

Research Article

Deep Convolutional Neural Network-Based Structural Damage Localization and Quantification Using Transmissibility Data

Sergio Cofre-Martel ^{1,2} **Philip Kobrich** ¹ **Enrique Lopez Droguett** ^{1,2}
and Viviana Meruane ¹

¹Mechanical Engineering Department, University of Chile, Santiago, Chile

²Center for Risk and Reliability, University of Maryland, College Park, USA

Correspondence should be addressed to Enrique Lopez Droguett; elopezdroguett@ing.uchile.cl

Received 9 May 2019; Accepted 8 August 2019; Published 8 September 2019

Academic Editor: Luca Pugi

Copyright © 2019 Sergio Cofre-Martel et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Damage diagnosis has become a valuable tool for asset management, enhanced by advances in sensor technologies that allows for system monitoring and providing massive amount of data for use in health state diagnosis. However, when dealing with massive data, manual feature extraction is not always a suitable approach as it is labor intensive requiring the intervention of domain experts with knowledge about the relevant variables that govern the system and their impact on its degradation process. To address these challenges, convolutional neural networks (CNNs) have been recently proposed to automatically extract features that best represent a system's degradation behavior and are a promising and powerful technique for supervised learning with recent studies having shown their advantages for feature identification, extraction, and damage quantification in machine health assessment. Here, we propose a novel deep CNN-based approach for structural damage location and quantification, which operates on images generated from the structure's transmissibility functions to exploit the CNNs' image processing capabilities and to automatically extract and select relevant features to the structure's degradation process. These feature maps are fed into a multilayer perceptron to achieve damage localization and quantification. The approach is validated and exemplified by means of two case studies involving a mass-spring system and a structural beam where training data are generated from finite element models that have been calibrated on experimental data. For each case study, the models are also validated using experimental data, where results indicate that the proposed approach delivers satisfactory performance and thus being an appropriate tool for damage diagnosis.

1. Introduction

Recent advances in sensors' technology and costs reduction have made them a valuable asset for engineers to monitor structures and equipment. Sensors can acquire relevant parameters of a system, such as velocity, temperature, pressure, or vibrations. The gathered data can be used for monitoring purposes, as well as to determine the health state of a system and thus support the implementation of preventive actions before catastrophic failures. To obtain an accurate damage diagnosis, it is important to detect, locate, and quantify the damage level that a system presents. However, managing large amount of data usually encompasses careful feature engineering to input into a damage

quantification model [1]. Furthermore, feature extraction and selection demand prior expert knowledge of the data for choosing which features to include or exclude within the model.

In this context, structural vibration-based damage assessment focuses in detecting and characterizing structural damage at the earliest possible stage to estimate the remaining time before failure in a structure is presented or when is no longer usable. Damage assessment has tremendous potential in providing life safety and economic benefits by reducing maintenance costs and enhancing safety and reliability. One of the main challenges in vibration-based damage assessment is the selection of an appropriate metric of the system response that is sufficiently sensitive to

small damage. This metric can be constructed in the time, frequency, or modal domains, the latter two being the most broadly used. The idea of directly using the transmissibility functions (TFs) has attracted many researchers [2–20]. TFs relate the responses between two points of the structure. Among all dynamic responses, TFs are the easiest to obtain in real-time because the in situ measurement is straightforward. As an advantage, no modal extraction is necessary; thus, contamination of the data with modal extraction errors is avoided, and they are identified from response-only data. Therefore, it does not involve the measurement of excitation forces.

Worden [2] presented the first investigation of TFs as indicators of structural damage. Here, for a simple lumped-parameter system, transmissibilities were able to detect small stiffness changes. Since then, the research group headed by Worden and Manson has done extensive research in this topic. In [3], Worden et al. used a representative aircraft skin panel to investigate the sensitivity of transmissibilities to damage. Damage detection was carried out via a statistical outlier analysis. Manson et al. [4, 5] verified the performance of the outlier analysis technique to detect damage in a Gnat aircraft inspection panel. Damage was simulated by holes and saw-cuts across the panel. Zhang et al. [6] proposed a procedure to detect structural damage using changes in the TF, which were derived from structural translations and curvatures, the latter being the most sensitive to damage. Johnson and Adams [7] also studied the use of TFs for detecting, locating, and quantifying damage. They demonstrated that since transmissibility functions are determined solely by the system's zeroes (antiresonant frequencies), they are potentially better indicators of localized damage. These results were employed to develop a framework for transmissibility-based damage identification using smart sensor arrays [8]. Maia et al. [9] presented a methodology for computing the transmissibility matrix from responses only. They showed that TFs are sensitive to damage, making them a possible approach for damage assessment. Sampaio et al. [10] implemented a similar approach to explore the ability of transmissibilities in detecting and localizing damage, concluding that it is possible to detect sensitive changes to damage, but further research is needed.

The most successful applications of vibration-based damage assessment are model updating methods based on global optimization algorithms [21–24]. The basic assumption is that damage can be directly related to a decrease in stiffness in the structure. Nevertheless, these algorithms are exceedingly slow making them impractical for real-time applications. As an alternative to these methods, neural networks (NNs) have been proposed as a tool for damage identification [25–27]. In recent years, the interest in applying machine learning algorithms for transmissibility-based damage assessment has increased [12], NN being the most frequently used. However, the number of spatial response locations and spectral lines in transmissibility measurements is overly large for traditional NN applications. The direct use of transmissibilities leads to NN with many input variables and connections, thus rendering them until now impractical. Hence, it has been necessary to extract

features from the transmissibilities and then use these features as inputs to the NN. Indeed, Worden et al. [13] trained an autoassociative NN for damage detection. The feature vector was constructed from transmissibility data, selecting spectral lines centered at a particular peak and then using PCA to reduce the dimension of the dataset. Chen et al. [14] used a comb data sampling technique to acquire amplitude and phase data of transmissibility functions randomly. These data were the input to a damage classification NN, which was validated using simulated data of a sandwich beam and a frame structure. Pierce et al. [15] and Worden et al. [16] identified spectral line windows with the largest variations due to damage as features to train a NN-based damage classifier. In both cases, the classifier was evaluated with experimental data of the aircraft (Gnat) wing. Lai and Perera [17] trained a damage classification NN using damage indicators extracted from the power spectrum density transmissibility. This methodology was evaluated using simulated data of a beam. Meruane [18] trained an online sequential extreme learning machine (OS-ELM) algorithm to detect, locate, and quantify structural damage using antiresonant frequencies extracted from transmissibility measurements. The approach is illustrated with two experimental cases: an eight-degree-of-freedom (DOF) mass-spring system and a beam under multiple damage scenarios. Meruane and Ortiz-Bernardin [19] presented another algorithm for real-time damage assessment that uses a linear approximation method in conjunction with antiresonant frequencies that are identified from transmissibility functions. The performance is validated by considering three experimental structures: an eight DOF mass-spring system, a beam, and an exhaust system of a car.

All the aforementioned works rely on identifying and extracting proper features. Nonetheless, the feature extraction process requires specialized knowledge of the problem under investigation and the best selection is case sensitive. Up to date, there is no consensus on what the best features for vibration-based damage assessment are. It is here where deep learning techniques have been proposed for automatic feature extraction and fault diagnostic analysis [28–39]. Jia et al. [40] applied autoencoders to pretraining and pretuning layers of neural networks for fault diagnosis of rolling element bearings, whereas a model with two layers of restricted Boltzmann machines is proposed in [41] where vibration signals are analyzed to obtain an automatic diagnosis system for rolling element. Of particular interest for reliability problems are convolutional neural networks (CNNs) as they are capable of obtaining better nonlinear representations of raw signals without the need of human intervention by providing a higher level of abstraction and avoiding biases in the feature extraction process. For example, Wang [29] applied a CNN architecture to detect faults, using scalograms as input parameter, obtained from vibration signals. In the context of structural damage, Abdeljaber et al. [31] implemented a one-dimensional CNN to detect and locate structural damage. Damage is simulated by loosening bolt connections in a framed structure; the CNN algorithm is trained to detect and to locate the damaged joint. The input data is the acceleration measured

at different locations when the structure is excited by random noise. Furthermore, Yu et al. [42] presented a new structural damage identification method that uses a DCNN to detect and localize damage in an n -level smart building. The input is a matrix containing by columns the frequency response at each building level, and the output is a vector consisting on the health condition of each floor. The results obtained with numerically generated data demonstrate the DCNN method outperforms traditional neural network approaches. Khodabandehlou et al. [43] used a CNN for vibration-based structural condition assessment. The input data is a matrix containing the raw time response measured at different locations, and the output is a global classification of the structure in different damage levels (no damage, minor, moderate, and extensive). The classification algorithm is trained with experimental data of a scaled bridge model under seismic and random excitations.

In most of the abovementioned approaches, the researchers have used specific spectral lines from the transmissibility functions. In damage detection, it is important to find the spectral lines which are highly sensitive to damage, whereas for damage localization, an additional requirement is to find spectral lines that are also sensitive to the damage location [20]. A major problem of this approach is that these spectral lines cannot be determined a priori. Therefore, it requires a deep investigation of each application case. Another approach is to extract features such as the antiresonant frequencies and use them to detect, locate, and quantify damage. Nevertheless, the antiresonant identification process is not automatic and requires human intervention. In this paper, we propose a novel deep CNN-based approach for the detection, localization, and quantification of structural damage that operates on raw transmissibility functions. The main advantage over previous investigations is that this approach makes automatic feature extraction. Therefore, the input to the proposed algorithm are the full transmissibility functions, and it is not necessary to select spectral lines or to extract antiresonant frequencies. The proposed CNN-based approach is validated and exemplified with two case studies: an eight-degree-of-freedom (DOF) mass-spring system and a beam under multiple damage scenarios. To demonstrate the potential of the proposed algorithm over existing ones, the obtained results are compared against conventional approaches using neural networks.

The remaining of this paper is structured as follows. Section 2 introduces deep learning and convolutional neural networks. Section 3 reviews the definition and characterization of transmissibility functions. Then, the proposed approach is presented in Section 4, followed by a description of the datasets used for the CNN-based approach training, validation, and testing in Section 5, as well as the metrics to measure its performance in Section 6. The proposed approach is then exemplified by two case studies and the corresponding discussions on the approach's performance are presented in Sections 7 and 8 for the mass-spring system and the structural beam, respectively. A comparison of the CNN-based approach performance with a shallow multi-layer perceptron model is discussed in Section 9, and Section 10 presents some concluding remarks.

2. Deep Learning and Convolutional Neural Networks' Background

Deep learning (DL) techniques have become a popular approach for numerous tasks involving image recognition and computer vision, due to its high performance. These techniques have shown superior performance in image classification [44], natural sentence classification [45], and image segmentation [46] than previous methods based on shallow architectures. Even though most DL techniques are capable of automatic feature extraction, great care must be taken when choosing which technique to use when dealing with a specific task. Within these techniques, CNNs have proven to be superior to deep neural networks at obtaining a representation of the input data involving grid type data such as images or matrixes.

To understand how CNNs work, first it is important to know how a one-layer feedforward network (FFN) works. Consider, for example, the neural network shown in Figure 1. A FFN takes the input data vector \mathbf{x} , a weight matrix \mathbf{W}^1 (for the input layer), and a bias vector \mathbf{b}^1 to obtain a vector of values for the hidden layer. This is represented in equation (1), where f is an activation function such as the sigmoid function or a rectifier linear unit (ReLU) function. The output vector \mathbf{y} is computed from the hidden unit vector \mathbf{h} and an additional weight matrix \mathbf{W}^2 and bias vector \mathbf{b}^2 (characterizing the connections between the hidden and the output layers) using equation (2):

$$\mathbf{h} = f(\mathbf{W}^1 \mathbf{x} + \mathbf{b}^1), \quad (1)$$

$$\mathbf{y} = \mathbf{W}^2 \mathbf{h} + \mathbf{b}^2. \quad (2)$$

The weights and biases are adjusted (i.e., optimized) by minimizing the error between the predicted value and the real value based on a training dataset, usually known as a cost function. The error is represented by a cost function, where regression models usually use the mean squared error. The minimization is usually done with the gradient descent method, and the gradients are calculated with the back-propagation algorithm [47]. The FFN architecture can be expanded for use in deep learning problems by adding additional hidden layers or increasing the number of hidden units. Deeper FFNs allow for a higher level of abstraction but require more computational resources.

A CNN is a deep learning neural network that uses convolution operations instead of matrix multiplication in its layers. The convolution is performed using a weights matrix \mathbf{K} , also known as filter or kernel. The kernel is used to obtain a feature map \mathbf{S} from the input vector \mathbf{A} using the convolution operation as shown in the following equation:

$$\mathbf{S} = \mathbf{A} * \mathbf{K} \text{ where } \mathbf{S}(i, j) = \sum_n \sum_m \mathbf{A}(i - m, i - n) \cdot \mathbf{K}(m, n). \quad (3)$$

Figure 2 shows a representation of the convolution operation using a 2×2 kernel and a 3×3 input data matrix to obtain a 2×2 output matrix. A bias matrix \mathbf{B} is added to the convolution, and an activation function is applied to the

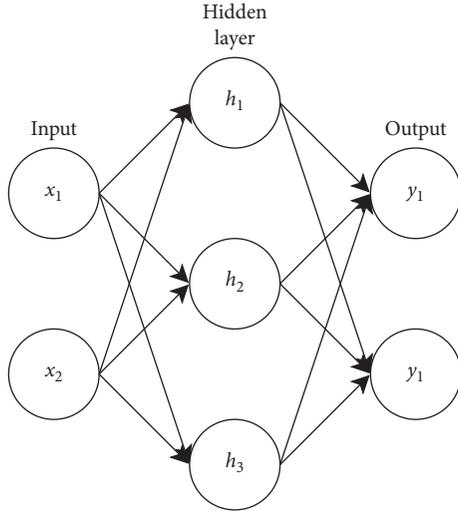


FIGURE 1: Feedforward neural network with 3 hidden units and 2 output values.

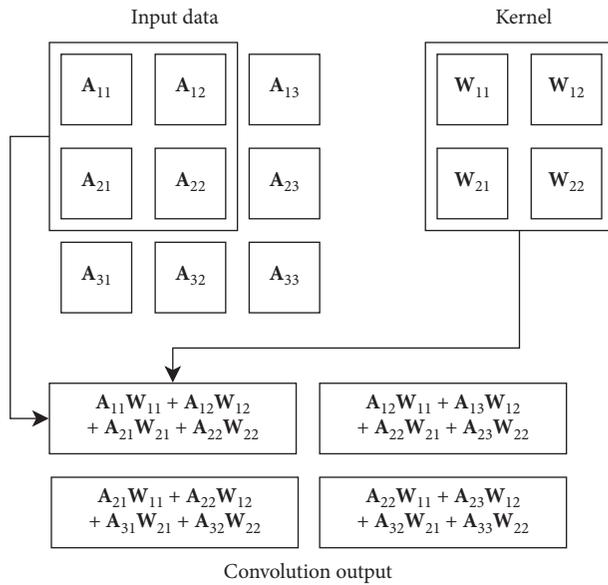


FIGURE 2: Convolution operation of a 3×3 input matrix and a 2×2 kernel.

result to form the feature map \mathbf{H} as shown in equation (4). The training of the biases and weights is the feature extraction process from the original input data. If a value in the feature map gets activated, it indicates that an important learned feature is in that position. In the case of image analysis, activation in the feature map can indicate the location of features like edges or specific shapes.

A convolution layer in a CNN consists of several kernels and biases applied to a single input matrix to generate a set of feature maps in a hidden layer. Every component in the feature map is computed using the same kernel, thus reducing the amount of weights that need to be calculated. Also, each component of the output feature map is calculated only from a subset of the input matrix, reducing the

amount of connections and, therefore, decreasing the required computation resources. To achieve higher levels of abstraction and more complex relations between features, feature maps can be used as input to adjacent convolution layers:

$$\mathbf{H} = f(\mathbf{A} * \mathbf{K} + \mathbf{B}). \quad (4)$$

The last section of a CNN is a feedforward neural network that is responsible for generating the predicted labels as the output vector. Figure 3 shows an architecture of a CNN with three 5×5 convolutional filters as the first layer, one 2×2 pooling layer, and a fully connected feedforward layer. The CNN is trained in the same way as a FFN defining a cost function and then performing gradient descent to minimize the cost function.

Due to the usually high degrees of freedom of CNN architectures, one should prevent overfitting, i.e., over adjustment of the weights to the training data resulting in poor generalization performance to unseen data. This can be accomplished via regularization techniques. One of commonly used such techniques to tackle overfitting when training CNN architectures is dropout. Furthermore, another regularization technique is early stopping, which stops the training cycle when training and validation errors begin to diverge. These two techniques together greatly reduce overfitting and prevent the network from identifying noise and use it as a distinguishing feature.

3. Transmissibility Functions

In vibration analysis, transmissibility functions (TFs) are the ratio in the frequency domain between two responses when an excitation force is applied. Transmissibility functions have shown a strong relation with a system's damage and have been previously used for damage assessment in different studies [2–20]. TF can be computed from experimental measurements or from a numerical model of the structure. Since it is not feasible to produce large enough datasets to train a CNN from experiments, the CNN models presented in this work are trained with data generated from numerical models of the structures and then have been validated with experimental data. The next sections describe the computation of experimental and numerical transmissibilities.

3.1. Experimental Transmissibilities. The experimental TFs are calculated using equation (5), where $T_{ir}^k(\omega)$ is the transmissibility function between measuring points i and r subject to an excitation force at point k , and $X_{ik}(\omega)$ is the response in the frequency domain of point i due to the excitation at point k :

$$T_{ir}^k(\omega) = \frac{X_{ik}(\omega)}{X_{rk}(\omega)}. \quad (5)$$

The main advantage of TF is that the magnitude of the excitation force is not required, but only its location. This makes it easier to obtain in situ measurement when compared with other methods. In practice, there are advantages

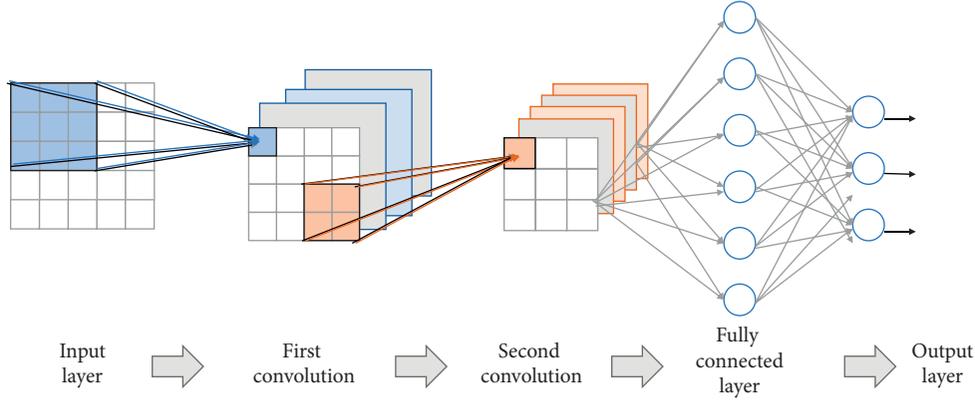


FIGURE 3: CNN architecture illustration.

in using alternative ways of calculating the TF using the auto- and cross-power spectrums:

$$T_{ir}^k(\omega) = \frac{X_{ik}(\omega)X_{rk}^*(\omega)}{X_{rk}(\omega)X_{rk}^*(\omega)} \quad (6)$$

where $X_{rk}^*(\omega)$ is the complex conjugated of $X_{rk}(\omega)$. The main reason for calculating the TF with equation (6) and not equation (5) is the reduction of uncorrelated noise. Figure 4 shows an example of the logarithm for three experimental transmissibility functions calculated through equation (6) for a given structure. It can be seen a similar behavior among the functions, where shifts in the peaks, deeps, and magnitude are related with the system's damage [24].

3.2. Numerical Transmissibilities. A numerical model of a linear structure is represented by the following $n \times n$ matrices: mass (\mathbf{M}), stiffness (\mathbf{K}), and damping (\mathbf{C}), where n is the number of degrees of freedom (DOFs). The motion of a linear system is described by

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{C}\dot{\mathbf{x}} + \mathbf{K}\mathbf{x} = \mathbf{f}(t), \quad (7)$$

where \mathbf{x} , $\dot{\mathbf{x}}$, and $\ddot{\mathbf{x}}$ are the displacement, velocity, and acceleration vectors, respectively. $\mathbf{f}(t)$ represents a vector of time-dependant external forces. We can write equation (7) in the frequency domain as

$$(-\omega^2\mathbf{M} + j\omega\mathbf{C} + \mathbf{K})\mathbf{X}(\omega) = \mathbf{F}(\omega), \quad (8)$$

where ω is the frequency in rad/s and j is the imaginary unit. From equation (8), the frequency response function matrix ($\mathbf{H}(\omega)$) is computed by

$$\begin{aligned} \mathbf{X}(j\omega) &= \mathbf{H}(j\omega)\mathbf{F}(j\omega), \\ \mathbf{H}(\omega) &= (-\omega^2\mathbf{M} + j\omega\mathbf{C} + \mathbf{K})^{-1}. \end{aligned} \quad (9)$$

The element at the i -th row and k -th column of $\mathbf{H}(\omega)$ and $\mathbf{H}_{ik}(\omega)$ corresponds to the frequency response function when the structure is excited at i and the response is measured in k , or vice versa:

$$\mathbf{H}_{ik}(\omega) = \mathbf{H}_{ki}(\omega) = \frac{X_{ik}(\omega)}{F_k(\omega)} = \frac{X_{ki}(\omega)}{F_i(\omega)}. \quad (10)$$

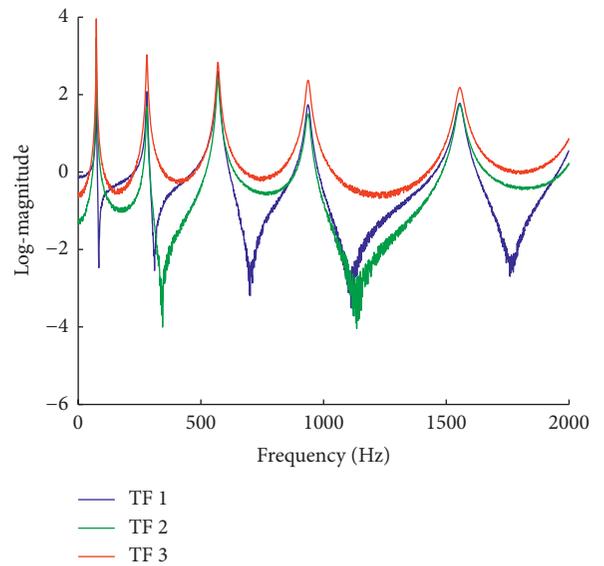


FIGURE 4: Example of three transmissibility functions (TFs) of a system.

Lastly, the transmissibility function between measuring points i and r subject to an excitation force at point k is computed by

$$T_{ir}^k(\omega) = \frac{X_{ik}(\omega)}{X_{rk}(\omega)} = \frac{\mathbf{H}_{ik}(\omega)}{\mathbf{H}_{rk}(\omega)}. \quad (11)$$

4. Proposed CNN Approach for Structural Damage Localization and Quantification

The proposed approach is intended to analyze any structure that can be divided into a discrete number of elements for identifying and quantifying the damaged elements by processing raw transmissibility functions. In the following sections, we discuss the different modeling choices made for the damage representation, input data format, and the proposed CNN architecture.

4.1. Structural Damage. Damage is represented as a reduction in the stiffness of an element of the structure. This is a simple representation of structural damage but has demonstrated good results in damage identification algorithms [48]. If the element's stiffness reaches zero, it is considered to have catastrophic failure. Defining y_i as the stiffness reduction of element i , undamaged and completely damaged states can be represented by $y_i = 0$ and $y_i = 1$, respectively. This definition can be expressed in equation (12), where \mathbf{K}_i and \mathbf{K}_i^d are the undamaged and damaged stiffness of the i -th element, respectively:

$$\mathbf{K}_i^d = (1 - y_i)\mathbf{K}_i. \quad (12)$$

4.2. Transmissibility Function Images as Input Data. To fully take advantage of the CNN's feature extraction capabilities, the TFs are represented by small-sized images which include the information of the raw TFs represented by the intensity of each pixel. The images only contain the values of the logarithmic magnitude of the TFs at a given frequency range. For this purpose, the magnitude is normalized to a value between 0 and 255 and is represented as the intensity of a pixel in a grayscale image. In the image, each row indicates the magnitude of a single TF and the columns indicates a specific frequency. Rows are arranged in numerical order. The width of the images was reduced using bicubic interpolation in order to reduce the number of input values and training parameters. Note that this is just for convenience, since reducing the reduction on the number of pixels of the images imply a smaller number of parameters to train in the CNN models. Thus, no feature extraction is done to the gathered TFs.

Figure 5(a) shows 10 different transmissibility functions measured on a structural beam on a 0–2000 Hz frequency range, while Figure 5(b) shows the corresponding grayscale pixel representation of these measurements. The generated input image has a size of 10×96 pixels, where each row is a distinct TF, and the frequency range was converted from 2000 to 96 frequencies using a bicubic interpolation. The proposed CNN model uses the TFs in this image format to localize and quantify damage.

4.3. Proposed Convolutional Neural Network Architecture. The proposed architecture encompasses a two-layer CNN for automatic feature extraction purposes. The first convolutional layer consists of 32 different filters, whereas the second layer applied 64 filters. The first convolutional layer's filter sizes are one pixel wide with a height equal to the number of transmissibility functions' pixels. That is, for each column representing a range of frequencies, the first filter processes all transmissibility functions simultaneously. Padding, where the filter is bounded within the image matrix when computing the convolution, is not applied. The aim of this architecture is to have the first layer extracting meaningful relations between different transmissibility functions at a given frequency. Note that given the shape of the first filter (10×1), the arrangement order of the TF in the input

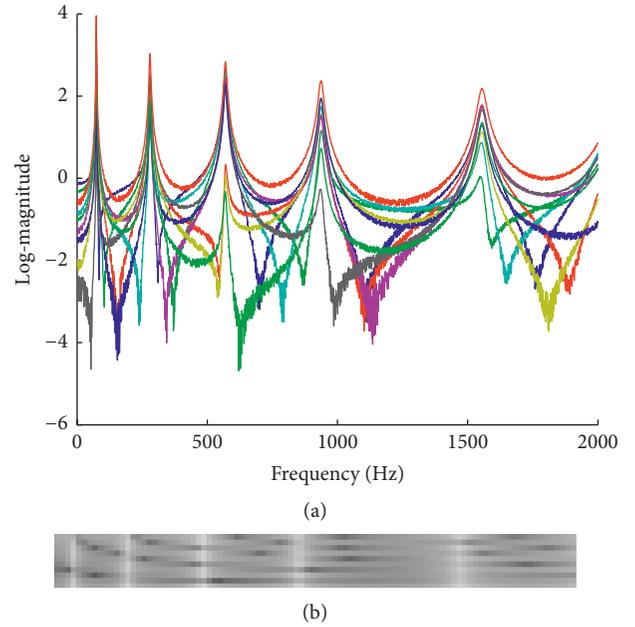


FIGURE 5: Ten TFs measured in a structural beam (a) and the image representation of the TFs used as input by the CNN (b).

image is irrelevant since during the optimization process, the kernel's weights will adjust accordingly to relevant features. Thus, the only precaution that must be taken is to feed the network the TF in the same order as it was trained. The second convolutional layer, with filter size of 1×5 pixels, is designed to detect peaks and dips in the input feature maps that are related to the antiresonant frequencies. It must be noted that the proposed CNN architecture does not receive antiresonant frequencies as input, only the TF image. After each convolution, a ReLU (rectified linear unit) function is applied as the activation function.

The assessed structure is divided into elements, and each element can have an amount of damage represented by a real number. Therefore, the proposed architecture also has a feedforward neural network that processes the feature maps provided by the last convolutional layer. Thus, the final step consists of a neural network with 1024 hidden units with ReLU as activation function, an output layer with units equal to the number of elements that the structure is divided into, and no activation function. As we are interested not only in damage localization but also damage quantification, the proposed CNN-based approach is trained for regression, i.e., each output node provides an estimate of the amount of damage in the corresponding structural element. Figure 6 shows a diagram of the proposed deep CNN architecture where 10 different TF measurements are used. Notice that the input data consist of 10×96 pixel images.

5. Datasets and Training

We present two different experimental cases: a spring-mass system and a structural beam, presented in Sections 7 and 8, respectively. The training datasets for both cases are obtained via a finite element (FE) model to generate transmissibility functions as discussed in Section 3.2. The FE

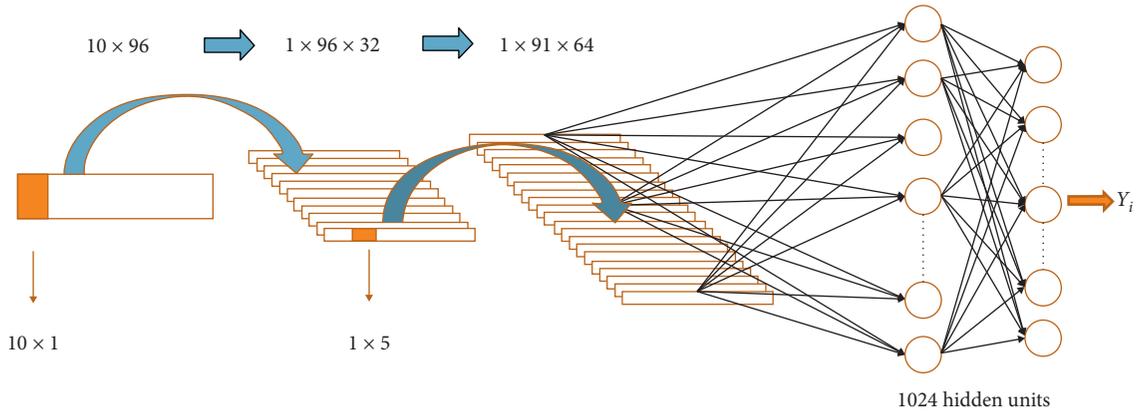


FIGURE 6: Deep CNN architecture for damage localization and quantification.

model has been calibrated with experimental data. In a FE model, the physical continuous domain of a complex structure is discretized into small components called finite elements, the physical properties of each individual element can be adjusted to simulate different damage conditions. This type of training data has been previously used in [18, 24] to assess damage using TF. Using this FE model, we can generate large amounts of training data with different damaged elements and corresponding damage magnitudes. For each of the two case studies, four different datasets were generated with zero, one, two, and three damaged elements, respectively. The stiffness reduction and damaged elements were independently and randomly selected to obtain a uniform distribution of damages in each dataset.

To deal with the problem of experimental noise in real measurements and to improve the robustness of the proposed models to randomness, random amounts of noise were added to the input signal generated by the FE models. The noise was applied as a percentage of the magnitude of the TF, and the amount of noise added was randomly selected from 0 to 6 percent uniformly distributed. This upper limit was chosen because when measuring vibrations, noise does not usually exceed the 6% threshold [49].

For both case studies, 10,000 images with zero damage and 30,000 images for each of the scenarios with one, two, and three damaged elements were generated, totaling 100,000 images for training and testing. In both case studies, the proposed architecture is trained with different datasets to detect different number of damaged elements. Table 1 summarizes the different training and test sets used for these scenarios. Note that for all the scenarios in Table 1, the trained models share the same architecture as presented in Section 4, with the only difference being in terms of the learnable parameters due to the use of different training datasets.

In the training of each of the proposed models, random truncated normal is implemented to set the initial values of the parameters. The multivariate mean squared error function shown in equation (13) is used as the cost function during the training phase. To minimize this cost function, many optimizers have been developed based on the back-propagation error to train machine learning architectures. The gradient descend optimizer is usually implemented in

TABLE 1: Trained models.

Model	Dataset	Number of images
Model 1	No damage and 1 damaged element	40,000
Model 2	No damage, 1 and 2 damaged elements	70,000
Model 3	No damage, 1, 2, and 3 damaged elements	100,000

neural networks, while RMSProp has largely been used for time-series analysis [50]. Others such as Nesterov accelerated gradient and adaptive gradient [51, 52] have also been used. However, the Adam optimizer [53] has been the most successful optimizer when dealing with CNNs, given its use of combined momentum and automatic learning rate updating. Hence, the proposed models were trained with the Adam adaptive gradient-based optimization technique, starting with a learning rate of 0.0001:

$$\text{cost} = \frac{1}{\text{NO}} \sum (y_i - o_i)^2. \quad (13)$$

Moreover, when implementing deep learning techniques, one should deal and control overfitting that occurs when the model almost perfectly fits the training data to its labels, thus leading to poor generalization to unseen data. This is usually due to the large number of learnable parameters in a deep learning model. Regularization is then used to prevent overfitting. In the context of CNNs, dropout has been reported to be efficient in controlling overfitting [20]. The training of the proposed models encompassed dropout with a 50% keep probability for regularization purpose. Additionally, an early-stopping criterion was also utilized to the training algorithm that stops training when the results do not improve in three consecutive steps, iterating up to 100 epochs. This prevents overfitting by not allowing the model to excessively learn features from the training set that do not necessarily represent the desired target label and therefore are not relevant to the application under consideration. Models were fully optimized by ADAM adaptive gradient-based optimization algorithm. Training is performed on an Intel Core i7 6700K CPU, 32 GB DDR4 RAM with NVIDIA Titan XP GPU with 12 GB and with Tensorflow 1.0, cuDNN 5.1, and Cuda 8.0. Ubuntu 64bits 16.06 LTS was used as the operating

system. Higher training time was reported for model 3, since it is trained with a larger dataset, with an average training time of 30 minutes.

6. Performance Metrics

Given that the proposed CNN-based approach performs both damage localization (classification task) and quantification (regression task), it is important to compute metrics for: the model's accuracy at quantifying the damage detected in each element, the model's precision at not missing damaged elements and thus preventing false negatives, and the model's effectiveness detecting damaged elements and thus avoiding false alarms. Hence, three different metrics are implemented to evaluate the model's performance: Mean sizing error (MSE), damage missing error (DME), and false alarm error (FAE), which are all defined in [54]. MSE is the average quantification error of the outputs defined as follows:

$$\text{MSE} = \frac{1}{\text{NO}} \sum |y_i - o_i|, \quad (14)$$

where y_i and o_i are the estimated and real outputs of the node i , respectively, and NO is the number of output nodes. DME, on the other hand, represents the fraction of damaged elements that are wrongly diagnosed as undamaged. Thus, high DME values correspond to a great number of false negatives, which is not desirable from a safety perspective where a conservative model is clearly preferable in damage assessment. Thus, DME is given by

$$\text{DME} = \frac{1}{\text{NT}} \sum \varepsilon_i^l, \quad 0 \leq \text{DME} \leq 1, \quad (15)$$

where NT is the number of true damaged elements and ε_i^l is equal to 0 if the i -th element is correctly detected and 1 otherwise. ε_i^l is mathematically defined as follows:

$$\varepsilon_i^l = \begin{cases} 1, & \text{if } o_i > 0, y_i \leq \alpha_c, \\ 0, & \text{otherwise.} \end{cases} \quad (16)$$

Damage is considered as detected if the value of y_i is greater than a prescribed critical value α_c . In this work, α_c is considered as the MSE of the test set, which represents the margin of damage the model can accurately assess.

False alarm error (FAE) is defined as

$$\text{FAE} = \frac{1}{\text{NF}} \sum \varepsilon_i^h, \quad 0 \leq \text{FAE} \leq 1, \quad (17)$$

where NF is the number of predicted damage locations and ε_i^h is 0 if the detected damage corresponds to the true damaged element and 0 otherwise. It is calculated as

$$\varepsilon_i^h = \begin{cases} 1, & \text{if } y_i \geq \alpha_c, o_i = 0, \\ 0, & \text{otherwise.} \end{cases} \quad (18)$$

7. Case Study 1: Spring-Mass Structure

Los Alamos National Laboratory (LANL) designed a structure to study different vibration-based damage

identification techniques [55]. The setup consists of an 8-DOF spring-mass system where masses are separated by identical springs as shown in Figure 7. Each mass consists of an aluminum disk with a 76.2 mm diameter and 25.4 mm thickness. The first mass is also connected to a shaker that provides the excitation force. Each mass has an accelerometer that measures the horizontal acceleration data that are used to obtain the transmissibility functions. Experimental data are acquired in a frequency range from 10 Hz to 110 Hz with a frequency resolution of 0.125 Hz. Note that the possible rotation of the masses is not included when obtaining the transmissibility functions. Table 2 shows the physical properties of the structure used in the finite element model for generating the training datasets (see Section 5). The FE model is built using concentrated masses for the discs and linear spring elements for the springs; it considers only horizontal displacement and has a total of seven spring elements and eight degrees of freedom. In our system representation, damage is represented by a spring stiffness reduction (e.g., change of one of the springs for a softer one). Thus, each spring represents one element of the system; hence, a stiffness reduction in one of the springs is equivalent to the damage level y_i described in Section 4.2. For instance, a 20% reduction on the i -th spring's stiffness would correspond to a damage level $y_i = 0.2$.

7.1. Results. Two experimental measurements are available from LANL. The first measurement corresponds to a spring series system where all the springs have the same stiffness (i.e., the system has no damage), whereas in the second one, the fifth spring from the original setup has been changed by another with 55% less stiffness (i.e., system has a 55% damage on the fifth element). The dataset with the undamaged system has been used to calibrate the numerical model to obtain the transmissibility functions as discussed in Section 3.2. All models presented in Table 1 are trained and used to predict the experimental scenario, i.e., the localization and amount of damage of one element.

Model 1 was trained to detect no damage or one damaged element in the spring series system, utilizing a total of 40,000 images, whereas models 2 (for detecting up to 2 damaged elements) and 3 (able to detect up to 3 damaged elements) were trained with 70,000 and 100,000 images, respectively. In this case, the system possesses eight elements, and transmissibility functions were obtained by exciting the first mass of the system with a shaker and measuring the response from the seven remaining masses, thus resulting in a total of seven transmissibility functions. Figure 8 shows two examples of those transmissibility functions that are obtained following the procedure described in Section 3.2. The transmissibility functions were obtained for the fifth element of the system for the cases with one damaged element and no damaged elements (orange and blue, respectively). It can be seen how the transmissibility functions' peaks shift to the left, and at the same time, it is possible to observe a reduction of their magnitudes. As it was previously discussed, a traditional feature extraction from these functions consists in obtaining the

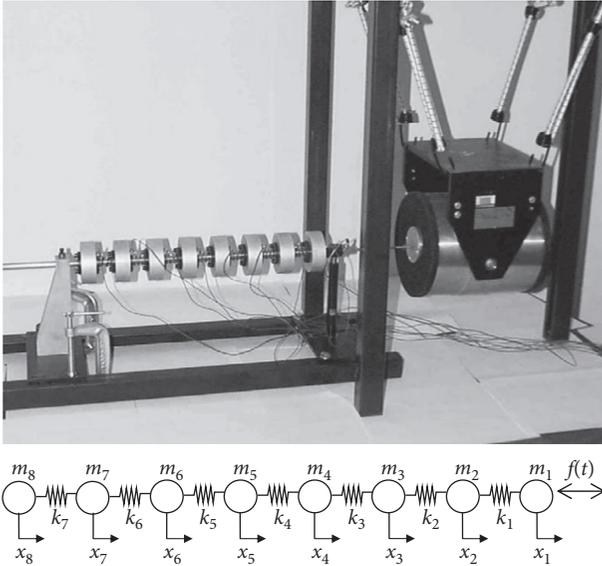


FIGURE 7: 8-DOF mass-spring system at Los Alamos National Laboratory.

TABLE 2: Mass and strength values of the experimental setup.

Mass 1	559.3 g
Masses 2–8	419.4 g
Spring Young modulus k	56.7 kN/m

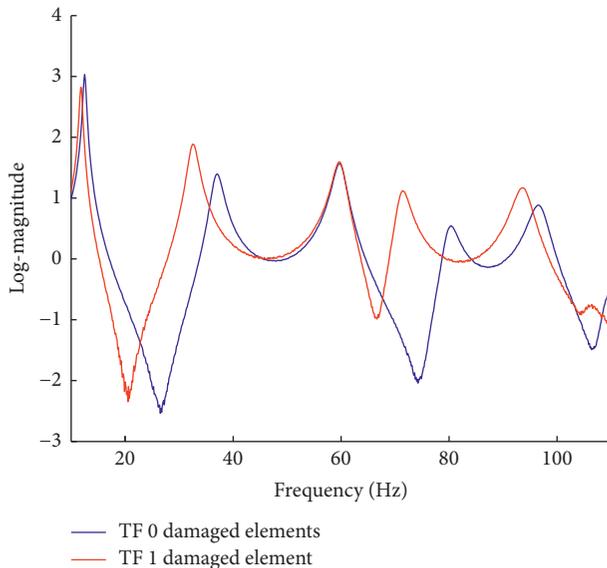


FIGURE 8: Transmissibility functions measured from the fifth element of the spring series system. Blue line corresponds to the no damage scenario, whereas in orange, the system has one randomly damaged element.

antiresonant frequencies, which are strongly associated with the peaks of the functions. Furthermore, these frequencies have been proven to be strongly related with the stiffness of a material or structure [4, 5]. However, the extraction of these features is not only time consuming but also requires the

intervention of an analyst with domain specific knowledge to interpret the results, which is liable to potential subjectivism.

Using these transmissibility functions, training images are generated for each of the models described in Table 1 based on the approach discussed in Section 4. With the finite element model discussed in the previous section, 10,000 images were generated simulating a system with no damaged elements, as well as three other datasets of 30,000 images each, with one, two, and three randomly damaged elements with random noise from 0% to 6%. When training the models, each training dataset is comprised of 85% of the generated images, leaving the remaining 15% of the images to test the model. An example of the images generated from the transmissibility functions is shown in Figure 9, where Figure 9(a) shows an image representation of the seven transmissibility functions corresponding to the system with no damaged elements, whereas Figure 9(b) illustrates a representation of these transmissibility functions when one element is randomly damaged. The colors represent the magnitude of the transmissibility functions at each frequency, as explained in Section 4.2.

Also, note that when the proposed architecture misses to detect a damaged element for any of the trained models, the associated damage to those elements is minimal. On the other hand, Table 3 shows that the maximum value for the FAE corresponds to model 3 with 34.063%. However, Figures 10 and 11 also show that all false alarm damages range between 0% and 10%. Thus, the results obtained when evaluating the test set show that the trained models are reliable when predicting damage level over 10% as these are detected with 100% accuracy (DME and FAE) and low MSE.

For validation purposes, the damage localization and quantification performance of the proposed approach are evaluated using experimental data corresponding to the spring-mass system with a 55% damage level in the fifth element. These results are presented in Figures 12 and 13. We can see that model 1 accurately predicts a 54% damaged in the fifth element. Moreover, no false alarms were detected.

Figure 13(a) shows the results for model 2. This is more conservative than model 1, since it accurately detects a 58% damage at the fifth element, a slightly higher amount of damage than that detected by model 1 (54%) and the real damage (55%). Model 2 also detects three small false negatives, which is not desirable but expected since Figure 11(a) shows that 96% of the damages detected from 0% to 10% correspond to false positives. Figure 13(b) shows the results when evaluating the experimental data with model 3. Once again, we can see small false positives at elements 1, 3, and 4, all of them under 10% as expected from the results shown in Figure 11(b). The predicted damage level is 62% for the fifth element. Hence, model 3 is the most conservative out of the three trained models, and it also presents the highest quantification error (MSE) of 1%, the highest false negatives (DME) with 2,067% and 34.063% of false positives (FAE), all of them in the range from 0% to 10%.

All trained models take an average training time of 18 seconds per epoch. When experimental data are fed to the trained model, the average assessment time is 0.31 seconds for a new image (data point). Therefore, the proposed CNN-

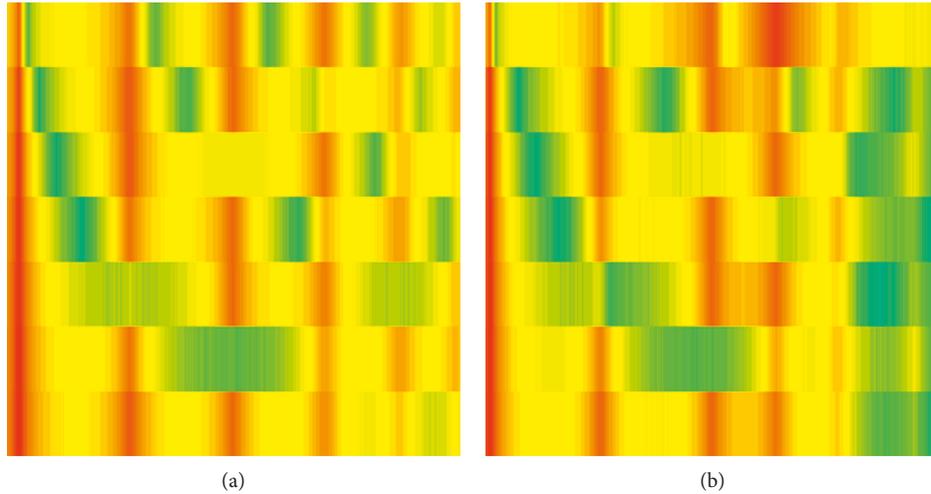


FIGURE 9: Images for the spring-mass system with (a) no damaged elements and (b) one randomly damaged element.

TABLE 3: Training result summary for the spring-mass system when evaluating the test set.

Model	MSE (%)	DME (%)	FAE (%)
Model 1	0.271	0.916	32.548
Model 2	0.478	2.139	28.057
Model 3	1.003	2.067	34.063

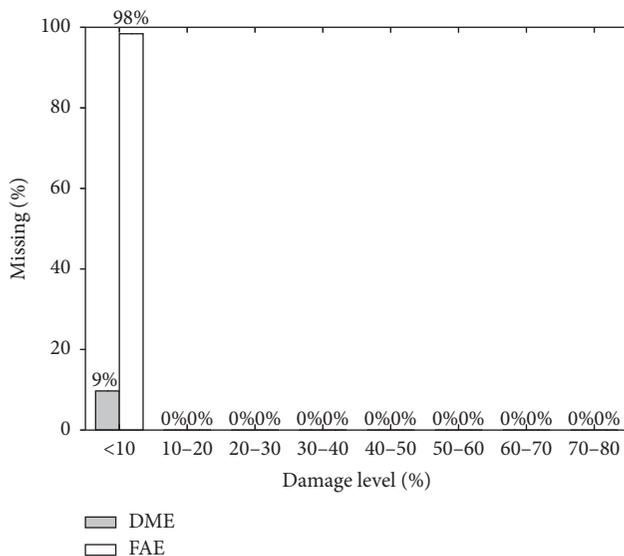


FIGURE 10: DME and FAE for model 1, spring series case study.

based approach satisfactorily detects and quantifies damage above the 10% level for the considered spring-mass system, delivering more conservative results when asked to detect elements with higher damaged levels. It might also be considered for online damage monitoring, given that it takes under one second to yield an accurate diagnosis for new unseen measurements [19].

In addition to the injected noise in the training dataset (see Section 5), the model's robustness to randomness due to noise contamination is also evaluated by means of three test

images. Each image is generated according to the procedure described in Section 4.2 and simulating a 55% damage level at element 5. Moreover, the noise contamination level applied to each image is 2%, 4%, and 6%. Figure 14 shows the results from model 1 when evaluated with these images as its input. The model outputs the exact same results for all three images, predicting a 53% damage level at element 5 and no false positives. That is, the convolutional layers from model 1 eliminate the noise entirely from the input images regardless of their contamination level.

8. Case Study 2: Structural Beam

Meruane and Mahu [27] proposed an experimental setup to identify damage in a structural beam through transmissibility data and antiresonant frequencies. The experiment was set up at the Laboratory of Mechanical Vibrations and Rotordynamics at the University of Chile. The experiment consists of a structural beam where damage is generated by saw cutting the beam. The transmissibility data are recorded with accelerometers along the structural beam. Figure 15(a) presents the experimental beam of 1-meter longitude and a rectangular cross-sectional area of $25 \times 10 \text{ mm}^2$. Both ends of the beam are suspended on soft springs to simulate a "free-free" boundary condition. The excitation force is generated by a shaker at one end of the structure, and the response is measured with 11 accelerometers (therefore, 10 transmissibility functions are obtained). Experimental data are acquired for a frequency range from 1 to 2000 Hz with a frequency resolution of 1 Hz.

The beam is modeled using unidimensional beam elements with two nodes per element and two degrees of freedom (DOFs) per node. The beam (and its finite element model) was divided into 20 elements of 5 cm each, as shown in Figure 16, resulting in a FE model with 42 DOFs. The transmissibilities are computed using the translational DOF at nodes 1, 3, 5, . . . , 21, node 1 being the reference. To simulate stiffness reduction (i.e., damage) in beams, saw cuts

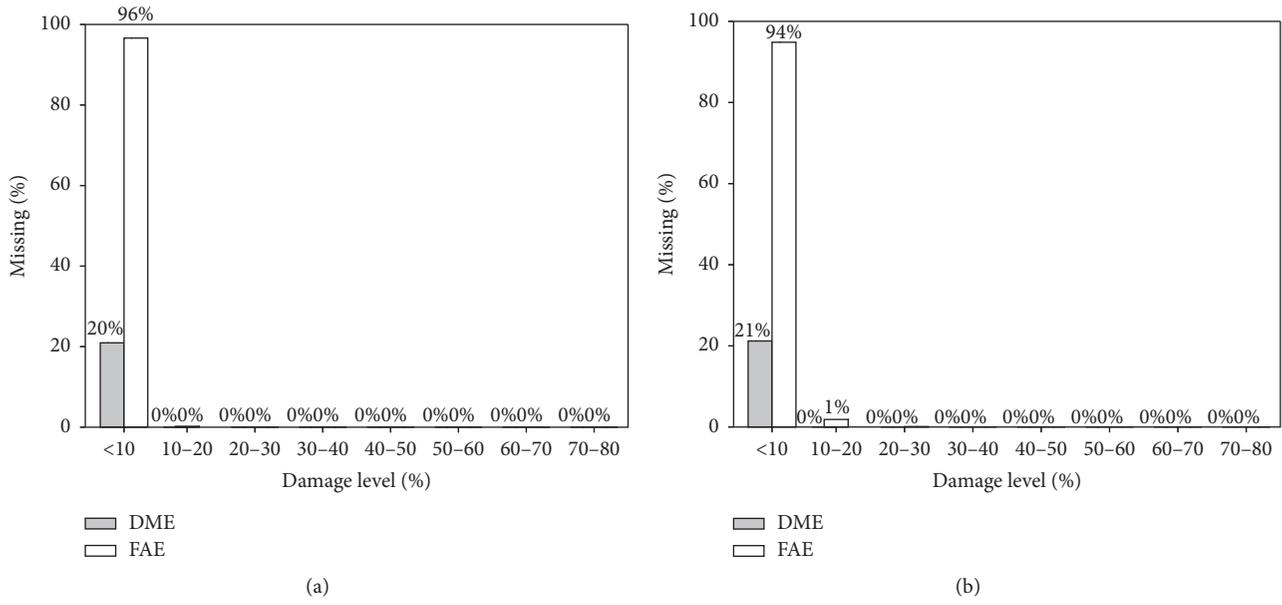


FIGURE 11: DME and FAE for model 2 (a) and model 3 (b), spring series case study.

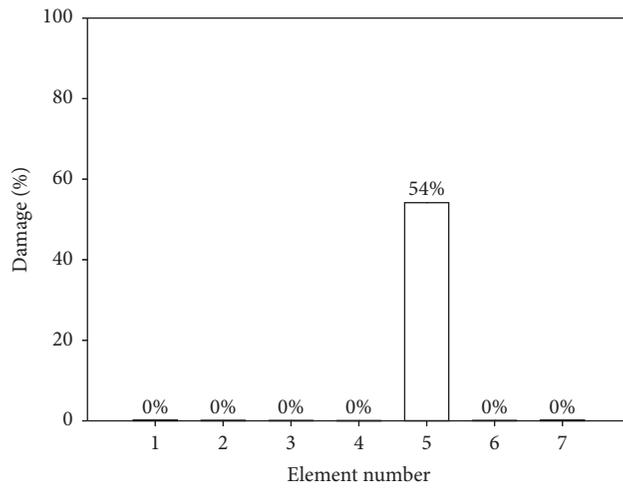


FIGURE 12: Predicted damage for the spring-mass system when evaluating model 1. Expected damage level of 55% at the fifth element.

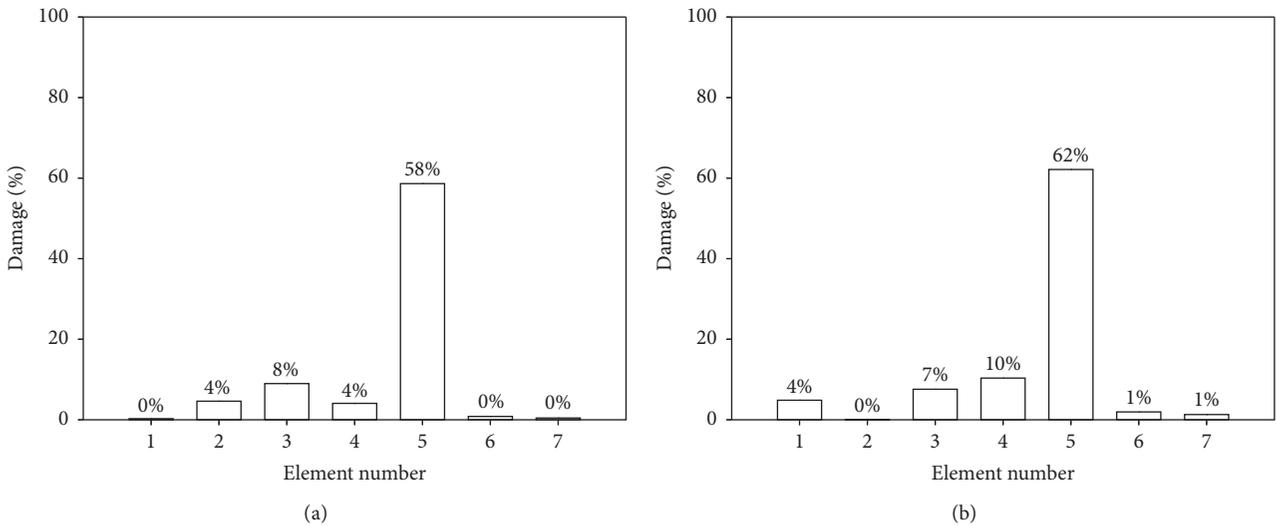


FIGURE 13: Predicted damage for the spring-mass system when evaluating model 2 (a) and model 3 (b). Expected damage level of 55% at the fifth element.

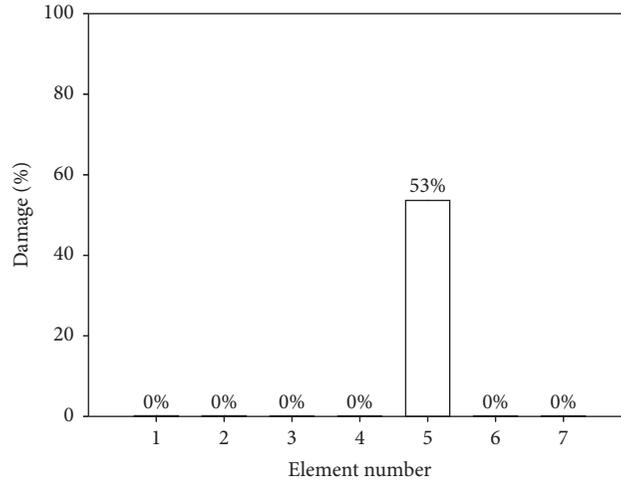


FIGURE 14: Spring system. Damage quantification results from the proposed CNN model based on three different images with noise levels of 2%, 4%, and 6%.

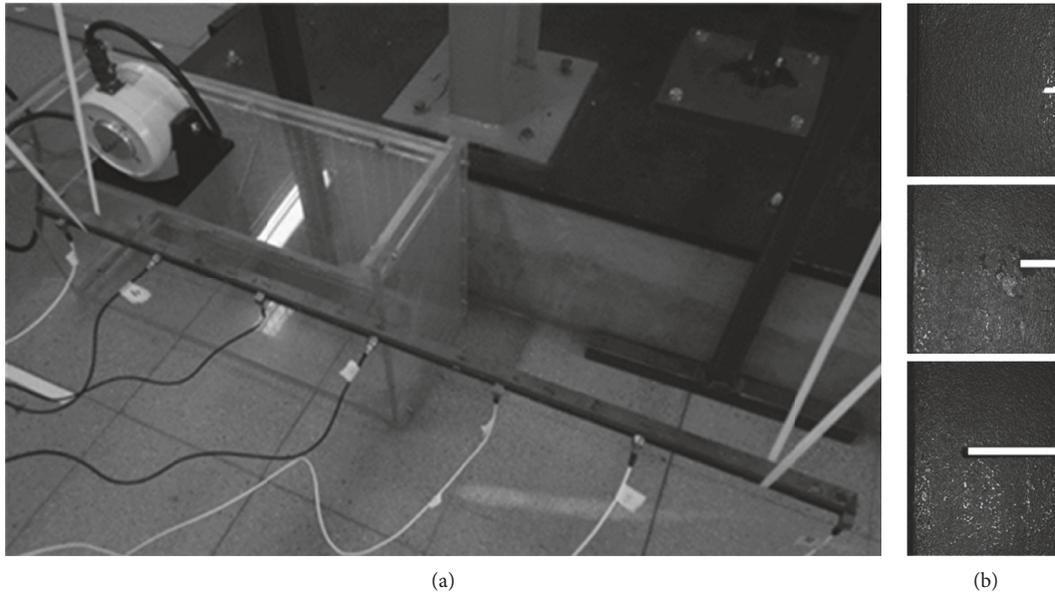


FIGURE 15: (a) Experimental beam setup [19]. (b) Example of saw cuts inflicted to the experimental beams [19].

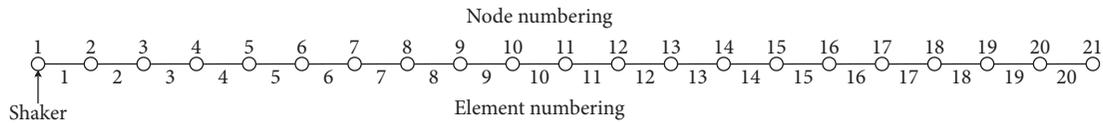


FIGURE 16: FEM element and node numbering.

of different lengths were inflicted in a set of beams. Four damage scenarios are studied, where damage scenarios 1 and 2 correspond to two different beams with one saw cut each (i.e., one damaged element), whereas damage scenarios 3 and 4 consist of two different beams with two and three damaged elements, respectively. Figure 15(b) shows three examples of possible saw cuts inflicted in the experimental

beams. Since there is no direct relationship between saw cuts and stiffness reduction of the structure, as there was for the spring-mass series system discussed in the previous section, the damage level is represented by the saw cuts' length. Details for each scenario are provided in Table 4. To detect these damages, all three models described in Section 5 are considered in this section.

TABLE 4: Experimental beam damage scenarios.

Damage scenario	Number of cuts	Distance from the left side (mm)	Damaged element	Cut depth (mm)
1	1	313	7	7
2	1	637	13	9
3	2	361	8	8
		812	17	15
4	3	363	8	13
		574	12	8
		696	14-15	6

8.1. Results. As previously discussed in Sections 3 and 7, transmissibility functions are known for their strong relationship with the stiffness of a material. Peaks of the TF are related with the antiresonant frequencies, which normally need to be manually extracted by an analyst with specific domain knowledge. Figure 17 shows two transmissibility functions measured for the 10th element of the beam. The first function corresponds to the beam with no damage inflicted in any element and the other shows the TF for three randomly damaged elements. As in the previous case study, the peaks shift to the left when damage is present along with a change in their magnitudes.

When compared to the previous experiment, where the system behaved like a discrete 8-mass system, the beam behaves as a continuous solid when vibrating. In addition, there are 10 transmissibility functions available for 20 possible damaged elements, instead of one transmissibility function per spring element as in case study 1. Thus, we have less information per damaged element. Given the four damage scenarios presented in Table 4, all three models in Table 1 are trained using 40,000, 70,000, and 100,000 images, respectively, generated from transmissibility functions based on the finite element model discussed in Section 3.2. Figure 18 shows two examples of the generated images, with ten transmissibility functions, where Figure 18(a) represents a beam with no damage, and Figure 18(b) is obtained from a beam with three randomly damaged elements.

Table 5 shows the overall MSE, DME, and FAE when evaluating the test set for each trained model. Note that the MSE is smaller when the training examples have fewer damaged elements since it is more challenging for the CNN-based model to detect and quantify several damage levels at the same time. In turn, this translates in the CNN holding more information from the transmissibility functions in its weights and biases. Thus, the MSE reaches a minimum at 0.27% for model 1 and a maximum value of 1.3% for model 3.

From Table 5, we can also see that the FAE is the only metric with the minimum obtained for model 3. Thus, even though the accuracy at localizing and quantifying damage decays (i.e., higher MSE), the false alarm rate improves when the model is trained with more damaged elements (i.e., lower FAE): the extra information given to the CNN-based model with more damaged elements in the training process allows it to be more reliable at detecting damaged elements. Furthermore, Figures 19 and 20 show that the DME is below 10% in all three models, for damage levels above 20%. Thus, we can say that most elements with a damage level above 20% can be accurately

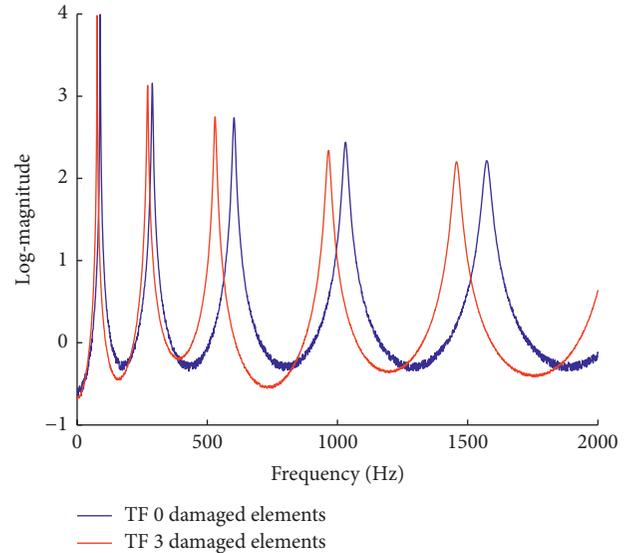


FIGURE 17: Transmissibility functions measured for the tenth element of the structural beam. Blue line corresponds to the no damage scenario, whereas in orange, the system has three randomly damaged elements.

detected and quantified. Furthermore, the FAE values are higher than the DME for all the damage ranges shown in Figures 19 and 20, which indicates that false positives are more recurrent than false negatives for low damage levels, making the proposed CNN-based approach a conservative one.

To evaluate the trained models, we evaluate the all experimental scenarios described in Table 4. Figures 21–24 show the results of each model for the different damaged scenarios. We compare the results from the proposed CNN-based model, with a simple multilayer perceptron. This comparison is further discussed in Section 9.

First, we evaluate the results from model 1 for every scenario in Table 4. This model can accurately detect the damage in scenarios 1 and 2 both of which contain one damaged element. However, for scenario 3, where the beam was damaged with two saw cuts, Figure 23(a) shows that this model accurately diagnoses a high damage level of 86% at element 17, and it also detects a small level of damage of 1% at element 8, which can be interpreted as a false positive. This result is expected since model 1 is trained to detect only one damaged element per beam. Hence, the results indicate the necessity of training a model with multiple damaged elements. The latter is corroborated by the results shown for

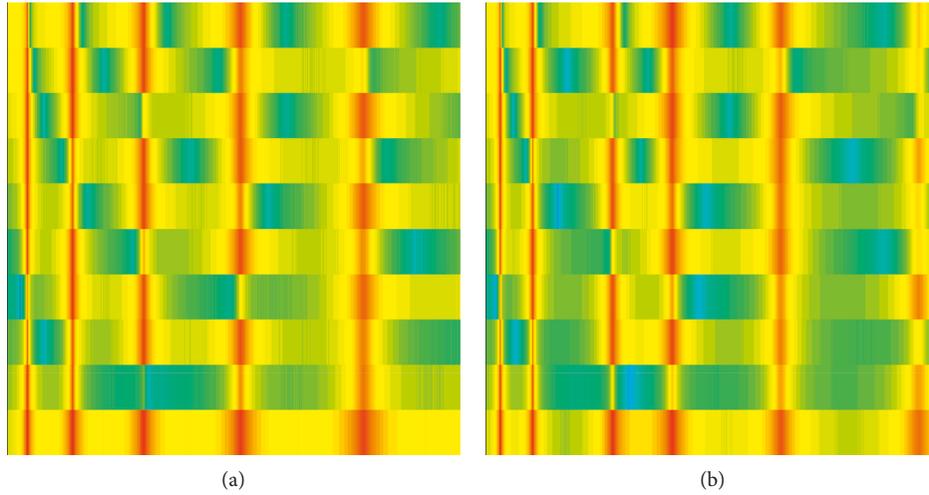


FIGURE 18: Images for structural beam with (a) no damaged elements and (b) three randomly damaged elements.

TABLE 5: Training result summary for the experimental beam.

Model	MSE (%)	DME (%)	FAE (%)
Model 1	0.274	2.85	80.52
Model 2	0.680	6.24	59.19
Model 3	1.334	6.70	57.55

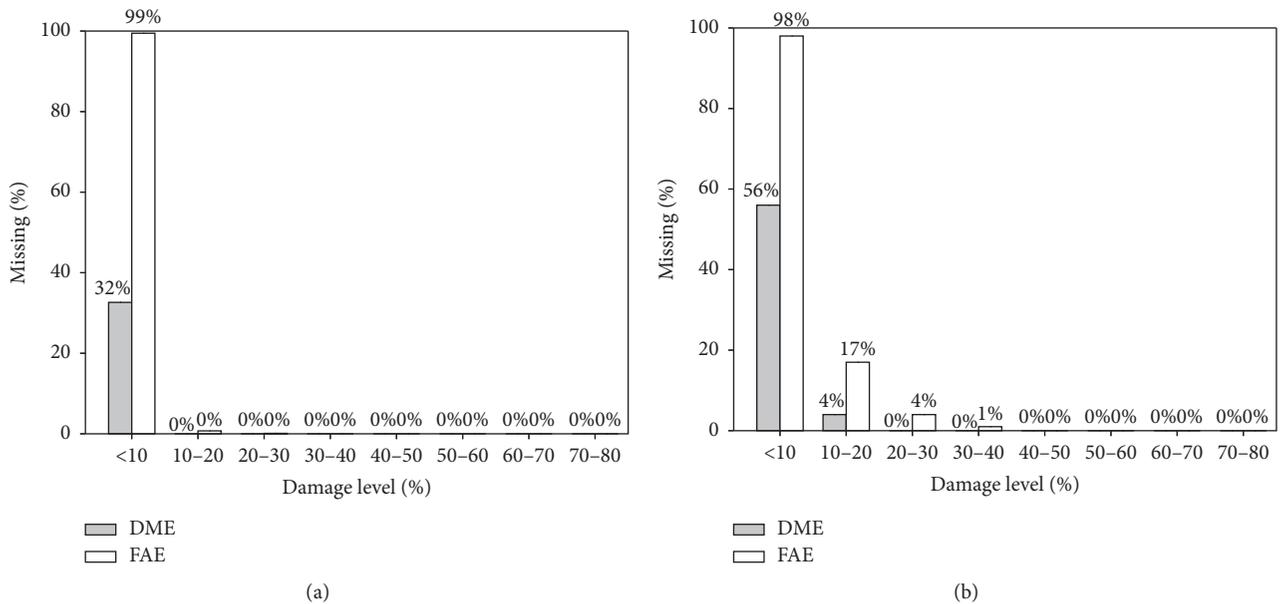


FIGURE 19: DME and FAE versus damage level for the structural beam using (a) model 1 and (b) model 2.

scenario 4 in Figure 24(a), as model 1 fails to detect the three damaged elements by delivering small quantification of the expected damages when compared with scenarios 1 and 2, which involve saw cuts of similar or greater lengths than in scenario 4.

Furthermore, models 2 and 3 are trained according to Table 1. From the aforementioned figures, we can see that model 2 accurately predicts the damage in element 7 for

scenario 1 (Figure 21(b)), with a slight increase in the damage level when compared with the results obtained by model 1, giving out a small false positive of 12% at element 17. A similar result is observed in Figure 22(b) for scenario 2, where, similar to model 1, model 2 detects damages at elements 13 and 14, but this time, the detected damage is evenly distributed between these two elements. Thus, when trained to detect more damaged elements, the proposed

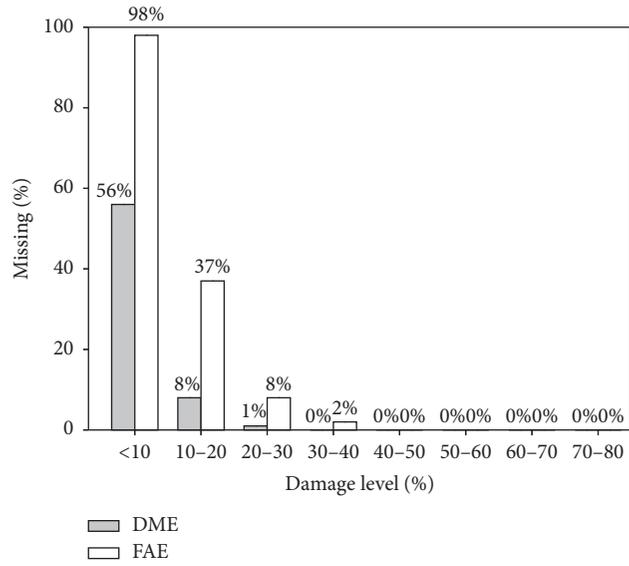
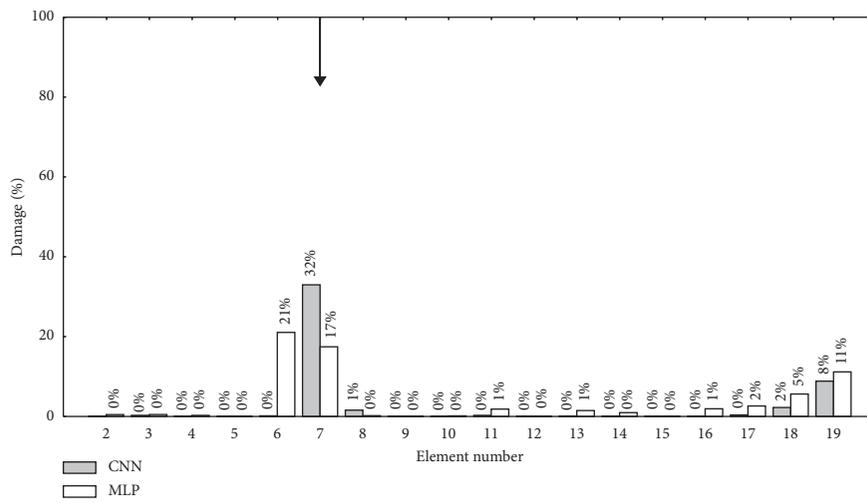
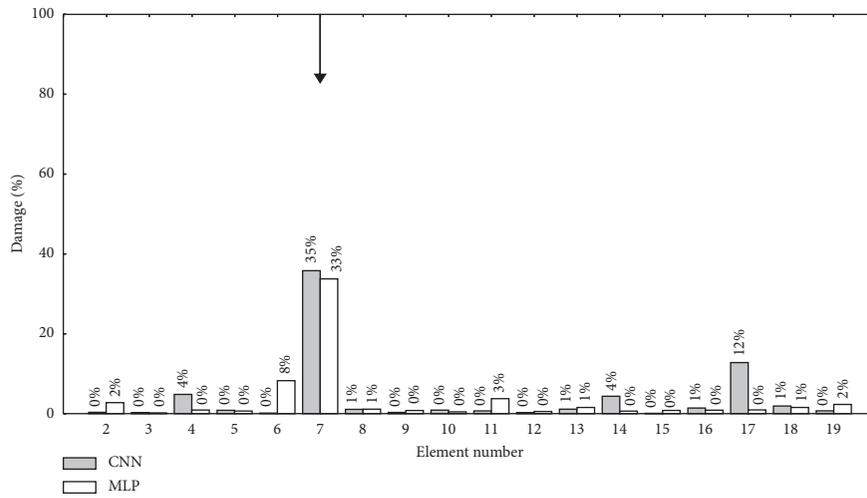


FIGURE 20: DME and FAE versus damage level for structural beam using model 3.

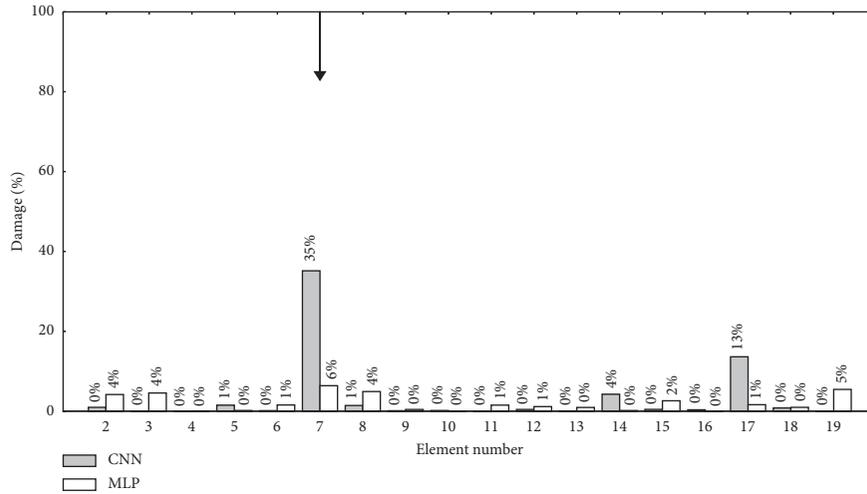


(a)



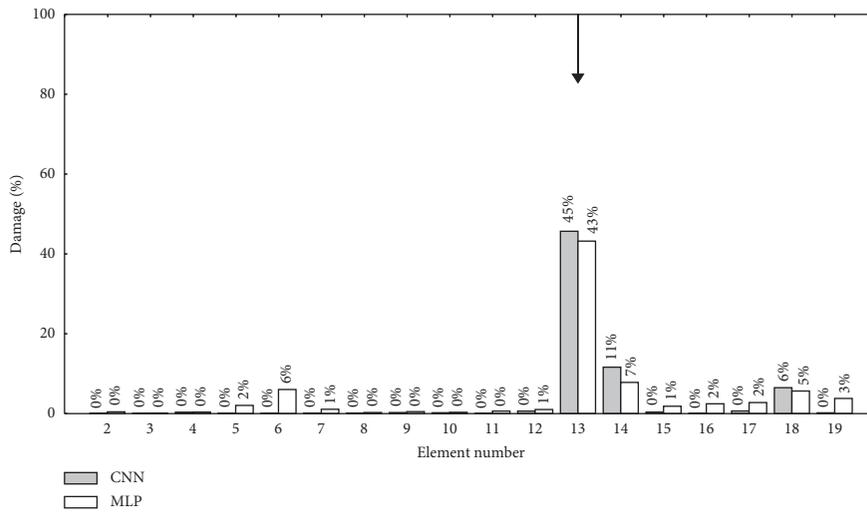
(b)

FIGURE 21: Continued.

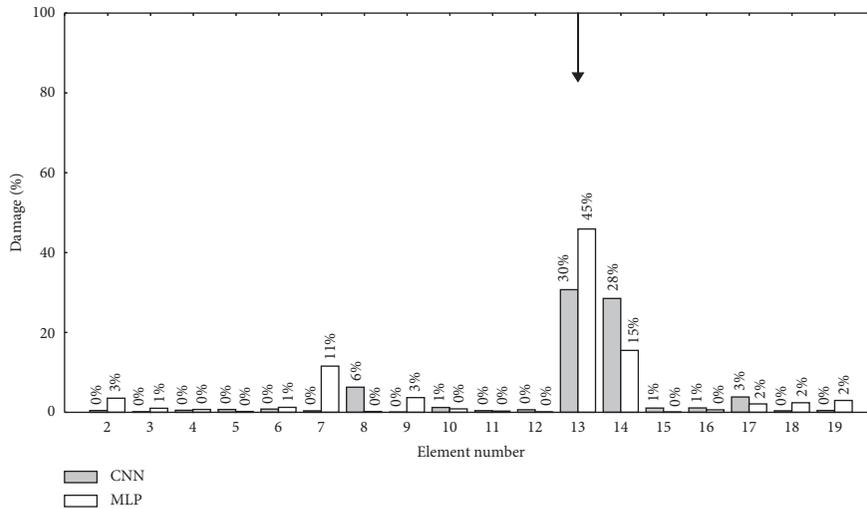


(c)

FIGURE 21: Results for damage scenario 1 according to (a) CNN and MLP models 1, (b) CNN and MLP models 2, and (c) CNN and MLP models 3.



(a)



(b)

FIGURE 22: Continued.

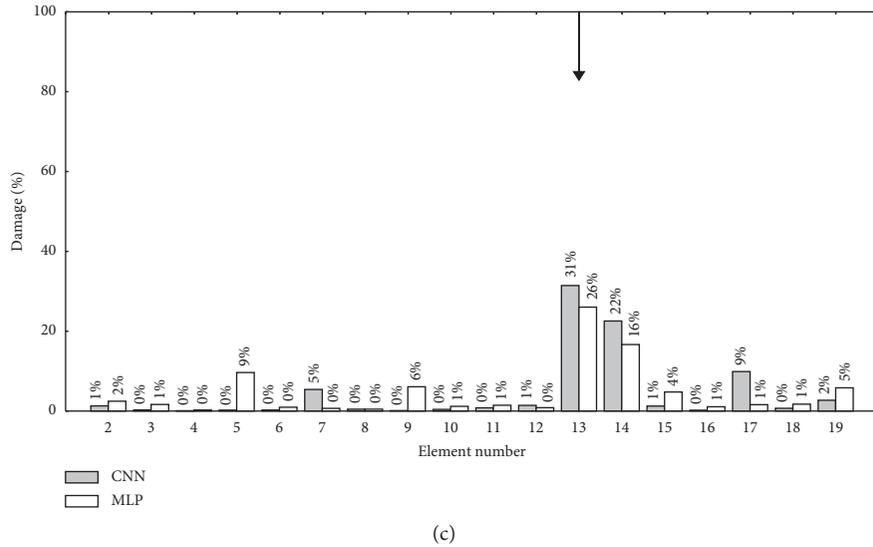


FIGURE 22: Results for damage scenario 2 according to (a) CNN and MLP models 1, (b) CNN and MLP models 2, and (c) CNN and MLP models 3.

CNN architecture is robust at detecting one damaged element, predicting small false positives.

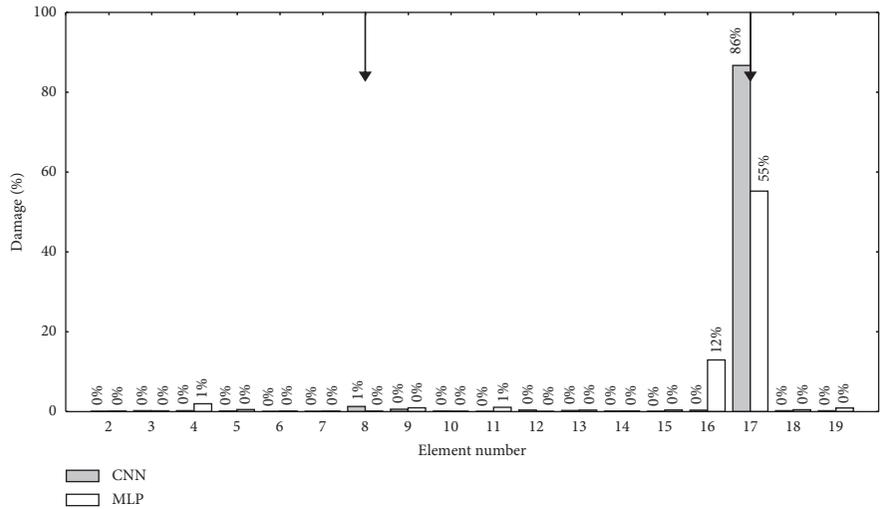
Results obtained for scenario 3 are also satisfactory as seen in Figure 23(b), where model 2 not only detects a high damage level at element 17 but also detects a 17% damage level at element 8 as it was expected (see Table 4). The model also gives out a small false positive for the element adjacent to where the real damage is. As discussed in [27], in continuous systems, it is natural to detect a false-positive damage in an element adjacent to the real damaged element, since the elements in the structure are not independent (as it is the case for the spring series system). Furthermore, Figure 24(b) shows that even though model 2 is trained to detect up to two damaged elements, it is still capable of accurately detecting three damaged elements, assessing damages at elements 8, 12, and 14 as expected from the experimental setup. We can observe the same false-positive effect for the 7th and 11th elements, which are adjacent to the damaged elements.

Lastly, when analyzing the experimental scenarios with model 3, we can see in Figures 21(c) and 22(c) how similar the results are when compared with models 1 and 2 for experimental beams with one saw cut, which is another indication of the proposed CNN-based approach's robustness at detecting one damaged element. On the other hand, even though Figure 23(c) shows a correct prediction of the damaged elements, two other false positives arise when evaluating scenario 3. Nevertheless, although the latter is not desirable, this is a more conservative result, and based on the results for the test set shown in Figure 20(b), these small false positives are to be expected. Thus, model 2 and model 3 correctly predict the true damaged locations for all scenarios, but they tend to yield small false damages, all of them lower than 15% damage. Also, when trained to detect more damaged elements, the CNN-based models tend to propagate the damage to adjacent elements.

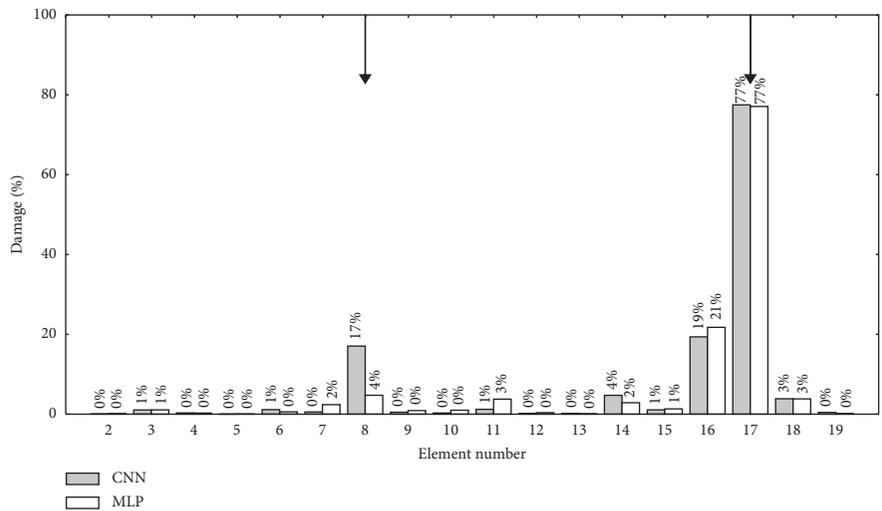
Similar to the spring-mass system in Section 7, we evaluated the CNN models' robustness to noise contamination, which represents randomness in the measurements due to experimental noise, by analyzing two different cases. In the first one, proposed model 2 was applied to generated images from transmissibility functions for a beam with a 40% damage at elements 8 and 12. For this configuration, three different images were generated: one with 2% noise contamination and the others with 4% and 6%. In the second case, a beam with a 55% damage at elements 4, 12, and 18 was analyzed by means of proposed model 3 and with three images with the same noise contamination levels as in the first case. This is summarized in Table 6.

Figure 25(a) shows the results from model 2 when evaluated using the images with different noise levels. Note that the model correctly identifies the damage at elements 8 and 12 regardless of the noise level applied to the transmissibility functions. Small false-positive damage detection arises at element 11. However, according to Figure 19(b), these false positives under 10% are to be expected. Furthermore, Figure 21 shows that the model tends to detect damage at the adjacent elements of the real damage, since the beam is a rigid body and a damage level at one element would likely affect the integrity of its neighboring elements. As for the damage level quantification, it is clear from Figure 25 that noise level does not have a major effect on the model's predicted damages, where a small variation can be observed for the damaged elements as well as a negligible increase on the damage level at element 11.

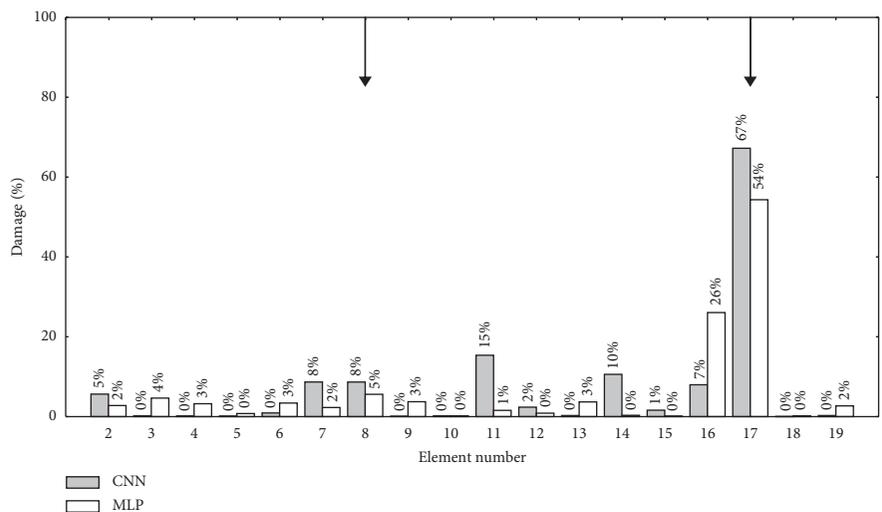
A similar result is obtained when evaluating model 3 with its corresponding noise contaminated images. Figure 25(b) shows how model 3 correctly identifies damage location at elements 4, 12, and 18. However, note that the model underestimates the damage level at the three damaged elements. Nevertheless, the elements adjacent to those with



(a)

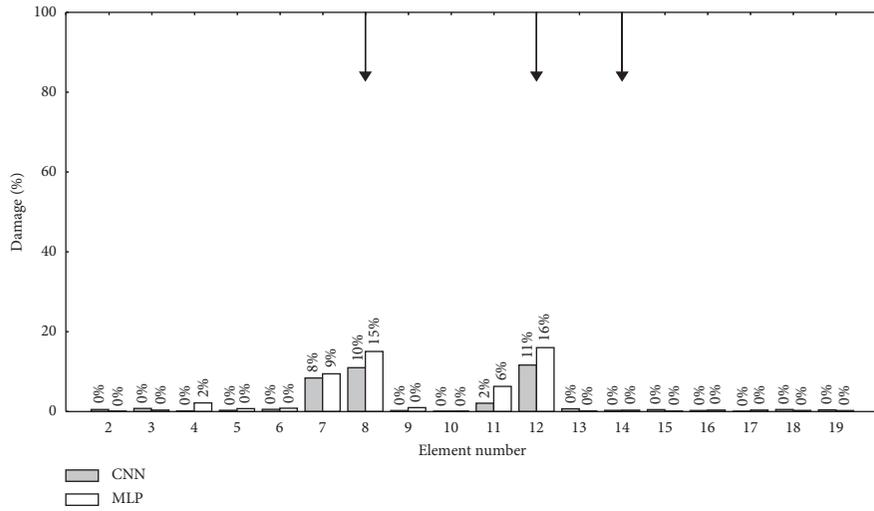


(b)

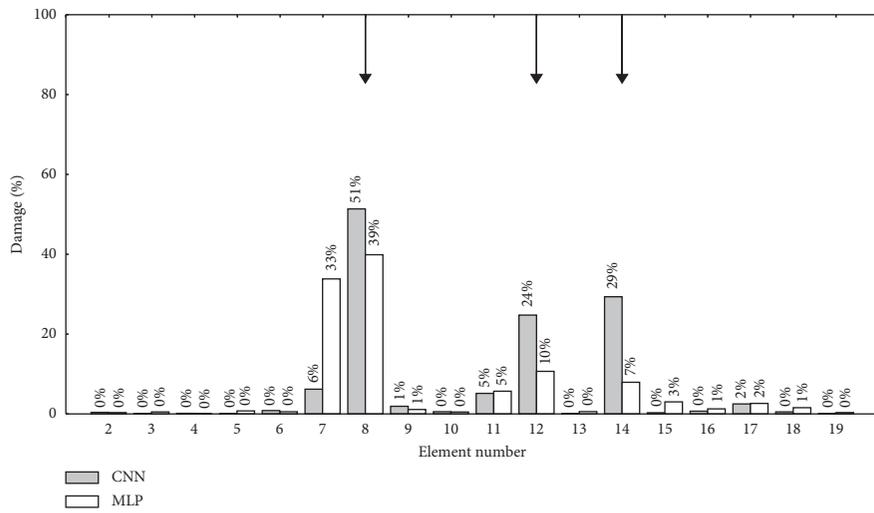


(c)

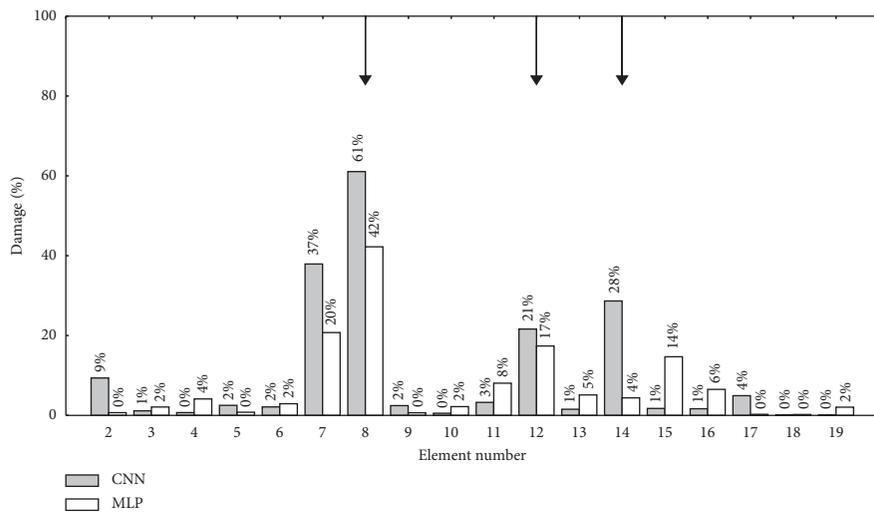
FIGURE 23: Results for damage scenario 3 according to (a) CNN and MLP models 1, (b) CNN and MLP models 2, and (c) CNN and MLP models 3.



(a)



(b)

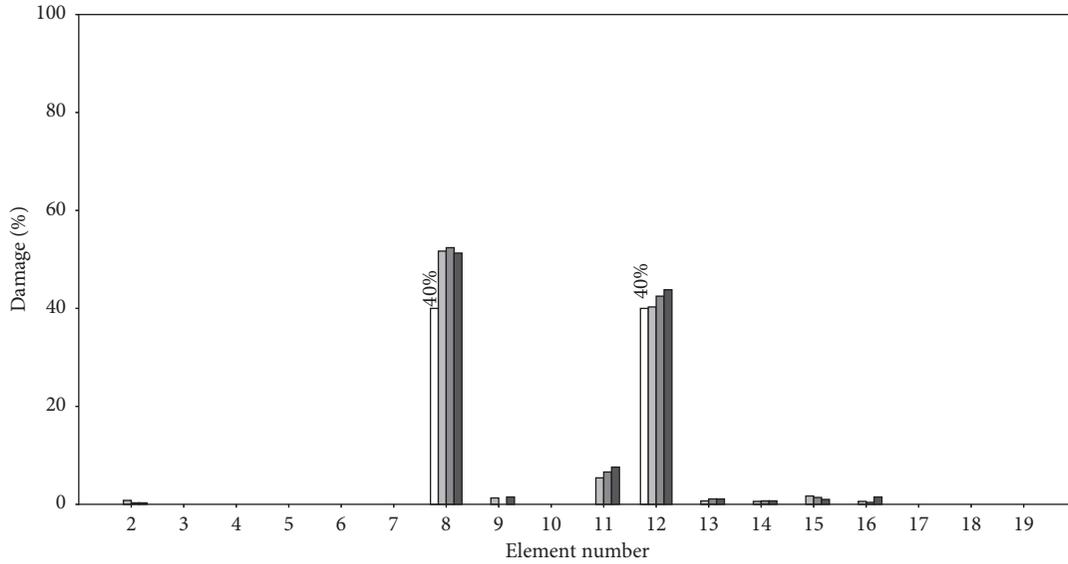


(c)

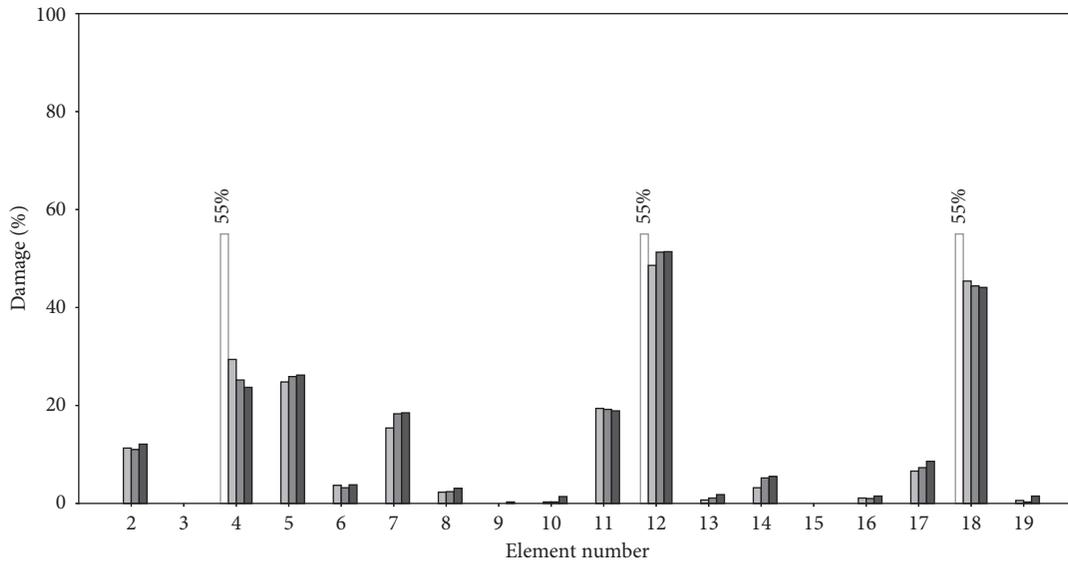
FIGURE 24: Results for damage scenario 4 according to (a) CNN and MLP models 1, (b) CNN and MLP models 2, and (c) CNN and MLP models 3.

TABLE 6: Elements with corresponding damage level of images for the noise robustness assessment.

	Model 2			Model 3	
Damaged element	8	12	4	12	18
Damage level (%)	40	40	55	55	55



(a)



(b)

FIGURE 25: Noise robustness for model 2 (a) and model 3 (b).

damage present false positives, e.g., elements 5, 11, and 17. We can ascribe these false detections to a distribution of damage from one element into its neighbors, observing that when summing up the damage quantification of elements 4-

5, 11-12, and 17-18, we obtain a slightly higher estimation than the expected damage. Moreover, damage level quantification does not present significant variation with higher noise contamination. Hence, we can assert that the

convolutional layers in the proposed CNN-based models successfully eliminate the noise contamination injected to the images, thus allowing one to argue that the models are considerably robust to randomness due to noise.

Finally, to test the robustness of the CNN architecture, we evaluate its performance when training each model with different dataset sizes. From the original datasets, a train and test set are defined just as it was done for the previous models. Different models are trained for different proportions of the train set, keeping the test set intact so the results can be comparable. Training is done with 100%, 75%, 50%, and 25% of the training set. Table 7 presents the results obtained from this approach. As expected, the performance of the model changes when using a smaller portion of the dataset, since the models are given less information to train themselves. However, we can see that even when using 25% of the training set, all models yield a small error. This shows a great robustness from the trained model, especially considering that the test set was kept constant, which means that when training the models with 25% of the training dataset, the number of training and testing images is almost the same. Hence, we would expect the CNN model to perform poorly. These results show that even when no large amount of data is available, the proposed CNN-based architecture can still be implemented.

9. Comparison with Other Models

From a practical point of view, it is interesting to compare the results from the deep CNN-based models with other shallow approaches as the one proposed in [27], i.e., a shallow multilayer perceptron- (MLP-) based models.

9.1. Comparison with Shallow Multilayer Perceptron. To have a fair basis for comparison, the MLP models have the same architecture as the models based on the proposed approach but without the convolutional layers. That is, we evaluate the MLPs with one hidden layer of 1024 units and an output layer with a total number of units equal to the possible damaged elements. The input layer is fed directly with the generated images. No features are automatically generated, but one has a considerably simpler model due to the reduced number of weights and biases, making this shallow MLP easier and faster to train. Moreover, the MLP models were fully optimized by Adam adaptive gradient-based optimization algorithm and regularized via dropout (with 50% keep probability) and early stopping with up to 100 epochs.

Taking as basis for comparison the more challenging case represented by the structural beam discussed in Section 8, the MLP is trained for the three cases presented in Table 1, thus resulting in three models that only differ in terms of the values of the weights, and then evaluated for the four experimental scenarios shown in Table 4. Indeed, Table 8 shows the obtained results for these MLPs (i.e., for MLP 1, 2, and 3) after the training process. We can observe that not only the accuracy decreases (i.e., higher MSE) compared with the results presented in Table 5 for the models based on

TABLE 7: Model performance comparison for different training set sizes.

Model	Dataset portion (%)	MSE (%)	DME (%)	FAE (%)
Model 1	100.00	0.27	2.46	72.14
	75.00	0.26	2.62	69.74
	50.00	0.37	3.21	75.74
	25.00	0.87	13.84	70.49
Model 2	100.00	0.59	4.42	66.22
	75.00	1.01	13.67	62.91
	50.00	0.69	5.55	63.61
	25.00	0.96	7.04	66.69
Model 3	100.00	0.98	6.59	55.66
	75.00	1.52	15.82	55.93
	50.00	1.20	6.77	59.80
	25.00	1.39	7.87	60.93

TABLE 8: Training result summary for structural beam with a MLP.

Models	MSE (%)	DME (%)	FAE (%)
MLP 1	1.35	13.39	76.35
MLP 2	2.61	16.52	57.43
MLP 3	4.42	24.29	71.99

the proposed approach but also the FAE does not improve with more damaged elements per image. Furthermore, as we can see in Figures 26 and 27, when evaluating the test set, MLP 1 manages to accurately predict damage levels over 30%. However, MLP 2 and MLP 3 fail to predict damaged elements with a high confidence level as one can infer from the high values for the DME and FAE metrics for damage levels between 0% and 50%.

We now compare the shallow MLP models with the proposed deep CNN-based models when predicting the location and the amount of damage for the experimental beam damage scenarios in Table 4. Results for damage scenario 1 were previously presented in Figure 21. Note that both the MLP- and the CNN-based models provide good results when trained to detect one or two damaged elements (i.e., CNN and MLP models 1 and 2). However, MLP 3 fails to predict one damaged element when trained to detect three damaged elements, whereas the CNN-based model 3 delivers a precise location and quantification of the damaged element with a small false positive of 12% at element 17.

Figure 22 reports the results for damage scenario 2. Overall, in this case, the MLP models yield similar results as the CNN-based models in terms of localization of damage. Nevertheless, Figures 22(b) and 22(c) show that most of the detected damage levels are under 40% for the MLP-based models 2 and 3, and given the CNN-based models' results obtained for the DME and FAE shown in Figures 26(b) and 27, one can argue that the MLP models do not perform as well as the CNN-based models for these damage levels due to the high rate of false positives. Moreover, Figure 23 contains the results for damage scenario 3, where Figure 23(b) corresponds to CNN and MLP models 2 trained to detect two damaged elements. Note that the MLP model does not identify the damage at the 8th element, but it accurately predicts the damage level at element 17, whereas

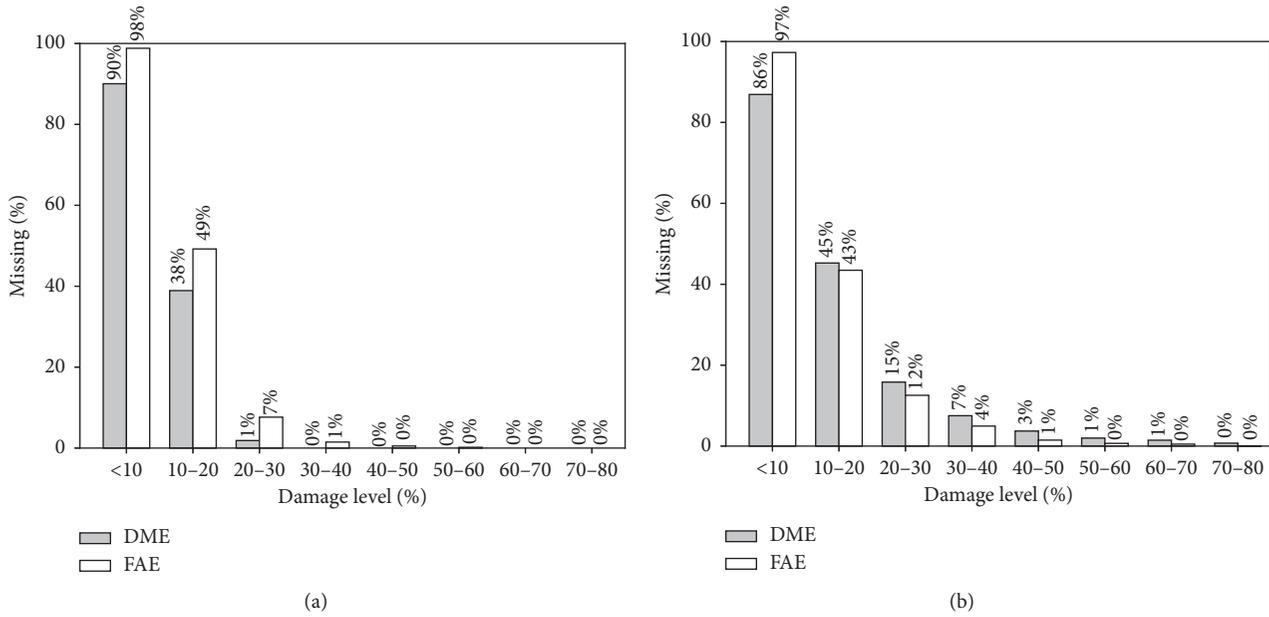


FIGURE 26: MLP’s DME and FAE versus damage level for the structural beam using (a) MLP 1 and (b) MLP 2.

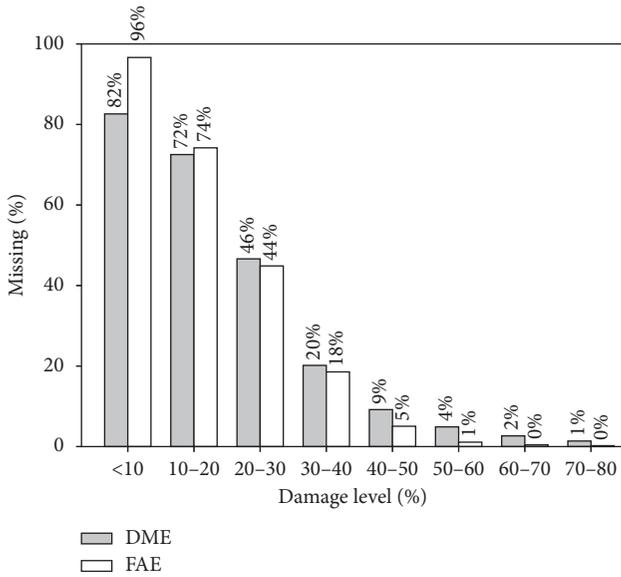


FIGURE 27: MLP’s DME and FAE versus damage level for the structural beam using MLP 3.

the CNN-based model satisfactorily identifies both damaged elements.

This same trend in the results can be observed for damage scenario 4, as shown in Figure 24. Although both the MLP and CNN models fail to detect the three damaged elements when trained to detect only one element (i.e., CNN and MLP models 1), the MLP predicts most of the damage level under 30% which, as it was discussed in Section 8.1, is not a reliable diagnosis performance based on the DME and FAE values shown in Table 8 and Figure 27. At the same time, the CNN-based models deliver more accurate results for every damaged element and with

better scores for DME and FAE metrics as presented in Figure 20 and Table 5. Hence, even though the MLP models can identify and localize some of the damaged elements in the presented scenarios, these results are inferior when compared to the CNN-based models, particularly when dealing with beams with multiple damaged elements.

9.2. Comparison with a Multilayer Perceptron Trained Using Antiresonant Frequencies. Damage assessment using transmissibility functions has been done in the past. In particular Meruane and Mahu [27] used the frequency response functions to manually extract the antiresonant frequencies through the “dip-picking” method. These frequencies are then used as input to a regression neural network which is trained to quantify damage in each element of a beam. The training data were generated using an updated FE model to obtain numerical antiresonances.

A 1.5% noise was numerically injected to the antiresonant frequencies. The multilayer perceptron had 20 input nodes, 80 hidden nodes in the hidden layer, and 18 nodes in the output layer. The model was trained using input data from beams that have up to 2 damaged elements the same way as model 2 was trained. Figure 28 shows a comparison of the DME and FAE between the proposed model and the results obtained with the antiresonant frequency-based MLP [27].

Antiresonant frequency-based MLP has a mean sizing error of 1.53% [27], greater than the one obtained with the model 2 CNN (0.68%). This shows that using the full transmissibility functions, the CNN can extract more relevant features than just the antiresonant frequencies. Also, the CNN has a much lower chance to miss a damaged element but shows a higher amount of false alarms when the

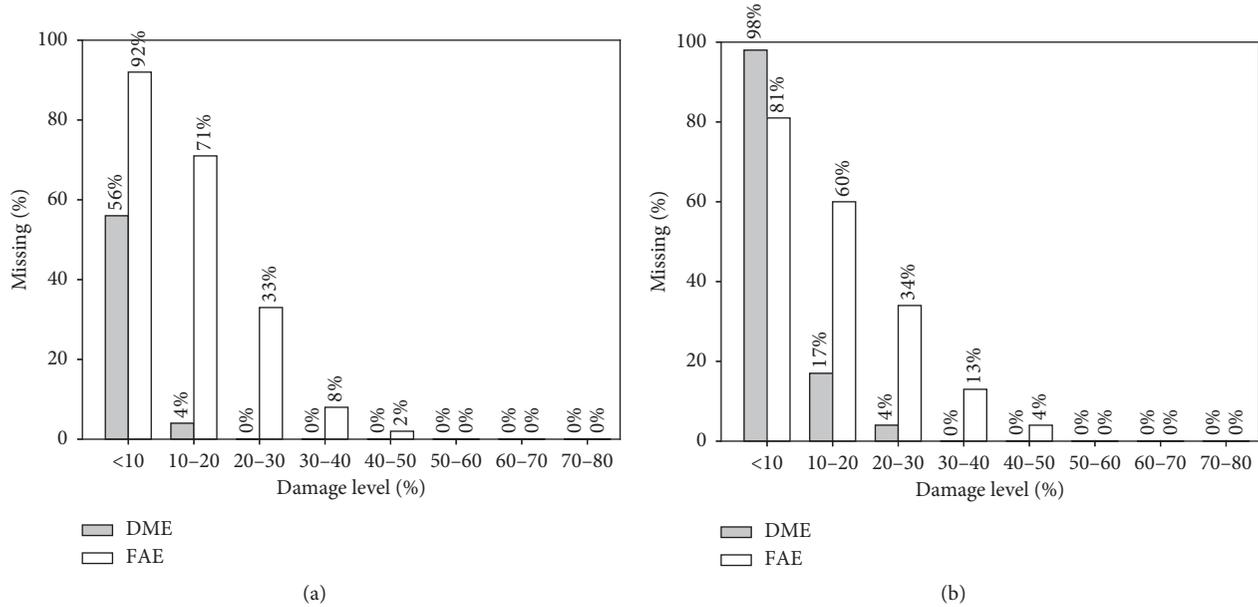


FIGURE 28: Comparison between the proposed CNN model 2 and the antiresonant frequency-based MLP [27]: DME (a) and FAE (b).

damage level is below 10% but has lower FAE once the damage exceeds the 10% threshold and has an overall better performance.

From Meruane and Mahu [27], the experimental results obtained for cases 1 and 2 from the structural beam are directly comparable with the results yielded by model 2 when evaluating case studies 1 and 2 from Table 6. Figure 29 shows that similar results are achieved when comparing the models' performances over these experimental setups. We can see from Figure 29(a) that when evaluating experimental case 1, both models accurately predict the location of the damage. However, the CNN outputs three small false positives, while the MLP yields two false predictions. Similar results are obtained when testing the models with experimental case 2, where the MLP does not output any false positive, while some small ones can be seen for the CNN's output, which also splits the real damage at element 13 into elements 13 and 14. The better adjustment of the proposed CNN-based model to the numerical FE model (Figure 28) can serve as an explanation for the differences in the presented experimental results, where the MLP presented slightly better results than the proposed CNN model regarding the false damage detected. Nevertheless, the proposed CNN-based model and the MLP with manually extracted features have comparable results when identifying the location and quantification of the damaged element.

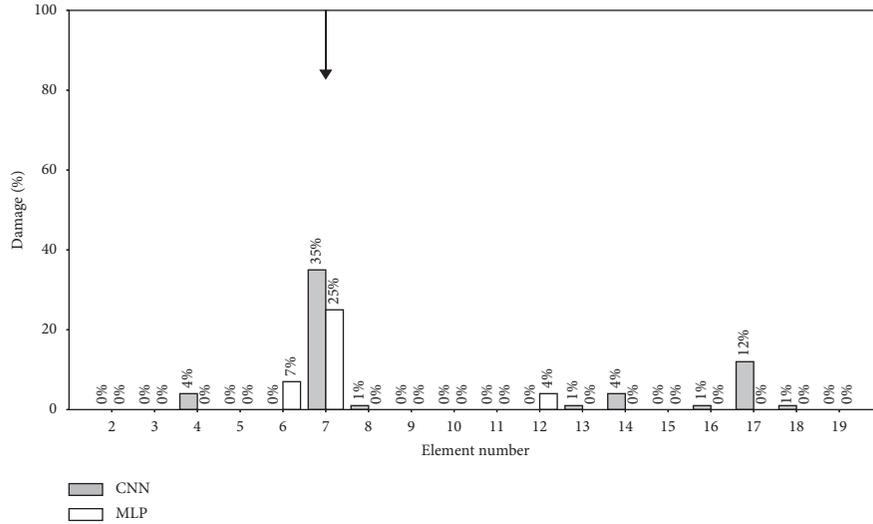
Given that the CNN does not need manual extraction of any feature from the transmissibility functions to train and test the model, it presents a major advantage over the MLP model from [27]. This characteristic from the proposed CNN-based model is of great importance when dealing with complex structures presenting noise contamination, due to the difficulty of extracting the antiresonant frequencies in those cases.

10. Concluding Remarks

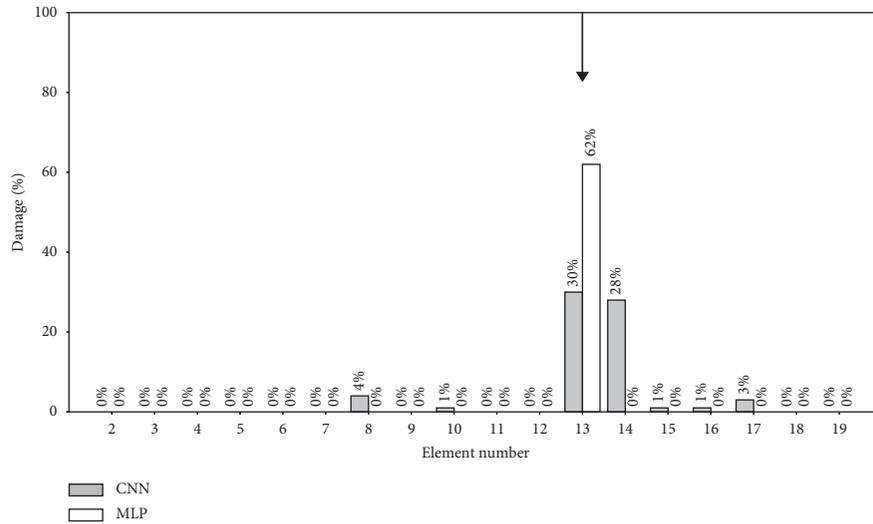
In this paper, we have proposed a new approach for structural damage assessment based on deep convolutional neural networks. The model processes raw transmissibility functions-based images to detect damage in discretized elements of a structure. Damage is represented as reduction of stiffness, which has been shown to be related to transmissibility functions. The CNN-based models' parameters were trained with data generated from a FEM model calibrated with experimental data, simulating structures containing up to three randomly damaged elements. To take into account the source of uncertainty due to randomness, additional noise was injected into the TF signals with the goal of increasing the robustness of the model. One of the novelties of this approach lies on the capability of the CNN to locate and quantify structural damage using only raw vibrational data.

A relevant contribution of the proposed CNN-based approach is to take advantage of the automatic feature extraction enabled by the stacking of convolutional layers, thus eliminating the need to perform manual feature extraction from the transmissibility functions as done in the literature. Therefore, a CNN delivers non-linear representations of the input transmissibility-based images to a higher level of abstraction and complexity isolated from the touch of human engineers directing the training of the models. The proposed CNN model architecture is designed to create a high-dimensional representation without the need for handcrafted feature extraction.

The usefulness of the proposed CNN-based approach in damage localization and quantification was evaluated and validated by means of two case studies: an eight-degree-of-freedom mass-spring system and a structural beam. Based



(a)



(b)

FIGURE 29: Comparison of CNN-based model 2 and MLP from [27]. Experimental cases 1 (a) and 2 (b).

on the resulting performance metrics FAE, DME, and MSE for the test set as shown in Figures 10 and 11 for the spring mass system, the proposed CNN-based approach delivers accurate assessments for damage levels above 10%. This was corroborated by the predictions of the trained CNN-based models for the experimental spring mass system with a 55% reduction of stiffness (damage level) at the fifth element that resulted in a predicted 54% damage level for that same element. In the case of the structural beams, Figures 21–24, as well as Table 4, show that the CNN-based approach provides satisfactory results at localizing the damaged elements and at assigning greater damage level to those elements with deeper saw cuts (i.e., greater damage level). These results can also be attributed to the ability of the CNN architecture to successfully and automatically extract relevant features from the transmissibility functions.

Trained models from both case studies showed great robustness when evaluating new scenarios with 2%, 4%, and

6% of noise contamination. For the eight-degree-of-freedom mass-spring system, results yield an accurate detection and quantification when evaluating the model with one damaged element, showing no variation regardless of the added noise level. A similar result was obtained when testing the robustness of models trained with TF from the structural beam. In this case, two different scenarios were tested with two and three damaged elements, respectively. Once again, the models’ output give a correct prediction for both location and quantification of the damaged elements, showing negligible false positives when noise level increases; however, these are to be expected according to Figures 19(b) and 20.

Moreover, the proposed CNN-based approach and a shallow multilayer perceptron were compared using the structural beam case study as a basis. The results showed that the CNN-based approach delivered better accuracy and fewer false positives and false negatives than the MLP. Hence, the CNN-based approach provided more accurate

and reliable damage assessments than the MLP when trained to detect one, two, and three damaged elements. The proposed CNN-based method was also compared to an antiresonant frequency-based MLP [27]. The proposed CNN-based method has a significantly lower DME in all damage ranges. Once the damage surpasses the 10% threshold, the CNN shows fewer false alarms. Moreover, similar results are obtained from both the proposed CNN model and the MLP model when evaluating experimental cases with one damaged element. However, the proposed CNN skips the requirement of preprocessing the antiresonant frequencies from the transmissibility functions and can be trained to detect more than two damaged elements.

A disadvantage of using transmissibility measurements is their dependency on the force location. Therefore, in a real application, a requisite is to have the structure excited always on the same location, which can be a problem in structures where the input excitations are not controlled, such as ambient or seismic excitations. A solution is to use the value of the transmissibility measurements evaluated only at the natural frequencies, which has been demonstrated to be independent on the force location, or in narrow bands around these frequencies [20]. In addition to the force variability, it should be noted that the proposed models have been evaluated in simple structures under controlled conditions. Therefore, a topic of future research is to evaluate the proposed models with more complex structures considering variable environmental and excitation conditions. Furthermore, the performance of the CNN-based models may drop significantly if evaluated with images generated from structures different from the one used to generate the training datasets. Under these circumstances, the proposed CNN architecture can be used as a starting point to explore new structural damage contexts. Moreover, the models resulting from the proposed CNN architecture discussed in this paper might need to be retrained with the new dataset so as to avoid degradation of the generalization capacity.

Data Availability

The transmissibility data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

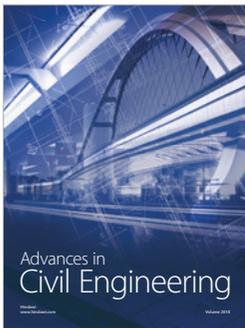
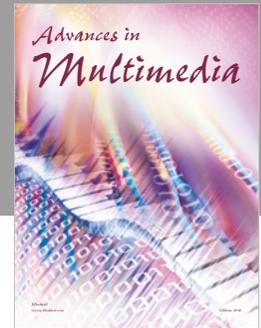
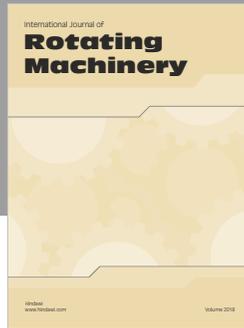
The authors acknowledge the partial financial support of the Chilean National Fund for Scientific and Technological Development (FONDECYT) under grant nos. 1160494 and 1170535.

References

- [1] D. Verstraete, A. Ferrada, E. L. Droguett, V. Meruane, and M. Modarres, "Deep learning enabled fault diagnosis using time-frequency image analysis of rolling element bearings," *Shock and Vibration*, vol. 2017, Article ID 5067651, 17 pages, 2017.
- [2] K. Worden, "Structural fault detection using a novelty measure," *Journal of Sound and Vibration*, vol. 201, no. 1, pp. 85–101, 1997.
- [3] K. Worden, L. Cheung, and J. Rongong, "Damage detection in an aircraft component model," in *Proceedings of the International Modal Analysis Conference IMAC XIX*, pp. 1234–1241, Orlando, FL, USA, February 2001.
- [4] G. Manson, K. Worden, and D. Allman, "Experimental validation of a structural health monitoring methodology: part III. Damage location on an aircraft wing," *Journal of Sound and Vibration*, vol. 259, no. 2, pp. 365–385, 2003.
- [5] G. Manson, K. Worden, and D. Allman, "Experimental validation of a structural health monitoring methodology: part II. Novelty detection on a Gnat aircraft," *Journal of Sound and Vibration*, vol. 259, no. 2, pp. 345–363, 2003.
- [6] H. Zhang, M. J. Schulz, A. Naser, F. Ferguson, and P. F. Pai, "Structural health monitoring using transmittance functions," *Mechanical Systems and Signal Processing*, vol. 13, no. 5, pp. 765–787, 1999.
- [7] T. J. Johnson and D. E. Adams, "Transmissibility as a differential indicator of structural damage," *Journal of Vibration and Acoustics*, vol. 124, no. 4, pp. 634–641, 2002.
- [8] T. J. Johnson, R. L. Brown, D. E. Adams, and M. Schiefer, "Distributed structural health monitoring with a smart sensor array," *Mechanical Systems and Signal Processing*, vol. 18, no. 3, pp. 555–572, 2004.
- [9] N. M. M. Maia, J. M. M. Silva, and A. M. R. Ribeiro, "The transmissibility concept in multi-degree-of-freedom systems," *Mechanical Systems and Signal Processing*, vol. 15, no. 1, pp. 129–137, 2001.
- [10] R. P. C. Sampaio, N. M. M. Maia, A. M. R. Ribeiro, and J. M. M. Silva, "Transmissibility techniques for damage detection," in *Proceedings of SPIE, the International Society for Optical Engineering*, pp. 1524–1527, San Diego, CA, USA, August 2001.
- [11] A. P. V. Urgueira, R. A. B. Almeida, and N. M. M. Maia, "On the use of the transmissibility concept for the evaluation of frequency response functions," *Mechanical Systems and Signal Processing*, vol. 25, no. 3, pp. 940–951, 2011.
- [12] K. Worden and G. Manson, "The application of machine learning to structural health monitoring," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 365, no. 1851, pp. 515–537, 2007.
- [13] K. Worden, G. Manson, and D. Allman, "Experimental validation of a structural health monitoring methodology: part I. novelty detection on a laboratory structure," *Journal of Sound and Vibration*, vol. 259, no. 2, pp. 323–343, 2003.
- [14] Q. Chen, Y. W. Chan, and K. Worden, "Structural fault diagnosis and isolation using neural networks based on response-only data," *Computers & Structures*, vol. 81, no. 22–23, pp. 2165–2172, 2003.
- [15] S. G. Pierce, K. Worden, and G. Manson, "A novel information-gap technique to assess reliability of neural network-based damage detection," *Journal of Sound and Vibration*, vol. 293, no. 1–2, pp. 96–111, 2006.
- [16] K. Worden, G. Manson, and T. Denceux, "An evidence-based approach to damage location on an aircraft structure," *Mechanical Systems and Signal Processing*, vol. 23, no. 6, pp. 1792–1804, 2009.
- [17] Z. Lai and R. Perera, "Transmissibility based damage assessment by intelligent algorithm," in *Proceedings of the 9th*

- International Conference on Structural Dynamics, EURO-DYN 2014*, pp. 2443–2448, Porto, Portugal, July 2014.
- [18] V. Meruane, “Online sequential extreme learning machine for vibration-based damage assessment using transmissibility data,” *Journal of Computing in Civil Engineering*, vol. 30, no. 3, article 04015042, 2016.
- [19] V. Meruane and A. Ortiz-Bernardin, “Structural damage assessment using linear approximation with maximum entropy and transmissibility data,” *Mechanical Systems and Signal Processing*, vol. 54–55, pp. 210–223, 2015.
- [20] S. Chesné and A. Deraemaeker, “Damage localization using transmissibility functions: a critical review,” *Mechanical Systems and Signal Processing*, vol. 38, no. 2, pp. 569–584, 2013.
- [21] Y. J. Yan, L. Cheng, Z. Y. Wu, and L. H. Yam, “Development in vibration-based structural damage detection technique,” *Mechanical Systems and Signal Processing*, vol. 21, no. 5, pp. 2198–2211, 2007.
- [22] M. I. Friswell, “Damage identification using inverse methods,” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 365, no. 1851, pp. 393–410, 2007.
- [23] V. Meruane and V. del Fierro, “An inverse parallel genetic algorithm for the identification of skin/core debonding in honeycomb aluminium panels,” *Structural Control and Health Monitoring*, vol. 22, no. 12, pp. 1426–1439, 2015.
- [24] V. Meruane, “Model updating using antiresonant frequencies identified from transmissibility functions,” *Journal of Sound and Vibration*, vol. 332, no. 4, pp. 807–820, 2013.
- [25] C. González-Pérez and J. Valdés-González, “Identification of structural damage in a vehicular bridge using artificial neural networks,” *Structural Health Monitoring: An International Journal*, vol. 10, no. 1, pp. 33–48, 2011.
- [26] J. L. Zapico, M. P. González, and K. Worden, “Damage assessment using neural networks,” *Mechanical Systems and Signal Processing*, vol. 17, no. 1, pp. 119–125, 2003.
- [27] V. Meruane and J. Mahu, “Real-time structural damage assessment using artificial neural networks and antiresonant frequencies,” *Shock and Vibration*, vol. 2014, Article ID 653279, 14 pages, 2014.
- [28] X. Guo, L. Chen, and C. Shen, “Hierarchical adaptive deep convolution neural network and its application to bearing fault diagnosis,” *Measurement*, vol. 93, pp. 490–502, 2016.
- [29] J. Wang, “A multi-scale convolution neural network for featureless fault diagnosis,” in *Proceedings of the International Symposium on Flexible automation*, pp. 1–3, Cleveland, OH, USA, August 2016.
- [30] D. Lee, V. Siu, R. Cruz, and C. Yetman, “Convolutional neural net and bearing fault analysis,” in *Proceedings of the International Conference on Data Mining DMIN’16*, pp. 194–200, Las Vegas, NV, USA, July 2016.
- [31] O. Abdeljaber, O. Avci, S. Kiranyaz, M. Gabbouj, and D. J. Inman, “Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks,” vol. 388, 2016.
- [32] O. Janssens, V. Slavkovikj, B. Vervisch et al., “Convolutional neural network based fault detection for rotating machinery,” *Journal of Sound and Vibration*, vol. 377, pp. 331–345, 2016.
- [33] Z. Chen, C. Li, and R.-V. Sanchez, “Gearbox fault identification and classification with convolutional neural networks,” *Shock and Vibration*, vol. 2015, Article ID 390134, 10 pages, 2015.
- [34] C. Zhang, P. Lim, A. K. Qin, and K. C. Tan, “Multiobjective deep belief networks ensemble for remaining useful life estimation in prognostics,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2306–2318, 2017.
- [35] L. Liao, W. Jin, and R. Pavel, “Enhanced restricted Boltzmann machine with prognosability regularization for prognostics and health assessment,” *IEEE Transactions on Industrial Electronics*, vol. 63, no. 11, pp. 7076–7083, 2016.
- [36] G. Sateesh Babu, P. Zhao, and X.-L. Li, “Deep convolutional neural network based regression approach for estimation of remaining useful life,” *Database Systems for Advanced Applications*, vol. 9642, pp. 214–228, 2016.
- [37] L. Guo, H. Gao, H. Huang, X. He, and S. Li, “Multifeatures fusion and nonlinear dimension reduction for intelligent bearing condition monitoring,” *Shock and Vibration*, vol. 2016, pp. 1–10, 2016.
- [38] F. Zhou, Y. Gao, and C. Wen, “A novel multimode fault classification method based on deep learning,” vol. 2017, Article ID 3583610, 14 pages, 2017.
- [39] H. Liu, L. Li, and J. Ma, “Rolling bearing fault diagnosis based on STFT-deep learning and sound signals,” *Shock and Vibration*, vol. 2016, pp. 1–12, 2016.
- [40] F. Jia, Y. Lei, J. Lin, X. Zhou, and N. Lu, “Deep neural networks: a promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data,” *Mechanical Systems and Signal Processing*, vol. 72–73, pp. 303–315, 2016.
- [41] M. Gan, C. Wang, and C. Zhu, “Construction of hierarchical diagnosis network based on deep learning and its application in the fault pattern recognition of rolling element bearings,” *Mechanical Systems and Signal Processing*, vol. 72–73, pp. 92–104, 2016.
- [42] Y. Yu, C. Wang, X. Gu, and J. Li, “A novel deep learning-based method for damage identification of smart building structures,” *Structural Health Monitoring*, vol. 18, no. 1, pp. 143–163, 2019.
- [43] H. Khodabandehlou, G. Pekcan, and M. S. Fadali, “Vibration-based structural condition assessment using convolution neural networks,” *Structural Control and Health Monitoring*, vol. 26, no. 2, p. e2308, 2018.
- [44] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 1–9, Lake Tahoe, NV, USA, December 2012.
- [45] Y. Kim, “Convolutional neural networks for sentence classification,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, October 2014.
- [46] C. Farabet, C. Couprie, L. Najman, and Y. Lecun, “Learning hierarchical features for scene labeling,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1915–1929, 2013.
- [47] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.
- [48] M. I. Friswell and J. E. T. Penny, “Crack modeling for structural health monitoring,” *Structural Health Monitoring: An International Journal*, vol. 1, no. 2, pp. 139–148, 2002.
- [49] C. Zang and M. Imregun, “Structural damage detection using artificial neural networks and measured FRF data reduced via principal component projection,” *Journal of Sound and Vibration*, vol. 242, no. 5, pp. 813–827, 2001.
- [50] Y. N. Dauphin, H. De Vries, J. Chung, and Y. Bengio, “RMSProp and equilibrated adaptive learning rates for non-

- convex optimization,” in *Proceedings of the Conference on Neural Information Processing Systems*, pp. 1–10, Montreal, Canada, December 2015.
- [51] Y. Nesterov, “Gradient methods for minimizing composite functions,” *Mathematical Programming*, vol. 140, no. 1, pp. 125–161, 2013.
- [52] M. D. Zeiler, “ADADELTA: an adaptive learning rate method,” 2012, <https://arxiv.org/abs/1212.5701>.
- [53] D. Kingma and J. Ba, “Adam: a method for stochastic optimization,” 2014, <https://arxiv.org/abs/1412.6980>.
- [54] C.-B. Yun, J.-H. Yi, and E. Y. Bahng, “Joint damage assessment of framed structures using a neural networks technique,” *Engineering Structures*, vol. 23, no. 5, 2001.
- [55] T. A. Duffey, S. W. Doebling, C. R. Farrar, W. E. Baker, and W. H. Rhee, “Vibration-based damage identification in structures exhibiting axial and torsional response,” *Journal of Vibration and Acoustics*, vol. 123, no. 1, p. 84, 2001.



Hindawi

Submit your manuscripts at
www.hindawi.com

