Damage detection in lightweight bridges with traveling masses using machine learning

Georgios Dadoulis[1], George D. Manolis[1], Konstantinos Katakalos[1], Kosmas Dragos[2], Kay Smarsly[2]

[1]Laboratory for Strength of Materials and Structures, Department of Civil Engineering, Aristotle University, 54124 Thessaloniki, Greece

[2]Institute of Digital and Autonomous Construction, Hamburg University of Technology, Blohmstrasse 15, 21079 Hamburg, Germany

Abstract

Damage detection via vibration testing typically relies on damage-sensitive features, which serve as "damage indicators", and decisions upon the existence of damage are based on comparing the damage indicators retrieved from two distinct structural states. However, the relatively low sensitivity of damage indicators to the onset of structural damage remains an open question, despite the considerable research efforts in vibration testing over the years. Low-sensitivity problems may be particularly exacerbated by the complex dynamic behavior of lightweight structures, such as lightweight bridges subjected to vehicular traffic. In particular, due to material (and, by extension, mass) reduction in lightweight bridges, vehicles essentially act as "traveling masses", which are comparable to the structural mass and result in a coupled complex dynamic motion problem that may obscure typical damage indicators used in vibration testing. This paper presents a damage detection approach for lightweight bridges with traveling masses, leveraging the powerful feature-extraction capabilities of machine learning (ML). In particular, a convolutional neural network (CNN) is trained to classify acceleration response data, collected from vibration testing, into damage scenarios. The training data for the CNN are created via simulations of damage scenarios, using calibrated analytical models. The damage detection approach is validated in laboratory tests on a continuous beam, showcasing the capability of the CNN to classify damage scenarios of the

beam. The outcome of this paper aims to serve as a starting point towards employing ML for damage detection in the context of vibration testing as well as structural health monitoring.

Keywords

Damage detection, vibration testing, machine learning, analytical modeling, structural dynamics, artificial intelligence

1. Introduction

Vibration testing is frequently employed for system identification and damage detection purposes as part of non-destructive evaluation of structures. For example, vibration testing is usually applied in the form of experimental modal analysis (typically, input-output modal identification) as well as operational modal analysis (typically, output-only modal identification), based on collecting acceleration response data under controlled excitation and under ambient excitation, respectively [1]. Drawing from the field of structural health monitoring (SHM), in which vibration testing holds a central role, common ground among most vibration-based damage detection (VBDD) methods is the extraction of damage-sensitive features, which essentially serve as "damage indicators". Decisions on the existence of damage are based on comparisons between damage indicators between two structural states, one designated as "undamaged" and one as "current" [2].

In structural engineering, the application of VBDD dates back to the 1970s, with research focusing first on damage-proofing off-shore oil/gas platforms, with subsequent extensions to critical infrastructure, including bridges [3]. Despite the promising results of early laboratory-validated approaches using modal parameters (eigenfrequencies, mode shapes) as damage indicators, transferring these approaches to full-scale real-life SHM has proven challenging, due to the low sensitivity of dominant (typically, low-frequency) modal parameters to the onset of structural damage [4]. Furthermore, in computing modal parameters, there is an implicit requirement that the excitation conditions (e.g. acting loads) and structural parameters (e.g. structural mass) are relatively stationary; this requirement rarely holds true, due to operational and environmental effects. Finally, tracking eigenfrequencies and mode shapes may be particularly difficult in

structures exhibiting complex dynamic behavior, in which modal parameters can hardly be intuitively inferred via engineering judgment. Complex dynamic behavior is particularly pronounced in lightweight bridges subjected to vehicular traffic. In particular, the introduction of high-performance materials in bridge construction encourages material reduction to satisfy budgetary, operational or even aesthetic design requirements, resulting in vehicular masses being comparable to structural masses of lightweight bridges. Vehicles traversing decks of lightweight bridges essentially behave as "traveling masses", resulting in a complex dynamic motion problem with strong coupling between the traveling mass and the structural mass, which is hardly amenable to conventional VBDD using modal parameters [5]. At a first glance, the aforementioned problem seems to affect predominantly lightweight bridges with complex geometries; however, as will be shown in this paper, the problem may also concern simple beam-like bridges.

With respect to lightweight bridges with traveling masses, VBDD essentially entails solving a "moving-load/mass" problem. Lots of research has been done on understanding and modeling the moving-load/mass problem, which dates back to the days of steam locomotives from two centuries ago, when the first mathematical formulations have been developed as a means of analyzing railway traffic over iron bridges [6]. Starting from the 1960's onwards with the seminal work of Fryba [7], the basic problem of a single load moving with constant speed over a single beam span has extensively been investigated, including extensions to multiple spans, multiple moving loads, and loads that accelerate/decelerate. Furthermore, the moving point load problem – representing a moving vehicle – has evolved into a problem analyzing the behavior of a structural sub-system with its own mass, stiffness and damping [8]. Numerical methods for analyzing the vibrations of structures under moving inertial loads with applications to railway problems, including the development of space-time finite elements for dynamic analysis of both Bernoulli-Euler and Timoshenko beams, have been proposed by Bajer and Dyniewicz (2012) [9]. The applicability of the moving point load approaches to bridge decks has also drawn research attention [10-13]. A detailed summary on the hierarchy of the mathematical models used for train-track-bridge interaction, covering probabilistic dynamic analysis of single bridges as well as bridge networks, can be found in Johansson (2013) [14]. Particular research focus has been placed on data analysis methods for the moving load/mass problem, such as the short-time Fourier transform (STFT), otherwise known as the Gabor transform [15]. Other techniques for analyzing acceleration

response data from bridges under moving loads are the wavelet transform [16] and the Hilbert transform [17], the latter being used for analyzing acceleration response data over large time intervals in single span railway bridges for tracing changes in the eigenfrequencies and damping ratios. The possibility of correlating damage indicators of bridges to the energy band variation of the acceleration time history of a moving vehicle using STFT has been presented in He et al. (2018) [18]. Regarding damage detection in pre-stressed concrete bridges, the relaxation of pre-stress in a continuous beam has been examined with the aid of data recovered from vehicles travelling over such bridges [19]. A thorough review of studies on moving loads in structures for the purpose of damage detection can be found in Dadoulis and Manolis [20].

Approaches on VBDD in bridges from a SHM perspective have been proposed since the 1980s. For instance, Doebling et al. [3] have provided an overview of VBDD for bridges in the last two decades of the 20th century. Following the advances in SHM, case studies of vibration-based SHM systems installed in bridges also have been reported [21]. However, vibration-based SHM largely has been restricted to facilitating system identification and enhancing the level of knowledge on existing structural conditions rather than actually detecting damage. As stated in [21], by 2011, no cases of damage detection using vibration-based SHM in bridges had been reported. The apparent failure of vibration-based SHM to resonate with practical problems in bridge maintenance has been attributed to the stark differences between simple laboratory tests, used to validate vibration-SHM approaches, and real-life structures operating in dynamic surroundings [22]. Recent approaches on vibration-based SHM (including VBDD) on bridges have been leveraging machine leaning (ML), building upon the seminal work of Farrar and Worden [2], who have cast the SHM problem within a statistical pattern recognition paradigm and, later, within a machine learning framework [23]. A recent review of the transition from traditional methods to ML-based methods for VBDD can be found in Avci et al. [24].

Adding to the argument of Cawley [22] that laboratory tests on simple beams may fail to replicate real-world phenomena, it is argued in this paper that the comparability of traveling masses to structural masses poses further challenges even during laboratory tests with beams. In particular, the strong coupling between a traveling mass and the structural mass of a beam obscures the frequency content of acceleration response data, rendering traditional VBDD methods, such as

frequency shifts and tracking of mode shapes, impractical. To circumvent this problem, the damage detection approach presented herein, builds upon feature extraction for identifying structural damage during laboratory-level vibration-testing of lightweight bridges, represented as simple beams traversed by traveling masses. The damage detection approach, which is essentially an extension of the concept presented by the authors [25], is based on machine learning for extracting features from acceleration response data and for classifying the features into predefined damage scenarios. Specifically, a convolutional neural network (CNN) is created and trained with acceleration time histories, i.e. acceleration responses obtained from simulations of a steel beam, representing a lightweight bridge deck, under different damage scenarios. Thereupon, the CNN is fed with acceleration response data recorded using a laboratory setup, including a steel beam, instrumented with accelerometers, and traveling masses, comparable with the mass of the beam. In [25], the classification capabilities of the CNN have been proven for damage scenarios involving "softening" of the supports. In this paper, the capabilities of the CNN in distinguishing between softening of supports and cracking close to the midspan of the deck are corroborated. The remainder of the paper starts with a description of the simulations and pre-processing of acceleration time histories, conducted to produce training data for the CNN. Next, the training process is described, followed by the implementation and validation tests. The paper concludes with a summary of the main findings and an outlook on future research.

2. Modeling and simulation of the moving-mass problem

Machine-learning classification using convolutional neural networks falls into the category of supervised learning, which relies on procuring "labeled" training data, i.e. data that correspond to the scenarios ("classes") envisaged for the phenomenon being investigated. With regard to VBDD, obtaining acceleration response data from scenarios involving structural damage is practically impossible in real-world structures. Solutions to this problem involve modeling and simulation either via laboratory down-scaled physical models of structures or via simulations using numerical or analytical modeling. Limitations related to the former solution include scaling problems and accurate representation of damage scenarios, while the latter solution introduces simplifications due to the linearization of the generally non-linear behavior of structures, particularly in the event of structural damage. In this paper, the labeled training data consists of acceleration time histories,

obtained via analytical simulation of the moving-mass problem. The reasoning behind selecting analytical modeling is that the partial differential equations of the continuous structural system are capable of capturing the moving-mass problem in its totality as opposed to piecewise solutions offered by finite element modeling. Specifically, analytical models are capable of integrating the effects of the speed of the traveling mass, which influences the structural response. With respect to linearization-induced limitations, as will be shown in this paper, the CNN exhibits robust performance and the ability to discern subtle damage-induced patterns even in the linear components of the structural response. In what follows, the analytical modeling process for obtaining the labeled training data is illuminated.

The analytical modeling of a lightweight bridge traversed by a traveling mass can be modeled as a continuous Euler-Bernoulli beam, shown in Figure 1. The structural response of the beam is described through the following partial differential equation of motion:

$$\rho A\ddot{w}(x,t)+c\dot{w}(x,t)+EIw''''(x,t)=f_c\delta(x-vt) \tag{1}$$

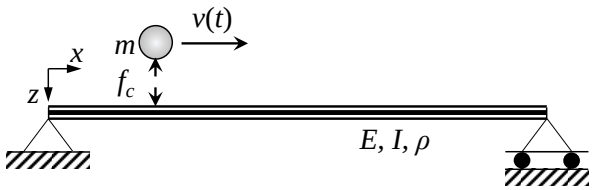

Figure 1. Euler-Bernoulli beam with traveling mass.

As can be seen in Equation (1), the governing parameters of motion are the material density (ρ), the modulus of elasticity (E), the moment of inertia (I), the section area of the beam (A), the viscous damping (c), and the speed of the traveling mass (v). The response of the beam is expressed via the vertical displacement (w), while the acting force is defined as the product of the contact force (f_c) and the Dirac delta function (δ) of the coordinate in the longitudinal direction of the beam (x) and the speed of the traveling mass over time (t). The derivatives are defined as follows:

$$\dot{w}(x,t)=\frac{\partial w}{\partial t}, \qquad \ddot{w}(x,t)=\frac{\partial^2 w}{\partial t^2}, \qquad w''''(x,t)=\frac{\partial^4 w}{\partial x^4} \tag{2}$$

For estimating the contact force, the equilibrium relationship at the contract point between the traveling mass and the beam is considered as follows:

$$\begin{aligned} w_T(x,t) &= w(x,t) + r(x) \\ m\ddot{w}_T(x,t) &= mg - f_c \end{aligned} \tag{3}$$

The first of the two expressions in Equation (3) corresponds to the total displacements of the traveling mass $w_T(x, t)$ comprising the vertical displacement of the bridge and a component representing the surface roughness $r(x)$. The second expression is the dynamic equilibrium, which comprises the gravitational force of the traveling mass (g is the gravitational acceleration), the inertial force of the beam, and the contact force. In Equation (3), the longitudinal coordinate is a function of time, depending on the speed of the traveling mass ($x(t) = vt$). The differentiation in the second expression results in:

$$\begin{aligned} &\dot{w}_T(x(t),t) = \frac{\partial w}{\partial t} + \frac{\partial w}{\partial x}\frac{\partial x}{\partial t} + \frac{\partial r}{\partial x}\frac{\partial x}{\partial t} = \frac{\partial w}{\partial t} + \frac{\partial w}{\partial x}v + \frac{\partial r}{\partial x}v \Leftrightarrow \\ &\ddot{w}_T(x(t),t) = \frac{\partial^2 w}{\partial t^2} + \frac{\partial^2 w}{\partial x^2}v^2 + \frac{\partial^2 w}{\partial x \partial t}v + \frac{\partial^2 w}{\partial t \partial x}v + \frac{\partial^2 r}{\partial x^2}v^2 = \frac{\partial^2 w}{\partial t^2} + \frac{\partial^2 w}{\partial x^2}v^2 + 2\frac{\partial^2 w}{\partial x \partial t}v + \frac{\partial^2 r}{\partial x^2}v^2 \end{aligned} \tag{4}$$

$$f_c = m\left(g - \frac{\partial^2 w}{\partial x^2}v^2 - 2\frac{\partial^2 w}{\partial x \partial t}v - \frac{\partial^2 w}{\partial t^2} - \frac{\partial^2 r}{\partial x^2}v^2\right) \tag{5}$$

By substituting the expressions from Equations (4) and (5) into Equation (1), the resulting equation of motion is

$$\rho A\ddot{w}(x,t) + c\dot{w}(x,t) + EIw''''(x,t) = m\left(g - \frac{\partial^2 w}{\partial x^2}v^2 - 2\frac{\partial^2 w}{\partial x \partial t}v - \frac{\partial^2 w}{\partial t^2} - \frac{\partial^2 r}{\partial x^2}v^2\right)\delta(x - vt) \tag{6}$$

Modal superposition is employed for solving Equation (6) as follows:

$$w(x,t) = \sum_{i=1}^{\infty} \varphi_i(x) q_i(t) \tag{7}$$

Specifically, the vertical displacement, which is a function of a spatial (x) and a temporal variable (t) is decomposed into a spatial function (mode shape) and a "generalized" temporal function. The complexity of the solution depends on the number of mode shapes considered, with a small number being sufficient for describing the structural response. For the Euler-Bernoulli beam, the ith mode shape $\varphi_i(x)$ is a function of the form:

$$\varphi_i(x) = C_{1,i}\sin(k_i x) + C_{2,i}\cos(k_i x) + C_{3,i}\sinh(k_i x) + C_{4,i}\cosh(k_i x), \quad 0 < x < L, \quad i \in \mathbb{N}^* \tag{8}$$

where L is the length of the beam, and k_i and $C_{1,i}$, $C_{2,i}$, $C_{3,i}$, $C_{4,i}$ are the wave number and constants, respectively, both depending on the boundary conditions of the beam. Mode shapes are characterized by the orthogonality condition[26]:

$$\rho A\int_0^L \varphi_i(x)\varphi_j(x)dx = \delta_{ij}, \tag{9}$$

where δ_{ij} is the Kronecker delta. To utilize the orthogonality condition, the vertical displacement terms (and respective derivatives) are decomposed using Equation (7), pre-multiplied and integrated according to Equation (9). As a result,

the terms of Equation (6) are converted as follows (notations for time t and coordinate x are dropped for simplicity):

$$\int_0^L \varphi_j \rho A\ddot{w}(x,t) = \left(\rho A\int_0^L \varphi_j\varphi_j dx\right)\ddot{q}_j = \ddot{q}_j \tag{10}$$

$$\int_0^L \varphi_j c_j \dot{w}(x,t) = \left(c_j\int_0^L \varphi_j\varphi_j dx\right)\dot{q}_j = 2\xi_j\omega_j\dot{q}_j, \qquad c_j = 2\rho A\omega_j\xi_j \tag{11}$$

$$\int_0^L \varphi_j EIw''''(x,t) = \left(EI\int_0^L \varphi_j\varphi_j''''dx\right)q_j = \omega_j^2 q_j \qquad \omega_j^2 = \frac{k_j^4 EI}{\rho A} \tag{12}$$

$$\int_0^L \varphi_j m\frac{\partial^2 w}{\partial x^2}v^2\delta(x-vt) = mv^2\left(\int_0^L \varphi_i''\varphi_j\delta(x-vt)dx\right)q_i = mv^2\psi_i''\psi_j q_i \tag{13}$$

$$\int_0^L \varphi_j 2m\frac{\partial^2 w}{\partial x\partial t}v\delta(x-vt) = 2mv\left(\int_0^L \varphi_i'\varphi_j\delta(x-vt)dx\right)\dot{q}_i = 2mv\psi_i'\psi_j\dot{q}_i \tag{14}$$

$$\int_0^L \varphi_j m\frac{\partial^2 w}{\partial t^2}\delta(x-vt) = m\left(\int_0^L \varphi_j\varphi_i\delta(x-vt)dx\right)\ddot{q}_i = m\psi_i\psi_j\ddot{q}_i \tag{15}$$

$$\int_0^L \varphi_j m \frac{\partial^2 r}{\partial x^2} v^2 \delta(x - vt) = mv^2 \left(\int_0^L \varphi_j r'' \delta(x - vt)\, dx \right) = mv^2 \psi_j \zeta'' \tag{16}$$

In the above, $\psi = \varphi(vt)$ and $\zeta = r(vt)$. By substituting the terms of Equations (10) - (16) in Equation (6), the formulation can be recast in matrix form:

$$\mathbf{M}(t)\ddot{\mathbf{q}}(t) + \mathbf{C}(t)\dot{\mathbf{q}}(t) + \mathbf{K}(t)\mathbf{q}(t) = \mathbf{F}(t) \tag{17}$$

with $\mathbf{q}(t)$ representing the vector of generalized temporal functions of the mode shapes. The matrices appearing in Equation (17) are given as follows:

$$\mathbf{M}(t) = \mathbf{I} + m\psi_i\psi_j \tag{18}$$

$$\mathbf{C}(t) = 2\left[diag\left(\xi_j \omega_j\right) + mv\psi_i'\psi_j \right] \tag{19}$$

$$\mathbf{K}(t) = diag\left(\omega_j^2\right) + mv^2\psi_i''\psi_j \tag{20}$$

$$\mathbf{F}(t) = m\left[g - v^2\zeta'' \right]\psi_j \tag{21}$$

Equation (17) is re-formulated in the state-space, which consists of a first-order system of differential equations

$$\dot{\mathbf{y}}(t) = \left[-\mathbf{A}^{-1}(t)\mathbf{B}(t) \right]\mathbf{y}(t) + \mathbf{A}^{-1}(t)\mathbf{h}(t) \tag{22}$$

where the matrices **A**, **B** and the vectors **y**, **h** are computed as follows:

$$\mathbf{A}(t) = \begin{bmatrix} \mathbf{0} & \mathbf{M}(t) \\ \mathbf{M}(t) & \mathbf{C}(t) \end{bmatrix},\ \mathbf{B}(t) = \begin{bmatrix} -\mathbf{M}(t) & \mathbf{0} \\ \mathbf{0} & \mathbf{K}(t) \end{bmatrix},\ \dot{\mathbf{y}}(t) = \begin{bmatrix} \dot{\mathbf{q}}(t) \\ \mathbf{q}(t) \end{bmatrix},\ \mathbf{h}(t) = \begin{bmatrix} \mathbf{0} \\ \mathbf{F}(t) \end{bmatrix} \tag{23}$$

Solving Equation (22) for N discrete time intervals $\kappa\Delta t$ ($\kappa = 1 \ldots N$) assumes that the matrices **A**, **B** and the vector **h** remain constant over each time step Δt and are updated at the end of it. For each time interval κ by calculating the eigenvalues $\mathbf{\Lambda}_\kappa$ and eigenvectors $\mathbf{\Psi}_\kappa$ of matrix $(-\mathbf{A}_\kappa)^{-1}\mathbf{B}_\kappa$, the

fundamental matrix $\boldsymbol{\Theta}_\kappa(t)$, $t \in [0, \Delta t]$ is determined. Using the fundamental matrix, the values of vector $\mathbf{y}$ at time instance κ are computed as follows:

$$\mathbf{y}_\kappa(\Delta t) = \boldsymbol{\Theta}_\kappa(\Delta t)\left(\boldsymbol{\Theta}_\kappa(0)\right)^{-1}\mathbf{y}_\kappa(0) + \boldsymbol{\Theta}_\kappa(\Delta t)\int_0^{\Delta t}\left(\boldsymbol{\Theta}_k(\tau)\right)^{-1}\mathbf{A}_\kappa^{-1}\mathbf{h}_\kappa d\tau \tag{24}$$

Upon computation of vector $\mathbf{y}$ for N time instances, the corresponding vector $\mathbf{q}$ is easily obtained, and its second derivative is combined with spatial functions of mode shapes to yield the vertical accelerations. Thereupon, the accelerations are used for preparing the labeled training data for the CNN, as shown in the next section.

3. Damage identification using a convolutional neural network

In this section, the identification of structural damage scenarios using a convolutional neural network is presented. First, the preparation of labeled training data for the CNN based on the acceleration time histories, obtained via the analytical modeling and simulation process previously shown, is described, followed by the CNN training process.

3.1. Preparation of labeled training data

The acceleration time histories obtained from modeling and simulation are typically smooth, i.e. impervious to external interferences, such as ambient noise and measurement errors. As such, mapping based on raw acceleration time histories, fed as "sequences" to the CNN, may increase the risk of poor performance when transferring the CNN to real vibration testing, in which real-world acceleration response data may be affected by noise or other external factors. Alternatively, for the training process proposed herein, the acceleration time histories are pre-processed in an attempt to expose features in the data that facilitate the training process. By ensuring that the decisions of the CNN are driven by the features, the CNN is expected to perform better in real-world vibration testing, making its decisions based on similar features exposed in real-world acceleration response data while neglecting noise and spurious factors. The pre-processing method

used in this study is the Gabor transform, shown below, which is suitable for non-stationary phenomena, such as bridges traversed by traveling masses:

$$
\begin{aligned}
&G_o\left(u\right) = S_p W_u W_u^* \\
&W_u = \sum_{\gamma=0}^{\frac{N-N_o}{N_o-H}} \sum_{h=1}^{N_o} \ddot{w}\left(x, h+\gamma\left(N_o-H\right)\right) z\left(h\right) e^{-i2\pi uh/N_o} \qquad u = \frac{N_o \cdot f}{f_s} \\
&S_p = \begin{cases} \dfrac{2A_w^2}{N_o^2 B_e}, & u \neq 0 \\ \dfrac{A_w^2}{N_o^2 B_e}, & u = 0 \end{cases}, \qquad A_w = \frac{N_o}{\sum_{h=0}^{N_o-1} z\left(h\right)}, \quad B_e = \frac{f_s \sum_{h=0}^{N_o-1} \left[z\left(h\right)\right]^2}{\left[\sum_{h=0}^{N_o-1} z\left(h\right)\right]^2}
\end{aligned}
\tag{25}
$$

In Equation (25), G_o is the Gabor coefficient at the u-th frequency "bin", corresponding to frequency f, of a N_o-sized window of acceleration time history $\ddot{w}(x, h + \gamma(N_o - H))$, with H points overlap, collected with sampling frequency f_s. The Gabor window is denoted by $z(h)$, and S_p is a scaling factor. The notation $\bullet^*$ indicates complex conjugate. The Gabor coefficients are typically illustrated in color map images, which essentially converts damage identification to image classification, which is well-defined in the field of machine learning. An exemplary illustration of an acceleration time history (left) with its Gabor transform color map (right) is shown in Figure 2.

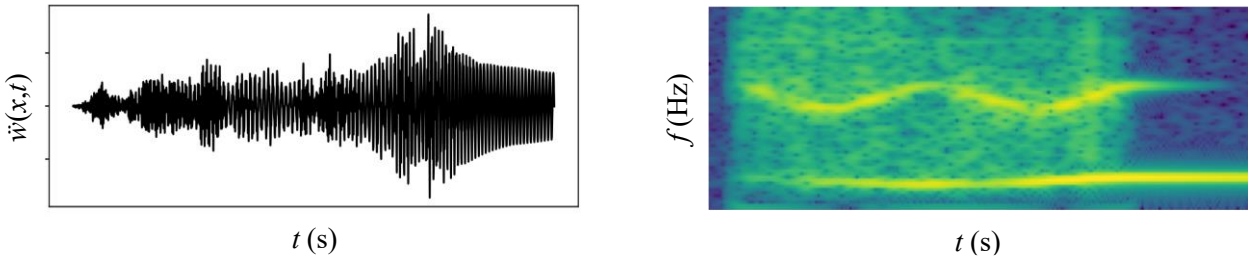

Figure 2. Acceleration time history and Gabor transform color map.

3.2. Design and training of the convolutional neural network

The layout of a typical CNN for image classification is shown in Figure 3. The *input layer* accepts a tricolor red-green-blue (RGB) image with aspect ratio $N_I \times N_I$, resulting in an input vector of size $N_I \times N_I \times 3$. The input layer is followed by a *convolution layer*, which consists of sliding "kernels", essentially square matrices of size $n_I \times n_I \times 3$, with $n_I << N_I$ in the general case. Each kernel scans the entire image with a predefined step, termed "stride" (s_c), and performs a convolution operation (dot product) in each area of the image of size $n_I \times n_I$, between the RGB values of the area and the values of the kernel matrix. Each dot product is stored as a "feature" into an activation map. The number of activation maps depends on the number of kernels used. The dimension of each activation map is equal to $W = (N_I - n_I)/s_c + 1$. The features are passed through *activation functions*, which introduce non-linearity to the CNN, thus enabling mapping complex relationships between inputs and outputs. Traditional activation functions, typically used in feedforward neural networks, such as multi-layer perceptrons, include the hyperbolic tangent and the sigmoid function. In convolutional neural networks, a popular activation function is the "rectifier linear unit" (ReLU), which is a linear positive function a_h that rejects negative values, i.e., $a_h(x_i) = \max(0, x_i)$, where x_i is the input to the activation function. The outputs of the activation function ("activations") are forwarded to a *pooling layer*, which performs sliding-window operations. Specifically, the activations corresponding to each activation map are "pooled" using a window of size $n_p \times n_p$, either by keeping the maximum value of each window ("maximum pooling") or by averaging the values of each window ("average pooling"). The window in the pooling layer moves across each activation map in predefined strides (s_p). A typical CNN includes a succession of convolution layers and pooling layers, depending on the target "depth" of the CNN.

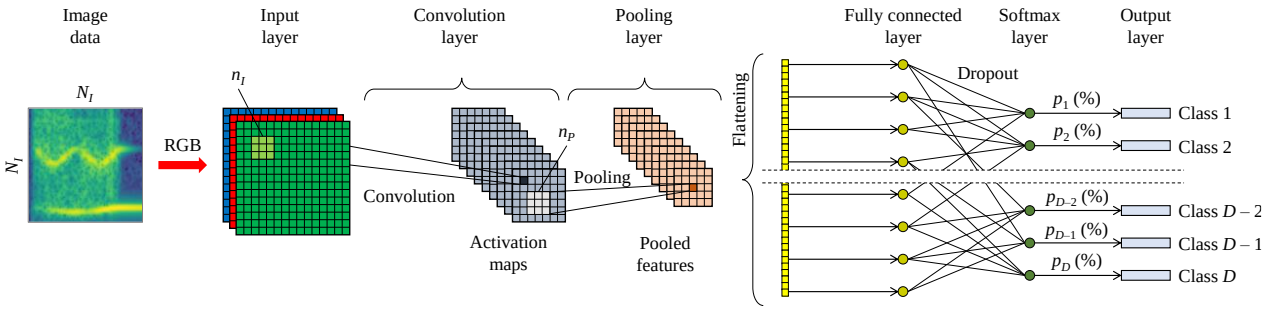

Figure 3. Layout of a typical CNN with one convolution layer and one pooling layer.

The output dimension of the last (o-th) pooling layer, following the o-th convolution layer, is of size $W_{p,o} = (W_o - n_p)/s_p + 1$ and is "flattened", i.e. converted into a single-column layer, whose activations are passed to a *fully connected layer*. The last part of the CNN involves combining the activations of the fully connected layer, in a manner similar to feedforward neural networks, before passing the activations to the *output layer*. Since the output layer contains class names and the fully connected layer contains numerical values, the combinations of the activations of the fully connected layers are first converted to probability values for each class. The probability values are computed using a *Softmax layer*, which is placed between the fully connected layer and the output layer. Finally, it is common to add a *dropout layer* between the fully connected layer and the *Softmax layer*, which randomly rejects activations during training – based on a predefined percentage of acceptance – so as to avoid overfitting. Dropout layers may also be added between pooling layers and convolution layers.

The labeled training data set for the CNN consists of images produced via applying the Gabor transform to acceleration time histories, computed for several damage scenarios. Each image in the labeled training data set is accompanied by the class name of the corresponding damage scenario. For training, one subset is extracted, termed training subset, consisting of 80% of the images of the labeled training data set. The remaining 20% of the images are used for forming a testing subset. Training is performed by first initializing the kernel matrices, and then by sequentially propagating the images of the training subset through the CNN. Each forward propagation is a training "iteration", during which a number of images are forwarded, referred to as "mini-batch". In each iteration, the kernel matrices are updated based on the "learning rate" in an attempt to minimize the classification performance error (training error). The forward propagation of every image in the training subset is called "epoch". Upon completing a predefined number of iterations, a part of the training subset, termed validation subset (20% of the images in the training subset) is propagated through the CNN to compute the so-called "validation error"; the purpose of propagating the validation subset is to ensure that the training error and the validation error follow similar trends and that the CNN does not overfit to the training subset. Once a predefined threshold of classification accuracy has been reached (i.e. the training error and validation error have fallen below a predefined threshold), training is terminated. Thereupon, the

testing subset is propagated through the CNN to evaluate the final performance of the CNN. The proposed approach, including the analytical modeling and simulation and the CNN training, is validated in laboratory tests, described in the next section.

4. Laboratory validation tests

Validation tests are devised, using an experimental setup of a beam with a traveling mass, for showcasing the capability of the damage detection approach in classifying acceleration response data into damage scenarios. First, the experimental setup is presented, followed by a description of the modeling and simulation of the beam and of the generation of labeled training data for the CNN. Next, the CNN definition and training are shown, and the laboratory tests are presented. Finally, the tests results are discussed.

4.1. Experimental setup

The experimental setup, shown in Figure 4, consists of a simply supported steel beam of standardized cross section of type HEB100. The external dimensions of the beam cross section are 100 mm × 100 mm (width × height), the flange thickness is 10 mm, and the web thickness is 6 mm. The length of the beam, essentially defined by the effective length of the neutral axis, is L = 5830 mm. The material properties of the beam are computed via tests, which yield an elastic modulus of E = 198.5 GPa and a density of $\rho = 7.65 \cdot 10^3$ kg/m^3. The beam rests on a steel substructure, comprising an arrangement of rods and struts. The struts are adjustable to facilitate leveling the beam, i.e. for compensating for imperfections of the floor. It should be noted that only the struts at the end points of the steel beam offer vertical support to the bridge; the middle strut is only added to ensure the stability of the substructure and is not connected to the beam. The traveling mass is represented by a wheel with a steel shaft, to which variable weights are added. The wheel is connected via a twisted-strand steel cable to pulleys, one of which is attached to a motor that induces motion to the pulley, and, in turn, to the wheel. The speed of the wheel is regulated via a switch, attached to a panel that is connected to the motor, allowing speeds between 4 cm/s to 50 cm/s.

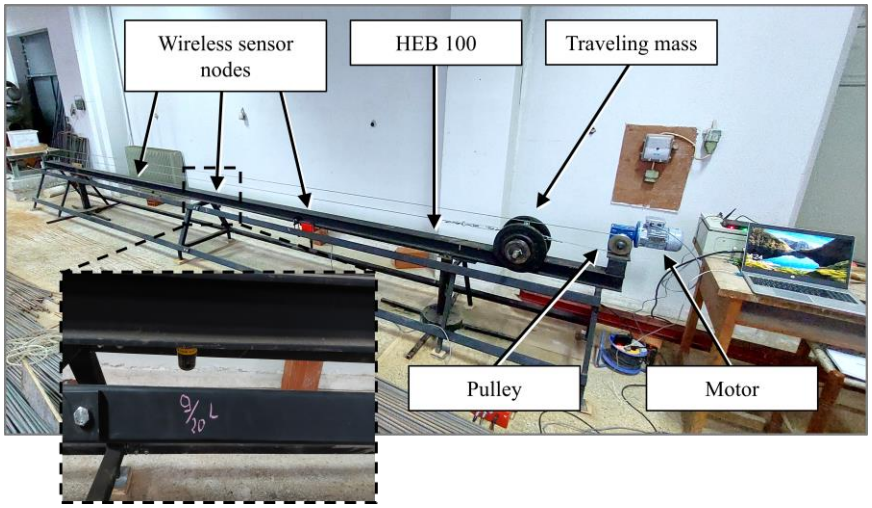

Figure 4. Experimental setup for the validation tests.

The beam is equipped with a wireless SHM system, consisting of three wireless sensor nodes, two of which (S1 and S3) are placed at the quarter-length points, i.e. at a distance of $L/4$ and $3L/4$ from one end of the beam, respectively, and one (S2) placed close to the midspan at $9L/20$ from one end. The reason for avoiding placing the sensor node exactly at the midspan is to avoid the nodal (i.e. zero-crossing) point of the second mode shape, which is expected to have a sinusoidal shape. The wireless sensor nodes are of type Lord Microstrain G-Link-200 [27]. Each sensor node includes a digital-output triaxial accelerometer with 20-bit resolution, capable of recording accelerations at a configurable range from ±2g to ±8g with 25μg/Hz^{-1} noise floor and sampling frequencies ranging from 1 sample per hour to 4,096 Hz. The wireless sensor nodes communicate with a base station, serving as "data aggregator", of type Lord Microstrain WSDA-2000 [28], over the license-free 2.405-2.480 GHz frequency with 16 channels. The management of the wireless SHM system is facilitated via dedicated software, running on a portable computer, to which the base station is connected.

4.2. Modeling and simulation of the beam

Using the modeling and simulation process, previously described, the beam is analyzed under five damage scenarios, corresponding to five damage classes for training the convolutional neural network, labeled L1, L2, L3, L4 and L5, in addition to the "intact beam scenario", labeled as class

L0. Two types of damage are considered, (i) cracking of the beam at three locations and (ii) "softening" of the supports, representing effects of scour on bridge abutments. In accordance with the aforementioned labels, the damage type for each scenario is illustrated in Figure 5. Cracks are modeled as discontinuities, represented by rotational springs, while the softening of the supports is modeled using translational springs in the vertical direction instead of pin/roller supports. The values for the rotational springs corresponding to the cracks (scenarios L1, L2, and L3) range between 340 kNm/rad 1,900 kNm/rad, and the translational springs assume values between 420 kN/m and 610 kN/m. These spring values result in a decrease of the fundamental eigenfrequency between 7% and 10%, which is represents low to moderate damage.

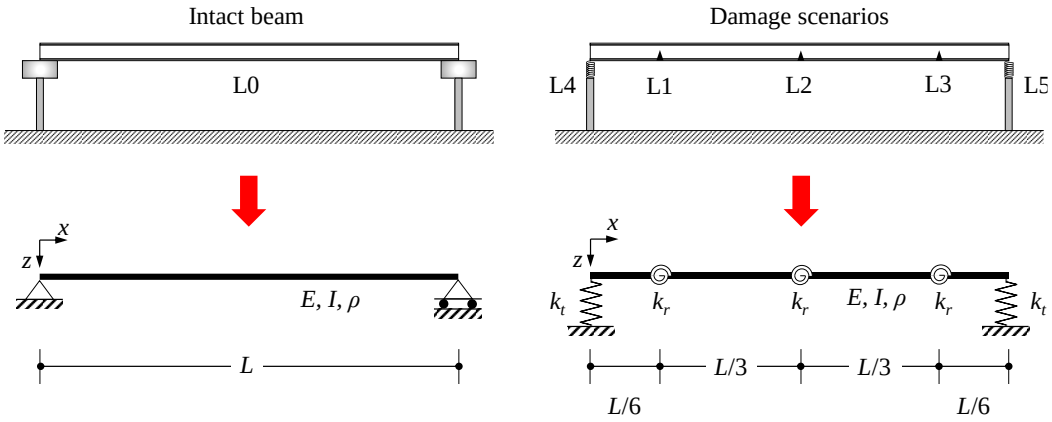

Figure 5. Intact beam and damage scenarios inflicted on the beam.

As mentioned previously, the modeling and simulation of the beam is conducted to generate labeled training data for the CNN, which must cover an adequate part of the solution space for the problem being studied. Therefore, several simulations are run by perturbing parameters of the problem; specifically, the traveling mass, the speed of the mass, and the values of the rotational and translational springs. Since samples of the beam material have been tested in the laboratory, the material properties are considered of reduced uncertainty and are, thus, excluded from the perturbations. The problem parameters in each perturbation are defined using dimensionless factors α_1, α_2 and α_3 with $0 \leq \alpha_1, \alpha_2, \alpha_3 \leq 1$, as follows:

(1) Traveling mass: $m = 0.01 + 0.03 \cdot \alpha_1$ (0.01 t $\leq m \leq$ 0.04 t)

(2) Mass speed: $v = 0.20 + 0.30 \cdot \alpha_2$ (0.20 m/s $\leq v \leq$ 0.50 m/s)

(3) Rotational springs at 1/6-length points: $k_r = 340 + 160 \cdot \alpha_3$ (340 kNm/rad $\leq k_r \leq$ 500 kNm/rad)

(4) Rotational springs at mid-span: $k_r = 1{,}300 + 600 \cdot \alpha_3$ (1,300 kNm/rad $\leq k_r \leq$ 1,900 kNm/rad)

(5) Translational springs: $k_t = 420 + 190 \cdot \alpha_3$ (420 kN/m $\leq k_t \leq$ 610 kN/m)

The values for the dimensionless factors are generated using the Latin Hypercube Sampling (LHS) method [29], which produces combinations of random samples of the factors assuming uniform distributions for all factors. The LHS method builds upon the "Latin square" concept, which is commonly applied in statistical sampling and dictates that each sample has its own row/column position in the square and is only used in one combination. The LHS method is essentially a generalization of the Latin square concept to an arbitrary number of dimensions, whereby only one sample is present in each axis-aligned hyperplane containing the sample. The LHS method has the advantage of requiring only the number of samples that are absolutely necessary for the combinations, thus avoiding redundant samples and contributing to the computational economy of the analyses. It is noted that the dimensionless factor α_3 is re-used in perturbations (3), (4) and (5), because the values of the rotational and the translation springs are not combined with each other. Instead, the values of k_r for scenarios L1, L2 and L3, as well as the values of k_t are individually combined with values of m and v.

The LHS method results in a total of 3,000 simulations, 500 simulations for each class. In each simulation, acceleration response data is computed at locations on the beam matching the locations of the sensor nodes, using Equations 22-24 and Equation 7 with two mode shapes, which, from preliminary tests, have been proven to adequately capture the dynamic behavior of the beam. The damping of the two mode shapes has experimentally been estimated to $\xi_1 = 2.2‰$ and $\xi_2 = 8‰$. It is noted that the damping coefficients are estimated for the intact beam; however, for low-to-moderate damage considered, the effect on the damping coefficients is expected to be negligible and are therefore kept constant in all scenarios [30]. The surface roughness $r(x)$ is also estimated from preliminary tests, conducted on the experimental setup, by moving a small mass across the beam length at a low velocity (4 cm/s) to avoid any dynamic effects, i.e. the transverse beam acceleration is $\ddot{w}(x, t) \approx 0$, and by measuring the acceleration response data directly on the traveling mass with a wireless sensor node. This acceleration response data is equated to the second time derivative of the roughness, assuming that the transmissibility between the contact surface of traveling mass on the beam and the wireless sensor node is close to unity. For more information on the estimation of roughness, the interested reader is referred to [31]. Thereupon, the spatial derivative of the roughness is approximated as:

$$\ddot{r}(t) = v^2 r''(x) \tag{26}$$

The acceleration time histories from all simulations are subjected to the Gabor transform to produce images for the labeled training data set of the CNN. For the Gabor transform, a Hanning window (z) of size N_o = 128 measurements is selected and a sampling rate of f_s = 128 Hz. The overlap H is set to 120 measurements and the scaling factor S_p is computed equal to $1.65 \cdot 10^{-4}$ and $3.31 \cdot 10^{-4}$ for $u = 0$ and $u \neq 0$, respectively. Exemplary Gabor transform images, plotted in logarithmic scale, are shown in Figure 6 below. The definition and training of the CNN using the labeled training data set with the Gabor images is described in the next subsection.

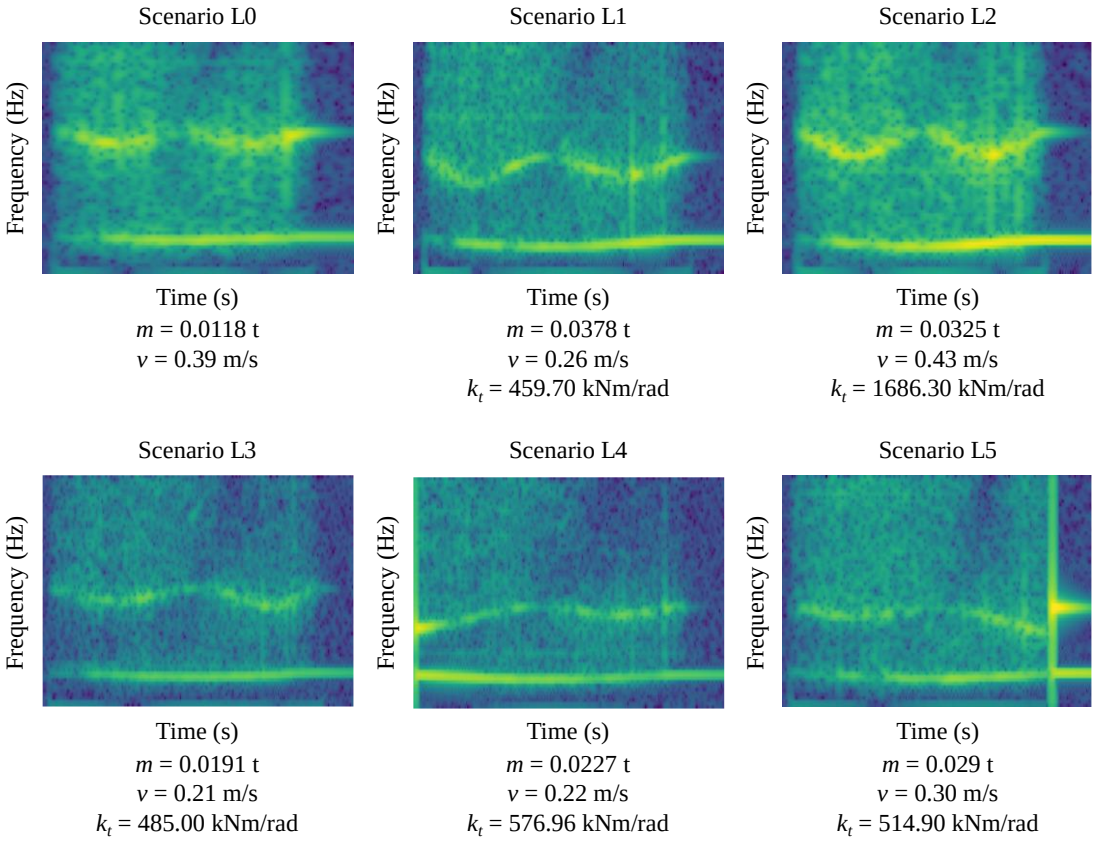

Figure 6. Spectrograms built from the numerical simulations for all scenarios.

4.3. Definition and training of the convolutional neural network

Based on trial-and-error analyses, the architecture defined for the CNN in this paper is shown in Figure 7. Since acceleration response data from three sensor nodes are collected, three CNN models are built, each dedicated to one sensor node. The reasoning behind using three sensor nodes and three CNN models is to compare the damage detection capabilities of the CNN models with different proximity to the location of the damage. The CNN architecture comprises an input layer,

accepting RGB images of dimensions 180×180, one convolution layer with 16 kernels of size 9×9 with stride 1, resulting in 16 activation maps of size 172×172, and one pooling layer, applying maximum pooling with a 8×8 window, stride 4 and 50% dropout, which results in 16 activation maps of size 21×21. The output of the pooling layer is flattened into a feature vector with 7,056 entries, which is followed by a fully-connected layer of 128 neurons, whose activations are forwarded with 50% dropout to a Softmax layer and eventually to an output layer, containing a number of classes equal to the scenarios previously defined.

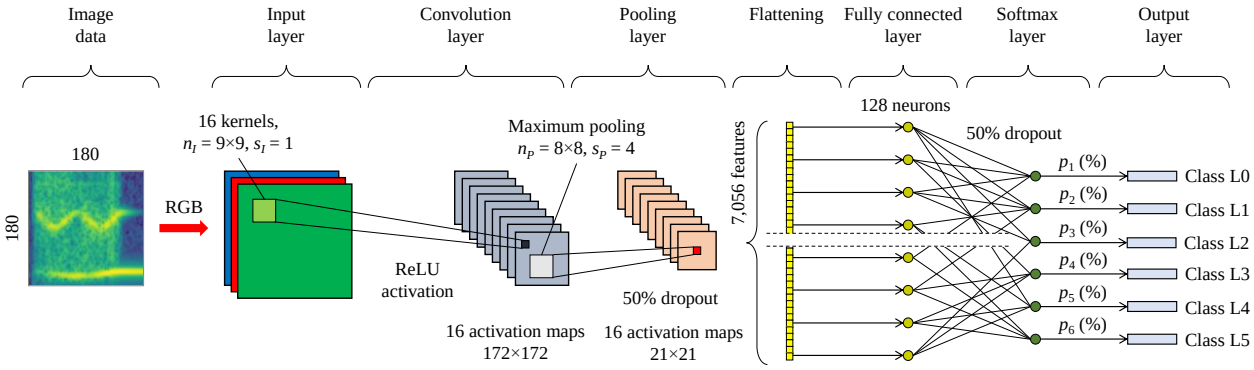

Figure 7. Architecture defined for the convolutional neural network.

As can be seen in Figure 7, the CNN architecture is kept relatively simple with low depth (i.e. "shallow learning"). The rationale for keeping the architecture simple is that the classification using the Gabor images is based on combining low-level features from the entire images, which is hardly possible with complex deep CNN architectures. In other words, the idea is to avoid classifying the images by identifying high-level features at specific regions (as typically done in image recognition), which in these validation tests would result in the CNN overfitting to regions of the Gabor images that may contain little information on the structural response. The training subset is formed using 80% of the Gabor images (i.e. 2,400 images), 20% of which (i.e. 480 images) are used for the validation subset. The rest 20% of the images in the labeled training data set (i.e. 600 images) forms the testing subset. The training is performed using the Python TensorFlow toolkit [32], the Adam optimizer with a sparse cross-entropy cost function, and a learning rate of 0.001. Training is completed after 7 epochs for all CNN models, with classification accuracy values, defined as percent ratio of correctly classified images over the total number of

images in the validation subset, reaching 97.5%. Thereupon, the 600 images of the testing subset are propagated through each CNN model; "confusion matrices" for the testing subset, showing the classification results for all CNN models are shown in Figure 8. The training accuracy histories for all CNN models are plotted in Figure 9.

Location — Actual scenario

Classification

L/4	**L0**	**L1**	**L2**	**L3**	**L4**	**L5**
L0	100					
L1		100				
L2			100			
L3				100		
L4					100	
L5						100

Location — Actual scenario

Classification

9*L*/20	**L0**	**L1**	**L2**	**L3**	**L4**	**L5**
L0	100					
L1		100				
L2			100			
L3				100		
L4					100	
L5						100

Location — Actual scenario

Classification

3*L*/4	**L0**	**L1**	**L2**	**L3**	**L4**	**L5**
L0	100					
L1		100				
L2			100			
L3				100		
L4					100	
L5						100

Figure 8. Confusion matrices for the testing subset.

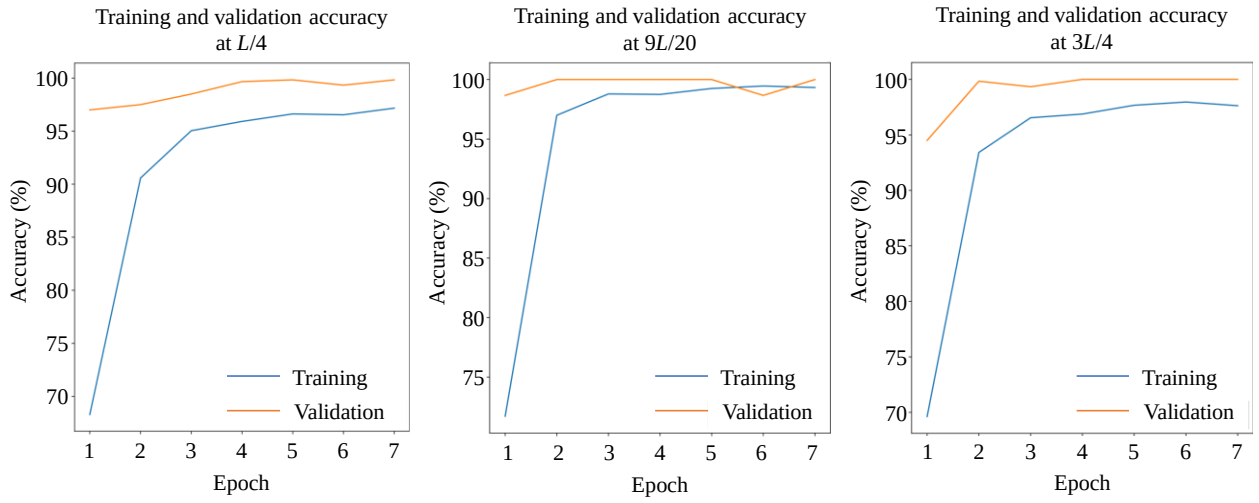

Figure 9. Training accuracy of the CNN models.

As shown in Figure 8, all CNN models classify the scenarios of the testing subset correctly, verifying that training is successful. The performance of the CNN models is validated using Gabor images generated from acceleration response data collected with the wireless sensor nodes and the experimental setup, as will be shown in the next subsection.

4.4. Application of the convolutional neural networks

The experimental setup shown in Figure 5 is used to conduct experiments to collect acceleration response data with the wireless sensor nodes. Using combinations of traveling mass values and speed values, a total of 120 experiments are devised, covering four scenarios (L0, L2, L4, and L5). For the scenarios L4 and L5, the supports at the left end and the right end, respectively, are substituted by translational springs of stiffness $k_{t,exp}$ = 457 kN/m (Figure 10 left). For the scenario L2, a cut is made 10 cm off the midspan of the bridge, extending through the bottom flange of the beam and part of the web, with a total height of 1.65 cm (Figure 10 right), resulting in an equivalent rotational spring of $k_{r,exp}$ = 3558 kNm/rad. Both damage types are devised to cause damage of severity that lies outside the upper and lower "severity bounds", considering in modeling and simulation, i.e. the 7% and 10% drop of the fundamental eigenfrequency, respectively. Specifically, the translational springs cause a drop in the eigenfrequency of 17%, while the respective drop for the cut at the midspan is 4%. The reason for selecting the damage types is to test the generalization capabilities of the CNN models for damage scenarios that are both less severe and more severe than the damage simulated. Five traveling mass values are used, 13 kg, 18 kg, 23 kg, 27 kg, and 38 kg, and six speed values from 0.25 m/s to 0.50 m/s with 0.05 m/s increment. The acceleration response data collected by the three wireless sensor nodes is used to produce Gabor transform images, similar to the images produced during modeling and simulation. The similarity in the features exposed by the Gabor transform between modeling and simulation and the experiments is shown in Figure 11. Moreover, Figure 12 shows the subtle differences between scenarios L0 and L2 in the features of the corresponding Gabor images that corroborate the necessity of machine learning for damage detection in the traveling mass problem. The 120 Gabor images are propagated through the CNN variants, previously trained. The classification results are shown in the form of a confusion matrix, illustrated in Figure 13.

Figure 10. Translation spring substituting pinned support (left) and cut close to midspan (right).

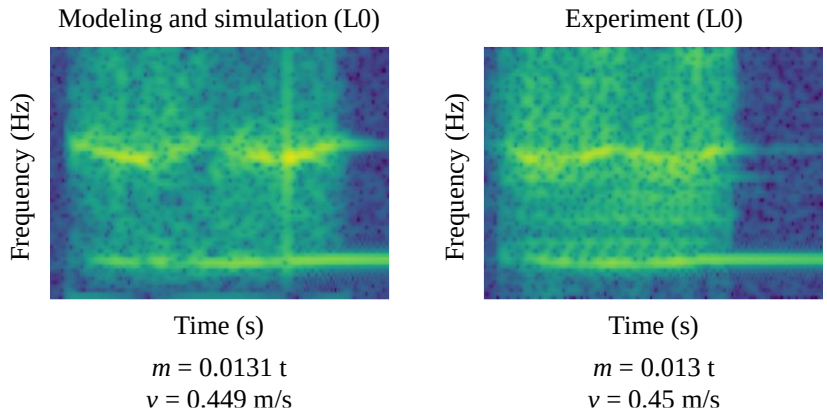

Figure 11. Comparison between Gabor images from modeling and simulation and corresponding experiment.

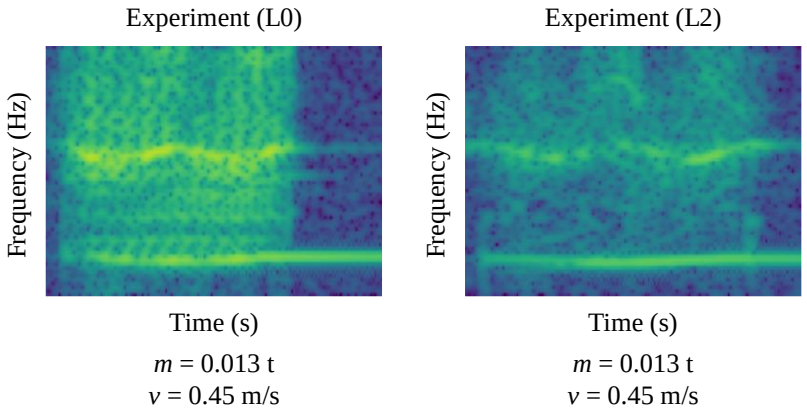

Figure 12. Differences in eigenfrequency-related features between scenario L0 and scenario L2.

Location — Actual scenario; Classification

S1	**L0**	**L1**	**L2**	**L3**	**L4**	**L5**
L0	30					
L1						
L2	4		26			
L3						
L4					30	
L5						30

Location — Actual scenario; Classification

S2	**L0**	**L1**	**L2**	**L3**	**L4**	**L5**
L0	27	3				
L1						
L2	16		14			
L3						
L4					30	
L5						30

Location — Actual scenario; Classification

S3	**L0**	**L1**	**L2**	**L3**	**L4**	**L5**
L0	30					
L1						
L2	3		27			
L3						
L4					29	1
L5						30

Figure 13. Confusion matrices for the experiments.

As can be seen in Figure 13, the best classification results are obtained for the acceleration response data from sensor nodes S1 and S3. Particularly concerning damage scenario L2, the classification using the data from sensor node S2 (close to the midspan) exhibits several "false negatives", which, at a first glance, would be unexpected considering the proximity of the sensor node to the crack. However, the false negatives are attributed to the nodal (zero-crossing) point of the second mode shape, whose participation in the structural response has been considered during modeling and simulation and has been observed in the acceleration response data of the other sensor nodes. The scarcity of features of the second mode shape in the Gabor images from sensor node S2 has resulted in the false classification of the images as "intact", since the classification has predominantly been driven by features of the first mode shape, which has low sensitivity to the onset of damage. As regards the damage scenarios L4 and L5, the classification results demonstrate the capability of the CNN models in detecting features indicative of softening of supports, as well as to generalize for damage severity that has not been accounted for during training. Satisfactory generalization capabilities are also observed for CNN models corresponding to sensor nodes S1 and S3 for damage scenario L2.

Nevertheless, the classification results of damage scenario L2 are subjected to further scrutiny in an attempt to examine whether the consensus of the three sensor nodes helps avoid false negatives. In Table 1, the 30 experiments, conducted for damage scenario L2 are listed, along with the classification results from each CNN model, represented by the corresponding wireless sensor node.

Table 1. Damage classification of experiments for the L2 damage scenario (shaded cells with bold font indicate false classification).

Experiment	Classification			Experiment	Classification		
	S1	S2	S3		S1	S2	S3
1	L2	**L0**	L2	16	L2	L2	L2
2	L2	**L0**	L2	17	L2	L2	L2
3	L2	**L0**	L2	18	**L0**	**L0**	**L0**
4	L2	L2	L2	19	L2	**L0**	L2
5	L2	**L0**	L2	20	L2	**L0**	L2
6	L2	**L0**	L2	21	L2	**L0**	L2
7	L2	**L0**	L2	22	L2	L2	L2
8	L2	L2	L2	23	L2	L2	L2
9	L2	L2	L2	24	**L0**	L2	L2
10	L2	L2	L2	25	L2	**L0**	**L0**
11	L2	**L0**	L2	26	**L0**	**L0**	L2
12	**L0**	**L0**	L2	27	L2	**L0**	**L0**
13	L2	L2	L2	28	L2	L2	L2
14	L2	L2	L2	29	L2	L2	L2
15	L2	**L0**	L2	30	**L0**	L2	L2

As evidenced by Table 1, only in one experiment a unanimous false classification is observed. Furthermore, even if the decision upon the existence of damage is based on majority voting, false classification in 2 out of 3 CNN variants is observed only in 4 experiments. As a result, assuming that the proposed damage detection approach is applied using acceleration response data from more than one location on the structure, false negative classifications are likely to be avoided. Finally, Figure 14 illustrates the classification probabilities for all the scenarios applied in the laboratory experiments and for the CNN models of sensor nodes S1 and S3. The high classification probabilities for damage scenarios L4 and L5 (close to 100%) show that the CNN models are capable of generalizing easily when the damage exceeds the lower bound of damage severity simulated for generating the labeled training data set. By contrast, when damage exceeds the

respective upper bound (damage scenario L2), the classification probabilities are lower, and false negatives are likely, since damage scenario L2 is close to the intact beam scenario L0. To enhance the classification capabilities of the CNN models, it is recommended to adopt a transfer learning method that would help accommodate the domain mismatch between simulations and real-world structural behavior.

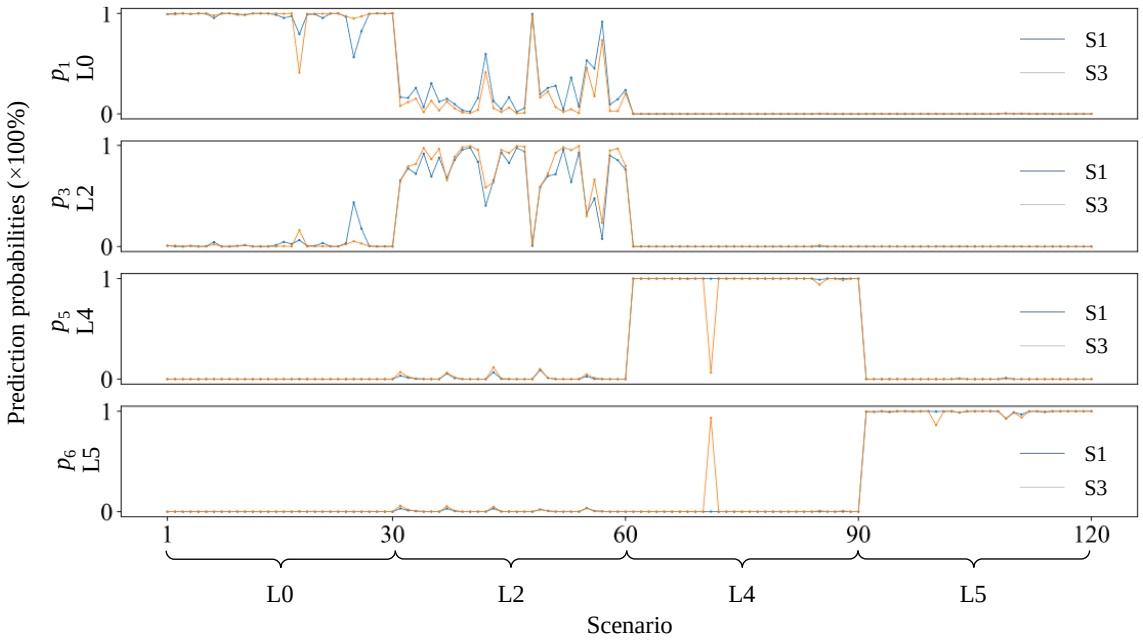

Figure 14. Classification probabilities for all classes considered in the laboratory experiments for sensor nodes S1 and S3.

5. Summary and conclusions

Traveling masses on lightweight bridges represent a non-stationary problem that requires sophisticated data analysis methods, extending beyond traditional data analysis methods applied in structural health monitoring and experimental testing. Drawing up on recent advances in informatics, this paper has proposed a damage detection approach using machine learning models, the mapping capabilities of which have been well established in scientific literature. In particular, convolutional neural network models have been employed for classifying acceleration response data, collected by sensors installed on structures, into damage scenarios. In lieu of acceleration

response data corresponding to actual structural damage, the CNN models are trained using acceleration time histories generated from an analytical modeling and simulation process, employing a computationally efficient Euler-Bernoulli beam model. Prior to being used for CNN training, the acceleration time histories are subjected to the Gabor transform, which exposes coupled time-frequency features in the accelerations that facilitate classification. Upon completing training, the CNN models are applied using real-world acceleration response data.

The proposed damage detection approach has been implemented and validated in laboratory tests, using an experimental setup, consisting of a simply supported steel beam, a moving mass (sliding along the beam), pulleys, and a motor for regulating the speed of the moving mass. The beam first has been modeled and analyzed for various combinations of moving masses and speeds under six scenarios, five of which have included damage and one representing the "intact" beam, using the analytical modeling and simulation process and considering the dynamic contribution of two mode shapes. The acceleration time histories from the modeling and simulation from three measurement locations of the beam have been used to train three CNN models, respectively, to classify the accelerations into the six scenarios. Upon completing training, the CNN models have been used for classifying acceleration response data collected from experiments conducted using the aforementioned setup and combinations of moving masses and speeds. The beam has been instrumented with three wireless sensor nodes, which have been attached to the same locations as the measurement locations considered for modeling and simulation. Acceleration response data has been collected from the intact beam and after inflicting damage, represented by a crack close to the midspan of the beam and by substituting the end supports with translational springs. In total, four out of the six scenarios, considered in modeling and simulations, have been devised for the experiments. Moreover, the damage severity in the experiments has deliberately been devised to fall outside the severity bounds considered in modeling and simulation to test the generalization capabilities of the CNN models. The test results have shown that two CNN models, corresponding to quarter-length locations of the beam, have exhibited high accuracy in classifying the acceleration response data from the experiments, with only four misclassification instances per variant. The CNN variant that has used acceleration response data from the midspan has performed well in classifying damage to the supports, but below average in classifying the crack damage close the midspan. The poor performance of this CNN variant is attributed to the scarcity of

frequency content of the second mode shape, which is more sensitive to damage than the first mode shape, due to a zero-crossing point of the second mode shape at the midspan. Nevertheless, as shown in the paper, the combined outcome of all CNN variants is capable of reducing the probability of false negatives, when the damage detection outcome follows a majority voting rationale. Finally, the test results have shown that the generalization capabilities of the CNN variants are better when damage severity exceeds the upper severity bound, considered in modeling and simulation, than when dropping below the lower severity bound. Future work will involve testing the damage detection approach in real-world conditions and integrating the approach into a structural health monitoring strategy.

Acknowledgments

This paper is the result of a collaborative effort of authors funded by different sources. The authors would like to gratefully acknowledge the support offered by the German Research Foundation (DFG) under grant SM 281/20-1 and of the Hellenic Foundation for Research and Innovation (HFRI) under grant 6255. Results presented in this work have been produced using the Aristotle University of Thessaloniki High-Performance Computing Infrastructure. Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the DFG or the HFRI.

Conflict of interest

The authors declare no conflict of interest.

Data availability statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

1. Cunha Á, Caetano E. Experimental modal analysis of civil engineering structures. *Sound and Vibration* 2006, 6(40): 12–20.

2. Farrar C, Worden K. An introduction to structural health monitoring. *Philosophical Transactions of the Royal Society A* 2007, 365(2007): 303–315.
3. Doebling S, Farrar C, Prime M., Shevitz, D. Damage identification and health monitoring of structural and mechanical systems from changes in their vibration characteristics. *Technical Report LA-13070-MS* 1996, Los Alamos, New Mexico, USA.
4. Chang PC, Flatau A, Liu SC. Review paper: Health monitoring of civil infrastructure. *Journal of Structural Health Monitoring* 2003, 2(3): 257–267.
5. Brandt A. *Noise and Vibration Analysis: Signal Analysis and Experimental Procedures*. Wiley, Chichester, UK, 2011.
6. Renaudot A. Etude de l'influence des charges en mouvement sur la résistance des ponts métalliques à poutres droites, *Annales des Ponts et Chaussées* 1861, 1: 145–204.
7. Fryba L. *Vibrations of Solids and Structures under Moving Loads*, Noordhoff, Groningen, 1972.
8. Liu K, Roeck GD, Lombaert G. The effect of dynamic train-bridge interaction on the bridge response during a train passage. *Journal of Sound and Vibration* 2009, 325: 240–251.
9. Bajer CI, Dyniewicz B. *Numerical Analysis of Vibrations in Structures under Moving Loads*, Springer, Heidelberg, 2012.
10. Green MF, Cebon D. Dynamic response of highway bridges to heavy vehicle loads: Theory and experimental validation, *Journal of Sound and Vibration* 1994, 70(1): 51-78.
11. Green MF, Cebon D. Dynamic interaction between heavy vehicles and highway bridges, *Computers and Structures* 1997, 62(2): 253-264.
12. Yang YB, Lin CW, Yau JD. Extracting bridge frequencies from the dynamic response of a passing vehicle. *Journal of Sound and Vibration* 2004, 272(3-5): 471-493.
13. Kehtarnavaz N. *Digital Signal Processing System Design, 2nd Edition*, Academic Press, Cambridge, Massachusetts, 2011.
14. Johansson C. *Simplified Dynamic Analysis of Railway Bridges under High-speed Trains*, Licentiate Thesis, Kungl Tekniska Hogskolan, Stockholm, Sweden, 2013.
15. Malekjafarian A, Obrien EJ. Identification of bridge mode shapes using Short Time Frequency Domain decomposition of the responses measured in a passing vehicle, *Engineering Structures* 2014, 81: 386-397.
16. Katounin A, Holewik F. Crack identification in composite elements with non-linear geometry using spatial wavelet transform. *Archives of Civil and Mechanical Engineering* 2013, 13: 287–296.
17. Ignacio G, Karoumi R. Analysis of the annual variations in the dynamic behavior of a ballasted railway bridge using Hilbert transform. *Engineering Structures* 2014, 60: 126–132.

18. He WY, He J, Ren WX. Damage localization of beam structures using mode shapes extracted from moving vehicle response. *Measurement* 2018, 121: 276-285.
19. Zhong M, Yang M, Gao Z. Dynamic responses of prestressed bridge and vehicle through bridge-vehicle interaction analysis, *Engineering Structures* 2015, 87: 116–125.
20. Dadoulis GI, Manolis GD. Dynamic Response of a Damaged Bridge Span Traversed by a Heavy Point Mass, *Journal of Sound and Vibration* 2023, 551: 117613. *doi.org/10.1016/j.jsv.2023.117613*
21. Brownjohn JMW, De Stefano A, Xu YL, Wenzel, H, Aktan, A. Vibration-based monitoring of civil infrastructure: Challenges and successes. *Journal of Civil Structural Health Monitoring* 2011, 1(3-4): 79–95.
22. Cawley P. Structural health monitoring: Closing the gap between research and industrial deployment. *Journal of Structural Health Monitoring* 2018, 17(5): 1225–1244.
23. Farrar C, Worden K. *Structural Health Monitoring: A Machine Learning Perspective*. Wiley, Chichester, UK, 2013.
24. Avci O, Abdeljaber O, Kiranyaz S, Hussein M, Gabbouj M, Inman DJ. A review of vibration-based damage detection in civil structures: From traditional methods to machine learning and deep learning applications. *Mechanical Systems and Signal Processing* 2021, 147: 107077.
25. Dadoulis G, Manolis GD, Katakalos K, Al-Zuriqat T, Dragos K, Smarsly K. Towards detecting damage in lightweight bridges with traveling masses using machine learning. *Proceedings of the 2024 European Conference on Computing in Construction (EC3)*. Chania, Crete, Greece, 14-17 July 2024.
26. Inman DJ. Engineering Vibrations, 4th Edition, Pearson Education, London, UK, 2013.
27. Parker Hannifin Microstrain Sensing (2020). Microstrain Sensing Product Datasheet: G-Link-200 Ruggedized Wireless Triaxial Accelerometer Node. Williston, VT, USA: Parker Hannifin Corp.
28. Parker Hannifin Microstrain Sensing (2020). Microstrain Sensing Product Datasheet: WSDA-2000 Wireless Sensor Data Aggregator. Williston, VT, USA: Parker Hannifin Corp.
29. Loh W-L. On Latin Hypercube Sampling. *The Annals of Statistics* 1996, 24(5): 2058-2080.
30. Manolis GD, Dadoulis G, Katakalos V. Experimental evaluation of damping in beams using the acceleration generalized coordinates: A comparison of the FDD and PCA methods. *Soil Dynamics and Earthquake Engineering* 2023, 175,108219.
31. Carrigan T, Talbot JP. A new method to derive rail roughness from axle-box vibration accounting for track stiffness variations and wheel-to-wheel coupling. *Mechanical Systems and Signal Processing* 2023, 192, 110232.
32. Tensorflow: An end-to-end platform for machine learning. Available at: https://www.tensorflow.org/ (accessed: 04/26/2024).