SMAR 2017 – Fourth Conference on Smart Monitoring,
Assessment and Rehabilitation of Civil Structures

SMAR 2017

Structural damage diagnosis with time-varying loads using convolutional neural networks

Nur Sila Gulgec[1], Martin Takáč[2], and Shamim N. Pakzad[3]

[1, 3] Department of Civil and Environmental Engineering, Lehigh University, USA
[2] Department of Industrial and Systems Engineering, Lehigh University, USA

ABSTRACT: Damage diagnosis of structures subjected to time-varying environmental and operational conditions has been a challenging task. This task involves a damage indicator selection to characterize the unknown relation between the measurements and damage patterns. The majority of the conventional methods adopt hand designed damage indicators that can be inefficient for some damage patterns and require manual effort to design. To address these challenges, this work uses a special kind of deep learning method called convolutional neural network (CNN) to learn complex damage features and create complex classifier boundaries. In the evaluation of the proposed methodology, multi-dimensional input samples are used — each dimension has an individual strain field resulting from a different force applied to the structure. The network is trained with several crack scenarios and the learned architecture is tested on completely unseen damage setups. Based on the findings of the paper, CNNs fed through multi-dimensional inputs improve the accuracy of the damage diagnosis. Furthermore, they give the opportunity to capture the behavior of the structures under variations in the loading conditions.

1 INTRODUCTION

Providing timely damage evaluation is important to establish lifetime safety of the infrastructures subjected to damage or deterioration (Fang et al., 2005). This process requires long term monitoring of the structures over a wide range of environmental and operational conditions (Sohn, 2007). Interpreting the health condition of structures from the collected sensor measurements often require selecting damage features. Traditional approaches primarily focus on hand-crafting damage features and classifiers which may not be efficient for diagnosing some damage patterns (Shi and Yu, 2012). Especially, when the data collected from large sensor networks are considered, damage feature extraction procedure might become computationally expensive (Yao et al., 2016; Gulgec et al., 2016). In order to overcome these problems, widely used biologically inspired soft-computing technique called neural networks were proposed in the 1990s (Flood and Kartam, 1994). Since then, they have been practiced to diagnose damages from static displacements (Szewczyk and Hajela, 1994; Zhao et al., 1998), the modal analysis of vibration response (Hearn and Testa, 1991; Hadzima-Nyarko et al., 2011) and statistical properties of vibration measurements (Shu et al., 2013). Other works demonstrated that more complex models are essential to use the full potential of the neural networks (Flood, 2008).
A special type of neural network called convolutional neural network (CNN) was first proposed by LeCun et al. (1998) to capture the visual patterns from handwritten digits. This network

architecture kept temporal features of the input and used fewer parameters to reduce memory requirements (LeCun, 1995). Then, it became more pronounced with the improvements in computing power and large-scale hierarchical image databases (Deng et al., 2009). CNNs kept evolving and got deeper and more complex which became the state-of-the-art technique in the image recognition (Szegedy et al., 2015; He et al., 2015). This breakthrough was also started to be practiced recently in structural health monitoring community (Abdeljaber et al., 2017; Gulgec et al., 2017; Cha et al., 2017).

In Gulgec et al. (2017), authors addressed the damage detection challenge by using strain fields as inputs to CNN. However, the study did not consider the effect of time-varying environmental and operational conditions which may significantly affect the characteristics of the structure. Instead of using only strain fields as inputs, this study proposes multi-dimensional inputs to feed the architecture — each dimension has an individual strain field resulting from a different force applied to the structure at a different time. The multi-dimensionality improves the damage detection accuracy and captures the behavior of the network under the time-varying loads. Besides, this study reduces the computational demand by exploiting the shared parametrization of CNN and highly parallel architecture of graphical processing units (GPU) which may enable real-time damage diagnosis challenge.

The rest of the paper is organized as follows. First, an overview of CNNs is provided in Section 2; then, the description of proposed CNN architecture is described in Section 3. In Section 4, the main findings are presented. Conclusions are given in Section 5.

2 OVERVIEW OF CONVOLUTIONAL NEURAL NETWORKS

Deep learning (deep neural network or multi-layer neural network) is a branch of machine learning which is gradually evolved from pattern recognition (Bishop, 2006). Multi-layer neural networks form a model by using deep graph organized in multiple linear layers followed with some nonlinear transformations (LeCun et al., 2015). Figure 1 shows a small DNN consisting of an input layer, two hidden layers, and an output layer. The network operates on the input instance $x = (x_1, x_2, ..., x_p)^T$ to obtain the output $y = (y_1, y_2, ..., y_p)^T$. Each layer is composed of neurons that are represented with circles. A connection from the output of one neuron to the input of another (i.e. weights) is shown with an arrow. Each neuron performs a weighted sum operation of the inputs followed by nonlinear mapping (e.g. sigmoid, tanh, etc.).

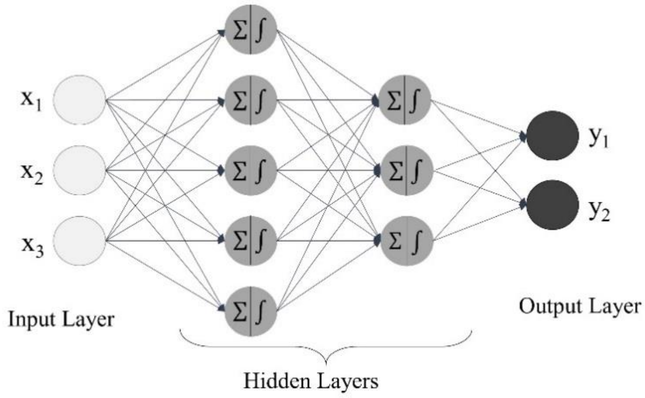

Figure 1. An example of deep neural network with four layers.

Convolutional neural network (CNN) is one of the most widely used multi-layer neural networks. This special type of architecture takes the advantage of the temporal correlation of the input (LeCun et al., 1998). CNN architectures can be built by using three types of layers such as convolutional layers (CONV), sub-sampling layers (POOL) and fully connected layers (FC).

(i) *Convolutional layer* parameters share the learnable weights and biases along the depth of the input. Feature maps are formed as these weights (filters or kernels) are passed through the entire input, and the dot product between the filter and the small region of the input in any position is computed. Then, nonlinear activation functions are used to activate the feature maps. This framework is called local receptive fields which allow extracting multiple feature maps as many as the number of filters. This property makes CNN robust to the translation and distortion of the input (i.e. the feature map will be shifted by the same amount of input shifting). Furthermore, these feature maps use shared weights and biases, which reduce the number of learned parameters and the memory needs.

(ii) *Sub-sampling (or pooling) layer* performs a downsampling operation (e.g. using maximum, average, sum operations) in the feature maps to reduce the resolution of the feature maps and avoid overfitting.

(iii) *Fully-connected layer* operates on the vectorized convolution or pooling layer outputs and computes the weighted sum of inputs composed with nonlinear mapping like multi-layer neural networks.

3 PROPOSED METHODOLOGY

3.1 Data Preparation

The neural network architecture should be trained with well-known damage states in order to have accurate damage detection (Elkordy et al., 1993). Thus, finite element model of steel structural connection is simulated in ABAQUS. The connection consists of 508 mm long C8x11.5 channels welded to a steel plate (711.2x355.6x6.4 mm) with 203.2 mm overlaps (Figure 2). Elastic-perfectly plastic steel which has a yield strength of 250 MPa is chosen as material and the model mesh size is adopted as 12.7 mm.

Training, validation and test data sets are created by simulating different loading and damage scenarios (i.e. "healthy" and "damaged"). Damages are modeled as 25.4 mm long cracks with a specified load in the range of ~U[-444.8 kN (compression), 533.8 kN (tension)]. In order to assess the effect of unseen damages, the coordinates of training samples are not used in the testing samples. Figure 2 shows the crack location coordinates used for training and test data sets.

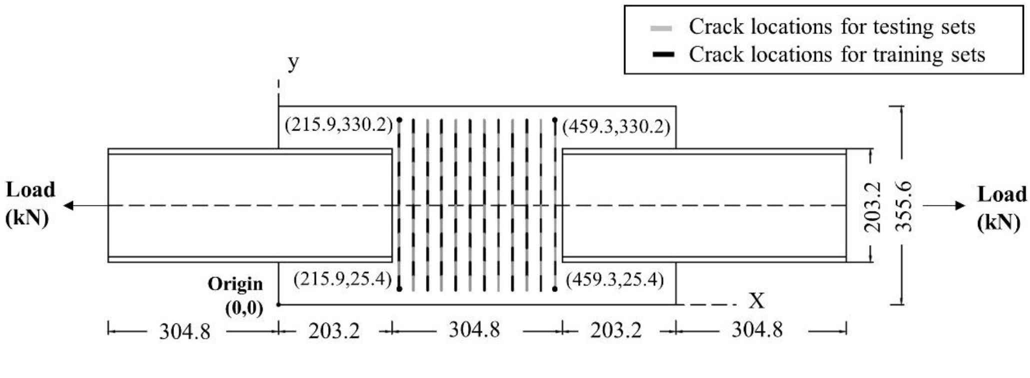

Figure 2. The damage locations for testing and training data sets.

The CNN receives the input instances as 3D volumes (i.e. width, height, depth). For instance, if the input was RGB (red-green-blue) image, it would have a depth of three (the number of the color channels). In this study, strain fields of the structural connection resulting from different loading conditions are stacked layer by layer to form a 3D volume. The first two layers (baseline) are constructed with the strain field from healthy cases, whereas last two layers (query) are concatenated from two unknown states. Figure 3 shows an example multi-dimensional input sample with two healthy strain distribution followed by two damaged state of the structure. A total 7,200 “healthy” and 7,200 “damaged” samples with the size of 28x56x4 are created and distributed to the training, validation, and test data sets. The multi-dimensional input is normalized for each strain field by dividing its maximum and then used to feed the network.

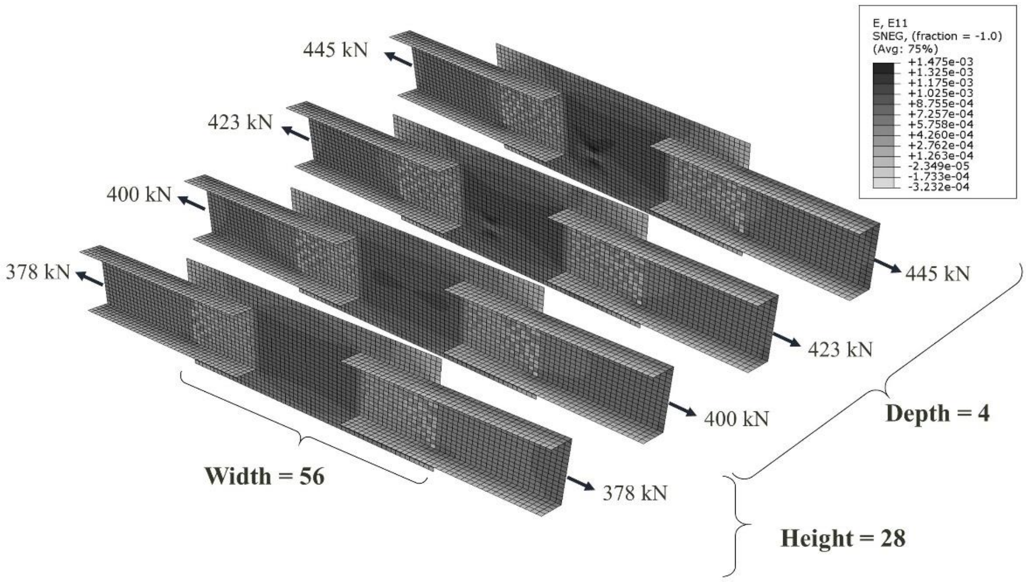

Figure 3. An example of multi-dimensional input.

3.2 Architecture Overview

3.2.1 Training

For a given input $x \in \Re^p$, prediction function $\theta = (x; w)$ can be parameterized by weights (w). After building the prediction function, a loss function is selected to measure the error between a prediction and the true value. The loss function can be denoted as $\ell(\theta = (x; w), y)$ where $y \in \Re^c$ is the true value (also called the label) of the input query x. The exact distribution of these inputs and labels are almost impossible to know in practice. Therefore, n sample data points, which is called training dataset, are used to minimize the empirical loss.

$$\min_x \frac{1}{n} \sum_{i=1}^{n} \ell(\theta(x_i; w), y_i) \tag{1}$$

In this study, the label of the final output is found by softmax classifier (Bridle, 1990). The class i of the input x is predicted by choosing the maximum probability of softmax function:

$$[\text{soft}\max(\theta(x;w))]_i = \frac{e^{[\theta(x;w)]_i}}{\sum_j e^{[\theta(x;w)]_j}} \quad ; \quad y_{prediction} = \arg\max_i([\text{soft}\max(\theta(x;w))]_i) \qquad (2)$$

Training process adopts back-propagation algorithm which is comprised of two phases: feed-forward computation and back-propagation (Rojas, 2013). The feed-forward process gets the input and evaluates the prediction function. Then, the back-propagation process updates the weights based on their contribution to the total error (Rumelhart et al., 1998). In this study, the stochastic gradient descent (SGD) methodology (Robbins and Monro, 1951) is implemented to calculate the gradients of the error with respect to the parameters (i.e. weights and biases) in the network. The negative log-likelihood is employed as the loss function of back-propagation process, where optimal model parameters are learned by minimizing the negative likelihood of the data. After the gradients are calculated with SGD, the model parameters are updated with the learning rate. No further optimization is implemented if the model performance is improved enough on the validation set (i.e. early stopping criteria). Proposed study is trained by using a Python library called Theano to optimize the mathematical expressions (Team T.T.D., 2016). The higher performance of these data-intensive calculations is achieved by NVIDIA Tesla K80 GPUs which enable parallelism for data processing.

3.2.2 Hyperparameter Optimization

The performance of the network highly depends on identifying a good set of hyperparameters (Li et al., 2016). These hyperparameters include the number of CONV and FC layers, the number of convolution filters and their sizes, pooling filter sizes, learning rate and the number of hidden layers. In this study, total 50 different networks are generated with different hyperparameters. The hyperparameters are randomly chosen from the list such as: the number of CONV Layers: [1,2 or 3]; the number of FC layers: [1,2 or 3]; convolutional filter size: [(3x3) or (5x5)], the number of filter (kernel) sizes [2 to 2^7]; max-pooling size: [(2x2) or (1x1)]; learning rate: [2 to 2^{-8}] and random hidden layer sizes. All networks are performed for one epoch, then, 10% of the networks which have the worst validation score are removed. The remaining networks are run for another epoch until the best 10 networks remain.

3.2.3 Putting It All Together - Proposed Network Architecture

Figure 4 shows the proposed network obtained by hyperparameter search mechanism with learning rate $\eta = 0.015$. The network architecture has one convolution layer. Here, the input is convolved with 8 filters (kernels) size of 5x5x4 to produce 8 feature maps, then these feature maps are activated by a nonlinear function *tanh()*. The CONV layer follows pooling layer where downsampling operation is performed with a maximum operator. Two fully connected layers with hidden layer size of 6207 and 1737 operate on the vectorized output of the pooling layer to classify the input as healthy or damaged.

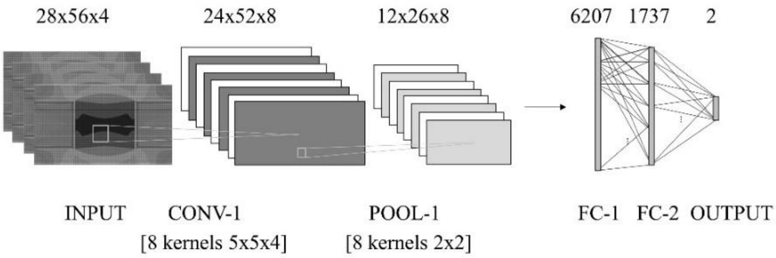

Figure 4. The proposed architecture for multi-dimensional input.

4 RESULTS AND DISCUSSION

The proposed architecture is tested on the previously unseen damage states and the learning curve is presented in Figure 5. As shown in the figure, the validation and testing error of the network is 0.36% and 0.62%, respectively. Learned filters can be investigated to better understand the behavior of the stacked layers (2 baseline + 2 query). This requires a simpler network architecture since it becomes difficult to visualize the filters when the network becomes deeper. In this case, all the nonlinearities and early stopping criteria are removed from the training procedure. One convolutional layer (with the filter size of 5x5), one pooling layer (filter size of 2x2) and one fully connected layer size of 618 are built and run for 1,000 epochs. The learning rate is adopted as $\eta = 0.011$.

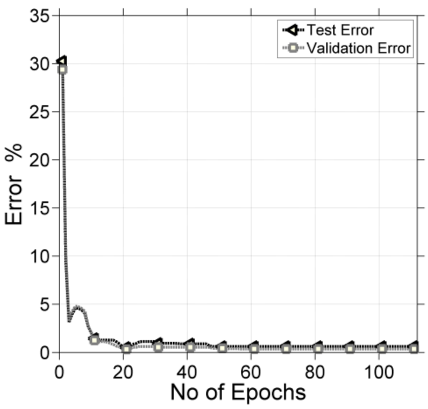

Figure 5. The learning curve of the proposed network.

It is observed that the first two learned filters show high similarities since the baseline layers belong to the layers solely created by healthy cases. On the other hand, the learned filters from two query inputs (either healthy or damaged) are not highly correlated like first two filters. This finding shows that even a small network can find some correlations of similar damage scenarios and it gives the opportunity to capture the behavior of the structures under the time-varying forces.

Lastly, the experiments are repeated with only one baseline (healthy) and two query (unknown) layers since first two filters are almost similar. The network is run through hypermeter search as described in Section 3 and the best network is presented in Figure 7. The best network has two convolutional layers with 3x3 filters followed by two fully-connected layers size of 3909 and 3420. The feature maps are transformed with *tanh()* function and learning rate is adopted $\eta = 0.022$. The validation and testing errors are found as 0.36% and 0.5%, respectively.

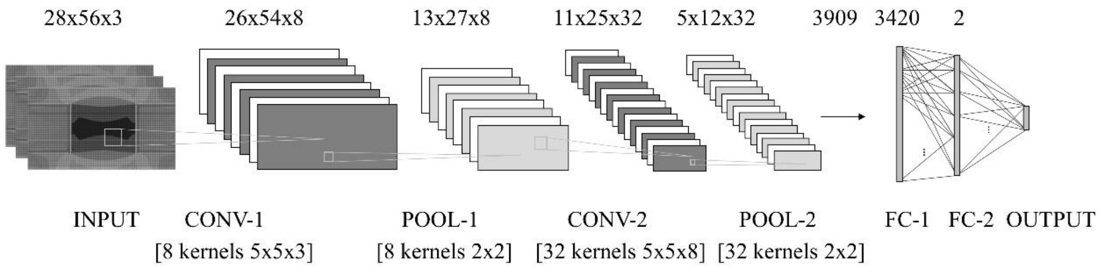

Figure 7. The network architecture for one healthy and two query layered inputs.

The findings are compared with the study (Gulgec et al. 2017) in Table 1. The table shows that the test error decreases considerably when the multi-dimensional input (normalized within each dimension) is adopted. Therefore, motivating the problem with multi-dimensional input improves the accuracy of the damage diagnosis. Furthermore, the obtained error rates show that a number of baseline layers do not affect the damage diagnosis accuracy significantly.

Table 1. The comparison of the test errors.

Model	Test Error (%)	Reference
1 query	2.06	Gulgec et al., 2017
2 baseline + 2 query	0.62	This study (Figure 4)
1 baseline + 2 query	0.50	This study (Figure 7)

5 CONCLUSIONS

This paper presents the damage evaluation of a structural connection which is subjected to time-varying loads by performing convolutional neural networks. The proposed methodology feeds the network architecture with multi-dimensional inputs where each dimension has an individual strain distribution resulting from a different force applied to the structure at a different time. The findings show that it is worthwhile to include samples from a wide range of operational and environmental conditions in the training dataset. This approach improves the damage detection accuracy and helps to capture the behavior of the network under the time-varying conditions.

Acknowledgment. Research funding is partially provided by the National Science Foundation through Grant No. CMMI-1351537 by Hazard Mitigation and Structural Engineering program, and by a grant from the Commonwealth of Pennsylvania, Department of Community and Economic Development, through the Pennsylvania Infrastructure Technology Alliance (PITA). Martin Takáč was supported by National Science Foundation grant CCF-1618717.

6 REFERENCES

Abdeljaber, O., Avci, O., Kiranyaz, S., Gabbouj, M. and Inman, D.J., 2017. Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks. *Journal of Sound and Vibration*, *388*, pp.154-170.

Bishop, C.M., 2006. Pattern recognition. *Machine Learning*, *128*, pp.1-58.

Bridle, J.S., 1990. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In *Neurocomputing* (pp. 227-236). Springer Berlin Heidelberg.

Cha, Y.J., Choi, W. and Buyukozturk, O., 2017. Deep learning-based crack damage detection using convolutional neural network". *Computer-Aided Civil and Infrastructure Engineering*, *32*(3), pp.2013-2014.

Deng, J., Dong, W., Socher, R., Li, L.J., Li, K. and Fei-Fei, L., 2009, June. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on* (pp. 248-255). IEEE.

Elkordy, M.F., Chang, K.C. and Lee, G.C., 1993. Neural networks trained by analytically simulated damage states. *Journal of Computing in Civil Engineering*, *7*(2), pp.130-145.

Fang, X., Luo, H. and Tang, J., 2005. Structural damage detection using neural network with learning rate improvement. *Computers & structures*, *83*(25), pp.2150-2161.

Flood, I. and Kartam, N., 1994. Neural networks in civil engineering. II: Systems and application. *Journal of Computing in Civil Engineering*, *8*(2), pp.149-162.

Flood, I., 2008. Towards the next generation of artificial neural networks for civil engineering. *Advanced Engineering Informatics*, *22*(1), pp.4-14.

Gulgec, N.S., Shahidi, S.G. and Pakzad, S.N., 2016. A Comparative Study of Compressive Sensing Approaches for a Structural Damage Diagnosis. In *Geotechnical and Structural Engineering Congress 2016* (pp. 1910-1919).

Gulgec, N. S., Takac M. and Pakzad, S. N., 2017. Structural Damage Detection Using Convolutional Neural Networks. In *Dynamics of Civil Structures, Volume 3*

Hadzima-Nyarko, M., Nyarko, E.K. and Morić, D., 2011. A neural network based modelling and sensitivity analysis of damage ratio coefficient. *Expert systems with applications*, *38*(10), pp.13405-13413.

He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770-778).

Hearn, G. and Testa, R.B., 1991. Modal analysis for damage detection in structures. *Journal of Structural Engineering*, *117*(10), pp.3042-3063.

LeCun, Y. and Bengio, Y., 1995. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, *3361*(10), p.1995.

LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P., 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), pp.2278-2324.

LeCun, Y., Bengio, Y. and Hinton, G., 2015. Deep learning. *Nature*, *521*(7553), pp.436-444.

Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A. and Talwalkar, A., 2016. Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization. *arXiv preprint arXiv:1603.06560*.

Robbins, H. and Monro, S., 1951. A stochastic approximation method. *The annals of mathematical statistics*, pp.400-407.

Rojas, R., 2013. *Neural networks: a systematic introduction*. Springer Science & Business Media.

Rumelhart, D.E., Hinton, G.E. and Williams, R.J., 1988. Learning representations by back-propagating errors. *Cognitive modeling*, *5*(3), p.1.

Shi, A. and Yu, X.H., 2012, July. Structural damage detection using artificial neural networks and wavelet transform. In *Computational Intelligence for Measurement Systems and Applications (CIMSA), 2012 IEEE International Conference on* (pp. 7-11). IEEE.

Sohn, H., 2007. Effects of environmental and operational variability on structural health monitoring. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, *365*(1851), pp.539-560.

Shu, J., Zhang, Z., Gonzalez, I. and Karoumi, R., 2013. The application of a damage detection method using Artificial Neural Network and train-induced vibrations on a simplified railway bridge model. *Engineering structures*, *52*, pp.408-421.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A., 2015. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1-9).

Szewczyk, Z.P. and Hajela, P., 1994. Damage detection in structures based on feature-sensitive neural networks. *Journal of computing in civil engineering*, *8*(2), pp.163-178.

Team, T.T.D., Al-Rfou, R., Alain, G., Almahairi, A., Angermueller, C., Bahdanau, D., Ballas, N., Bastien, F., Bayer, J., Belikov, A. and Belopolsky, A., 2016. Theano: A Python framework for fast computation of mathematical expressions. *arXiv preprint arXiv:1605.02688*.

Yao, R., Pakzad, S.N. and Venkitasubramaniam, P., 2017. Compressive sensing based structural damage detection and localization using theoretical and metaheuristic statistics. *Structural Control and Health Monitoring*, *24*(4).

Zhao, J., Ivan, J.N. and DeWolf, J.T., 1998. Structural damage detection using artificial neural networks. *Journal of Infrastructure Systems*, *4*(3), pp.93-101.