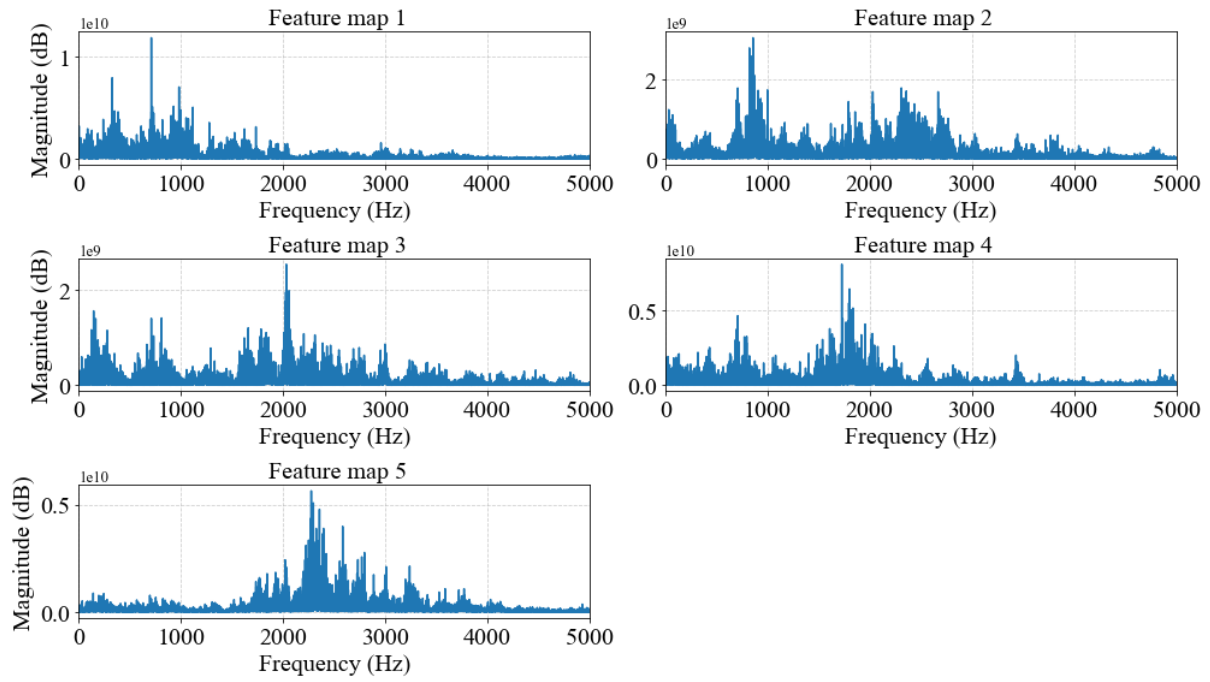


Fig. 5.24 convolutional layer output in frequency domain (example of the long steel girder bridge database)

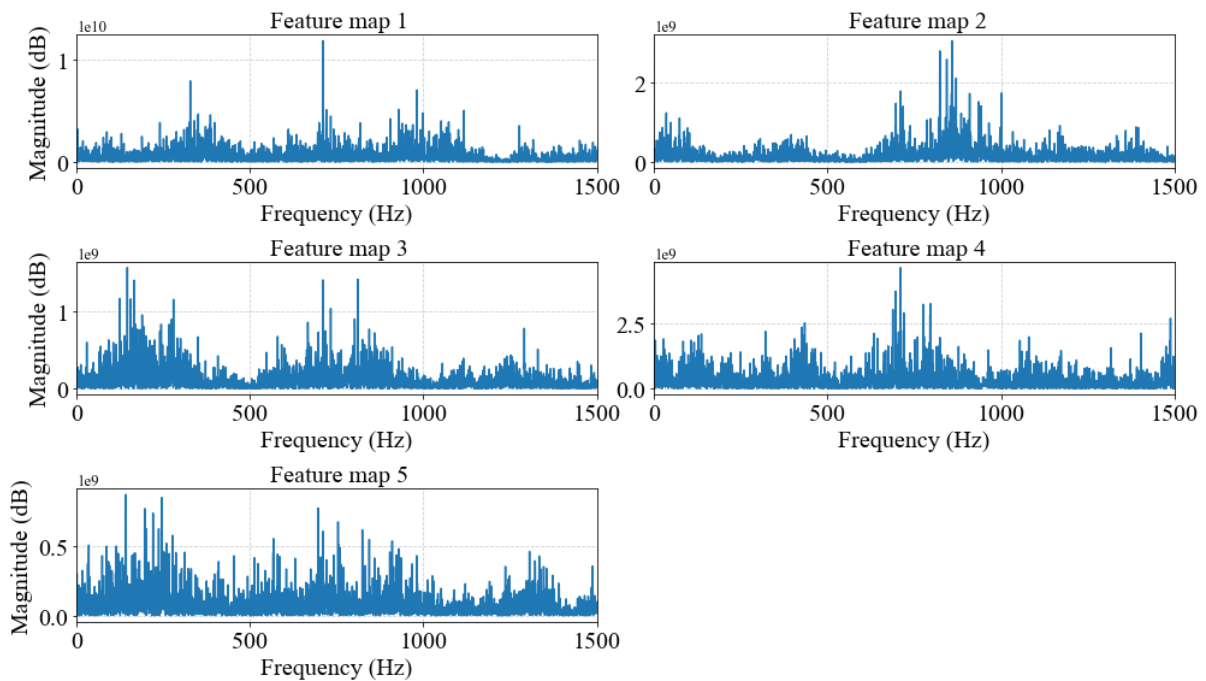


Fig. 5.25 convolutional layer output in frequency domain in range of 0 Hz -1500 Hz (example of the long steel girder bridge database)

Finally, the output of the max pooling layer is visualized in both time domain and frequency domain in Figs. 5.26 and 5.27. By the max pooling operation, the negative components of the data are slightly shrunk as shown in Fig. 5.26 when comparing to the output of convolutional layer output presented in Fig. 5.23. In frequency domain of the output of the max pooling layer as shown in Fig. 5.27, most features are also remained after the pooling operation. For example, the peaks in approximate 700 Hz in Channels 1-4 are maintained from the output of the convolutional layer.

In summary, the process of random vibration data flowing through the CNN for the long steel girder bridge are generally same to the cases of the T-shaped steel beam and the short steel girder bridge which use free damped vibration data as input data. Convolution and max pooling operations reduce the dimension of data and abstract features simultaneously. The only difference is that when using random vibration data as input or training data, more frequency components are ignored since the CNN must determine the most effective features from the complex raw data.

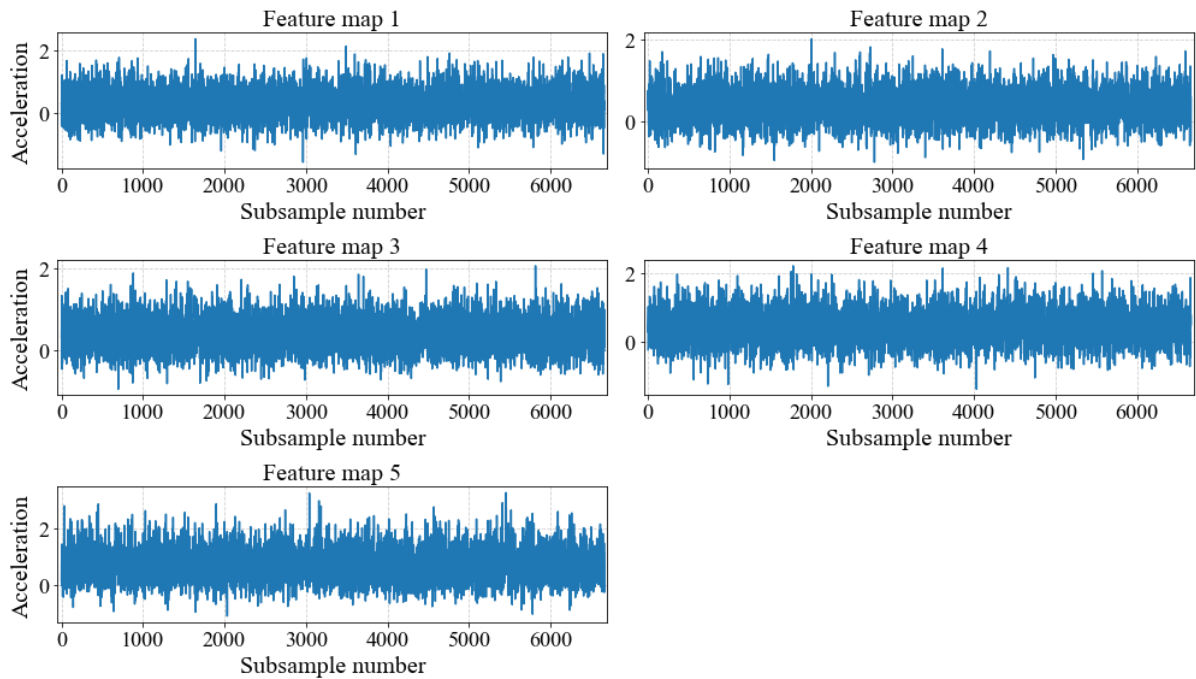


Fig. 5.26 Max pooling layer output in time domain (example of the long steel girder bridge database)

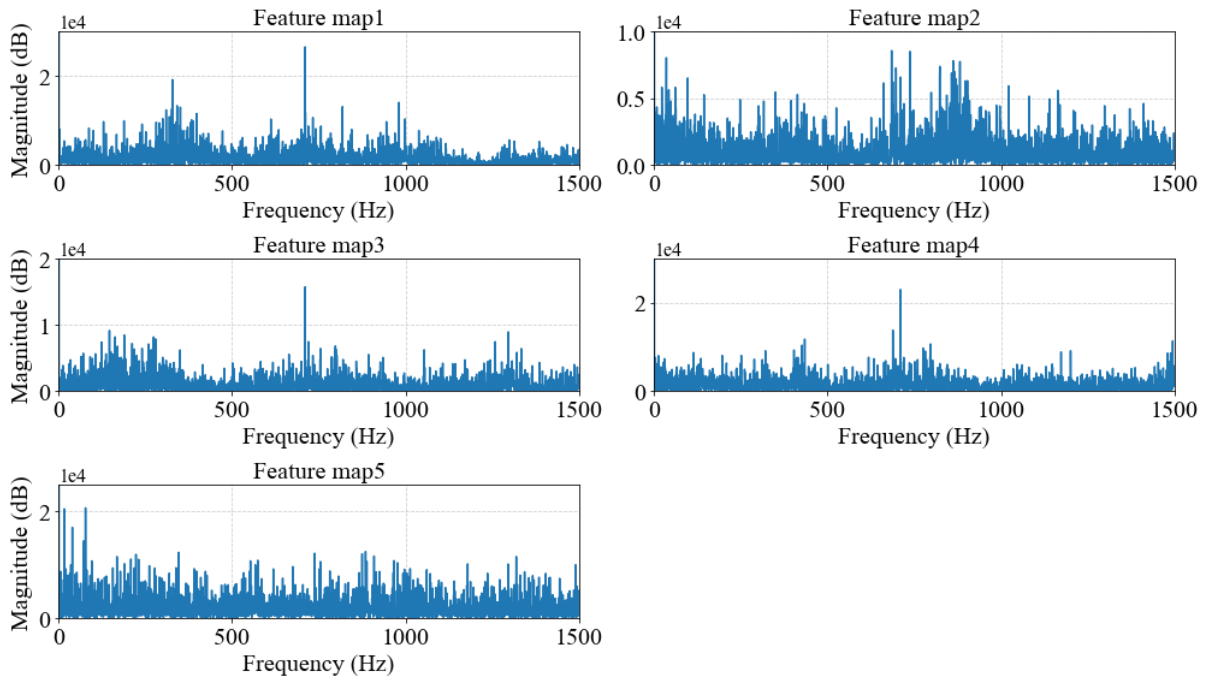


Fig. 5.27 Max pooling layer output in time domain (example of the long steel girder bridge database)

5.4 Variation of Data Structure in Each Layer

To understand the effect of each layer from the view of data structure, changes of data structure flowing through the CNN are investigated by visualizing the data structure variation of the CNNs for the three databases.

5.4.1 Data structure visualization of the CNN for the T-shaped beam

Data structures of the Fold 1 validation data of the T-shaped beam database from the input to the output via each hidden layer of the CNN are visualized in Figs. 5.28-5.33 by the T-SNE algorithm (introduced in Section 3.8) with same learning rates of 400.

The T-SNE algorithm is a nonlinear data dimensionality reduction algorithm which preserve local structure of the database and projecting the data in a new low-dimensional space which is much easier to see the clusters of same categories of data. In this study, the raw vibration data which are in 90000-dimensional space remain close in the new 2-dimensional space. As the projecting is highly nonlinear, the 2 dimensions can be seen as 2 highly nonlinear feature sets which cannot be physically explained.

In Figs. 5.28-5.33, each scatter plots a datum in the validation set of the T-shaped beam database. Different colours of the scatters indicate different structural states of data. The distance between two scatters means the correlation or similarity of the two data. Close distance between the scatters shows high correlation or similarity of the data.

Fig. 5.28 shows the data structure of the raw vibration data. The distribution of data in each structural state is highly discrete and irregular. Clear big clusters of a certain category of data of all the structural states are not formed.

Fig. 5.29 shows the data structure of the convolutional layer output. The clusters of each structural state starts to gather. Comparing to the data structure of input data as shown in Fig. 5.10, the obvious gap in Fig. 5.10 has been fulfilled.

Fig. 5.30 is the data structure of the batch normalization layer output. The clusters of each structural state gather obviously. The overall data distribution is more compact and dense than the outputs of previous layers.

Fig. 5.31 is the data structure of the max pooling layer outputs. The data distribution begins to change to discrete clusters. The clusters of each structural state begin to form.

Fig. 5.32 is the data structure of the FC layer. Obvious isolated clusters of different structural state are visible, such as data in States 0, 1, 2, 4, 5, and 8.

Fig. 5.33 is the data structure of the network outputs. Clusters of each structural states are classified with a shape of willow leaf. There is clear boundary between all the clusters.

Overview Figs. 5.28-5.33, for one thing, the first several layers (convolutional layer, batch normalization layer, and max pooling layer) only change the data structure slightly while operating. The data structure after the fully connected layer changes obviously. To some extent, this phenomenon supports why some people use the network the before FC layer as feature extractor, and the network after the FC layer as classifier. For another, Figs. 5.28-5.33 are evidences to show that the CNN can clearly define the boundary between each structural state and classify the structural state correctly, indicating good capacity of structural state identification of the CNN.

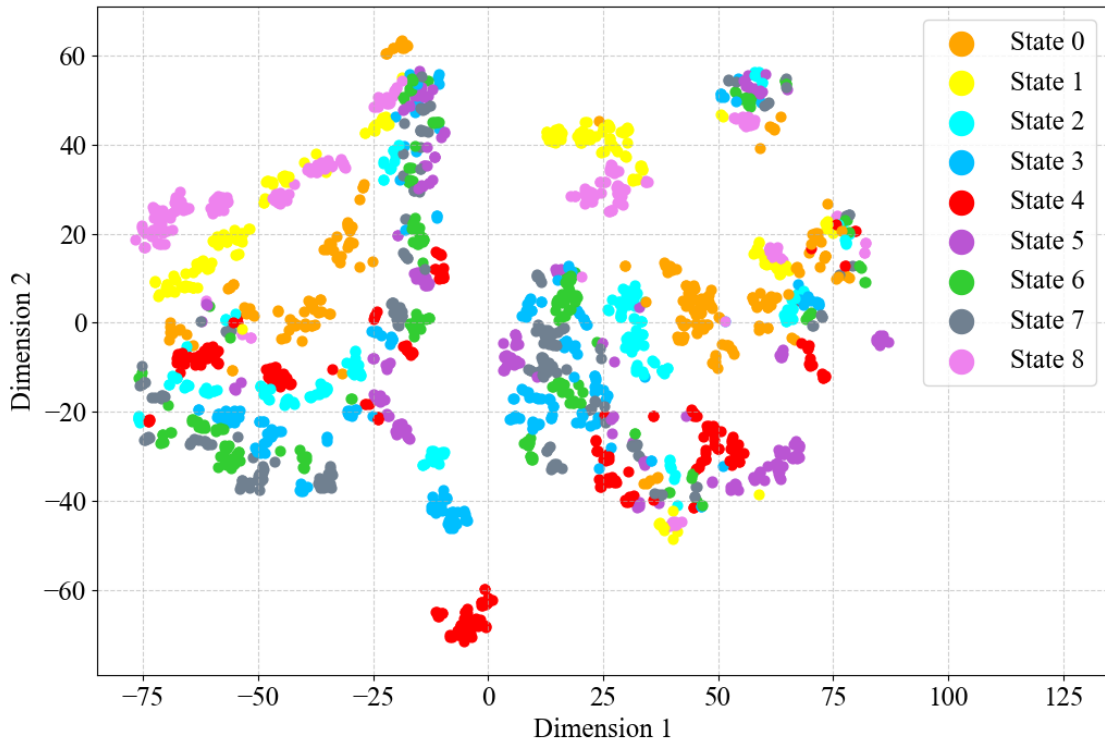


Fig. 5.28 Data structure of the input data (case of the T-shaped steel beam)

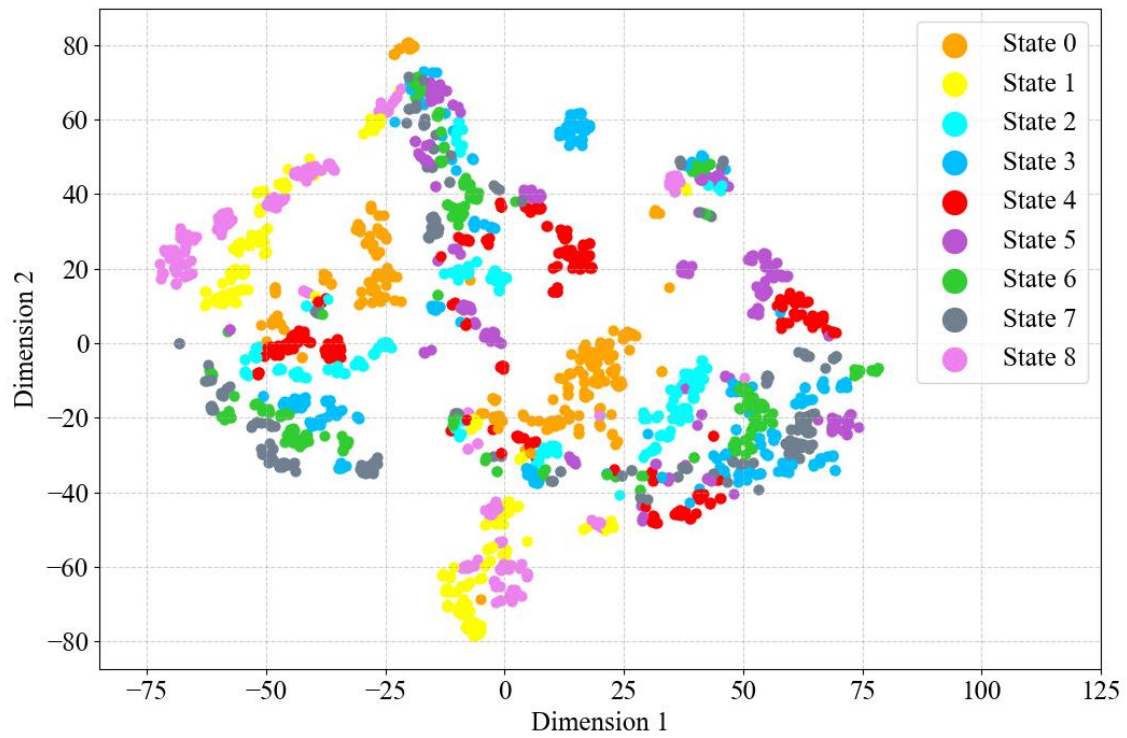


Fig. 5.29 Data structure of the convolutional layer outputs (case of the T-shaped steel beam)

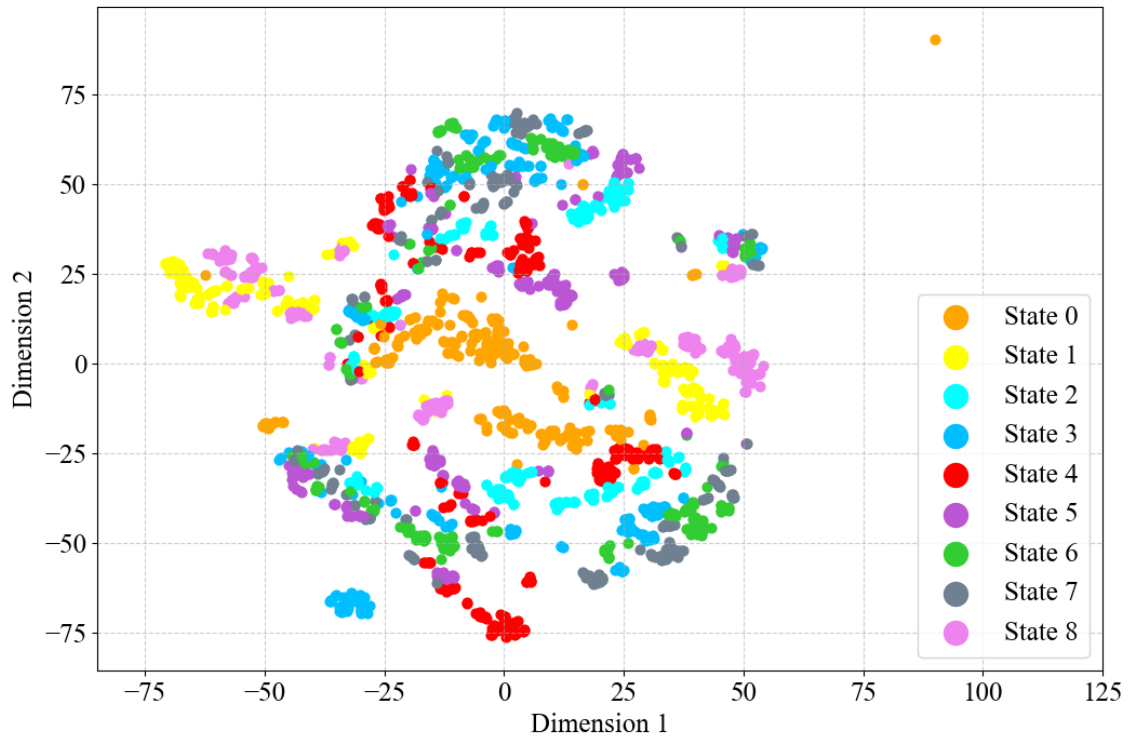


Fig. 5.30 Data structure of the batch normalization layer outputs (case of the T-shaped steel beam)

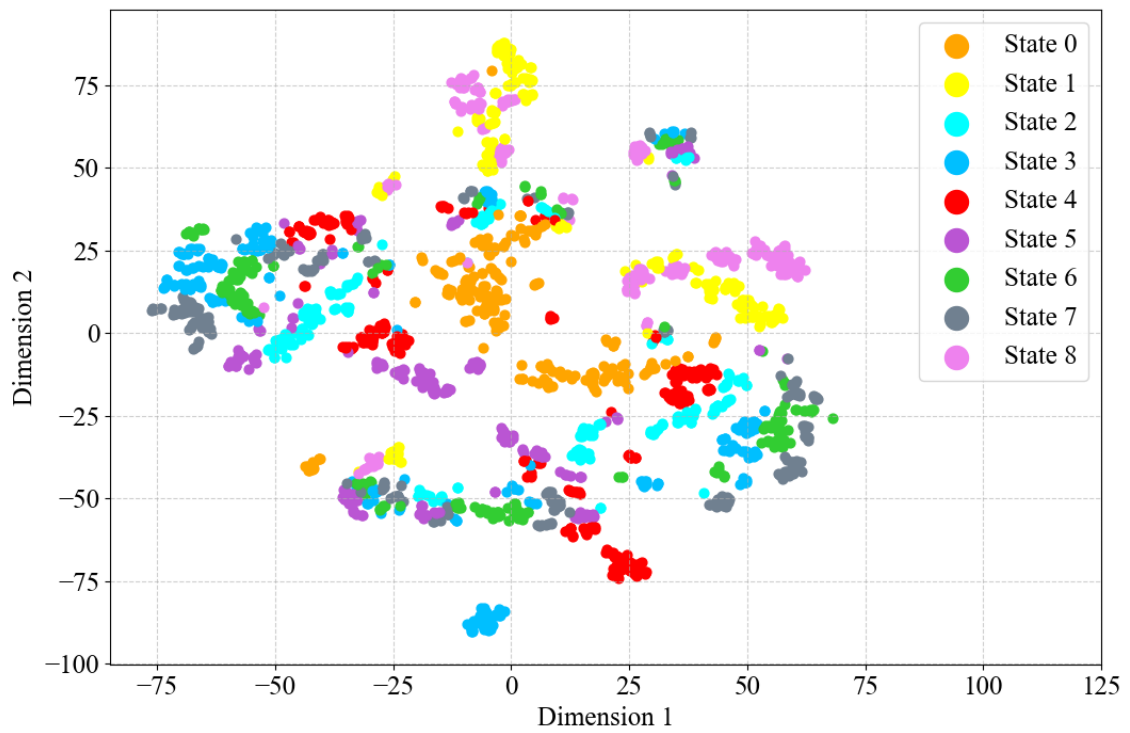


Fig. 5.31 Data structure of the max pooling layer outputs (case of the T-shaped steel beam)

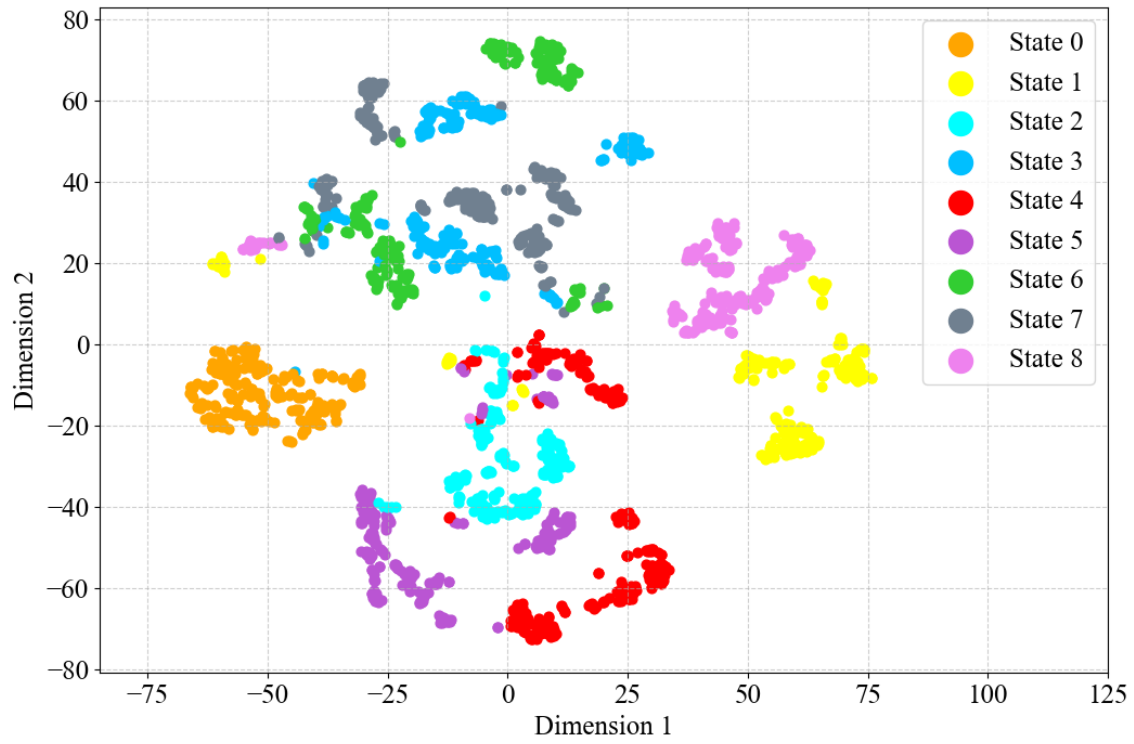


Fig. 5.32 Data structure of the FC layer outputs (case of the T-shaped steel beam)

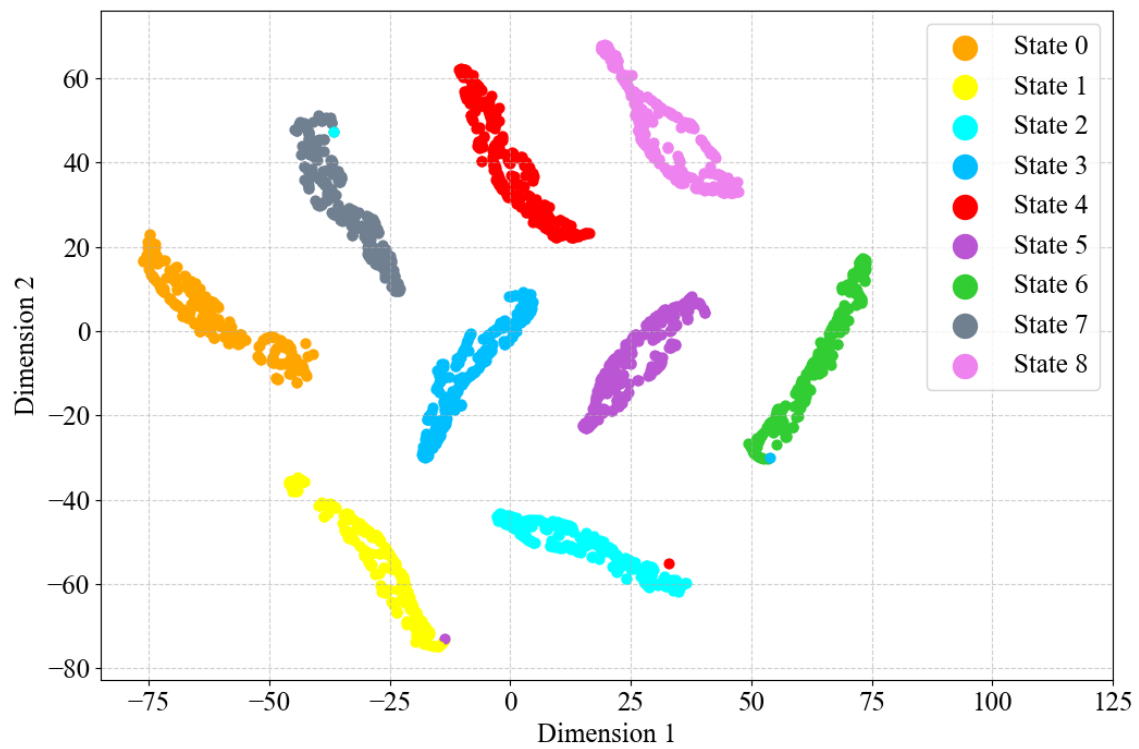


Fig. 5.33 Data structure of the network outputs (case of the T-shaped steel beam)

5.4.2 Data structure visualization of the CNN for the short steel girder bridge

The variation of data structure of the CNN model of the short steel girder bridge from the input to the output through the convolutional layer, batch normalization layer, max pooling layer FC layer, is visualized in Figs. 5.34-5.39 by the T-SNE method.

Overview the process of data structure variation as shown in Figs. 5.34-5.39, same phenomenon to the case of the T-shaped beam as introduced in Subsection 5.4.1 is presented, that in the first several layers (convolutional layer, batch normalization layer, and max pooling layer), the data structure only changes slightly comparing to the data structure of the input data. The dramatic data structure variation begins in the FC layer, and almost perfect clusters are obtained in the output layers, which shows the good performance of the CNN.

There is another interesting phenomenon that some data points are far away from the main group in the input, convolutional layer, and batch normalization layer. The reason should be the structural complexity of the short steel girder bridge is much higher than the T-shaped steel beam. Thus, there are more uncertainties of the structural response. Then after the max pooling layer those data points approach the main group, which demonstrates the good performance of the CNN to process the free damped vibration data acquired on a more complex structure.

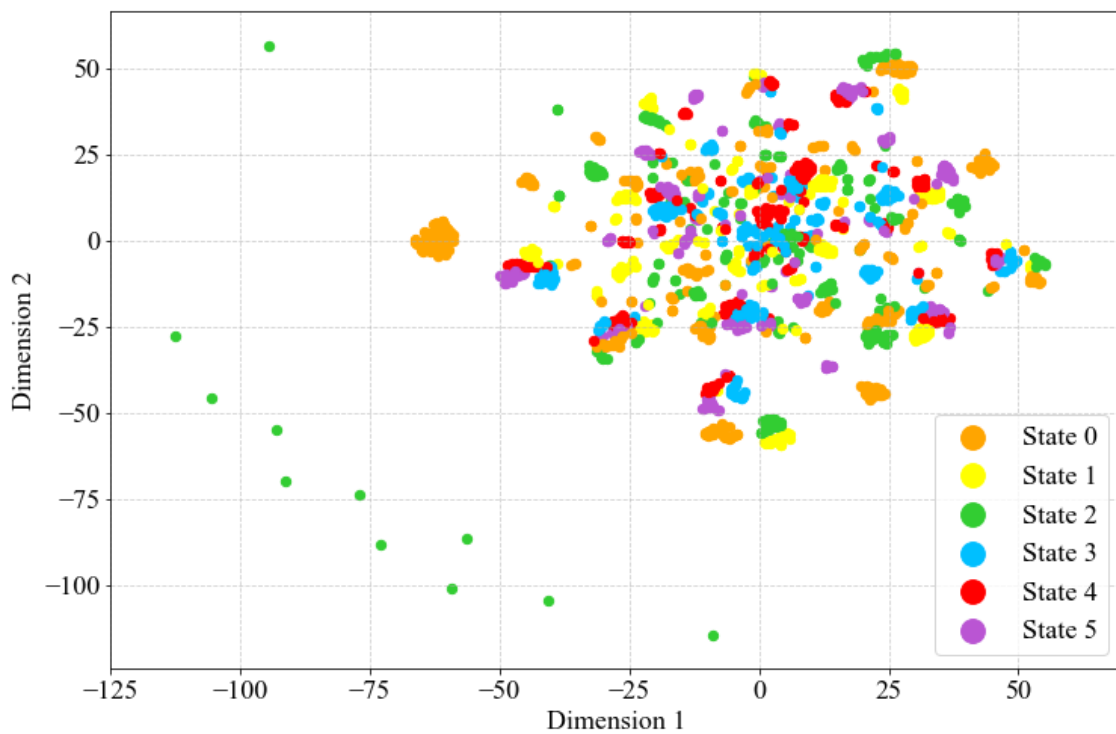


Fig. 5.34 Data structure of the input data (case of the short steel girder bridge)

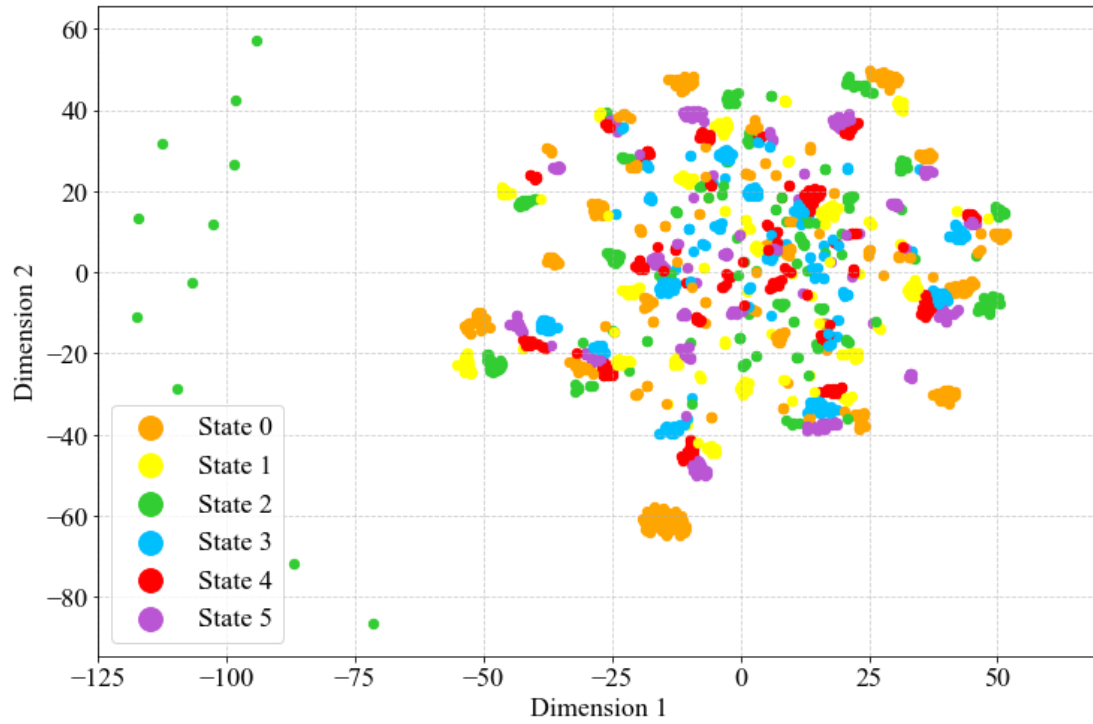


Fig. 5.35 Data structure of the convolutional layer outputs (case of the short steel girder bridge)

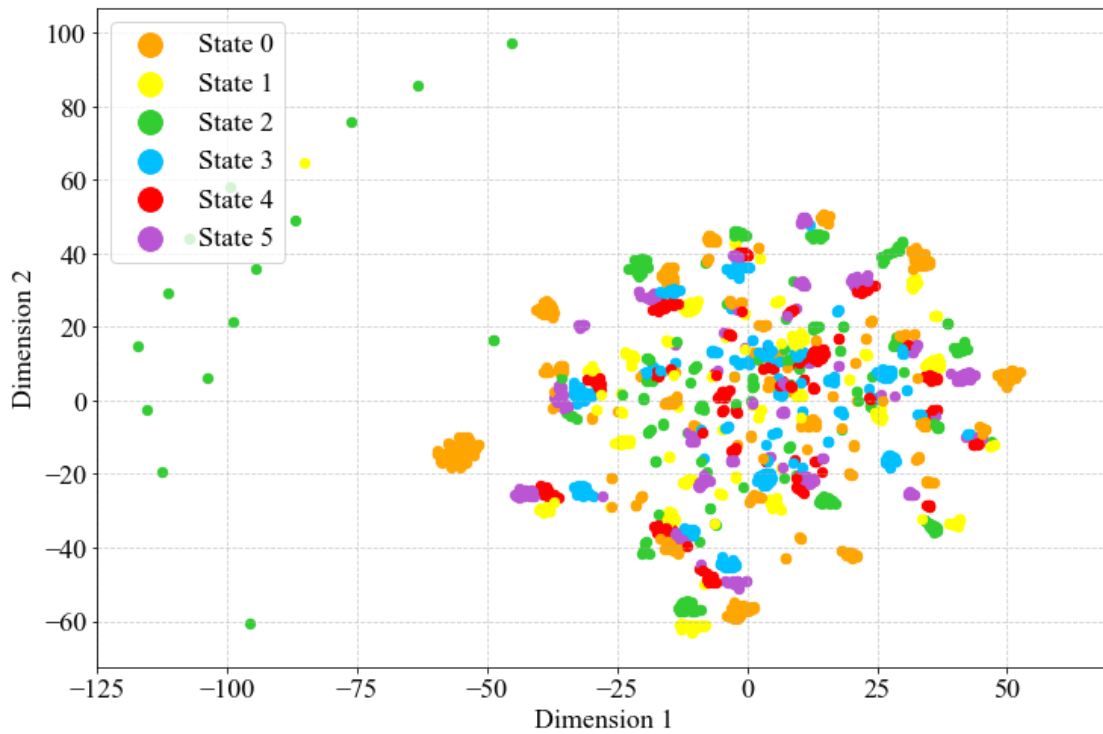


Fig. 5.36 Data structure of the batch normalization layer outputs (case of the short steel girder bridge)

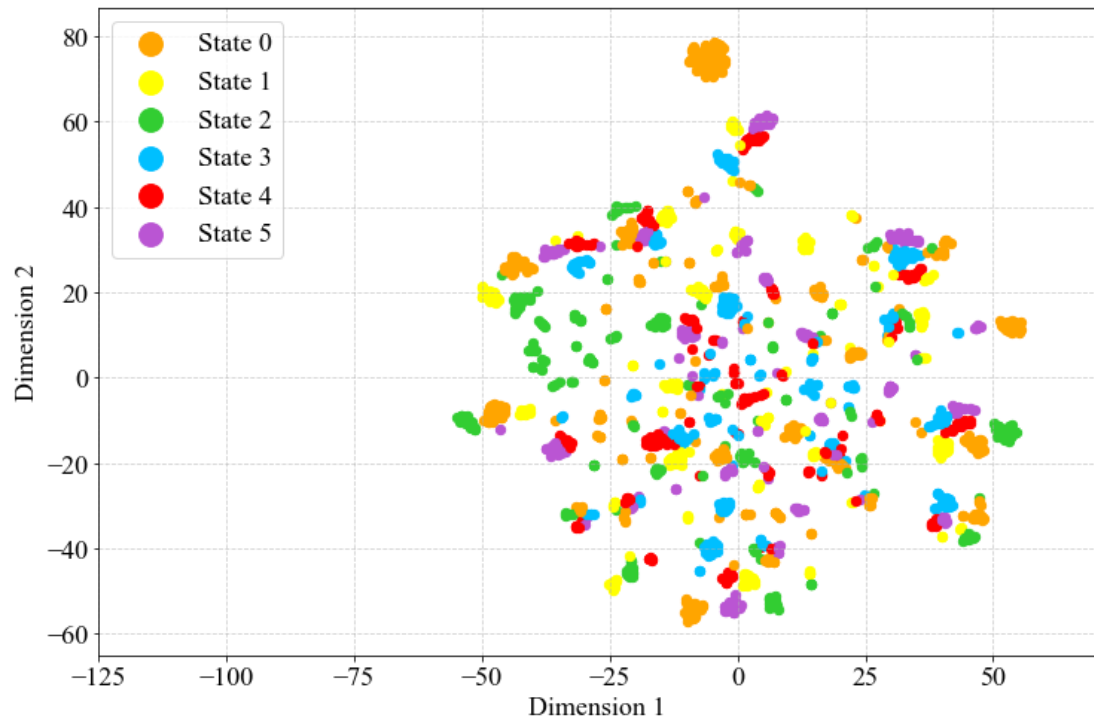


Fig. 5.37 Data structure of the max pooling layer outputs (case of the short steel girder bridge)

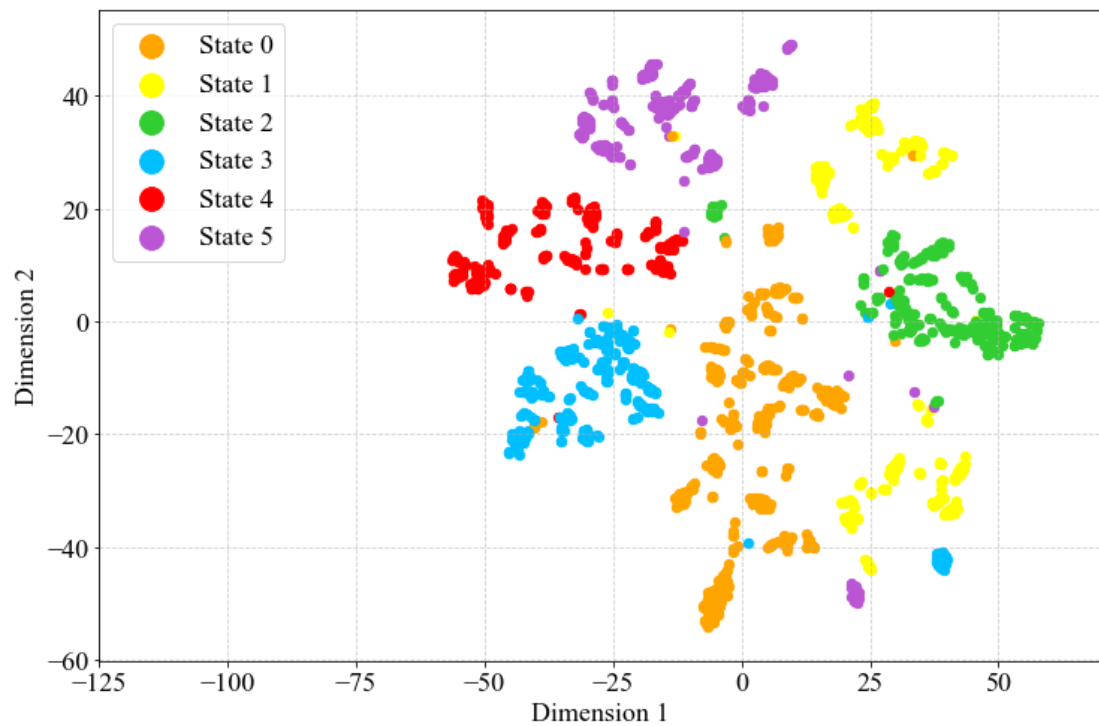


Fig. 5.38 Data structure of the FC layer outputs (case of the short steel girder bridge)

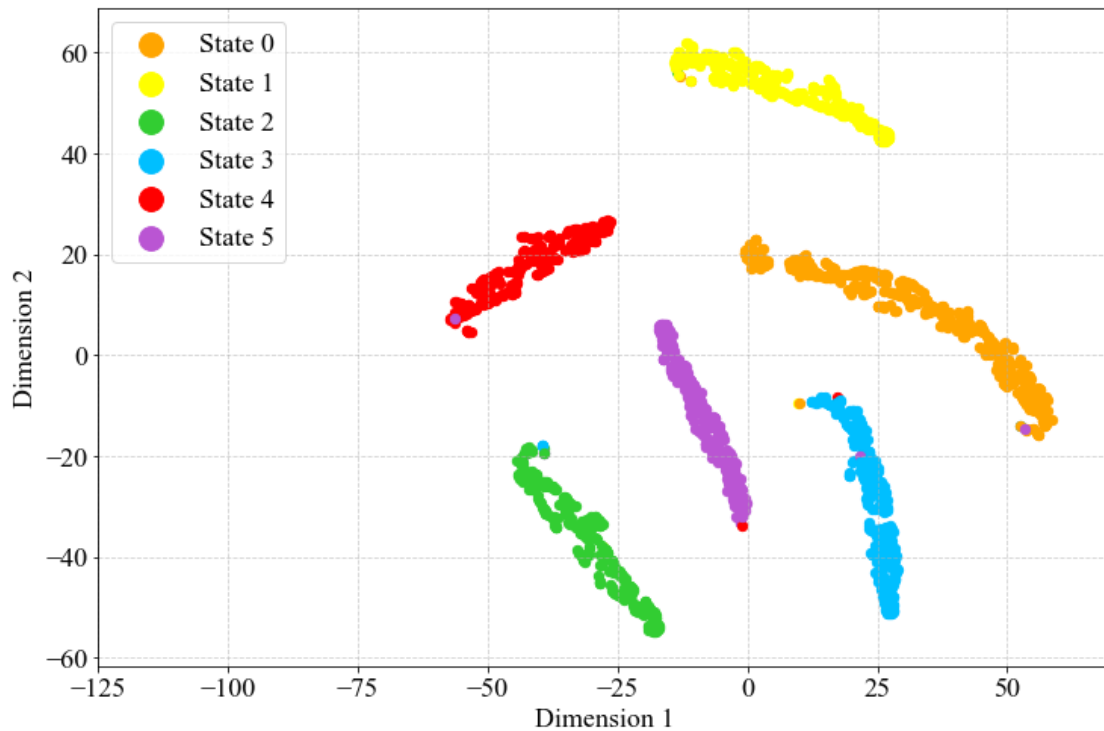


Fig. 5.39 Data structure of the network outputs (case of the short steel girder bridge)

5.4.3 Data structure visualization of the CNN model for long steel girder bridge

The process of data structure variation of the CNN for the long steel girder bridge is also visualized by the T-SNE method as shown in Figs. 5.40-5.45.

Since the input data is the bridge response of the white noise excitation and traffic load with a large data shape of 20000×15 , the complexity of the input data is much higher than the input data of the T-shaped beam database and the short steel girder bridge database which only consist of free damped vibration data. The data structures of the first several layers (input, convolutional layer, batch normalization layer, and the max pooling layer) are very complex as shown in Figs. 5.40-5.43. Highly discrete data points surround a core of data points. There is no clear small cluster of a certain category of data. The data structure changes dramatically in the FC layer. Clear clusters can be formed in Fig. 5.44. Some data of State 0 are wrongly clustered with data of State 1, the reason should be that the additional mass of State 1 is very close to the support and has very small effect to the structural state. Finally, the data structure of the output layer shows better clusters than the FC layer, even though there are overlapped area between the State 0 and State 1. The process of data structure variation also shows the CNN is also capable to process the complex random vibration data.

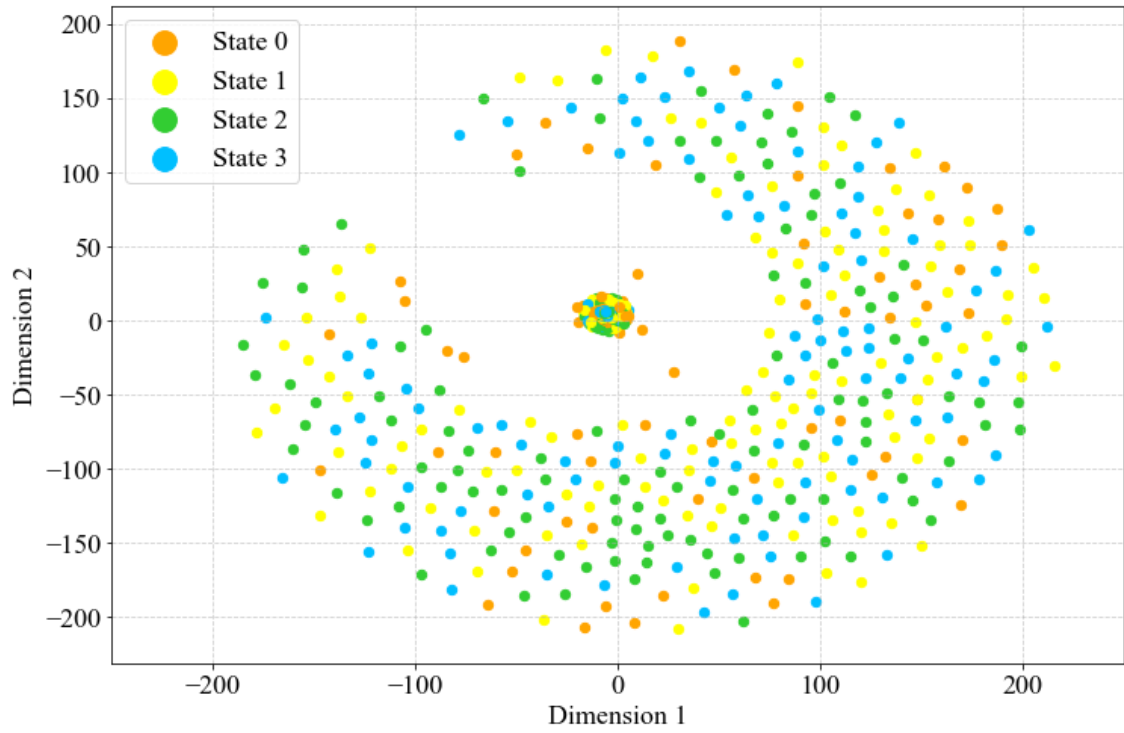


Fig. 5.40 Data structure of the input data (case of the long steel girder bridge)

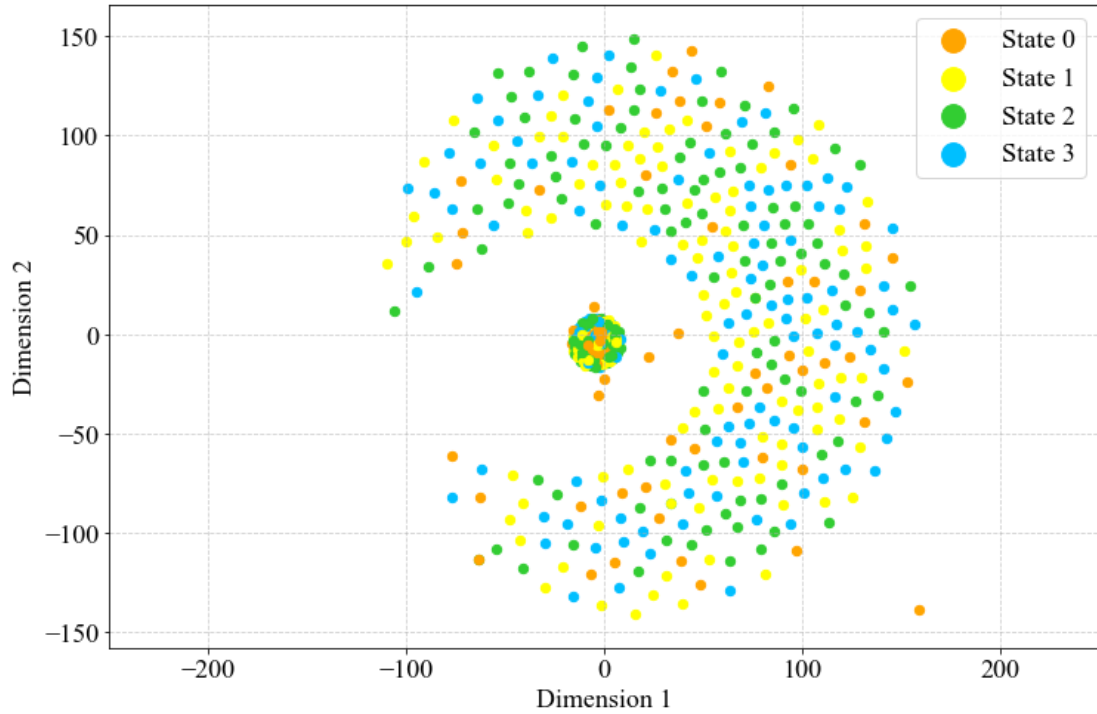


Fig. 5.41 Data structure of the convolutional layer outputs (case of the long steel girder bridge)

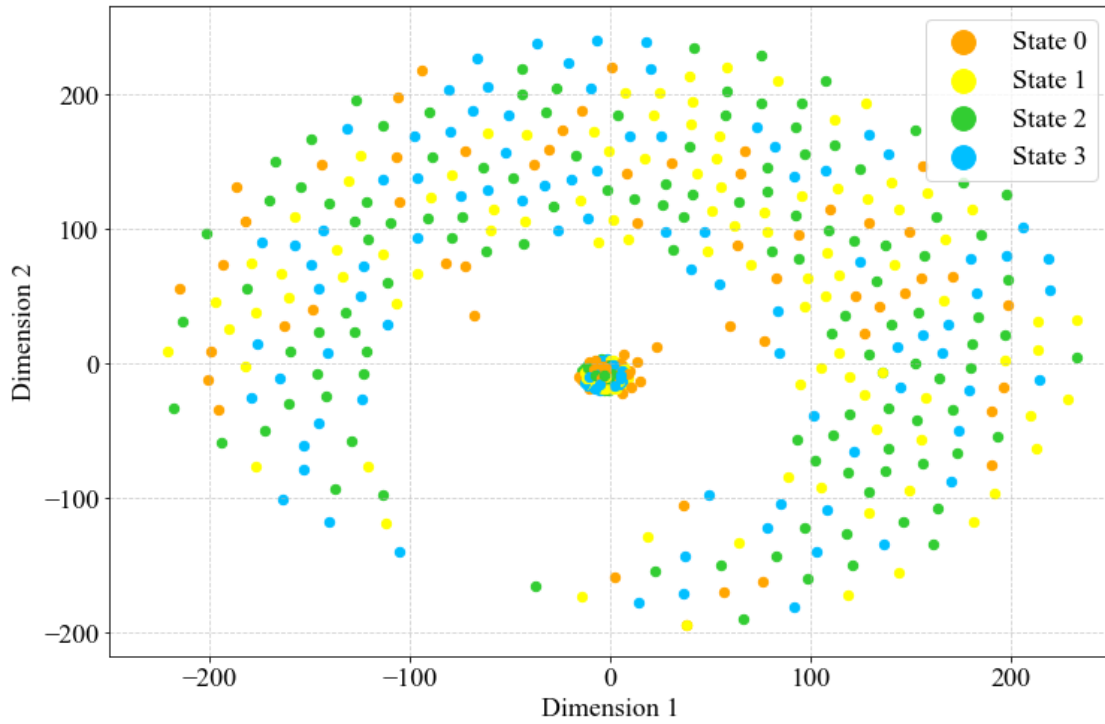


Fig. 5.42 Data structure of the batch normalization layer outputs (case of the long steel girder bridge)

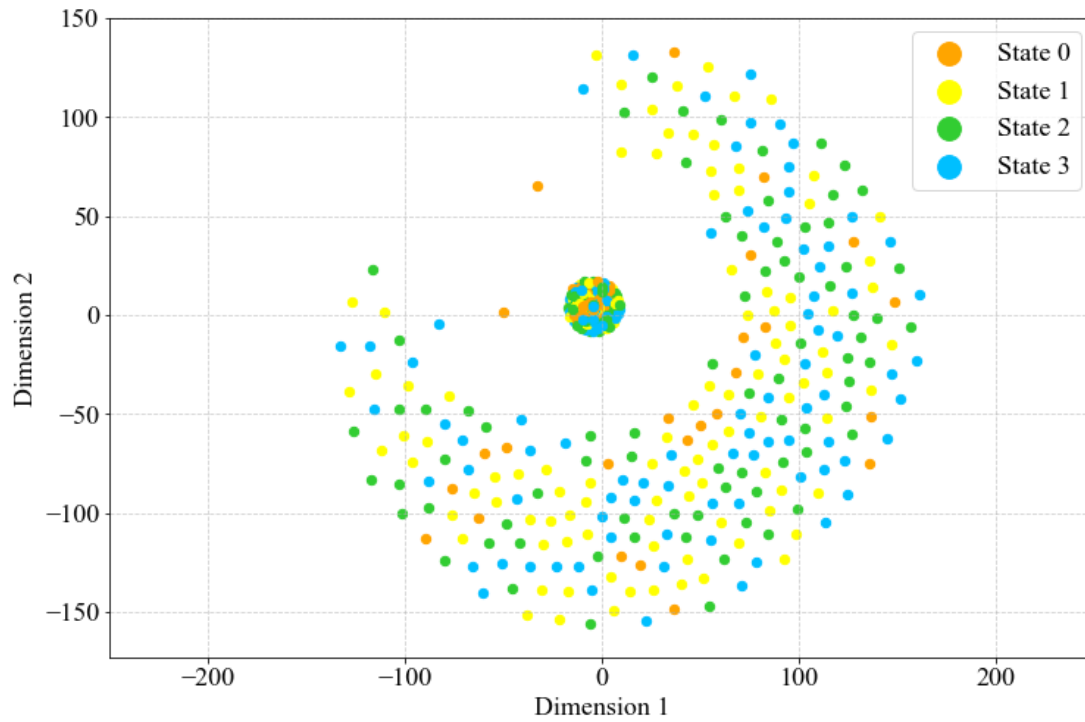


Fig. 5.43 Data structure of the max pooling layer outputs (case of the long steel girder bridge)

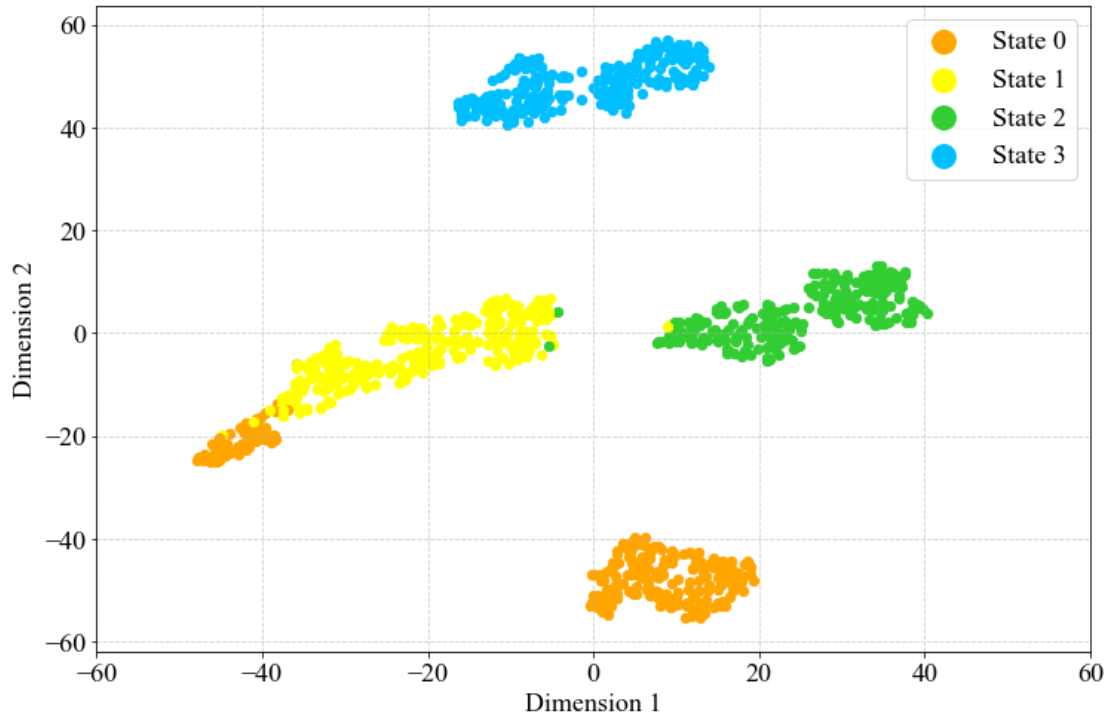


Fig. 5.44 Data structure of the FC layer outputs (case of the long steel girder bridge)

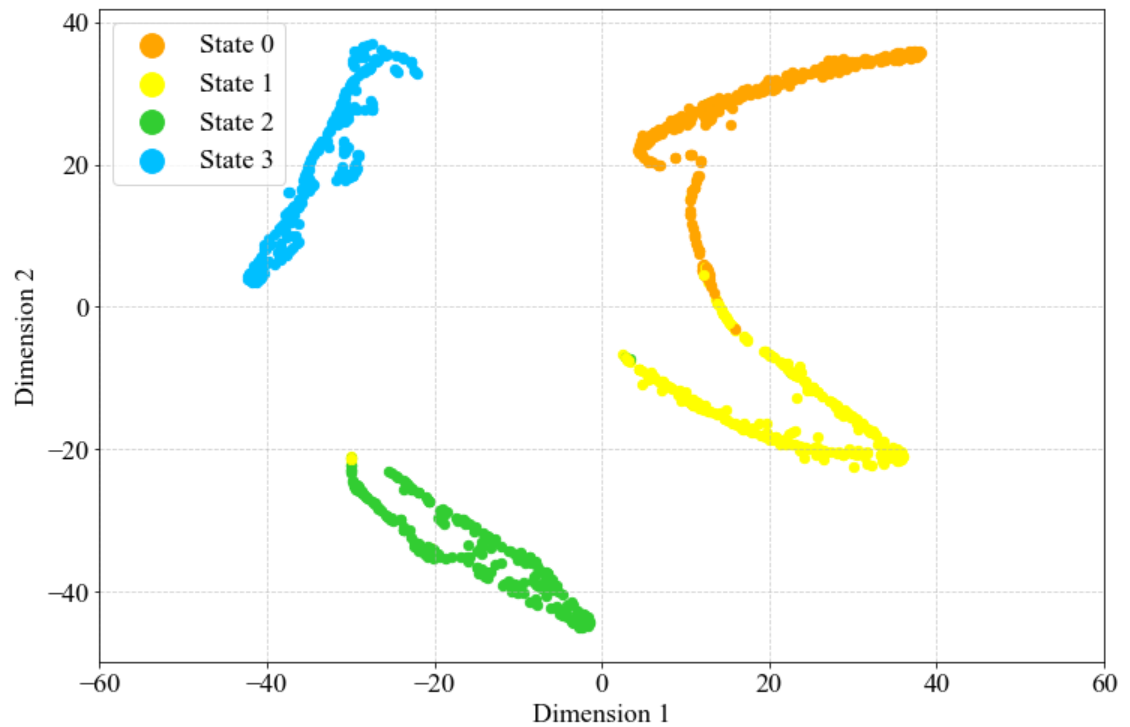


Fig. 5.45 Data structure of the network outputs (case of the long steel girder bridge)

5.5 Conclusions

In this chapter, visualizations of the CNN have been carried out. Some working mechanisms of the proposed CNN have been found. Firstly, the convolutional kernels extract and learn the vibration features in the acceleration data with approximately the same amplitude but different frequencies, computing the acceleration data as a band-pass filter. Secondly, the max pooling layer maintains the features of mode frequencies while reducing the size of the data. Finally, by observing the variation of data structure of the validation data flowing through each layer of the CNN, the effectiveness of structural state identification of the proposed CNN model has been proved.

Chapter 6

Robustness Tests of the Classification CNN Model

6.1 Introduction

In Chapter 4, the sensitivity of CNN to small local structural changes has been proved. Very high accuracies of identifying trained local structural changes are obtained. However, two points are unclear in Chapter 4: (1) the performance of classification CNN model predicting data in untrained structural states; (2) the performance of the CNN model predicting data acquired in untrained temperature. Meanwhile, Chapter 2 shows the low robustness of the dynamic parameters to the temperature effects which is one the major difficulties of the conventional SDD methods. Thus, the author is very interested in the robustness of the CNN to both location of structural change and the temperature effect.

In this chapter, robustness tests on (1) location of structural change and (2) temperature effect of the proposed classification CNN model have been performed. New vibration data acquired in untrained structural states and untrained temperature are fed into the classification CNN models to test the robustness of the proposed CNN model.

Firstly, two rounds of tests are performed to investigate the robustness of the CNN to the location of structural change. In the first round of the robustness tests, the CNN models introduced in Section 4.2 are used, which are trained by the T-shaped beam database. In the second round of the robustness tests, new CNN models trained by extended T-shaped beam database have been used. The purpose of expanding the database is to increase the diversity of the data and the range of data distributions of each structural state to promote the performance of the CNN model. Secondly, tests of the robustness to the temperature effect is performed on the CNN for the short steel girder bridge. The robustness of the classification CNN models to locations of structural changes and temperature effect are tested and discussed.

6.2. Tests of the Robustness to Locations of Structural Changes

The robustness tests of the proposed CNN to the locations of structural changes are performed on the T-shaped steel beam as introduced in Section 4.2. New vibration data acquired in untrained structural states are fed into the generated CNN models (also introduced in Section 4.2) to evaluate the robustness of the CNN to location of structural change.

6.2.1 Vibration experiment for test data generation

A vibration experiment was carried out on the T-shaped steel beam introduced in Section 4.2 to generate test data.

The data acquisition system and the excitation method are all identical to the experiment introduced in Section 4.2. The only difference to the experiment in Section 4.2 is the locations of the additional mass. The layout of the experiments is shown in Fig. 6.1. The database generated in Section 4.2 is named Experiment 1 here, marked with blue rectangles (S1-S8). The new experiment is named Experiment 2, and the locations of the additional mass are S1'-S9', marked with orange rectangles, generating the test data for the robustness tests. The detailed data distribution is shown in Table 6.1. There are about 240 data in each structural state. In total the amount of test data is 2151.

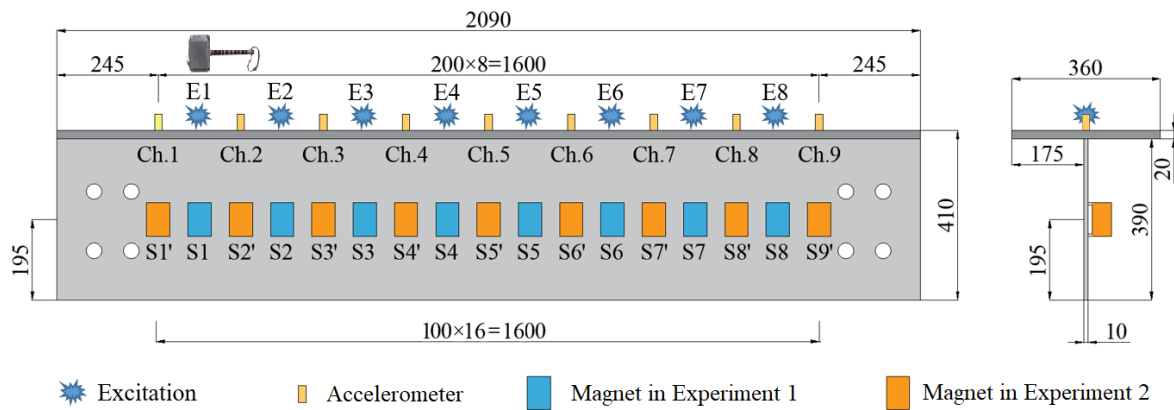


Fig. 6.1 Layout of the vibration experiments

Table 6.1 Data distribution of test data

| Category | Amount | Category | Amount |
|----------|--------|----------|--------|
| State 1' | 238 | State 6' | 239 |
| State 2' | 239 | State 7' | 239 |
| State 3' | 239 | State 8' | 238 |
| State 4' | 239 | State 9' | 240 |
| State 5' | 240 | | |
| In total | | | 2151 |

6.2.2 Robustness tests (Round 1)

6.2.2.1 Test set up

To exam the capacity of the 1-D CNN models to identify untrained local structural changes, the test data in Table 6.1 are fed into the 5-fold models trained and introduced in Section 4.2.

Here the author manually define that an accurate prediction should be in a range within the adjacent structural states. For example, the test data is in State 2', thus both States 1 and 2 are accurate predictions. In another word, a sample with an error of less than ± 10 cm is considered as a correct prediction.

6.2.2.2 Results

By evaluating the robustness with the correctness standard as defined in Subsubsection 6.2.2.1, the detailed results of the robustness tests of the 5-fold models are shown in Figs. 6.2-6.6, and the accuracies of the robustness tests are summarized in Table 6.2.

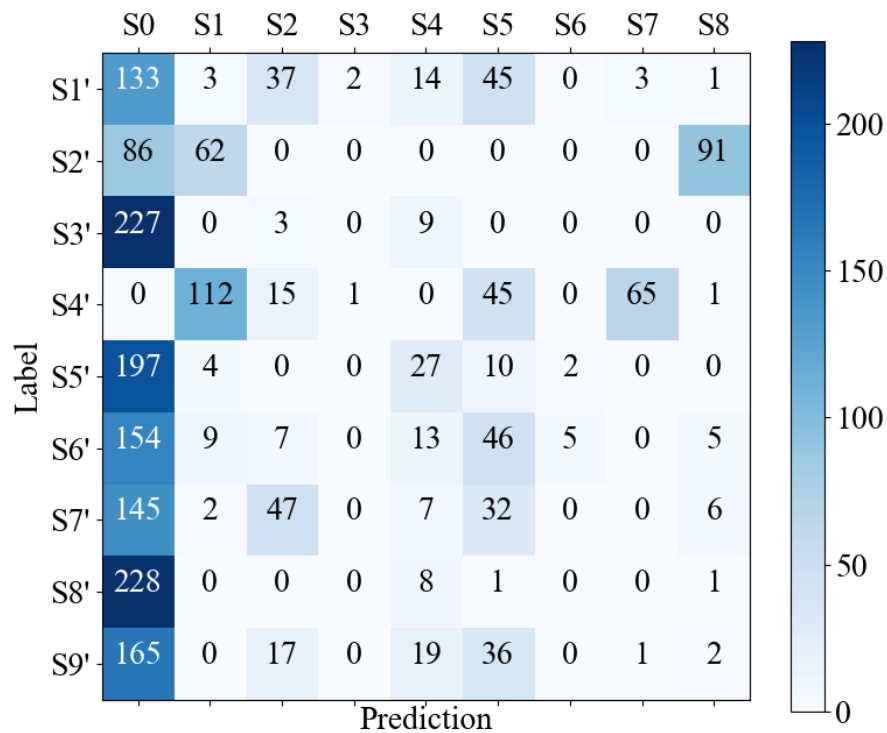


Fig. 6.2 Confusion Matrix of the robustness test results of Fold 1 model (round 1)

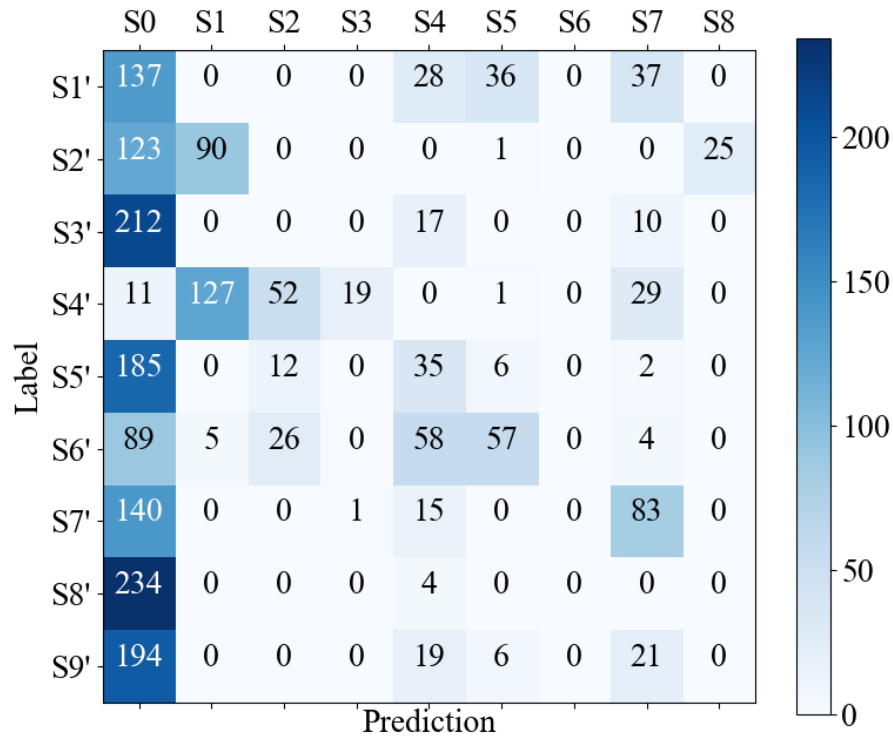


Fig. 6.3 Confusion Matrix of the robustness test results of Fold 2 model (round 1)

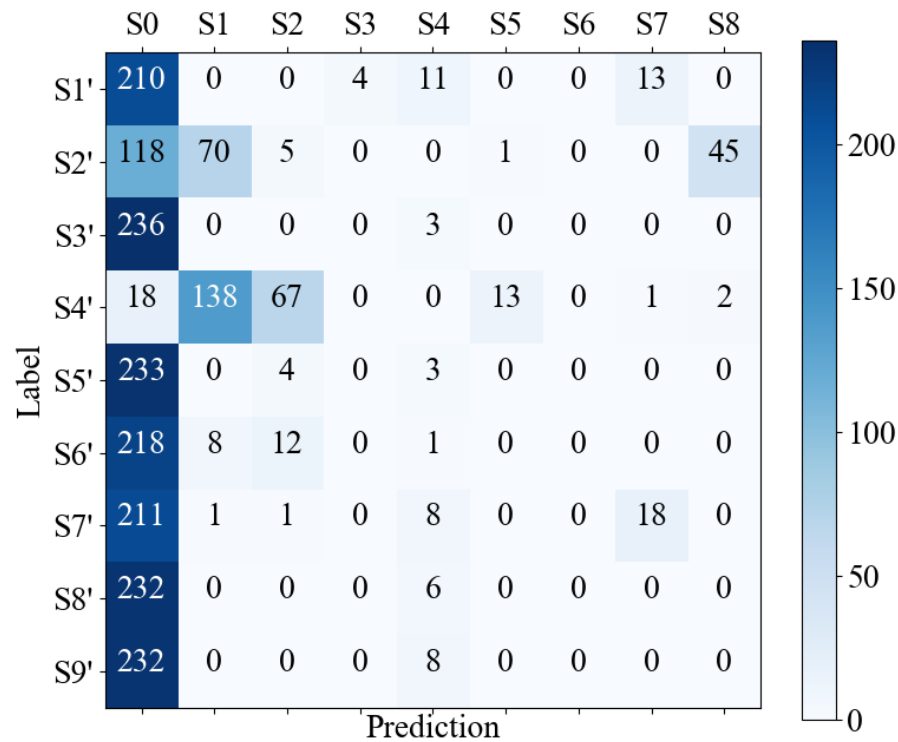


Fig. 6.4 Confusion Matrix of the robustness test results of Fold 3 model (round 1)

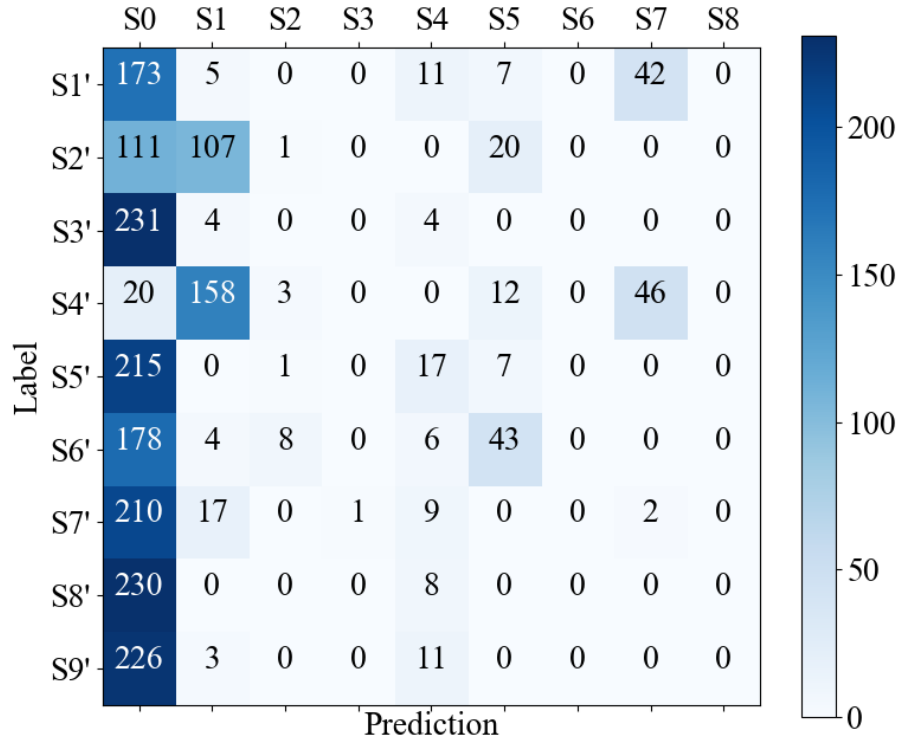


Fig. 6.5 Confusion Matrix of the robustness test results of Fold 4 model (round 1)

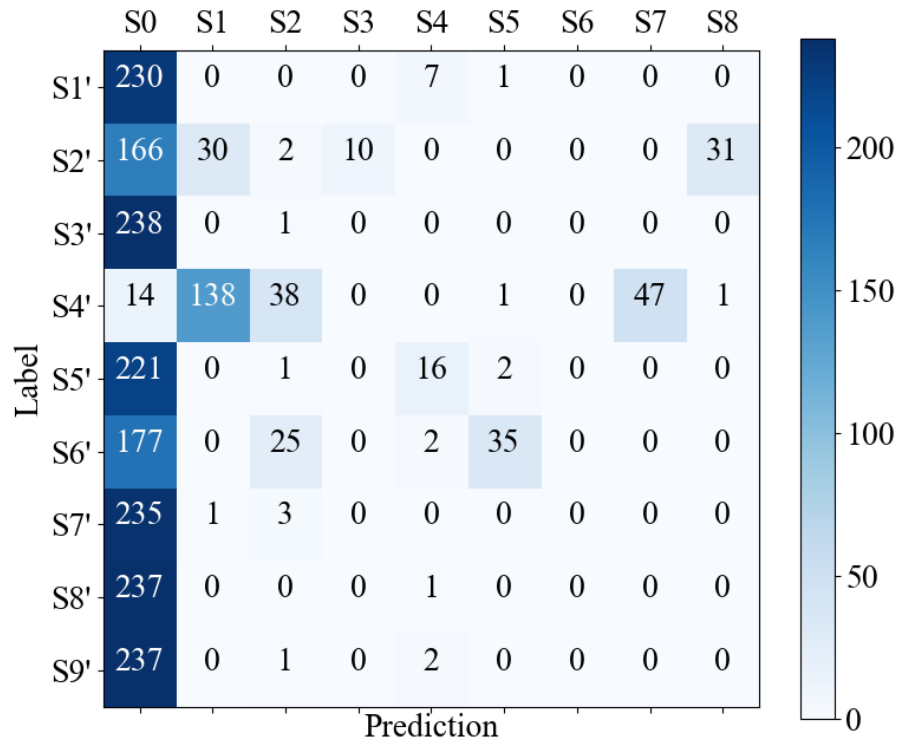


Fig. 6.6 Confusion Matrix of the robustness test results of Fold 5 model (round 1)

Table 6.2 Accuracies of the robustness tests (Round 1)

| <i>Model</i> | <i>Amount of accurate predictions</i> | <i>Accuracy (%)</i> |
|----------------|---------------------------------------|---------------------|
| <i>Fold 1</i> | 160 | 7.43 |
| <i>Fold 2</i> | 284 | 13.2 |
| <i>Fold 3</i> | 96 | 4.46 |
| <i>Fold 4</i> | 182 | 8.46 |
| <i>Fold 5</i> | 86 | 4.00 |
| <i>Average</i> | 162 | 7.51 |

It is expected that the prediction of the test data in untrained structural states should be the closest trained structural states, which are the diagonal and the upper secondary diagonal in Figs. 6.2-6.6. However, the test accuracies in all the 5 models are very low, with 162 accurate predictions and 7.51% accuracy on average, which shows the low robustness of the classification CNN models identifying untrained structural changes. Meanwhile, it was found that the results are more likely to be predicted as the intact state (S0). Therefore, the classification CNN models trained in Section 4.2 is not feasible to predict the structural state when vibration data is acquired in untrained structural state.

6.2.3 Robustness tests (Round 2) with expanded training data

Generally, increasing the amount of training data is a common way to improve the performance of a neural network, since the network is able to learn wider range of the data distribution and more features. Therefore, in this subsection, new 1-D CNN models are trained with expanded training data to improve the robustness of the CNN models to identify untrained local structural changes.

6.2.3.1 Introduction of the extended training data

To acquire additional training data, a new vibration experiment was performed on the T-shaped steel beam as introduced in Section 4.2. The detailed procedures are totally identical to the experiment as introduced in Subsection 4.2.1. In total there are 9 structural states, which are S0, S1..., and S8, as shown in Fig. 4.4. The only difference is the amounts of data. The amounts of the new acquired data are shown in Table 6.3. By emerging the existed database

introduced in Table 4.2 and the newly acquired data in Table 6.3, an emerged database is generated. The data distribution of the emerged database is shown in Table 6.4. Therefore, the amount of data increased 15.24% ($14465 \rightarrow 16670$) in the database.

Table 6.3 The amounts of the newly acquired data

| <i>Category</i> | <i>Amount</i> | <i>Category</i> | <i>Amount</i> |
|-----------------|---------------|-----------------|---------------|
| <i>State 0</i> | 299 | <i>State 5</i> | 238 |
| <i>State 1</i> | 238 | <i>State 6</i> | 239 |
| <i>State 2</i> | 238 | <i>State 7</i> | 238 |
| <i>State 3</i> | 239 | <i>State 8</i> | 238 |
| <i>State 4</i> | 238 | | |
| <i>In total</i> | | 2205 | |

Table 6.4 Data distribution of the emerged database

| <i>Category</i> | <i>Amount</i> | <i>Category</i> | <i>Amount</i> |
|-----------------|---------------|-----------------|---------------|
| <i>State 0</i> | 1886 | <i>State 5</i> | 1860 |
| <i>State 1</i> | 1833 | <i>State 6</i> | 1862 |
| <i>State 2</i> | 1833 | <i>State 7</i> | 1857 |
| <i>State 3</i> | 1850 | <i>State 8</i> | 1831 |
| <i>State 4</i> | 1858 | | |
| <i>In total</i> | | 16670 | |

6.2.3.2 CNN models preparation

After extending the amount of training data, new CNN models are trained and validated by using the new emerged database as introduced in 6.2.3.1. The CNN configurations are totally identical to the descriptions in Subsection 4.2.2. The only differences are the amounts of training and validation data. To follow the common way in Chapter 4, 80% of the new emerged database are used for training, and other 20% of the new emerged database are used for validation, with no overlap between the training set and validation set. Meanwhile, 5-fold validation is conducted to ensure the performance of the CNN models.

The new CNN models are trained with very high accuracies, as shown in Table 6.5, with 99.99% training accuracy and 99.68% validation accuracy on average. Those newly trained models are used to examine the robustness of identifying untrained local structural changes.

Table 6.5 Accuracies of 5-fold cross validation of models trained by the extended T-shaped steel beam database (%)

| | <i>Training</i> | <i>Validation</i> |
|----------------|-----------------|-------------------|
| <i>Fold 1</i> | 100.00 | 99.64 |
| <i>Fold 2</i> | 99.99 | 99.58 |
| <i>Fold 3</i> | 99.99 | 99.67 |
| <i>Fold 4</i> | 100.00 | 99.67 |
| <i>Fold 5</i> | 99.99 | 99.85 |
| <i>Average</i> | 99.99 | 99.68 |

6.2.3.3 Test set up

The procedure of robustness tests follow the way as introduced in Subsubsection 6.2.2.1. The test data generated in Subsection 6.2.1 are fed into the CNN models trained by the emerged database as introduced in Subsubsection 6.2.3.1. To evaluate the robustness of the CNN model, same correctness standard is used as Subsubsection 6.2.2.1.

6.2.3.4 Results

The detailed results of the robustness tests of the new 5-fold CNN models are shown in Figs. 6.7-6.11. As the same correctness standard as defined in Subsubsection 6.2.2.1 has been used, the predictions in the diagonal (except the S1'-S0 cell) and the upper secondary diagonal in Figs. 6.7-6.11 are correct predictions.

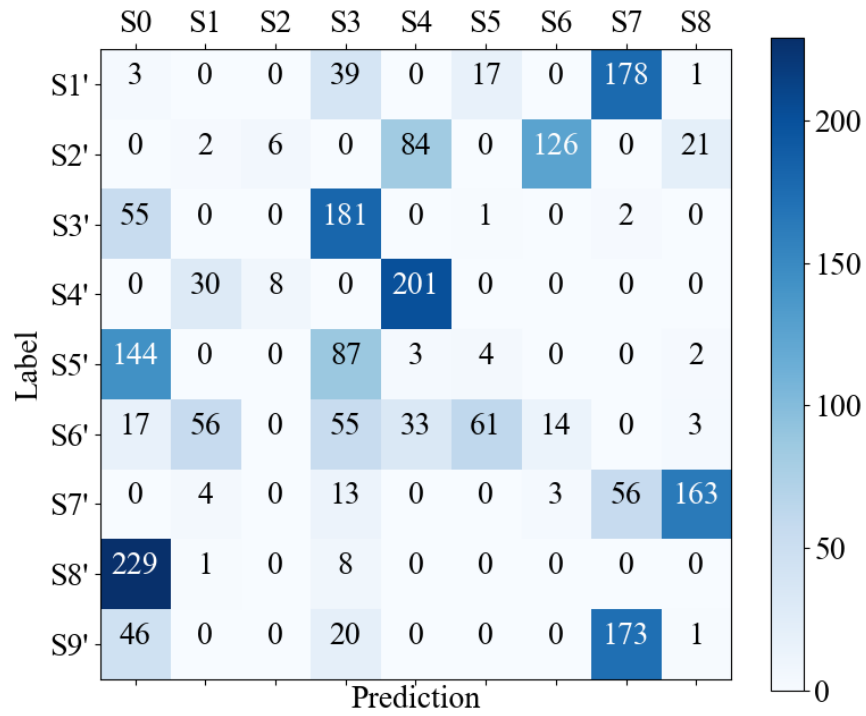


Fig. 6.7 Confusion Matrix of the robustness test results of Fold 1 model (Round 2)

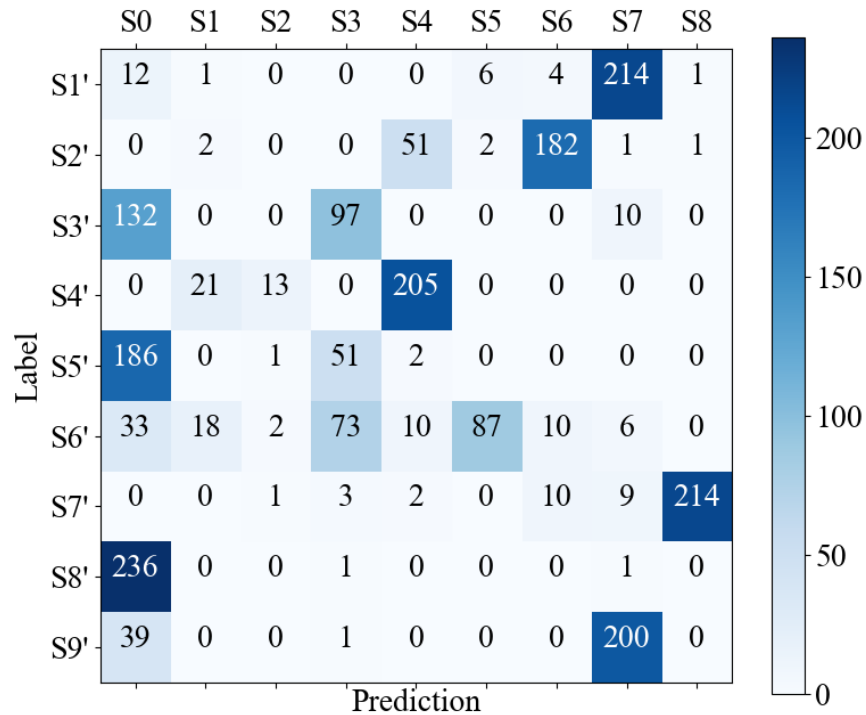


Fig. 6.8 Confusion Matrix of the robustness test results of Fold 2 model (Round 2)

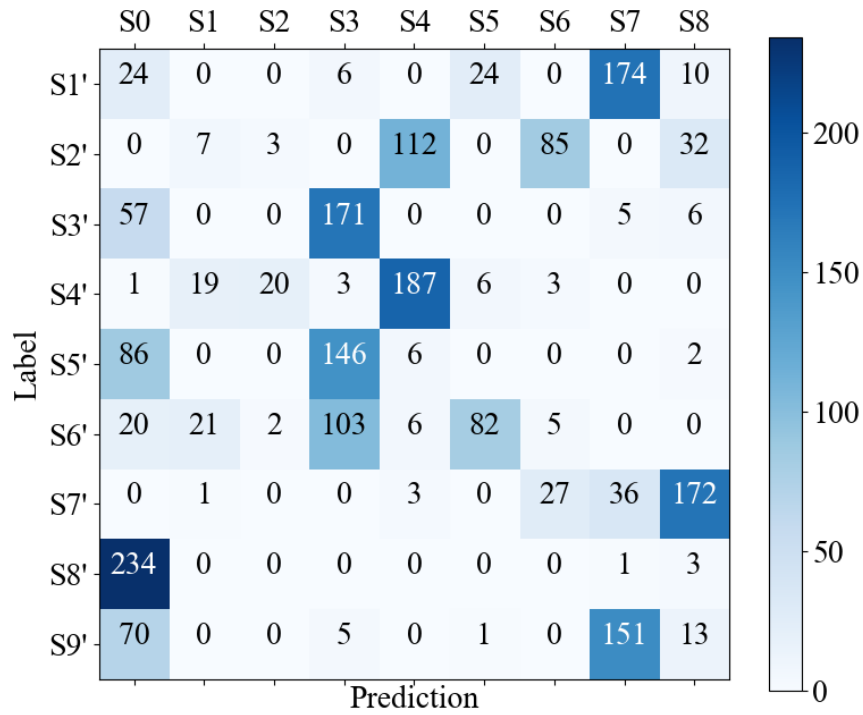


Fig. 6.9 Confusion Matrix of the robustness test results of Fold 3 model (Round 2)

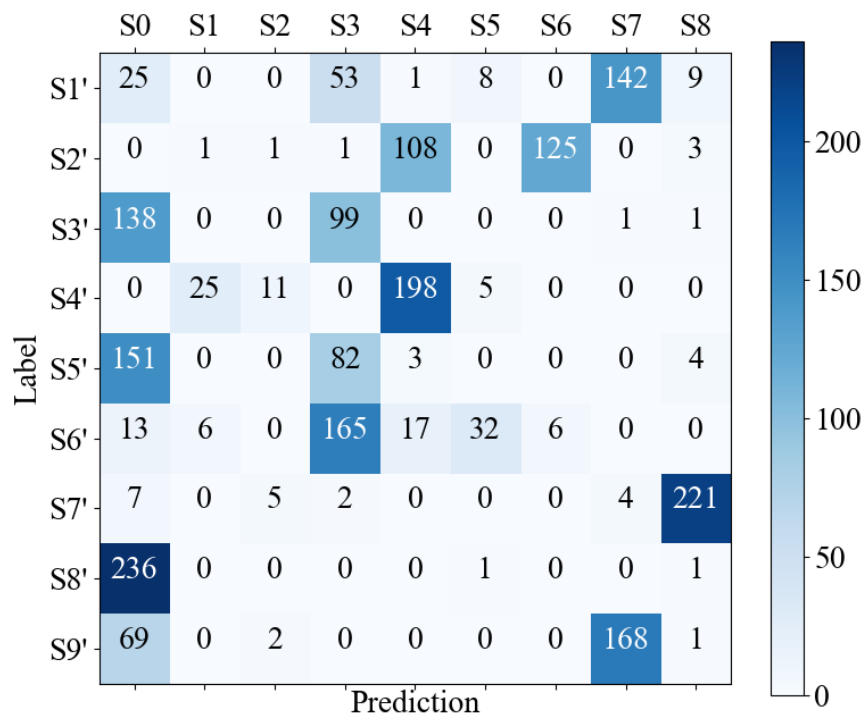


Fig. 6.10 Confusion Matrix of the robustness test result of Fold 4 model (Round 2)

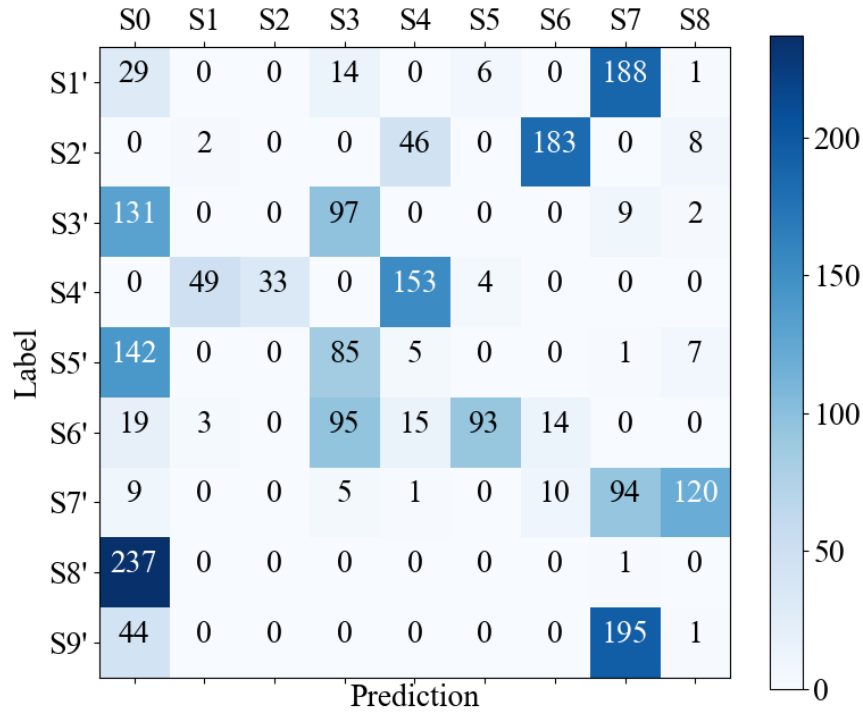


Fig. 6.11 Confusion Matrix of the robustness test results of Fold 5 model (Round 2)

Overview Figs. 6.7-6.11, and comparing to the Round 1 results in Figs. 6.2-6.6, the Round 2 results improved obviously. There are less negative errors in the Round 2 results. Meanwhile, there are also many correct predictions of the S3' and S4' data. The reason should be the increase of the amount of training data, so the CNN models learning more features of the vibration data, and determining more accurate boundary between the structural states.

The amount of accurate predictions and accuracies of the robustness tests are summarized in Table 6.6. Comparing to the Round 1 results, the amount of average accurate predictions increased from 162 to 463, and the corresponding accuracy increased from 7.51% to 21.53%. The improvement of the robustness of identifying untrained structural changes proves the effectiveness of the operation of increase the amount of training data. However, the robustness of the model is still not high. Therefore, update of the CNN architecture should be pay more attention to improve the robustness of the CNN to location of structural change.

Table 6.6 Accuracies of the robustness tests (Round 2)

| <i>Model</i> | <i>Amount of accurate predictions</i> | <i>Accuracy (%)</i> |
|----------------|---------------------------------------|---------------------|
| <i>Fold 1</i> | 532 | 24.73 |
| <i>Fold 2</i> | 424 | 19.71 |
| <i>Fold 3</i> | 544 | 25.29 |
| <i>Fold 4</i> | 346 | 16.09 |
| <i>Fold 5</i> | 470 | 21.85 |
| <i>Average</i> | 463 | 21.53 |

6.3 Tests of the Robustness to Temperature Effect

The robustness tests of the proposed CNN to the temperature effect are performed on the short steel girder bridge as introduced in Section 4.3. New vibration data acquired in different temperature conditions to the training data are fed into the generated CNN models (also introduced in Section 4.3) to evaluate the robustness of the CNN to temperature effect.

6.3.1 Vibration experiment for test data generation

A vibration experiment was carried out on the short steel girder bridge as introduced in Section 4.3 to generate test data. The data acquisition system and the excitation method are all identical to the experiment introduced in Section 4.3. The layout of the experiments is also identical to in Fig. 4.10. In total there are 6 different structural states, which are State 0 for the original state, States 1-4 for 4 structural states with a steel plate fixed on 4 locations on the lower flange, and State 5 for the state of a mass block attached on the center of upper flange. The detailed data distribution is shown in Table 6.7. There are about 290 data in each structural state. In total the amount of test data is 1733.

Table 6.7 The amounts of the newly acquired data

| <i>Category</i> | <i>Amount</i> | <i>Category</i> | <i>Amount</i> |
|-----------------|---------------|-----------------|---------------|
| <i>State 0</i> | 281 | <i>State 3</i> | 290 |
| <i>State 1</i> | 286 | <i>State 4</i> | 291 |
| <i>State 2</i> | 291 | <i>State 5</i> | 294 |
| <i>In total</i> | | | 1733 |

Owing to some technical problem, there no air temperature data acquired near the specimen. The air temperature data of the Amedas system is utilized as an alternative. Based on the author's experience, the difference of air temperature between the local measurement and the Amedas system is approximate 3-5 °C in average. Thus, using the air temperature data of the Amedas system for qualitative study is also a practical way. Fig. 6.12 shows the air temperature information of the experimental days. The data introduced in Section 4.3 is acquired on 2017.10.31 and 2017.11.1. The new test data is acquired on 2018.10.4. There is approximate 1.5 °C - 15.8 °C temperature difference between the two experiments.

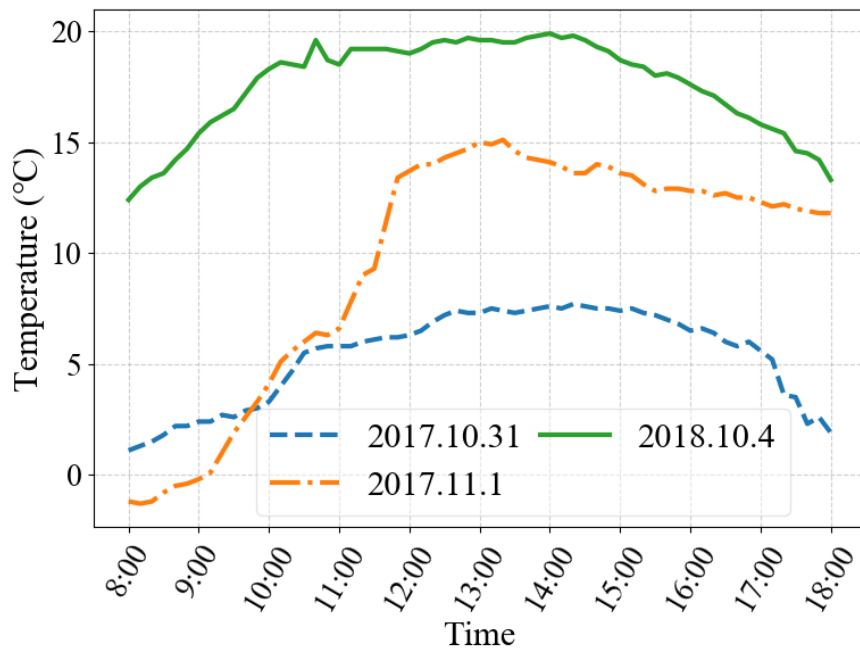


Fig. 6.12 Temperature of the three experimental days on Amedas system

6.3.2 Results

Feeding the newly acquired test data into the five generated CNN models introduced in Section 4.3, five sets of test results are obtained which are shown as confusion matrices in Figs. 6.13-6.18. Overview these confusion matrices, the numbers of correct predictions are very small. Most data are wrongly predicted as State 3 or 4. The number of correct predictions are summarized in Table 6. The average number of correct predictions is 297, and the average accuracy is 17.15% which is just a little higher than randomly guessing the result (16.67%). Thus, the robustness of the CNN to temperature effect is very low.

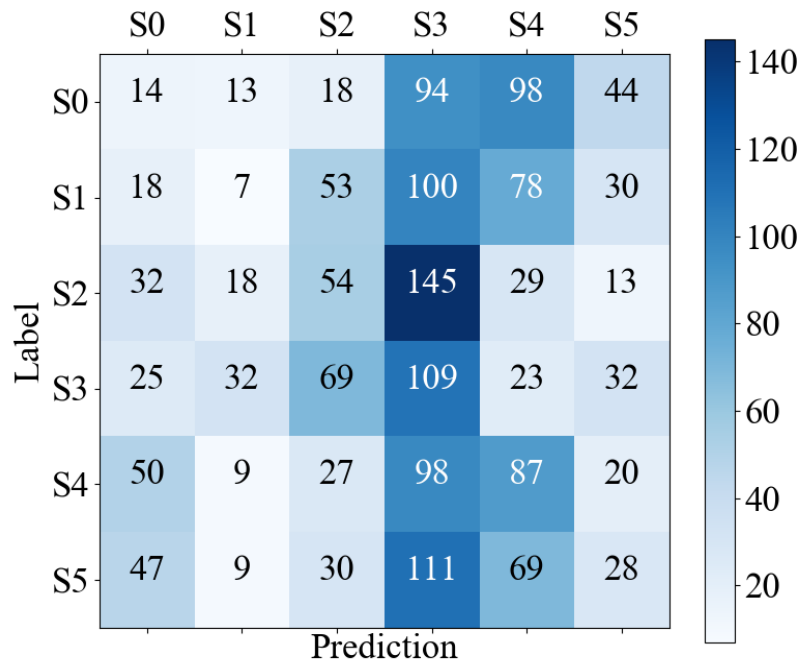


Fig. 6.13 Confusion Matrix of the robustness test result of Fold 1 model

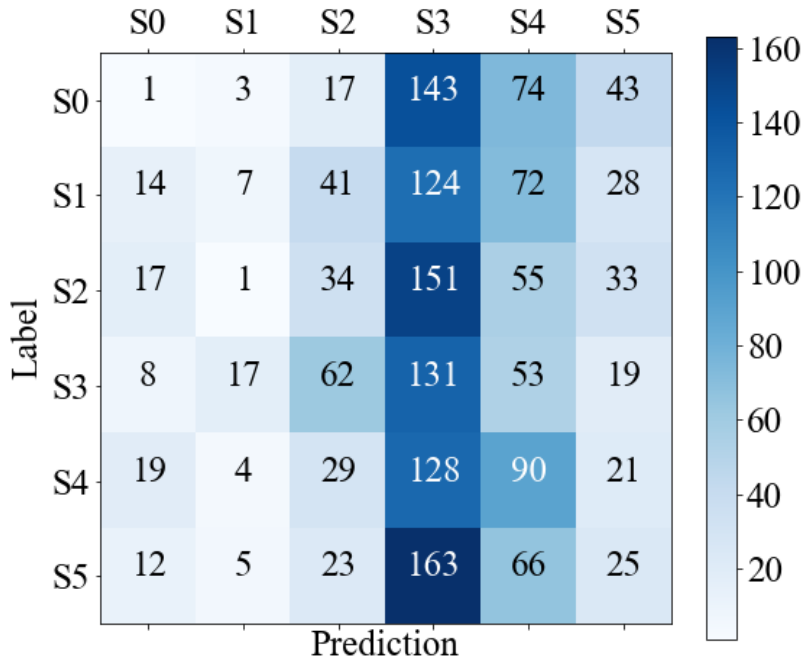


Fig. 6.14 Confusion Matrix of the robustness test result of Fold 2 model

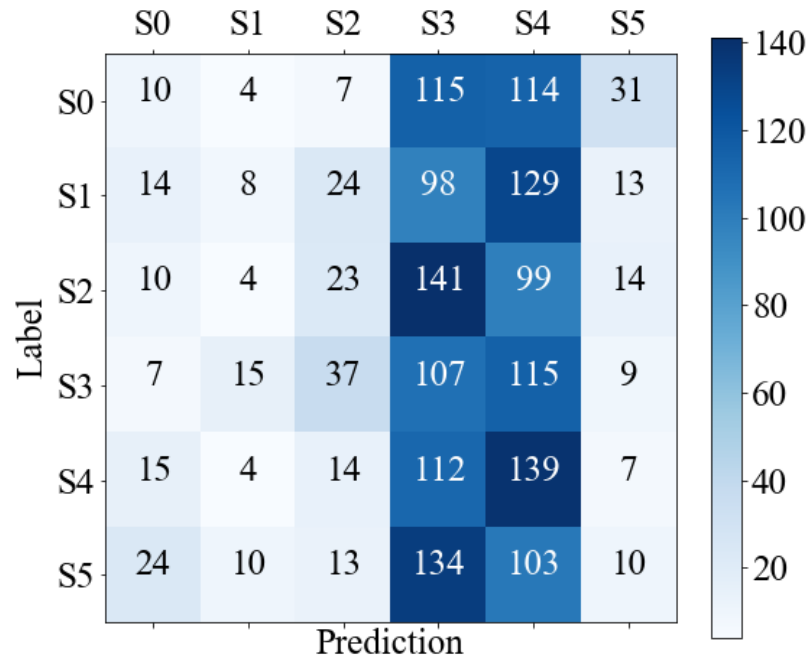


Fig. 6.15 Confusion Matrix of the robustness test result of Fold 3 model

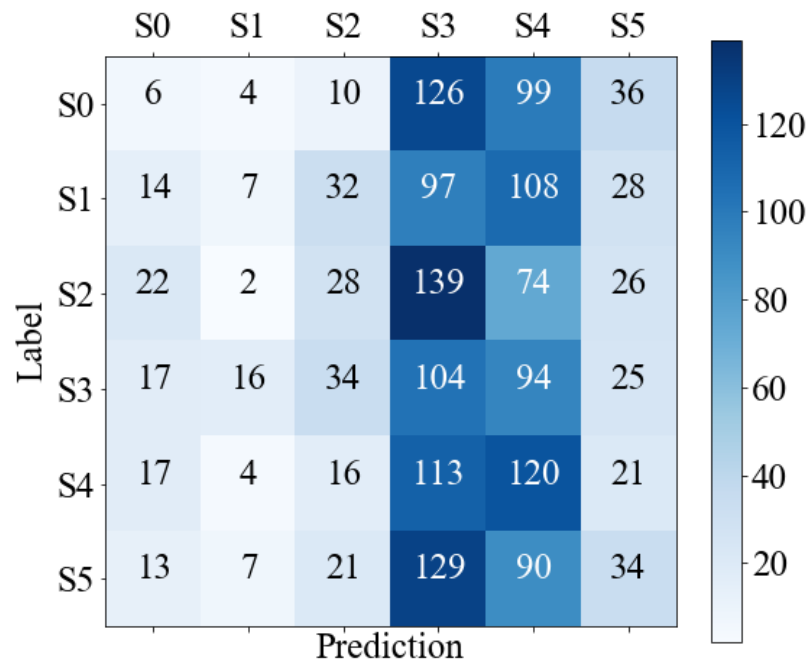


Fig. 6.16 Confusion Matrix of the robustness test result of Fold 4 model

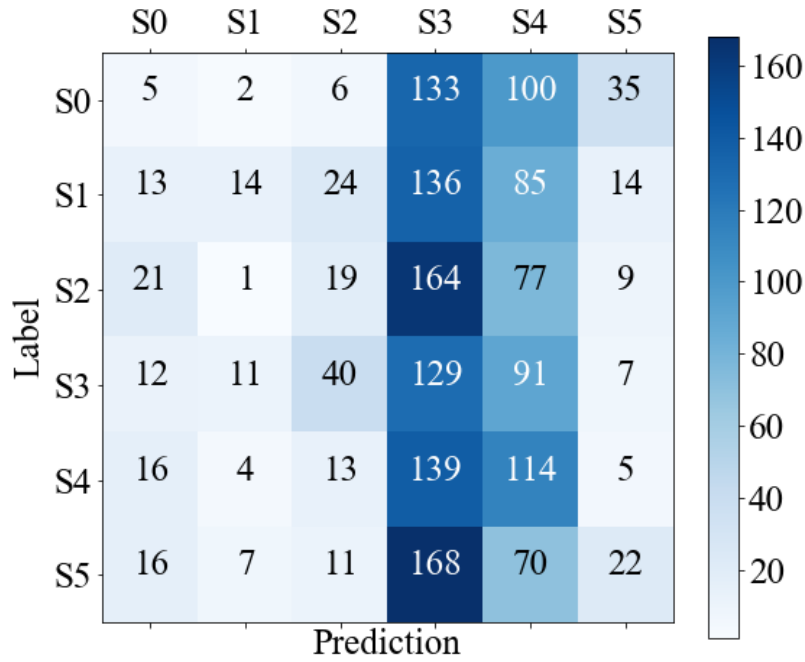


Fig. 6.17 Confusion Matrix of the robustness test result of Fold 5 model

Table 6.8 Accuracies of the robustness tests

| <i>Model</i> | <i>Amount of accurate predictions</i> | <i>Accuracy (%)</i> |
|----------------|---------------------------------------|---------------------|
| <i>Fold 1</i> | 299 | 17.25 |
| <i>Fold 2</i> | 288 | 16.62 |
| <i>Fold 3</i> | 297 | 17.14 |
| <i>Fold 4</i> | 299 | 17.25 |
| <i>Fold 5</i> | 303 | 17.48 |
| <i>Average</i> | 297 | 17.15 |

The reason of the low robustness of the CNN to the temperature effect should be same to be the reason of the low robustness of the dynamic parameters to the temperature effect as discussed in Chapter 2. Temperature influences the material properties, constrains, and support conditions of structure which vary the dynamic response of structure. Since the CNN has not been trained with these vibration data in diverse temperature conditions, the CNN are not able to identify the structural state correctly. Therefore, according to the above reason of the low robustness of the CNN to the temperature effect, an assumption of how to improve the robustness is proposed, that expanding the training data which are acquired in a wide range of temperature conditions.

6.3.3 Solution

Based on the assumption proposed in last paragraph, in this subsection, a new CNN is trained and validated to improve the robustness of the CNN to the temperature effect. The architecture and detailed settings of the CNN are all identical to the CNN as introduced in Section 4.3. The only differences are training data and validation data.

The data acquired in the first experiment (introduced in Subsection 4.3.1) is named Database 1, and the data acquired in the second experiment (introduced in Subsection 6.3.1) is named Database 2. The training data of the new CNN consists of 80% of the Database 1 (6876) and 80% of the Database 2 (1386). The validation data of the new CNN consists of other 20% of the Database 1 (1719) and other 20% of the Database 2 (347). In total, there are 8262 training data and 2066 validation data respectively. Note that there is no overlap between the training data and validation data.

The CNN was trained for 100 epochs. The training history is shown in Fig. 6.18. The training process is very smooth. Finally, the training accuracy reached 100%, and the validation accuracy obtained 98.11%. The detailed validation result is shown as a confusion matrix in Fig. 6.19. Most data are correctly predicted (2027 / 2066). The high validation accuracy proves the feasibility of the solution of improving the robustness of CNN to the temperature effect.

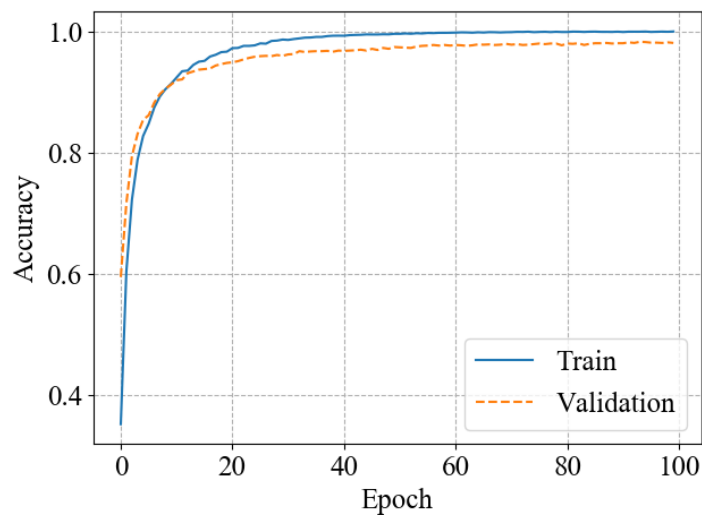


Fig. 6.18 Training history of the new CNN

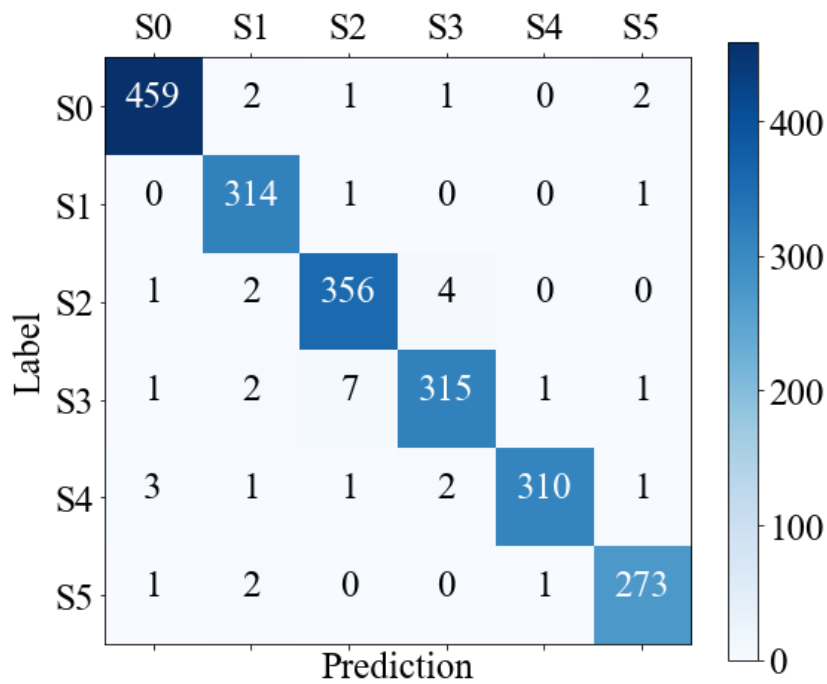


Fig. 6.19 Confusion Matrix of the validation results

To observe the validation results deeply, the components of the validation results in Database 1 and Database 2 are shown individually in Figs. 6.20 and 6.21. The component of Database 1 has 1707 accurate predictions (1707 / 1719), with 99.3% validation accuracy. The component of Database 2 has 320 accurate predictions (320 / 347), with 92.22% validation accuracy. There are about 7% difference between the two validation accuracies. The reason should be the amounts of training data of the two databases are quite difference (6876 and 1386), with an approximate ratio of 5:1. The training data in Database 2 is insufficient to learn the maximum amount of features of data. Thus, the CNN learned more features from the training data in Database 1 than Database 2. If increasing the amount of training data in Database 2, the validation result could be easily improved.

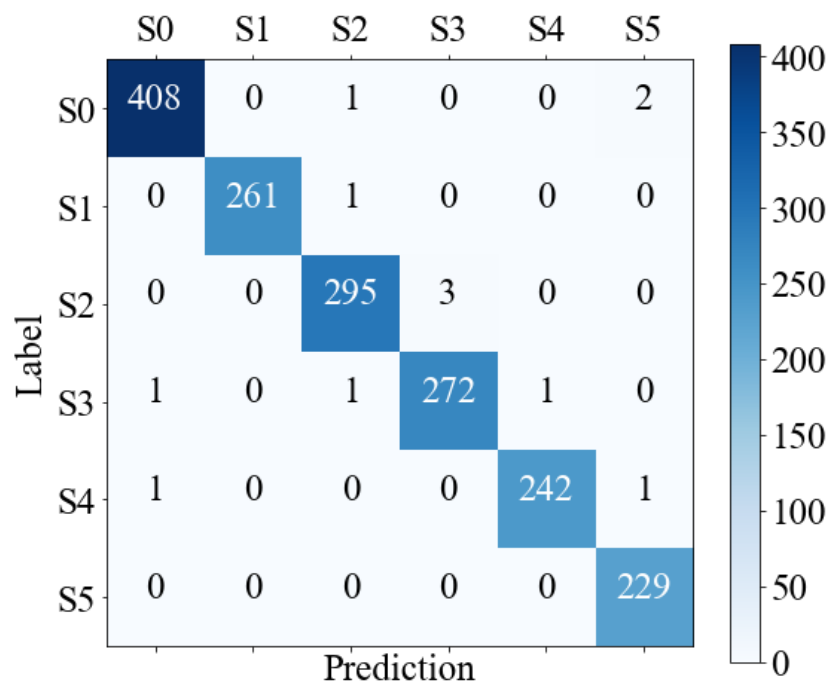


Fig. 6.20 Confusion Matrix of the validation results in Database 1

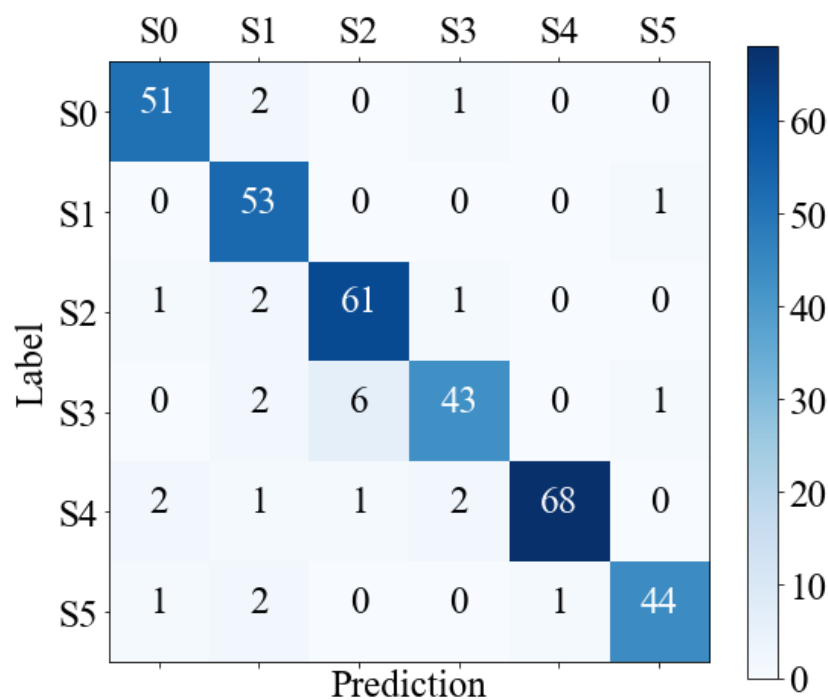


Fig. 6.21 Confusion Matrix of the validation results in Database 2

Overview the proposed solution of improving the robustness of CNN to the temperature effects, the concept can be summarized as Fig. 6.22. For the structural state identification problem, a distribution which can well express the target structural state is desired. However, by using limited amount of training data only partial features of the data can be abstracted to establish an approximate distribution. By expanding the diversity of training data, which means training a CNN by using data acquired in different conditions, the established distribution will become closer to the target distribution. The process can be seen as continuous complementation and improvements. When the training data have enough high diversity, the established distribution can be infinitely close to the target distribution. Therefore, the way of improving the robustness of CNN to the temperature effect can be simply expending the training data in diverse temperature conditions, which overcome the low robustness of the dynamic parameters to the temperature effect of the conventional SDD methods as introduced in Chapter 2.

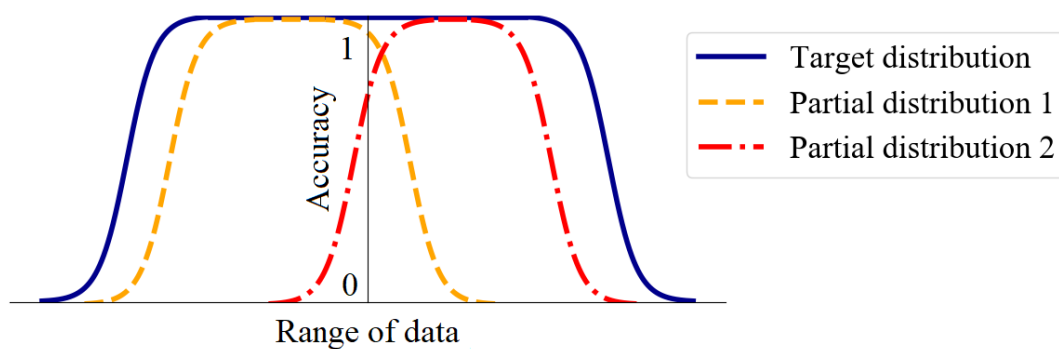


Fig. 6.22 Illustration of the mechanism

6.4 Conclusions

In this chapter, robustness tests of the proposed CNN to the location of structural change and temperature effect are performed.

For the tests of the robustness to the locations of structural changes, a vibration database contains untrained structural states is established for the two rounds of tests. In the first round, the new test data are feed in the CNN models introduced in Section 4.2. The results show very low robustness of the CNN models. In the second round, the training data are extended, and new CNNs are trained for better universality. The robustness of the newly trained CNN models

is examined. The robustness of the Round 2 tests improved obviously, which indicating the effectiveness of the operation of increase the amount of training data. However, the robustness is still not high, and further development are needed. Thus, using classification CNN models are difficult to identify untrained local structural changes, and updates of the CNN architecture are needed. The content of updates of the CNN architecture is presented in Chapter 7.

For the tests of the robustness to the temperature effect, a vibration database acquired in different temperature conditions was established. The CNNs show very low robustness to the temperature effect. To solve this problem, a solution is proposed that expanding the training data which are acquired in different temperature conditions. The solution is validated and very high accuracy of 92.22% is obtained. The approach of improving the robustness of the CNN based structural state identification method overcome the difficulty of the low robustness of the dynamics parameters to the temperature effect of the conventional SDD methods as discussed in Chapter 2.

Chapter 7

Updates of the CNN Model

7.1 Introduction

Since the 1-D classification CNN models proposed in Chapter 4 use softmax outputs and one-hot encoding labels as introduced in Subsections 4.2.2, 4.3.2, and 4.4.2, the CNN models can only identify or classify limited discrete trained structural states with very high accuracies, as shown in Subsections 4.2.3, 4.3.3, and 4.4.3. Based on the robustness test results in Chapter 6, the classification CNN models almost cannot predict approximate locations of the untrained structural changes accurately.

Therefore, this chapter aims to update the performance of the 1-D CNN to identify untrained structural states with better expression of location of structural change and higher robustness. Based on the classification CNN model as introduced in Chapter 4, several updates of the CNN have been proposed as follows:

- First, a regression CNN model has been proposed, which updates the encoding of labels and corresponding output layer of the CNN model.
- Second, to further improve the robustness of the CNN to identify untrained structural states, a deep network with multi-convolution blocks and multi-task outputs has been proposed.

7.2 Regression CNN Model

7.2.1 Details of the regression CNN model

The architecture of the regression CNN model is shown in Table 7.1. The architecture of the regression CNN model is almost same to the architecture of the classification CNN model as introduced in Subsection 4.2.2. Only the output layer and corresponding label encoding have been updated. The new label consists two units as shown in Fig. 7.1. The first unit is a value between 0 and 1 for the confidence of existence of structural change. The second unit is a value which is greater than 0 and less than 1 representing the relative location of structural change. The far left and the far right of the beam are 0 and 1. The updated output layer has only 2 units

corresponding to the new labels, and excited by the sigmoid function. The definition of the sigmoid function is shown in Equation 7.1, and corresponding figure is shown in Fig 7.2.

Table 7.1 Configuration of the classification and regression CNNs

| <i>Layers</i> | <i>Output Shape</i> | <i>Parameter</i> | <i>Activation</i> | <i>Variables</i> |
|----------------------------|---------------------|---|-------------------|------------------|
| <i>Input Layer</i> | 10000×9 | None | None | 0 |
| <i>Convolution 1-D</i> | 9991×5 | Kernel number: 5; Kernel size: 10×9; Stride: 1; Padding: Valid | Linear | 455 |
| <i>Batch normalization</i> | 9991×5 | None | None | 20 |
| <i>Max Pooling 1-D</i> | 3330×5 | Kernel number: 3; Stride: 1 | None | 0 |
| <i>Flatten</i> | 16650 | None | None | 0 |
| <i>Dropout</i> | 16650 | Rate: 0.25 | None | 0 |
| <i>FC</i> | 40 | None | ReLU | 666,040 |
| <i>Output</i> | 2 | None | Sigmoid | 82 |
| <i>Total parameters</i> | | | | 666,597 |

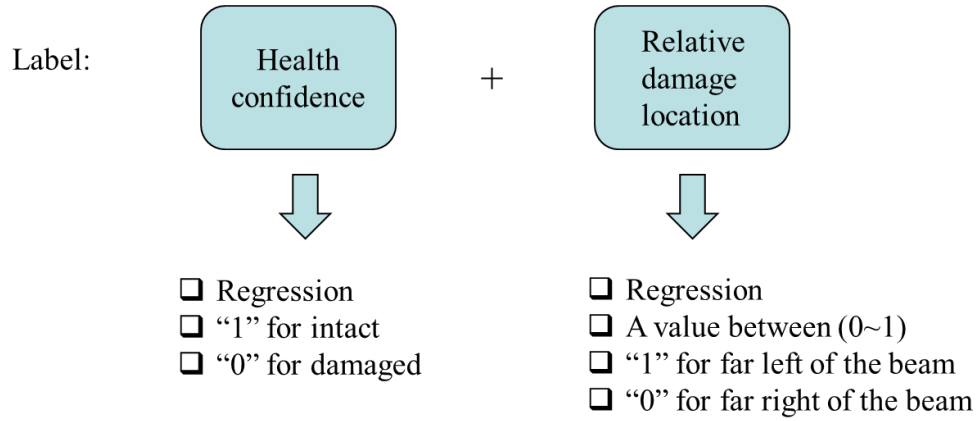


Fig. 7.1 Label of the regression CNN model

$$f(x) = \frac{1}{1+e^{-x}} \quad (7.1)$$

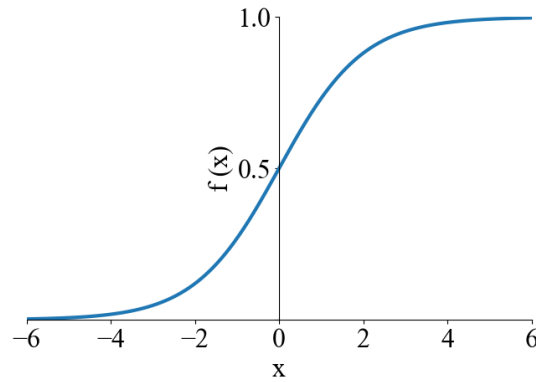


Fig. 7.2 Sigmoid function

The loss function of the new regression CNN is changed to mean squared error, as introduced in Subsection 3.4.2. Other parameters, such as optimizer (Adam, learning rate: 0.0001, ρ_1 : 0.9, ρ_2 : 0.999, δ : 1e-8, and decay: 0), initializers (weights: Variance Scaling, biases: zeros), and batch size (256) are all identical to the classification CNN model introduced in Section 4.2.

7.2.2 Validation set up

To validate the capacity of the proposed regression CNN model, the databases of T-shaped steel beam introduced in Tables 6.1 and 6.4 in Chapter 6 are used. The database in Table 6.4 consists of data in States 0-8, which are used for training the regression CNN model. The database in Table 6.1 consists of data in States 1'-9' is used for validating the regression CNN model. For easy reading, the layout of the structural state is shown in Fig. 7.3, and the training and validation data are shown in Tables 7.2 and 7.3 once again.

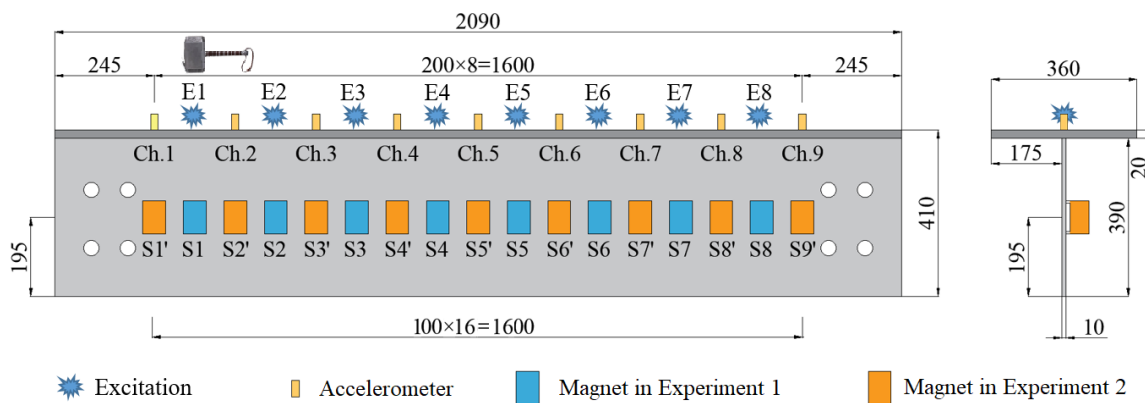


Fig. 7.3 Layout of the vibration experiments

Table 7.2 Data distribution of the Training data

| <i>Category</i> | <i>Amount</i> | <i>Category</i> | <i>Amount</i> |
|-----------------|---------------|-----------------|---------------|
| <i>State 0</i> | 1886 | <i>State 5</i> | 1860 |
| <i>State 1</i> | 1833 | <i>State 6</i> | 1862 |
| <i>State 2</i> | 1833 | <i>State 7</i> | 1857 |
| <i>State 3</i> | 1850 | <i>State 8</i> | 1831 |
| <i>State 4</i> | 1858 | | |
| <i>In total</i> | | | 16670 |

Table 7.3 Data distribution of validation data

| <i>Category</i> | <i>Amount</i> | <i>Category</i> | <i>Amount</i> |
|-----------------|---------------|-----------------|---------------|
| <i>State 1'</i> | 238 | <i>State 6'</i> | 239 |
| <i>State 2'</i> | 239 | <i>State 7'</i> | 239 |
| <i>State 3'</i> | 239 | <i>State 8'</i> | 238 |
| <i>State 4'</i> | 239 | <i>State 9'</i> | 240 |
| <i>State 5'</i> | 240 | | |
| <i>In total</i> | | | 2151 |

Based on the design concept of label as explained in Subsection 7.2.1, the encoding of the labels of data in each structural state are shown in Table 7.4. There are two units in each label. Unit 1 is the health confidence of the structure that “0” for intact and “1” for changed. Unit 2 is the relative location of structural change. For instance, structural mass change of State 1' occurred on 245 mm of the beam. Corresponding Unit 2 of label is 245 divided by the length of the beam 2090. Thus the Unit 2 is 0.117225.

Table 7.4 Encoding of the labels

| <i>Category</i> | <i>Label</i> | | <i>Category</i> | <i>Label</i> | |
|-----------------|---------------|---------------|-----------------|---------------|---------------|
| | <i>Unit 1</i> | <i>Unit 2</i> | | <i>Unit 1</i> | <i>Unit 2</i> |
| <i>State 0</i> | 0 | 0 | <i>State 1'</i> | 1 | 0.117225 |
| <i>State 1</i> | 1 | 0.165072 | <i>State 2'</i> | 1 | 0.212919 |
| <i>State 2</i> | 1 | 0.260766 | <i>State 3'</i> | 1 | 0.308612 |
| <i>State 3</i> | 1 | 0.356459 | <i>State 4'</i> | 1 | 0.404306 |
| <i>State 4</i> | 1 | 0.452153 | <i>State 5'</i> | 1 | 0.500000 |
| <i>State 5</i> | 1 | 0.547847 | <i>State 6'</i> | 1 | 0.595694 |
| <i>State 6</i> | 1 | 0.643541 | <i>State 7'</i> | 1 | 0.691388 |
| <i>State 7</i> | 1 | 0.739234 | <i>State 8'</i> | 1 | 0.787081 |
| <i>State 8</i> | 1 | 0.834928 | <i>State 9'</i> | 1 | 0.882775 |

7.2.3 Results

The progress of training is evaluated by monitoring the error of the validation results. Here error is defined as Equation 7.2. The mean abstract error between the predicted locations of structural changes y and the Unit 2 of label \hat{y} in each epoch was calculated, and n represents the amount of validation data. The CNN model in each epoch has been saved. The model with minimum error is chosen to be used.

$$E = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (7.2)$$

The model has been trained for 5000 epochs. The error of model in each epoch during training is shown in Fig. 7.4. The model in Epoch 4435 has been used according to its minimum error of 0.254.

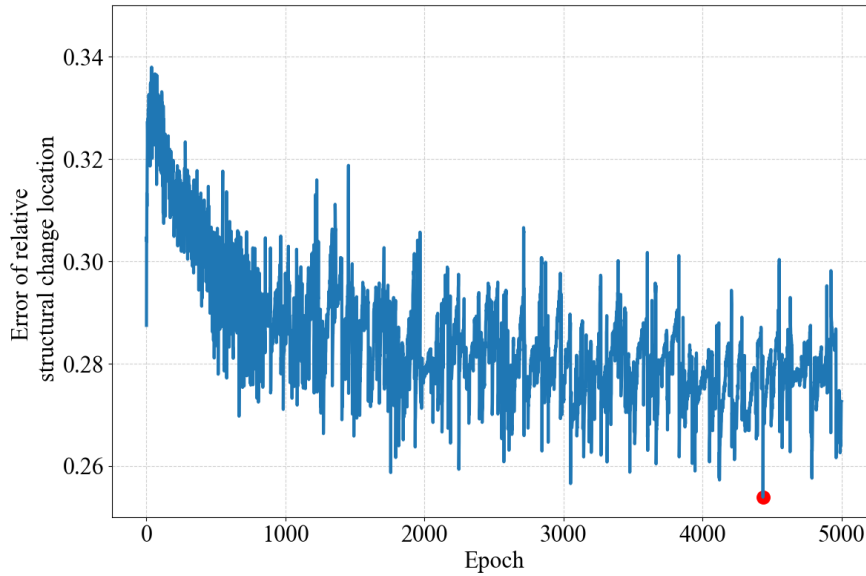


Fig. 7.4 Error of the regression model in each epoch

Fig. 7.5 is the predictions of the training data by using the regression CNN model in Epoch 4435, which shows that the regression CNN model has learned the features of the training data, since in Fig. 7.5 only values close to 0 or 1 are outputted in the Unit 1, and in the Unit 2 all the predicted locations of structural changes are close to the actual locations.

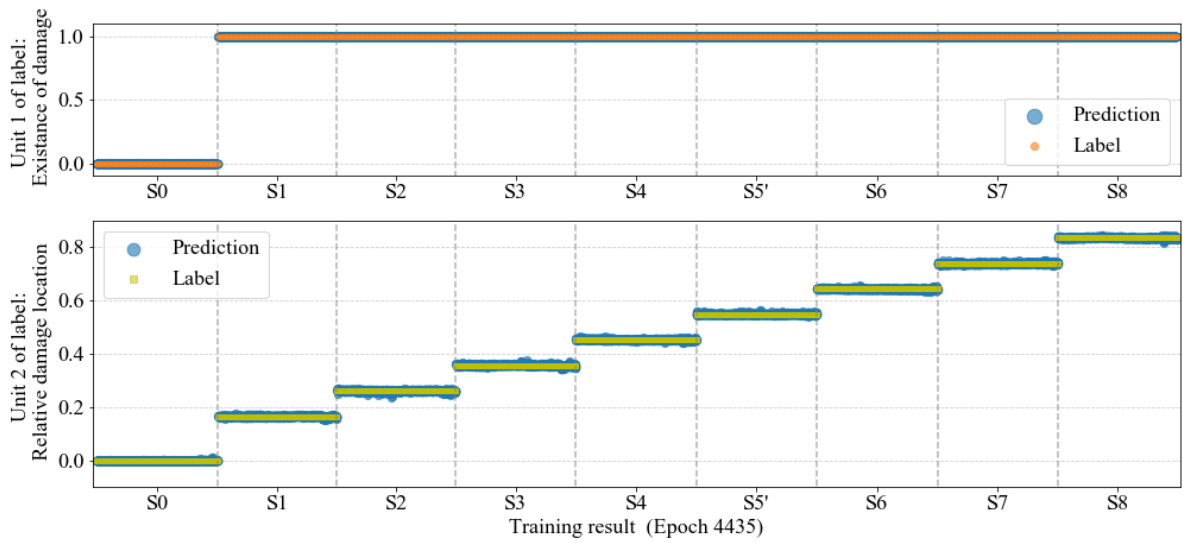


Fig. 7.5 Prediction of the training data

Fig. 7.6 shows the validation results of the regression CNN model in Epoch 4435. Generally, most data were correctly predicted as damaged cases. However, a large number of data in S8' are wrong predicted in intact state. The mean error of the Unit 1 predictions is 0.068. For the Unit 2 of the predictions of the validation data, relative location of the structural damage, data in S4', S5', and S7' show very good results that the predictions are mainly in the areas near the labels, indicating that for the validation data in S4', S5', and S7', the regression CNN model can predict the correct damage locations even though those cases are not trained. The error of the Unit 2 of the predictions of the validation data in S2', S3', S6' and S9' are in a range with 0.2 and 0.4 which are obviously higher than validation results of data in S4', S5', and S7'. The big errors of the predictions are in S1' and S8', since the structural changes on the two ends of the beam. The mean error of the predictions of the relative locations of structural is 0.254.

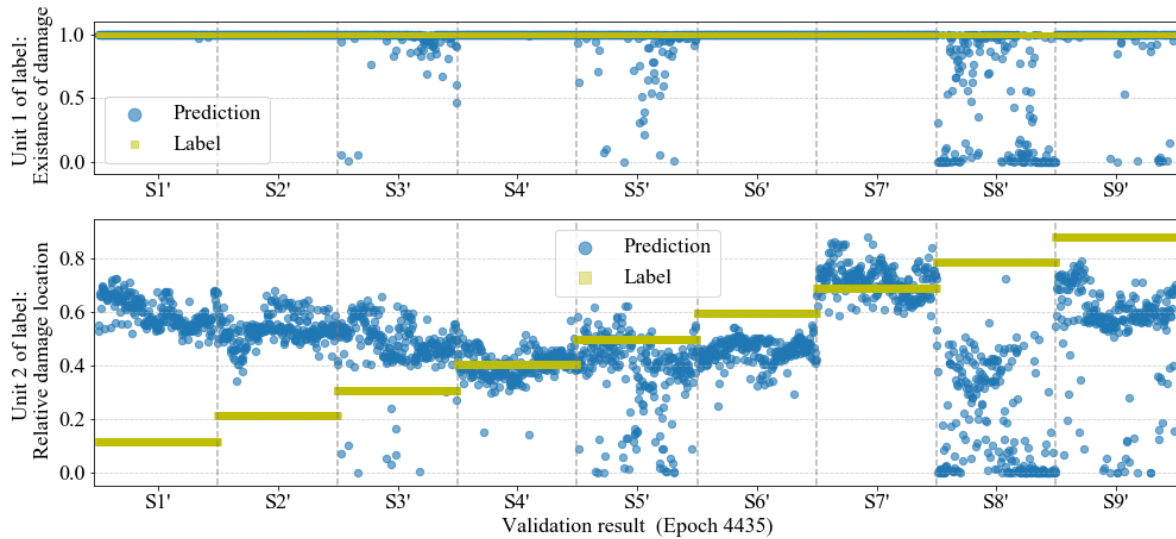


Fig. 7.6 Prediction of the validation data

Here the definition of accurate prediction is identical to that introduced in Subsubsection 6.2.2.1, that the predictions of structural states in adjacent locations of the actual structural state are considered as correct predictions. For example, the correct prediction of the validation data in State 2' can be either State 1 or State 2. In another word, a sample with an error of less than ± 10 cm is considered as a correct prediction. Thus, the amount of correct predictions of the regression CNN model is 607, and the accuracy is 28.22%, which shows better results than the classification CNN models as discussed in Section 6.2.

Comparing the results of the regression CNN model to the results of the classification CNN model introduced in Section 6.2, firstly, better expression capacity of the CNN has been achieved by the regression CNN model, since the locations of structural changes of most data in States 4', 5', and 7' are accurately predicted. While classification CNN model can only predict an approximate location of structural change to the best of its capacity in ideal condition. Secondly, since the regression CNN model use more 20% training data than the classification CNN model introduced in Subsection 6.2.3, the 6.69% accuracy increase shows good agreement of the phenomenon in Chapter 6 that, expanding training data is an effectiveness way to improve the performance of a CNN model. The classification model in Subsection 6.2.3 tends to predict more data as S0 (intact) state than the regression model, and the regression CNN model predicts most data as non-S0 states correctly. Overall, the regression model shows better performance than the classification model when predicting data acquired in untrained states.

Moreover, even though the data in States 1'-3', 6', and 9' are predicted in wrong locations, the data in each state are predicted in a small range of location, and data in different states are generally predicted in different range of locations. Above phenomenon shows the regression CNN model can classify the structural states of data, even though it fails to predict the accurate locations of structural change. In this case, the regression CNN doesn't learn the right features of location of structural change, but it still demonstrates the sensitivity of the regression CNN to tiny structural changes successfully.

7.3 Deep Network with Multi-Convolution Blocks and Multi-Task Outputs

7.3.1 Motivation

Chapter 4 presents the high accuracies of the classification CNN models for identifying tiny local structural changes in trained categories. Chapter 6 shows the low robustness of the classification CNN model identifying untrained structural states. Subsequently, a regression CNN model was proposed in Section 7.2 to improve the expression capacity of the classification CNN model and the robustness. Based on the previous results, in this section, multi-convolution block and multi-task outputs are designed, and a new network with two multi-convolution blocks and the multi-task outputs is proposed to further increase the robustness of CNN for identifying untrained structural states.

One of the advantages of neural network is automatic feature abstraction of data. Thus, CNNs are widely used for feature extraction tasks. For CNNs, convolutional kernels work as receptive fields which are sensitive to corresponding size of features. A neural network with only single-size convolution kernel may not be sensitive to larger or smaller features of data, which may lead to low performance of the neural network. In order to make the networks sensitive to different sizes of features of data, convolutional blocks with different sizes of kernels were proposed for image classifications tasks by Google. Such as GoogLeNet with Inception V1-V4 blocks (Szegedy et al., 2015; Szegedy et al., 2016; Szegedy et al., 2017).

For civil structures, when observing a free vibrating structure, the deflected shape can be expressed in terms of translational displacements at different levels (Clough and Penzien, 2003). The displacement vector v for the structure can be developed by superposing suitable amplitudes of normal modes, which means summing the modal vectors as Eq. 7.3. Here φ_n is the mode-shape vector, and Y_n is the modal amplitude. Eq. 7.4 shows that by using the mode

shapes and corresponding amplitudes, structure dynamic behavior can be expressed. If extend the 1-D weight vector Y_n to a 2-D weight matrix Y_{nt} with a time dimension, the procedure of structural vibration v_t can be described as mode-shape vector ϕ_n multiplied by 2-D weight matrix Y_{nt} . Expression of above approach is very similar to the mechanism of neural network. Since the mechanism of a neural network approximating a function is also mainly based on summing the features of data multiplied by different weights.

$$v = \phi_1 Y_1 + \phi_2 Y_2 + \cdots + \phi_N Y_N = \sum_{n=1}^N \phi_n Y_n \quad (7.3)$$

$$v_t = \sum_{n=1}^N \phi_n Y_{nt} \quad (7.4)$$

If considering vibration data from the view of data science, the normal modes of the structure should be dominant features. Generally, vibration measurement data consists of discrete acceleration samples. To show an example, Fig. 7.7 plots of a free damped vibration data of the steel beam introduced in Section 4.2, and the power spectral density of the measurement data is shown in Fig. 7.8. As we know, power spectral density describes the distribution of power into frequency components composing the signal, and frequency response functions reach extreme values approximately at the natural frequencies. In Fig. 7.8, natural modes can be identified on 184 Hz, 424 Hz, 636 Hz, 670 Hz, and 798 Hz, where have obvious peaks of magnitude and are marked with red scatters. Thus, Fig. 7.8 also can be explained as summing diverse features multiplied by different weights, which has the same concept as neural network.

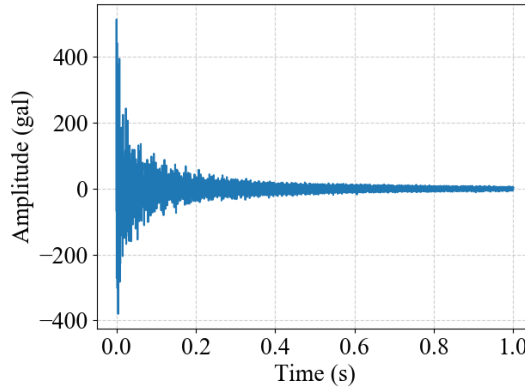


Fig. 7.7 Waveform of a vibration data

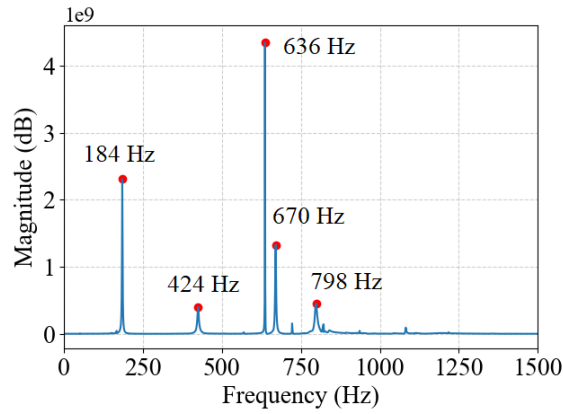


Fig. 7.8 Power spectral density the data

Meanwhile, according to the Nyquist-Shannon sampling theorem, the sampling rate has to be at least twice as high as the maximum frequency of interested modes. It is an evidence to show that: there is a limitation of the size of scoop to identify the interested mode. Thus, the author got the question that: when designing a CNN, should the CNN have different sizes of kernels which are sensitive to different sizes of features of vibration data accordingly?

Therefore, based on the results of previous works as introduced in Chapters 4, 6, and 7, and learning the experience from GoogLeNet, a deep network with multi-convolution blocks and multi-task outputs was proposed to fit the vibration-based structural state identification task, and was validated on the databases of the T-shaped steel beam.

7.3.2 Network details

The detailed architecture of the network is shown in Fig. 7.9. The network consists of three parts: two multi-convolution blocks, hidden layers, and multi-task output layers. The parameters of the three parts are shown in Tables 7.5-7.7 respectively.

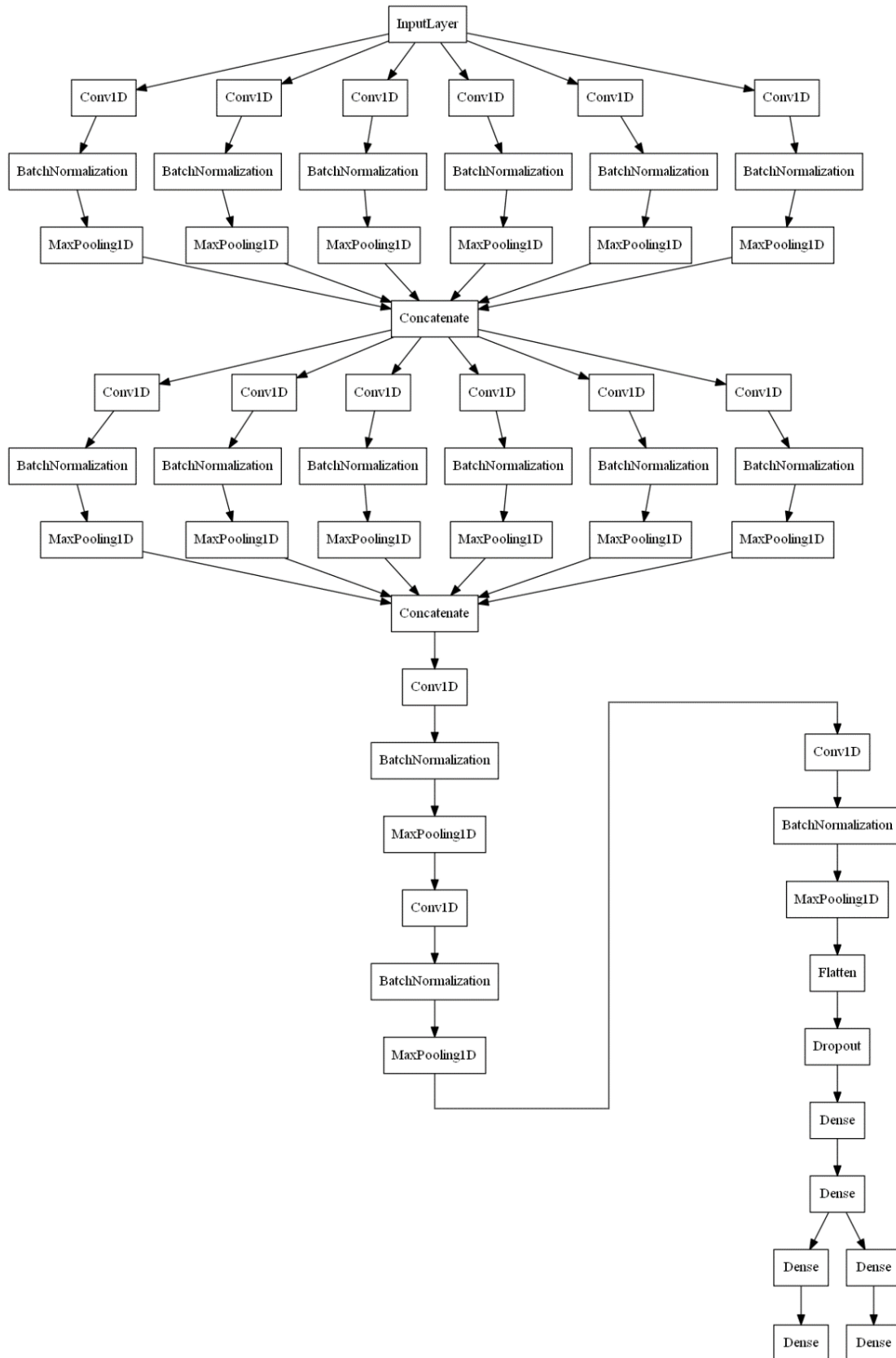


Fig. 7.9 Architecture of the network with multi-convolution blocks and multi-task outputs

Table 7.5 Architecture of the multi-convolution blocks

| | <i>Convolutions</i> | | | | | |
|----------------------------|---------------------|---------------|---------------|---------------|---------------|---------------|
| | <i>Conv_1</i> | <i>Conv_2</i> | <i>Conv_3</i> | <i>Conv_4</i> | <i>Conv_5</i> | <i>Conv_6</i> |
| <i>Kernel length</i> | 10 | 20 | 30 | 50 | 80 | 100 |
| <i>Number of kernel</i> | 5 | 5 | 5 | 5 | 5 | 5 |
| <i>Strides</i> | 1 | 1 | 1 | 25 | 40 | 50 |
| <i>Padding</i> | valid | valid | valid | valid | valid | valid |
| <i>Batch Normalization</i> | True | True | True | True | True | True |
| <i>Max pooling Strides</i> | 3 | 3 | 3 | 3 | 3 | 3 |
| <i>Parameters 1</i> | 475 | 925 | 1375 | 2275 | 3625 | 4525 |
| <i>Parameters 2</i> | 275 | 525 | 775 | 1275 | 2025 | 2525 |
| <i>Total Para.</i> | | | | | | 20,600 |

Table 7.7 Architecture of the multi-task output layers

| <i>Output 1</i> | | | | |
|-------------------------|---------------------|------------------|-------------------|------------------|
| <i>Layer</i> | <i>Output Shape</i> | <i>Parameter</i> | <i>Activation</i> | <i>Variables</i> |
| <i>FC</i> | 32 | None | ReLU | 2080 |
| <i>FC</i> | 16 | None | ReLU | 528 |
| <i>Output 1</i> | 2 | None | Softmax | 34 |
| <i>Output 2</i> | | | | |
| <i>Layer</i> | <i>Layer</i> | <i>Layer</i> | <i>Layer</i> | <i>Layer</i> |
| <i>FC</i> | 32 | None | ReLU | 2080 |
| <i>FC</i> | 16 | None | ReLU | 528 |
| <i>Output 2</i> | 1 | None | Sigmoid | 17 |
| <i>Total parameters</i> | | | | 666,884 |

Table 7.6 Architecture of hidden layers

| <i>Layer</i> | <i>Output Shape</i> | <i>Parameter</i> | <i>Activation</i> | <i>Variables</i> |
|----------------------------|---------------------|---|-------------------|------------------|
| <i>Convolution 1-D</i> | 10522×5 | Kernel number: 5; Kernel size: 10×9; Stride: 1; Padding: Valid | Linear | 255 |
| <i>Batch normalization</i> | 10522×5 | None | None | 20 |
| <i>Max Pooling 1-D</i> | 3507×5 | Strides: 3; | None | 0 |
| <i>Convolution 1-D</i> | 1749×5 | Kernel number: 5; Kernel size: 10×9; Stride: 1; Padding: Valid | Linear | 255 |
| <i>Batch normalization</i> | 1749×5 | None | None | 20 |
| <i>Max Pooling 1-D</i> | 583×5 | Strides: 3; | None | 0 |
| <i>Convolution 1-D</i> | 192×5 | Kernel number: 5; Kernel size: 10×9; Stride: 1; Padding: Valid | Linear | 255 |
| <i>Batch normalization</i> | 192×5 | None | None | 20 |
| <i>Max Pooling 1-D</i> | 64×5 | Strides: 3; | None | 0 |
| <i>Flatten</i> | 320 | None | None | 0 |
| <i>Dropout</i> | 320 | Rate: 0.25 | None | 0 |
| <i>FC</i> | 128 | None | ReLU | 41,088 |
| <i>FC</i> | 64 | None | ReLU | 8259 |
| <i>Total parameters</i> | | | | 50,172 |

The multi-convolution block consists of 6 parallel convolutional units with different kernel sizes and strides to fit different sizes of features, as shown in Table 7.5. The kernel lengths are in a range between 10 to 100. The first three convolutional units are designed for capturing small features of sizes, with only 1 stride. The last three convolutional units are designed for capturing large features, and the strides are set to half of the kernel length. Finally, the 6 parallel convolutional units are concatenated vertically to output. In the proposed network, two multi-convolution blocks have been used in the beginning.

The hidden layers are common layers of a typical feed-forward neural network, as shown in Table 7.6. Data flows through the layers straightly. Three convolutional units stack together in front of the flatten layer with 0.25 dropout ratio. Then data propagates to the two FC dense layers.

The multi-task outputs are updated according to the new label encoding, as shown in Fig. 7.10 and Table 7.7. One output is in one-hot encoding for a binary classification task, indicating the structure in intact or changed state. The other output has only one unit, representing the relative location of structural change. The detailed label encodings of each structural state are shown in Table 7.8.

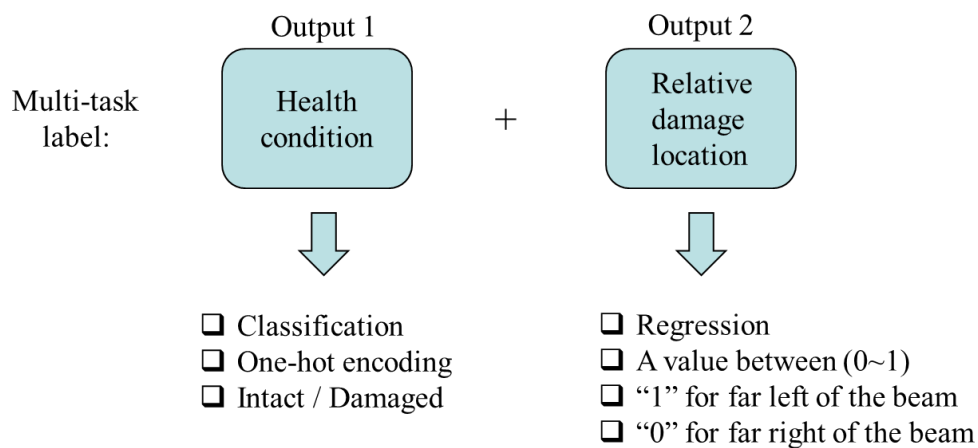


Fig. 7.10 Label encodings of the network with multi-task outputs

Table 7.8 Encoding of labels

| <i>Category</i> | <i>Label</i> | | | <i>Category</i> | <i>Label</i> | | |
|-----------------|---------------|---|---------------|-----------------|---------------|---|---------------|
| | <i>Task 1</i> | | <i>Task 2</i> | | <i>Task 1</i> | | <i>Task 2</i> |
| <i>State 0</i> | 1 | 0 | 0 | <i>State 1'</i> | 0 | 1 | 0.117225 |
| <i>State 1</i> | 0 | 1 | 0.165072 | <i>State 2'</i> | 0 | 1 | 0.212919 |
| <i>State 2</i> | 0 | 1 | 0.260766 | <i>State 3'</i> | 0 | 1 | 0.308612 |
| <i>State 3</i> | 0 | 1 | 0.356459 | <i>State 4'</i> | 0 | 1 | 0.404306 |
| <i>State 4</i> | 0 | 1 | 0.452153 | <i>State 5'</i> | 0 | 1 | 0.500000 |
| <i>State 5</i> | 0 | 1 | 0.547847 | <i>State 6'</i> | 0 | 1 | 0.595694 |
| <i>State 6</i> | 0 | 1 | 0.643541 | <i>State 7'</i> | 0 | 1 | 0.691388 |
| <i>State 7</i> | 0 | 1 | 0.739234 | <i>State 8'</i> | 0 | 1 | 0.787081 |
| <i>State 8</i> | 0 | 1 | 0.834928 | <i>State 9'</i> | 0 | 1 | 0.882775 |

There are two different loss functions for the two output tasks. The loss function of the first output (structural state) is categorical cross entropy, named L_1 . The loss function of the second output (relative location of structural change) is mean squared error, named L_2 . The total loss of the network is the sum of the two losses multiplied by different weights. In the network, the total loss is calculated as Equation 7.5. The weights 1 and 3 are determined by trial and error. As the relative locations of structural changes are paid more attention, when descending the gradient of the CNN, the loss of relative locations of structural changes (Unit 2) should have high weight than the structural state (Unit 1). Therefore, the multi-task outputs were designed to manually assign the ratio of each loss component of the two tasks.

$$L = L_1 \times 1 + L_2 \times 3 \quad (7.5)$$

The optimizer of the network is SGD. The learning rate is 0.0005 with decay of 1e-6, and momentum of 0.9 with Nesterov momentum applied.

All the weights in the network are initialized with Variance Scaling initializer, and all the biases are initialized with zeros.

7.3.3 Validation set up

The validation set up is totally identical to those introduced in Section 7.2. The training data and validation data are acquired from the T-shaped steel beam. Local structural changes are different between the training data and the validation data. The locations of structural changes are shown in Fig. 7.3. The detailed data distributions of training data and validation data are shown in Tables. 7.2 and 7.3.

The model has been trained for 9800 epochs. The error of the average relative location of local structural changes of model in each epoch has been monitored. The definition of error is identical to that introduced in Section 7.2, that a sample with an error of less than ± 10 cm are considered a correct prediction. The models with local minimum error are used for the structural state identification

7.3.4 Results

The error of the model in each epoch is shown in Fig. 7.11. There are two obvious local minimum errors which are marked with red scatters in Fig. 7.11. One is the model of Epoch 2824 with 0.160 error, and the other is the model of Epoch 5709 with 0.155 error. Therefore, above two models are used for validating the robustness of the network. The training results and validation results of the above two models are shown in Figs. 7.12-7.15. The amounts of accurate predictions and accuracies are summarized in Table 7.9.

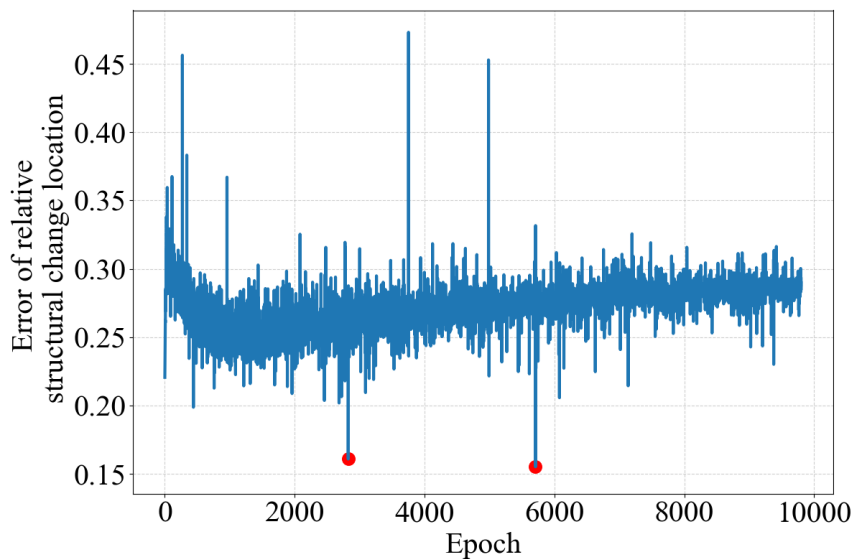


Fig. 7.11 Error of model in each epoch

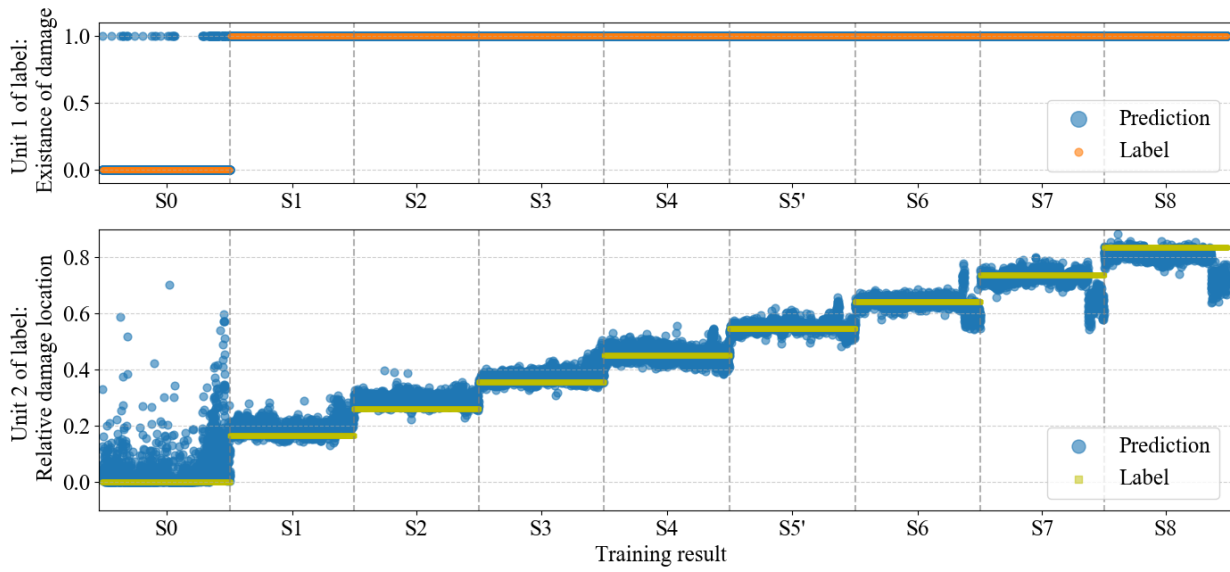


Fig. 7.12 Training result of the model in Epoch 2824

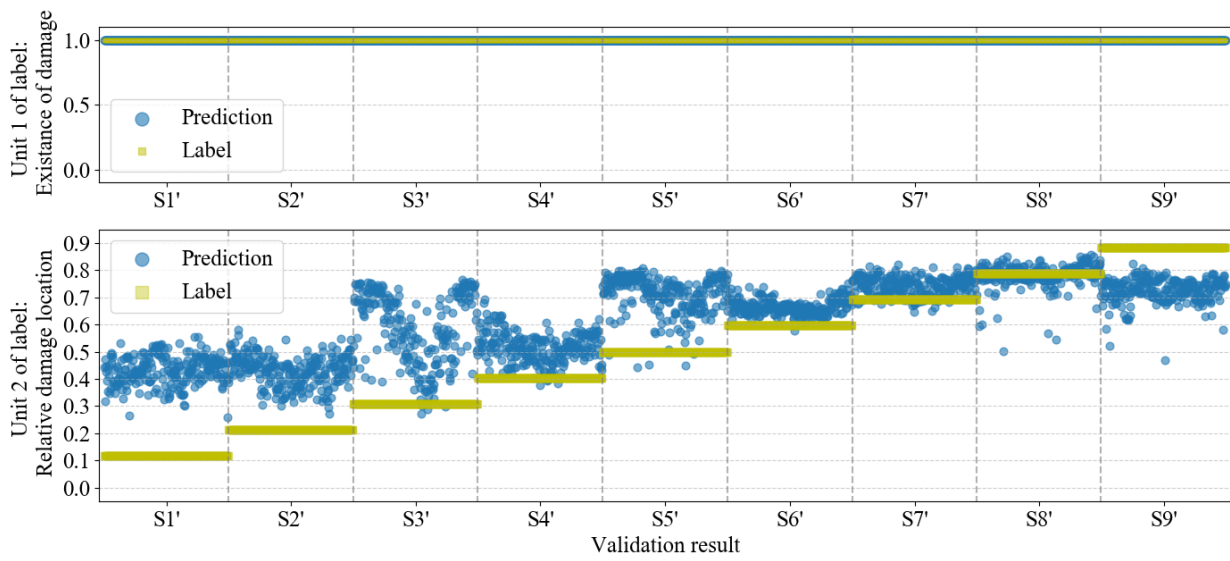


Fig. 7.13 Validation result of the model in Epoch 2824

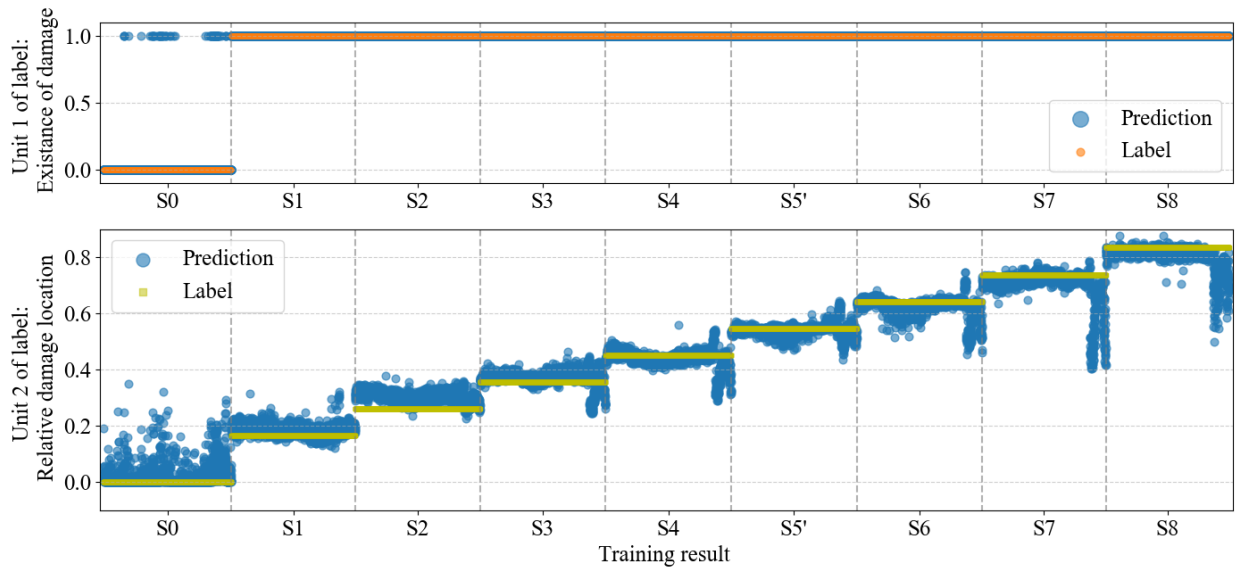


Fig. 7.14. Training result of the model in Epoch 5709

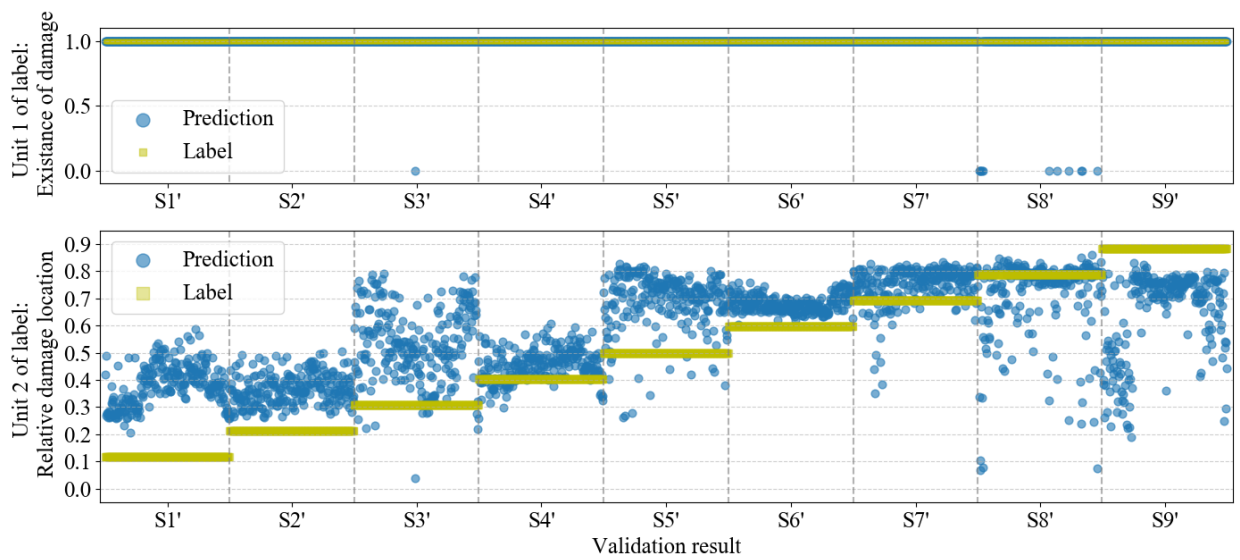


Fig. 7.15 Validation result of the model in Epoch 5709

Table 7.9. Accuracies of the validation results

| <i>Model</i> | <i>Amount of accurate prediction</i> | <i>Accuracy (%)</i> |
|----------------|--------------------------------------|---------------------|
| 1 | 1343 | 62.43 |
| 2 | 1481 | 68.85 |
| <i>Average</i> | 1422 | 65.64 |

Comparing the validation results of the networks with multi-convolution blocks and multi-task outputs to the regression CNN model discussed in Section 7.2, the accuracies are obviously increased from 28.22% to 65%. More data can be predicted accurately, such as data in States 6' and 8'. The results show the improvement of the performance of network for structural state identification. Meanwhile, the feasibility of using the multi-convolution blocks and multi-task outputs for structural state identification task is proved.

7.4 Conclusions

This chapter aims to improve the expression capacity and the robustness of the classification CNN model to identify untrained local structural changes. Several updates of the network are proposed.

Firstly, a regression CNN model is proposed, which redefines the encoding of labels and corresponding output layer in the CNN model. The results of the robustness tests show higher accuracy than the classification CNN models as shown in Chapter 6, which proves the effectiveness of using regression form of label and output layer for structural state identification task.

Secondly, to further improve the robustness of the CNN model, the CNN is going deeper with multi-convolution blocks and multi-task outputs. Higher results of the robustness tests are obtained comparing to all the previous results. The accuracy reaches 65%, which shows the effectiveness and rationality of using multi-convolution block and multi-task outputs in the CNN for structural state identification.

Chapter 8

Conclusions and Future Works

8.1 Conclusions

The rapid deterioration of the bridges has become a major global problem. To address on the above problem, this study proposed a vibration-based structural state identification method by using Deep Learning. This study aims to boost the development of new Deep-Learning-based SHM and SDD schemes, and promote the development of technology that guarantees human's daily safety.

In Chapter 1, the backgrounds of rapid deterioration of the infrastructure, SHM and SDD technologies, Deep Learning in civil engineering, conventional vibration-based SDD methodologies, and Deep-Learning-based SDD methodologies are introduced respectively. The purposes and layout of the thesis are also explained.

In Chapter 2, a series of vibration experiments are performed on a ballasted multi-span concrete railway bridge. In the first eight experiments, the railways and sleepers have been removed. In the last experiment, all the subsidiary elements (ballasts, walkways, handrails) of the bridge are removed. The acceleration data are acquired, and dynamic parameters are identified. Several conclusions are drawn as follows:

- First, all the basic dynamic parameters, such as natural frequencies, damping ratios, and mode shapes, are not stable indicators to present accurate structural state. The dynamic parameters are easily biased by the environmental changes, measurement errors, or other uncertainties. The robustness of the dynamic parameters to the temperature effect is very low. Thus, the dynamic parameters are difficult to be used as indicators of small local structural changes.
- Second, only arbitrary signal processing methods have been used to analyze the vibration data. The vibration data cannot be used effectively. Designing an indicator for identifying small local structural changes by using Deep Learning is highly needed.

In Chapter 3, the basic concept of Deep Learning is introduced. Subsequently, layers of neural network, mechanisms of learning, loss functions, optimizers, are explained in detail. Then, the layout of using Deep Learning for structural state identification is interpreted. Meanwhile, the mechanisms of SVM and T-SNE are also explained.

In Chapter 4, a 1-D CNN is developed to identify structural states by using raw vibration data. The proposed CNN is validated on three independent databases, based on a T-shaped steel beam (in lab), a short steel girder bridge (in test field), and a long steel girder bridge (in service) respectively. The structural states are varied by affixing additional elements on the structures. The proposed 1-D CNN accurately identified the different structural states in all the databases.

- First, the proposed 1-D CNN model is very sensitive to tiny changes in the local stiffness and mass values. Very high accuracies (99.79%, 99.36% and 97.23% in the T-shaped beam, the short steel girder bridge, and the long steel girder bridge experiments, respectively) are obtained even by the very simple CNN architecture. Such sensitivity demonstrates the high applicability potential of 1-D CNN models to detect structural damages in actual bridges in future phases.
- Second, the CNN-based method identifies structural states without pre-processing, post-processing, or manual feature extraction of the acceleration data. The raw acceleration data are directly input to the CNN. Feature extraction and structural state identification are also automated by the CNN.
- Third, the CNN has good inclusiveness for different excitation methods, since the vibration data acquired on the long steel girder bridge is excited by white noise with the component of traffic load.

In Chapter 5, the mechanism of the proposed CNN is investigated by visualizing the convolutional kernels and the outputs of the convolutional layer and the max pooling layer. Meanwhile, variation of the data structure flowing through the CNN via each layer is visualized by the T-SNE method. Following conclusions are drawn from the results:

- First, it is found that the convolutional kernels learn the vibration features with approximately the same frequencies but different amplitudes in each channel. The convolutional kernels perform as band-pass filters that properly fit the target structure and vibration data of the SDD problem.
- Second, the convolutional and max pooling layers extracted and maintained the features but adjusted the amplitudes of the modes, improving the expressions of the features.

- Third, by observing the variation of data structure flowing through the CNN via each layer visualized by the T-SNE method, it is found that clusters of data in each structural state can be formed, and clear boundary between each structural state can be determined. The variation of data structure proves the effectiveness of using CNN for structural state identification.

In Chapter 6, robustness of the proposed CNN to the locations of structural changes and temperature effect are tested. In the beginning, two rounds of tests are performed on the T-shaped steel beam to evaluate the robustness of the CNN to the location of structural change. The vibration data acquired from the T-shaped steel beam in untrained structural states have been used for examining the robustness of the classification CNN models. In the first round of robustness tests, the test data were fed into the CNN models trained in Section 4.2. In the second round of robustness tests, new CNN models trained by expanded database were used. Subsequently, the robustness of the proposed CNN to the temperature effect is investigated by feeding the test data acquired in different temperature conditions into the existed CNNs of the short steel girder bridge. Also a solution is proposed to improve the robustness of the CNN to the temperature effect. The conclusions are as follows:

- First, the classification CNN models has very low robustness to locations of structural changes, that the CNNs cannot have good performance to predict the approximate locations of the untrained structural changes. When the training data is not sufficient, the CNNs tend to predict the data in intact structural state (State 0).
- Second, increasing the amount of training data can improve the robustness of the CNN to the locations of structural changes. However, if only small amounts of training data are extended, the improvement of CNN model performance is also very limited. Therefore, updates of the architecture of the CNN model are needed.
- Third, the robustness of the CNN model to temperature effect is very low. Fortunately, the robustness to temperature effect can be easily improved by expanding the training data acquired in a wide range of temperature conditions. Thus, the difficulty of the conventional SDD method which caused by the low robustness of the dynamic parameters to the temperature effect can be overcome by using the CNN based structural state identification method.

In Chapter 7, to improve the performance of the CNN model for identifying local structural changes in untrained locations, some updates of the CNN model are proposed. Firstly, a regression CNN model is proposed which changes the output layer and corresponding label encoding of the classification CNN model introduced in Chapter 4. Secondly, to further improve the performance of the CNN model, a deep network with two multi-convolution blocks and multi-task outputs is proposed. Validations are carried out on the networks. Some conclusions are drawn as follows:

- First, the regression CNN model predict the locations of structural changes of many data in the inner area of the beam successfully. However, the structural changes in the two ends of the beam tend to be predicted with big errors. Overall, the regression CNN model has better expression capacity than the classification CNN model when predict the data in untrained categories, indicating the effectiveness of the regression encoding of label.
- Second, comparing the deep network to the classification CNN model in Chapter 4 and the regression CNN model in Section 7.2, the performance and the robustness of the deep network increase obviously. The effectiveness and feasibility of using the multi-convolution blocks and multi-task outputs for structural state identification are proved.

8.2 Future Works

Future work will target the limitations of the proposed structural-state identification method.

First, the resolution of detecting local changes in the structural state by the proposed CNN must be clarified.

Second, the experiments in this thesis have been performed by short-term measurements in relatively stable conditions, unlike most bridge SHM projects which measure the vibration continuously with more uncertainties. Thus, for future applications of the proposed CNN on long-term monitored bridges, representative training data could be acquired in various conditions to test the robustness of the CNN model.

Third, all supervised learning cases require a large volume of labeled training data to build a high-performance CNN. In other words, based on the success of supervised learning by the proposed CNN method, the labeled data burden of the CNN must be reduced by developing semi-supervised or unsupervised learning algorithms for SDD problems.

Fourthly, no vibration data acquired on actual damaged structural is used in these study, since the high difficulty of acquirement. To address this problem, vibration data of actual damaged structural states will be simulated by using FE models.

Finally, this thesis only proposed a method of identifying a single location of structural change. The method of identifying multiple structural changes by using Deep Learning should be updated in future works.

Acknowledgement

The author gratefully acknowledges the financial support of the **Ministry of Education, Culture, Sports, Science and Technology, Japan (MEXT)**, which sponsored my study in Japan.

I would like to express my gratitude to many people who supported this work and helped me during my doctoral period.

Prof. **Yasunori Miyamori** instructed my essential accomplishment of doing research. His influences made me strongly interested in the research of structural health monitoring and structural damage detection. It is also him who made me feel that programming is very cool, and inspired my interest in programming. He encouraged me to participate many academic activities, such as APESS summer school, international and domestic conferences, and collaboration research projects. Participating in the academic activities improved my skills on presentation and sociability, and broadened my horizon. He also spent significant amount of time co-authoring the publications. I am also very grateful to him for his helpful counsel and for serving my committee.

I would like to thank the rest of my thesis committee. The kind support of committee members Prof. **Mayuko Nishio**, Prof. **Yoshihiko Hayakawa**, Prof. **Shuichi Mikami**, Prof. **Masumi Inoue**, and Prof. **Heesup Choi** are greatly appreciated.

I am very grateful to Prof. **Mayuko Nishio** for her instruction and valuable comments of proposal writing. Also, I feel very happy to have the good memories of her lectures and the experiment she supervised in APESS 2017.

I also want to thank Prof. **Yoshihiko Hayakawa** for his invitation of presentation in his laboratory. It is a very good opportunity to have interdisciplinary communication and obtain professional suggestions and discussions.

The professors in Civil and Environmental Engineering Department are greatly acknowledged. Prof. **Shuichi Mikami** gave valuable comments on my research, provided the sensor system, and supported me to perform the vibration experiment on Hokushin Bridge with his laboratory members. Prof. **Masumi Inoue** and Prof. **Heesup Choi** encouraged me positively and continuously. The help of Prof. **Masumi Inoue** after the earthquake is

unforgettable. Meanwhile, many suggestions on my career are obtained from Prof. **Heesup Choi**.

Dr. **Tomoyuki Yamazaki** and **Yukina Hashizume** provided much technical support on the sensor system, and the vibration experiments.

I would like to thank my past co-workers: Dr. **Takehiko Saito** and **Shuta Shirakawa** for the Mukagawa Bridge vibration experiments in the beginning phase of my doctoral course.

I appreciate the technical support and valuable discussions of machine learning from my friend **Huangyi Qin**, and my wife **Ruijing Yang**.

Last but not least, I really appreciate the support of my family members. The love, trust, and encouragement from **my wife**, **my parents**, and **my grandparents** are driving force behind my achievements.

List of Publications

| | | |
|--|---|----------|
| ● Articles in refereed International Journals | : | 2 |
| ● Articles in refereed International Conferences | : | 4 |
| ● National Conference | : | 1 |
| ● Local Conferences | : | 3 |

International Journals:

1. **Zhang Y.**, Miyamori Y., Mikami S., & Saito T. (2019). Vibration-based Structural State Identification by a 1-Dimensional Convolutional Neural Network. *Computer-Aided Civil and Infrastructure Engineering*. DOI: 10.1111/mice.12447.
Impact Factor: 6.208
2. **Zhang Y.**, Miyamori Y., Kadota T., & Saito T. (2017). Investigation of Seasonal Variations of Dynamic Characteristics of a Concrete Bridge by Employing a Wireless Acceleration Sensor Network System. *Sensors and Materials*, 29(2), 165-178.
Impact Factor: 0.519

International Conferences:

1. **Zhang Y.**, Miyamori Y., Saito T., Mikami S., & Oshima T. (2019). Robustness Tests of a Vibration-based Structural State Identification Method Through a 1-D Convolutional Neural Network. “9th International Conference on Structural Health Monitoring of Intelligent Infrastructure Conference (SHMII9)”, Missouri, USA, August, 2019.
2. **Zhang Y.**, Miyamori Y., Mikami S., & Saito T. (2018). Vibration-based Structural Damage Detection by Deep Learning Method on a Small Steel Bridge Structure. Proceedings in 7th World Conference on Structural Control and Monitoring (7WCSCM), Qingdao, China, 2018.
3. **Zhang Y.**, Miyamori Y., Oshima T., Shirakawa Y., Mikami S., & Saito T. (2017). Effect of Ballast State on Natural Frequencies of a Multi-Span Ballasted Prestressed Concrete Railway Bridge. Proceeding in *Structural Health Monitoring of Intelligent Infrastructure Conference 2017 (SHMII 8)*, Brisbane, Australia, 2017.

4. **Zhang Y.**, Miyamori Y., Shirakawa Y., & Saito T. (2017). Significant Dynamic Parameter Variations of a Ballasted Railway Bridge Investigated by Employing a Wireless Smart Sensor Network System. Proceedings in *The 13th International Workshop on Advanced Smart Materials and Smart Structures Technology (ANCRiSST2017)*, July 22-23, The University of Tokyo, Japan, 2017.

National Conference:

1. **Zhang Y.**, Miyamori Y., Kadota T., Shirakawa Y., & Saito T. (2017). Study on Dynamic Characteristic Variations of a Ballasted Railway Bridge. Japan Society of Civil Engineering Annual Conference 2017, Fukuoka, Japan, 2017.

Excellent presentation award

Local Conferences:

1. **Zhang Y.**, Miyamori Y., Saito T. & Mikami S., (2019). Robustness Test of a CNN-Based Structural State Identification Method. Japan Society of Civil Engineering Hokkaido Branch Annual Conference 2018, Tomakomai, Japan, 2019.
2. **Zhang Y.**, Miyamori Y., Saito T. & Mikami S., (2018). Identification of Structural States by Acceleration Data based on a Convolutional Neural Network. Japan Society of Civil Engineering Hokkaido Branch Annual Conference 2017, Sapporo, Japan, 2018.
3. **Zhang Y.**, Miyamori Y., Kadota T. (2017). Dynamic Parameter Variations of a 5-Span Prestressed Railway Bridge. Japan Society of Civil Engineering Hokkaido Branch Annual Conference 2016, Kitami, Japan, 2017.

Excellent presentation award

References

- AASHTO. (2008), *Bridging the gap—restoring and re-building the nation's bridges*, American Association of State Highway and Transportation Officials, Washington, D.C.
- Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., et al.. (2016), Tensorflow: Large-scale machine learning on heterogeneous distributed systems, *arXiv preprint arXiv:1603.04467*.
- Abadi, M., Barham, P., Chen, J., Chen, Z., et al.. (2016), TensorFlow: A system for large-scale machine learning, In *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16)*, Savannah, GA, USA, **16**, 265-283.
- Abdeljaber, O., Avci, O., Kiranyaz, M. S., Boashash, B., Sodano, H., & Inman, D. J. (2018), 1-D CNNs for structural damage detection: verification on a structural health monitoring benchmark data. *Neurocomputing*, **275**, 1308-1317.
- Abdeljaber, O., Avci, O., Kiranyaz, S., Gabbouj, M., & Inman, D. J. (2017), Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks, *Journal of Sound and Vibration*, **388**, 154-170.
- Adeli, H., & Jiang, X. (2008). *Intelligent Infrastructure: Neural Networks, Wavelets, and Chaos Theory for Intelligent Transportation Systems and Smart Structures*, CRC Press, 305-332.
- Alampalli, S. (1998). Influence of in-service environment on modal parameters. *Proceedings-SPIE The International Society for Optical Engineering*. Colorado, American.
- Amezquita-Sanchez, J. P., Park, H. S., & Adeli, H. (2017). A novel methodology for modal parameters identification of large smart structures using MUSIC, empirical wavelet transform, and Hilbert transform. *Engineering Structures*, **147**, 148-159.
- Avci, O., Abdeljaber, O., Kiranyaz, S., Hussein, M., & Inman, D. J. (2018), Wireless and real-time structural damage detection: a novel decentralized method for wireless sensor networks, *Journal of Sound & Vibration*, **424**, 158-172.
- Barile, C., Casavola, C., Pappalettera, G., & Pappalettere, C. (2016), Analysis of crack propagation in stainless steel by comparing acoustic emissions and infrared thermography data, *Engineering Failure Analysis*, **69**, 35-42.

- Boller, C., Chang, F., and Fujino Y. (2009) Encyclopedia of Structural Health Monitoring, Vol. 1., Wiley, West Sussex.
- Cauchy, A. (1847). Méthode générale pour la résolution des systemes d'équations simultanées. Comp. Rend. Sci. Paris, **25**(1847), 536-538.
- Cawley, P., & Adams, R. D. (1979), The location of defects in structures from measurements of natural frequencies, *The Journal of Strain Analysis for Engineering Design*, **14**(2), 49-57.
- Cha, Y. J., Choi, W., & Büyüköztürk, O. (2017), Deep learning-based crack damage detection using convolutional neural networks, *Computer-Aided Civil and Infrastructure Engineering*, **32**(5), 361-378.
- Cha, Y. J., Choi, W., Suh, G., Mahmoudkhani, S., & Büyüköztürk, O. (2017), Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types, *Computer-aided Civil and Infrastructure Engineering*, **00**, 1-17.
- Chang, K. C., & Kim, C. W. (2016), Modal-parameter identification and vibration-based damage detection of a damaged steel truss bridge, *Engineering Structures*, **122**, 156-173.
- Chollet, F. (2015), Keras, <https://github.com/keras-team>.
- Clough, R.W., Penzien, J. (1995), Dynamics of Structures. P 220. Computers & Structures, Inc. University Ave. Berkeley, CA 94704 USA
- Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, **20**(3), 273-297.
- Crossbow. (2007). Imote2 High-Performance Wireless Sensor Network Node, <http://www.xbow.com>.
- Cruz, P. J. S., & Salgado, R. (2008), Performance of vibration-based damage detection methods in bridges, *Computer-Aided Civil and Infrastructure Engineering*, **24**(2008), 62-79.
- Deraemaeker, A., Reynders, E., De Roeck, G., & Kullaa, J. (2008). Vibration-based structural health monitoring using output-only measurements under changing environment. *Mechanical systems and signal processing*, **22**(1), 34-56.
- Döhler, M., Hille, F., Mevel, L., & Rücker, W. (2014), Structural health monitoring with statistical methods during progressive damage validation of S101 Bridge, *Engineering Structures*, **69**, 183-193.

- Dutta, A., & Talukdar, S. (2004), Damage detection in bridges using accurate modal parameters, *Finite Elements in Analysis and Design*, **40**(3), 287-304.
- Farahani, R. V., & Penumadu, D. (2016), Damage identification of a full-scale five-girder bridge using time-series analysis of vibration data, *Engineering Structures*, **115**, 129-139.
- Gao, Y., & Mosalam, K. M. Deep Transfer Learning for Image-Based Structural Damage Recognition. *Computer-Aided Civil and Infrastructure Engineering*, **00**, 1-21.
- Goldberg, Y. (2017), Neural network methods for natural language processing, *Synthesis Lectures on Human Language Technologies*, **10**(1), 1-309.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016), *Deep Learning*, MIT Press, 324-325.
- Graves, A., & Jaitly, N. (2014), Towards end-to-end speech recognition with recurrent neural networks, *International Conference on Machine Learning*, Beijing, China, 1764-1772.
- Graves, A., Mohamed, A. R., & Hinton, G. (2013), Speech recognition with deep recurrent neural networks, *IEEE International Conference on Acoustics, Speech and Signal Processing*, Vancouver, Canada, 6645-6649.
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).
- Hu, W. H., Caetano, E., & Álvaro Cunha. (2013), Structural health monitoring of a stress-ribbon footbridge, *Engineering Structures*, **57**(4), 578-593.
- Huang, Y., & Beck, J. L. (2018). Full Gibbs Sampling Procedure for Bayesian System Identification Incorporating Sparse Bayesian Learning with Automatic Relevance Determination. *Computer-Aided Civil and Infrastructure Engineering*, **33**(2018), 712-730.
- Ioffe, S., & Szegedy, C. (2015). Batch normalization: accelerating deep network training by reducing internal covariate shift, In *Proceedings of the 32nd International Conference on International Conference on Machine Learning*, Lille, France, 448-456.
- ISHMP. (2009). SHM-H Board High-Sensitivity Accelerometer Sensor Board Datasheet and User's Guide, <http://shm.cs.uiuc.edu/>.
- Khuc, T., & Catbas, F. N. (2017), Completely contactless structural health monitoring of real-life structures using cameras and computer vision, *Structural Control & Health Monitoring*, **24**(1).

- Kim, D., Oh, B. K., Park, H. S., Shim, H. B., & Kim, J. (2017). Modal identification for high-rise building structures using orthogonality of filtered response vectors. *Computer-Aided Civil and Infrastructure Engineering*, **32**(12), 1064-1084.
- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization, *arXiv preprint arXiv:1412.6980*.
- Koziarski, M., & Cyganek, B. (2017). Image recognition with deep neural networks in presence of noise—dealing with and taking advantage of distortions. *Integrated Computer-Aided Engineering*, **24**(4), 337-349.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012), Imagenet classification with deep convolutional neural networks, *International Conference on Neural Information Processing Systems*, Nevada, 1097-1105.
- Kumar, P. R., Oshima, T., Yamazaki, T., Mikami, S., & Miyamouri, Y. (2012), Detection and localization of small damages in a real bridge by local excitation using piezoelectric actuators, *Journal of Civil Structural Health Monitoring*, **2**(2), 97-108.
- LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition, *Proceedings of the IEEE*, **86**(11), 2278-2324.
- LeCun, Y., & Bengio, Y. (2003). Convolutional networks for images, speech, and time series, In *The handbook of brain theory and neural networks*, MIT Press, 276-279.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015), Deep learning, *Nature*, **521**(7553), 436-444.
- Li, Z., Park, H. S., & Adeli, H. (2017). New method for modal identification of super high-rise building structures using discretized synchrosqueezed wavelet and Hilbert transforms. *The Structural Design of Tall and Special Buildings*, **26**(3), e1312.
- Lin, T. Y., RoyChowdhury, A., & Maji, S. (2015). Bilinear cnn models for fine-grained visual recognition, In *Proceedings of the IEEE International Conference on Computer Vision*, Santiago, Chile, 1449-1457.
- Lin, Y. Z., Nie, Z. H., & Ma, H. W. (2017), Structural damage detection with automatic feature extraction through deep learning, *Computer Aided Civil and Infrastructure Engineering*, **32**(12), 1025-1046.
- Liou, C. Y., & Jeng, T. S. (1989). The determination of mode shapes from modern cross-spectral estimates. *Mechanical Systems and Signal Processing*, **3**(3), 291-303.

- Luong, T., Pham, H., & Manning, C. D. (2015), Effective Approaches to Attention-based Neural Machine Translation, In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, 1412-1421.
- Maaten, L. V. D., & Hinton, G. (2008), Visualizing data using t-SNE, *Journal of Machine Learning Research*, **9**(2008), 2579-2605.
- MLIT. *White paper on land, infrastructure, transport and tourism in Japan, 2017*. (2018), Ministry of Land, Infrastructure, Transport and Tourism, Japan
- Moughty, J. J., & Casas, J. R. (2017). A state of the art review of modal-based damage detection in bridges: development, challenges, and solutions, *Applied Sciences*, **7**(5), 510.
- Nabian, M. A., & Meidani, H. (2018). Deep Learning for Accelerated Seismic Reliability Analysis of Transportation Networks. *Computer-Aided Civil and Infrastructure Engineering*, **33**(6), 443-458.
- Nair, V., & Hinton, G. E. (2010), Rectified linear units improve restricted Boltzmann machines, In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, Haifa, Israel, 807-814.
- Ndambi, J. M., Vantomme, J., & Harri, K. (2002), Damage assessment in reinforced concrete beams using eigenfrequencies and mode shape derivatives, *Engineering Structures*, **24**(4), 501-515.
- Oh, B. K., Kim, D., & Park, H. S. (2017). Modal response-based visual system identification and model updating methods for building structures. *Computer-Aided Civil and Infrastructure Engineering*, **32**(1), 34-56.
- Oshima, T., Miyamori, Y., Mikami, S., Yamazaki, T., Beskhyroun, S., & Kopacz, M. F. (2013). Small damage detection of real steel bridge by using local excitation method. *Journal of Civil Structural Health Monitoring*, **3**(4), 307-315.
- Pandey, A. K., Biswas, M., & Samman, M. M. (1991), Damage detection from changes in curvature mode shapes, *Journal of Sound & Vibration*, **145**(2), 321-332.
- Perez-Ramirez, C. A., Amezcuita-Sanchez, J. P., Adeli, H., Valtierra-Rodriguez, M., Camarena-Martinez, D., & Romero-Troncoso, R. J. (2016). New methodology for modal parameters identification of smart civil structures using ambient vibrations and synchrosqueezed wavelet transform. *Engineering Applications of Artificial Intelligence*, **48**, 1-12.

- Rafiei, M. H., & Adeli, H. (2016). A novel machine learning model for estimation of sale prices of real estate units. *Journal of Construction Engineering and Management*, **142**(2), 04015066.
- Rafiei, M. H., & Adeli, H. (2017). A novel machine learning-based algorithm to detect damage in high-rise building structures. *The Structural Design of Tall and Special Buildings*, **26**(18), e1400.
- Rafiei, M. H., & Adeli, H. (2018), A novel unsupervised deep learning model for global and local health condition assessment of structures, *Engineering Structures*, **156**, 598-607.
- Rafiei, M. H., Khushefati, W. H., Demirboga, R., & Adeli, H. (2017). Supervised Deep Restricted Boltzmann Machine for Estimation of Concrete. *ACI Materials Journal*, **114**(2), 237-244.
- Ratcliffe, C. P. (1997), Damage detection using a modified laplacian operator on mode shape data, *Journal of Sound & Vibration*, **204**(3), 505-517.
- Reynders, E. (2012), System identification methods for (operational) modal analysis: review and comparison, *Archives of Computational Methods in Engineering*, **19**(1), 51-124.
- Reynders E., Wursten G., De Roeck G. (2014), Output-only fault detection in structural engineering based on kernel PCA. *BIL2014 workshop on data-driven modeling methods and applications*, Leuven, Belgium, 14-15 July.
- Rice, J. A., Mechitov, K., Sim, S. H., Nagayama, T., Jang, S., Kim, R., ... & Fujino, Y. (2010). Flexible smart sensor framework for autonomous structural health monitoring.
- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, **323**, 533-536.
- Salawu, O. S. (1997), Detection of structural damage through changes in frequency: a review, *Engineering Structures*, **19**(9), 718-723.
- Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- Sirca Jr, G. F., & Adeli, H. (2012). System identification in structural engineering. *Scientia Iranica*, **19**(6), 1355-1364.

- Siringoringo, D. M., & Fujino, Y. (2018). Seismic response of a suspension bridge: Insights from long-term full-scale seismic monitoring system. *Structural Control and Health Monitoring*, 25(11), e2252.
- Spiridonakos, M. D., & Chatzi, E. N. (2014). Polynomial chaos expansion models for SHM under environmental variability. In *Proceedings of the 9th European Conference on Structural Dynamics, EUROODYN 2014, Porto, Portugal, 30 June-2 July 2014* (pp. 2393-2398). Faculty of Engineering of University of Porto.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014), Dropout: a simple way to prevent neural networks from overfitting, *Journal of Machine Learning Research*, **15**(1), 1929-1958.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., & Anguelov, D., et al. (2015), Going deeper with convolutions. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, 1-9.
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2818-2826).
- Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-First AAAI Conference on Artificial Intelligence*.
- Tong, Z., Gao, J., & Zhang, H. (2017), Recognition, location, measurement, and 3D reconstruction of concealed cracks using convolutional neural networks, *Construction and Building Materials*, **146**, 775-787.
- Wattana, K., & Nishio, M. (2017). Traffic volume estimation in a cable-stayed bridge using dynamic responses acquired in the structural health monitoring. *Structural Control and Health Monitoring*, **24**(4), e1890.
- Xia, Y., Chen, B., Weng, S., Ni, Y. Q., & Xu, Y. L. (2012), Temperature effect on vibration properties of civil structures: a literature review and case studies, *Journal of Civil Structural Health Monitoring*, **2**(1), 29-46.
- Xue, Y., & Li, Y. (2018), A Fast Detection Method via Region-Based Fully Convolutional Neural Networks for Shield Tunnel Lining Defects, *Computer-aided Civil and Infrastructure Engineering*, **0**(2018), 1-17.

- Yan, A. M., Kerschen, G., De Boe, P., & Golinval, J. C. (2005). Structural damage diagnosis under varying environmental conditions—part I: a linear analysis. *Mechanical Systems and Signal Processing*, **19**(4), 847-864.
- Yao, X. J., Yi, T. H., Qu, C., & Li, H. N. (2018). Blind Modal Identification Using Limited Sensors through Modified Sparse Component Analysis by Time-Frequency Method. *Computer-Aided Civil and Infrastructure Engineering*, **33**(9), 769-782.
- Yin, T., Yuen, K. V., Lam, H. F., & Zhu, H. P. (2017). Entropy-Based Optimal Sensor Placement for Model Identification of Periodic Structures Endowed with Bolted Joints. *Computer-Aided Civil and Infrastructure Engineering*, **32**(12), 1007-1024.
- Zhang, A., Wang, K. C., Li, B., Yang, E., Dai, X., Peng, Y., ... & Chen, C. (2017, a). Automated pixel-level pavement crack detection on 3D asphalt surfaces using a deep-learning network. *Computer-Aided Civil and Infrastructure Engineering*, **32**(10), 805-819.
- Zaurin, R., & Catbas, N. (2010), Structural health monitoring using video stream, influence lines, and statistical analysis, *Structural Health Monitoring*, **9**(4), 309-332.
- Zhang, Y., Miyamori, Y., Kadota, T., & Saito, T. (2017, b), Investigation of seasonal variations of dynamic characteristics of a concrete bridge by employing a wireless acceleration sensor network system, *Sensors & Materials*, **29**(2), 165-178.
- Zhou, Y. T., & Chellappa, R. (1988), Computation of optical flow using a neural network, In *IEEE International Conference on Neural Networks*, **27**, 71-78.