Check for updates

Original Article

SHM Structural Health Monitoring

Structural Health Monitoring
1–14
© The Author(s) 2023
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/14759217231193102
journals.sagepub.com/home/shm
Sage

Damage identification using convolutional neural networks from instantaneous displacement measurements via image processing

Lucas Resende[1], Rafaelle Finotti[2], Flávio Barbosa[2], Hernán Garrido[3,4], Alexandre Cury[2] and Martín Domizio[3,4]

Abstract

This work investigates the effectiveness of using convolutional neural networks (CNNs) and instantaneous displacement measurements for damage identification in beams. The study involves subjecting laboratory beams to eight distinct damage scenarios and capturing the vertical positions of 60 points along the beam length during free-vibration tests using a high-speed camera. The data obtained was subsequently used to train a CNN in a supervised manner to estimate the level of damage at each point. Results showed that the CNN models were able to correctly localize and quantify the damage levels when trained on data from all damage scenarios. The soundness of the proposed methodology was demonstrated in a robustness assessment, where all eight damage scenarios were correctly identified even when two of them were excluded from the training dataset.

Keywords

Convolutional neural network, damage identification, high-speed camera, photogrammetry, instantaneous displacement, free vibration

Introduction

Throughout the entire life cycle of any civil engineering construction, the safety of users and the efficient functioning of structural systems must be a top priority for engineers. This is particularly crucial for large structures such as bridges, viaducts, tall buildings, and towers, where monitoring the structure condition on a continuous basis is imperative to detect abnormal behavior and prevent catastrophic failures. This ongoing process of identifying structural failures or malfunctions is known as structural health monitoring (SHM).

Avci et al.[1] conducted an extensive literature review and concluded that damage detection is the most critical aspect of an SHM system. Detection is the first step in the damage identification process, which also involves locating it and assessing its severity—according to the well-known Rytter's damage identification. In order to achieve the three steps above defined for damage identification, instead of using classical approaches based on modal tracking,[2–4] many researchers have turned to machine learning.[5] artificial neural networks (ANNs) undoubtedly deserve to be highlighted among the machine learning algorithms. ANNs are extremely versatile and can be applied to SHM problems through various architectures, such as fully connected networks,[6,7] autoencoders,[8,9] and, more recently, convolutional neural networks (CNNs).[10,11]

[1]Faculty of Engineering, Federal University of Juiz de Fora, Juiz de Fora, Brazil
[2]Graduate Program in Civil Engineering, Federal University of Juiz de Fora, Juiz de Fora, Brazil
[3]Consejo Nacional de Investigaciones Científicas y Técnicas CONICET, Argentina
[4]Facultad de Ingeniería, Instituto de Mecánica Estructural y Riesgo Sísmico, Universidad Nacional de Cuyo, Mendoza, Argentina

Corresponding author:
Hernán Garrido, Facultad de Ingeniería, Universidad Nacional de Cuyo, Centro Universitario, Parque Gral. San Martín, Mendoza M5502JMA, Argentina.
Email: carloshernangarrido@gmail.com

A CNN is a type of ANN commonly used in image recognition and processing tasks.[12] In the context of structural damage identification, a CNN is commonly applied to detect and classify damage patterns in structural time series, such as those obtained through various non-destructive testing techniques. The network uses convolutional layers to learn spatial features of the time histories by detecting local patterns associated with damage. These features are then fed into fully connected layers that perform the classification task.[13] Thus, the network determines the type and severity of the damage present in the structure from where the input data was taken. CNNs have shown promising results in the field of SHM, and several recent works have addressed this topic.[11,14–16]

Since the input data for these algorithms must be reliable, successful *identification* of damage in structures also depends on effective instrumentation. In addition, in order to be able to locate and quantify damage, it is usually necessary to have many instrumented points in the structure, which can add difficulties to the already complex problem of damage identification.[17] Hence, a successful *implementation* of damage identification requires instrumentation that is not only *effective*, but also *efficient*. In this sense, the use of cameras to capture displacements is an attractive approach to overcome the challenges associated with instrumentation composed of many sensors. In such cases, adequate image processing of videos from the tested structure can provide a substantial number of displacement time histories corresponding to different points in space. This approach can potentially replace a large number of instruments based on sensors such as displacement transducers.[18]

In this context, this paper presents a comprehensive analysis of beam damage identification based on CNN and structural displacements obtained via image processing. The authors have opted to use CNN since they were originally created to deal with images, which are the main inputs for the proposed methodology. This differs from classical CNN applied directly to images in that the images are preprocessed to extract displacement information. The preprocessing largely improves the signal-to-noise ratio of the training dataset and reduces its dimensionality (from millions to tens).

The study uses a high-speed camera to evaluate a dynamically monitored beam when subjected to eight different damage scenarios (healthy structure and seven damaged states). Time-series data of dynamic displacements for 60 distinct beam points are extracted for each scenario employing an ad hoc image processing method. The deformed configurations of the beam at different instants of time are then used to train a CNN to identify, localize, and quantify the damage level at each point. In addition, a robustness analysis is carried out to assess the ability of the proposed methodology to detect damage in samples from scenarios that were completely excluded from the training process.

Most of the previous works on CNN for damage identification use convolution in time to learn, precisely, *temporal patterns*. For example, in Chen et al.[10] a single-channel time history is preprocessed to generate time-frequency images (spectrograms) that are then used as inputs for a CNN. A distinctive aspect of the present work is that time dependency of displacements is not used as information at the input for the CNN. On the contrary, the convolutional layers are intended to learn *spatial patterns*. The algorithm is expected to work based on the assumption that the specimen is in free vibration and has proportional damping. Under this condition, the instantaneous structural response is a linear combination of mode shapes and deflection due to gravity. Since both types of deformation have shapes that are characteristic of the beam and its state, instantaneous displacements are expected to convey significant information about damage.

In simple words, the proposed methodology treats the frames of the video as individual and independent snapshots of the specimen. Not needing the time information is an important practical advantage of the algorithm for the following reasons. Firstly, free-vibration record segments can be automatically obtained using a dynamics-based algorithm[19] or using state-of-the-practice object detection (from the same video or using synchronized cameras) to detect the forcing objects on the structure. In this context, the shorter the free-vibration record segments required, the easier (more likely) they are to obtain.[20] Secondly, since time information is not necessary, there is no need for time order or time continuity of the instantaneous displacements obtained from video. Therefore, the algorithm can be fed with instantaneous displacements measured in any order and even on different days. These facts can leverage the use of a large number of samples corresponding to the same scenario at a low cost. This aspect of the approach has two consequences: on one hand, it yields a substantial increase in the training data, as each deformed configuration becomes an input sample; on the other hand, relatively high performance in terms of damage identification can be obtained by combining predictions in an *ensemble*[21] method for example, the median of predictions.

As a final note, it can be said that learning *spatial patterns* has the side advantage that they are expected to be less sensitive to environmental conditions than *temporal patterns*. Note that classical temporal patterns as the structure natural frequencies can be even more sensitive to day-night temperature changes than to severe damage.[22] On the contrary, classical spatial patterns as modal shapes and modal curvatures are more

sensitive to damage than to global changes (under the assumption of sparse damage[23]). Good examples of exploiting this phenomenon are the works of Deraemeker et al.,[24,25] where local modal filters are used as feature extractors for damage localization using ambient vibrations under a changing environment.

Convolutional neural networks

CNNs,[26] which are neural networks inspired by biology, are commonly utilized in various systems for object and event recognition in images or videos. Despite being around for almost 30 years since their inception, contemporary CNN architectures still incorporate similar characteristics as their predecessors, including convolutional and pooling layers. In addition, back-propagation remains the most widely used training method.[27]

The broad application domains and popularity of CNNs can be attributed to several advantages, such as their ability to merge parameter extraction and feature classification into a single learning body. Additionally, the sparsely connected neurons of CNNs allow for efficient processing of large inputs compared to fully connected Multi-Layer Perceptron networks. CNNs are also invariant to minor input data transformations, including translation, scaling, and skewing, and are able to adapt to various input sizes. These benefits are highly valuable when dealing with SHM issues. This is because the input data typically consists of temporal histories with varying high dimensionality specific to each problem, as well as characteristics that can change irrespective of the damaged state of the structure being analyzed, such as scale and amplitude. Although this is the approach used in the vast majority of works that address CNNs applied to SHM, it is not the case in this article, where the input data consists of deformed configurations of the structure instead of temporal histories, as previously highlighted.

A complete review can be found in the work of Goodfellow et al.[28] for a comprehensive understanding of CNNs. This review delves into the fundamental concepts of CNNs, their architecture, and their importance in the field of computer vision. Bhatt et al.[29] provide a good overview of the history of CNNs, how they work, and the various architectures developed over the years. These references are excellent resources for anyone seeking to understand the ins and outs of CNNs.

Experimental program

The CNN utilized in this study was architected and optimized through an iterative process consisting of training, testing, and validation, using only experimental data. These data were collected according to the experimental program described below.

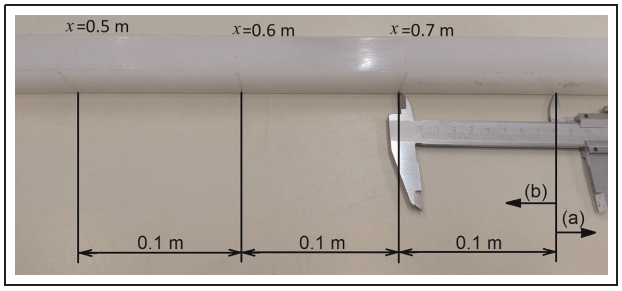


Figure 1. Zoom view of the specimen in the damaged zone. References: (a) clamped and (b) free.

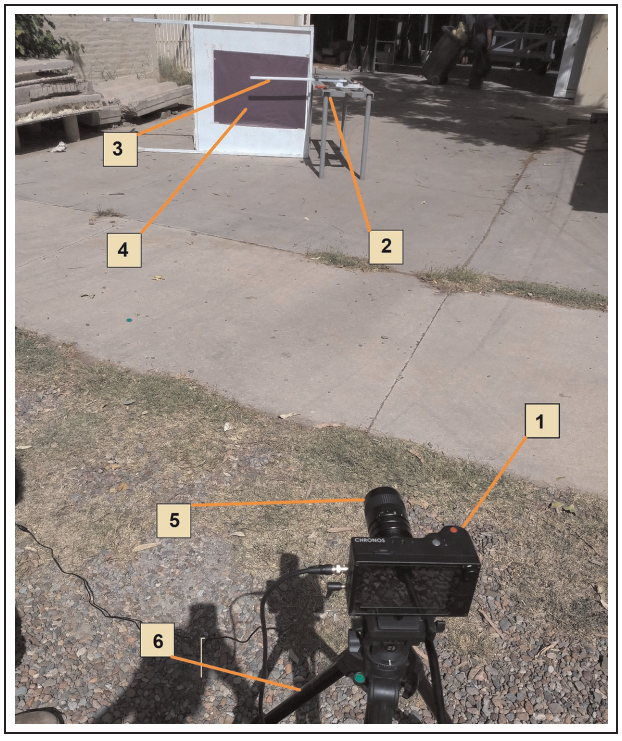


Figure 2. Experimental setup. References: (1) high-speed camera, (2) fixed support, (3) beam specimen, (4) background, (5) lens, and (6) tripod.

Test setup

The beam specimen under test (Figure 1, and reference 3 in Figure 2) consisted of a 1 m long straight cantilever beam made of polypropylene with a 25.4 mm × 25.4 mm squared cross-section. In the reference frame used, the variable x represents the position along the longitudinal axis of the beam. Specifically, $x = 0$ m corresponds to the free end of the beam, while $x = 0.8$ m corresponds to the

Table 1. Damage scenarios.

Scenario	Depth $d_c(x)$ (mm) at			Actual $D_{EI}(x)$ at		
	$x_d = 0.5$ m Position 36	$x_d = 0.6$ m Position 44	$x_d = 0.7$ m Position 53	$x_d = 0.5$ m Position 36	$x_d = 0.6$ m Position 44	$x_d = 0.7$ m Position 53
#1	0	0	0	1.000	1.000	1.000
#2	0	0	5	1.000	1.000	1.930
#3	0	0	10	1.000	1.000	4.487
#4	0	0	13	1.000	1.000	8.595
#5	0	5	13	1.000	1.930	8.595
#6	5	5	13	1.930	1.930	8.595
#7	5	10	13	1.930	4.487	8.595
#8	11	10	13	5.488	4.487	8.595

clamping point (see Figure 1). The resulting free span was 0.8 m long. The first natural frequency under healthy condition was $f_1 = 7.69$ Hz.

As shown in Figure 2, the scene was illuminated with natural daylight. On the other hand, the camera pose was aligned such that the line of sight was normal to the specimen axis.

Definition of damage scenarios

The beam was artificially damaged by producing vertical and rectangular Mode I (opening) cracks on its upper surface using a hand bow saw. The actual widths of the cracks were between 0.5 and 1 mm (Figure 1); so they were small enough to be considered infinitesimal, and large enough to avoid crack breathing. As a combination of crack locations x_d and depths d_c, eight successive damage scenarios were defined and executed, which are summarized in Table 1. This table also presents the damage index $D_{EI}(x_d)$, which is directly related to d_c and is defined further on. As shown in Figure 1, cracks were produced at 0.1, 0.2, and 0.3 m from the fixed end (x = 0.7, 0.6, and 0.5 m, respectively), and their depths were measured using a Vernier caliper for the quantification of the actual damage.

The region where the cracks were made was near the fixed support of the beam. This is because higher stresses are expected in that region, making it the most likely area for damage. Additionally, that region generally exhibits greater curvature, increasing the likelihood of detecting cracks using a spatial pattern-based method. In any particular application, the damage scenarios used for training should reflect the more likely situations, preferably from statistical data of actual damage observed in the past.

Data collection

For each of the eight damage scenarios, two trials were performed while captured in video. In these trials, free vibrations were produced by imposing an initial displacement of 4–5 cm to the beam tip. The incorporation of two trials was intended to make the experiment more representative of real-world scenarios, as it exposed the experiment to different lighting conditions although the camera pose remained invariant (image plane parallel to the plane containing the displacements). In a particular actual application, the number and conditions of data-collection trials should encompass all the likely variabilities. For example, if the camera pose relative to the beam or the boundary conditions of the macrostructure could change, the training dataset should include trials that capture these variabilities in each damage scenario. In this way the CNN could identify the damage being insensitive to such variabilities.

Videos were captured using a high-speed camera Chronos 2.1-HD (as shown in Figure 2), with a total duration of 2.7 s (approximately 20 times the period corresponding to f_1), at a frame rate of 1000 fps, with an exposure time of 100% (1/1000 s), and a resolution of 1280 px × 1024 px. The region of interest was considered to be 1200 px wide out of the total resolution. The distance between the camera and the specimen was about 8 m. This was chosen simply as that allows covering more than a half of the free part of the specimen with the available lens. Finally, the optical setup resulted in a pixel scale of approximately 0.6 mm per pixel.

Methodology definition

Input data

The proposed methodology relies on the number of data points considered, that is, with more data points available, the results tend to be better. This represents a strong advantage for camera-based monitoring compared to instrumented sensor-based approaches, since it allows observing several data points at the same time. In parallel, the larger the size of the horizontal window

used for calculating vertical displacements, the higher the accuracy of the results, as more pixels are involved in the calculations of each point. However, fewer vertical displacements are allowable from each snapshot. On the other hand, smaller horizontal window sizes allow for a larger number of vertical displacements along the beam to be measured, but with reduced precision. Finding the ideal balance between the number of monitored points and the number of pixels used for each point is a challenge inherent to camera-based instrumentation. In the current case, using 60 monitored data points and a horizontal window with 20 px per monitored point resulted in precise measurements. This choice of the proposed methodology is based on Saint-Venant's principle and the study by Garrido et al.[30] The latter confirmed, numerically, statistically, and experimentally, that a crack induces a reduction of relative flexural rigidity not as a Dirac's delta but in a region around the crack. Moreover, the corresponding apparent damage extent (along the longitudinal axis) depends on the beam cross-section height and not on the crack depth. Therefore, a general rule-of-thumb could be to choose a point spacing smaller than the apparent damage extent, which can be approximated as the beam height (considering a piecewise linear model[30]). For instance, in the specimen under test, the cross-section height is 0.0254 m while the point spacing resulting from using 60 points is approximately 0.01 m.

The first step in the proposed methodology is to define a vector of 60 displacement measurements, denoted as $\mathbf{y} = [y_1 \quad y_2 \quad y_3 \quad \cdots \quad y_{60}]$, which is used as input data for training the CNN. This input vector was obtained using image processing techniques, and a summary of this procedure is presented below.

For extracting displacements from video, in the particular case of a straight beam, an axial edge is a good natural feature if the background is uniform and of a different color/intensity than the beam.[23] An additional requirement is that one face corresponding to the selected edge is always hidden from the camera perspective. Thus, the edge presents high contrast and is not confused with the beam face.

In this work, the method described in Garrido et al.[23] for a single image, which uses the beam edge as a natural feature, was applied to each video frame to obtain displacement records. The main steps of the method are detailed below for the sake of completeness.

The image, that is composed of rows and columns, is partitioned in horizontal windows which are groups of image columns. Each horizontal window corresponds to a position in the beam where displacement is required. In the present work, 60 horizontal windows are considered.

For each video frame:

1. The image is converted to a gray scale by combining the three 8-bit color channels into a single double-precision matrix of intensity values.
2. For each image column: The horizontal edge is detected using the smooth Gaussian gradient calculated along the vertical direction. The vertical Gaussian window is 15 px long and ranges ± 3 standard deviations.
3. The image gradient is binarized using a threshold, and then the largest connected component is identified using the flood-fill algorithm. This component is a region of the image that will be used as a mask.
4. For each horizontal window:
   (a) For each image column:
      i. Where the mask is active: Gradient peaks are estimated and located with sub-pixel resolution by using cubic splines.
   (b) A straight line is fitted using the peak locations as sample points and their amplitudes as weights in a weighted least-squares linear fitting.[1]
   (c) The vertical positions y_i are estimated as the y-intercepts of each fitted straight line (assuming the y_i-axis is at the middle of each horizontal window).

After obtaining temporal histories at 60 points of the beam, the identification of the input vectors for the CNN is done by directly observing the displacements measured at each of the 60 points monitored for a given instant of time. Figures 3 and 4 describe typical input vectors $\mathbf{y} = [y_1 \quad y_2 \quad y_3 \quad \ldots \quad y_{60}]$ of data points and their respective positions on the beam. As depicted in Figure 4, the change in curvature induced by a crack is too subtle to be discerned with the naked eye. However, CNN architecture can internally amplify such changes through convolution.

Each experimental trial lasted for 2.7 s, with a data acquisition frequency of 1000 Hz, yielding a total of 2700 input data vectors per trial $\mathbf{y}$ for each damage scenario. To mitigate the impact of noise in the monitored signals, the authors decided to utilize deformed configurations that had more substantial displacements. Specifically, out of 2700 deformed configurations, only 1350 configurations with the highest displacement amplitudes, based on the displacement at the free end of the beam, were used in this work. Since all free-vibration tests were performed twice, the total number of samples per damage scenario was 2700.

Labeling

In the present work, damage is quantified numerically using the following damage index:

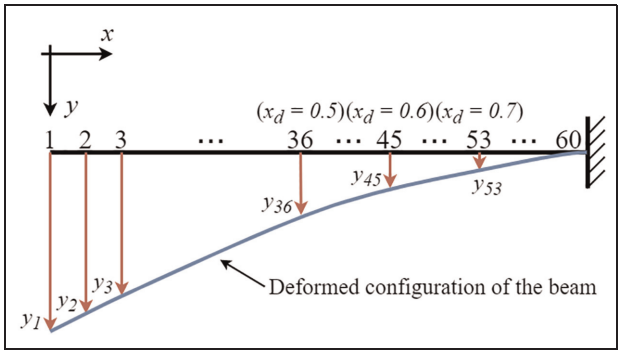


Figure 3. Description of input data **y**, numbering and position on the beam. Points 36, 45, and 53 (x_d = 0.5, 0.6 and 0.7 m, respectively), represent crack positions.

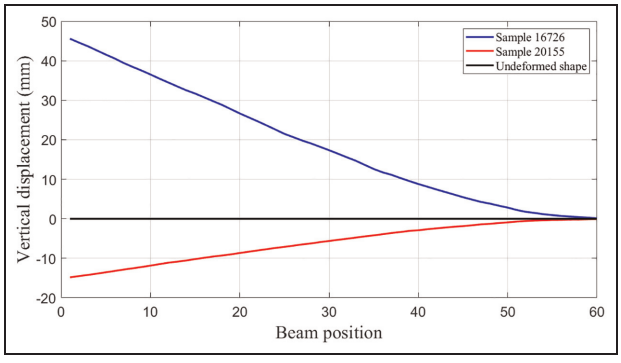


Figure 4. Two samples of the training dataset.

$$D_{\mathrm{EI}}(x) = \frac{\mathrm{EI}_{\mathrm{h}}(x)}{\mathrm{EI}_{\mathrm{d}}(x)} \tag{1}$$

where $\mathrm{EI}_{\mathrm{d}}(x)$ is the flexural rigidity at x; and $\mathrm{EI}_{\mathrm{h}}(x)$ is the flexural rigidity of the healthy beam, being constant with respect to x in this case of uniform beam; and E is the Young's modulus, also constant.

To use the CNN in a supervised learning setting, actual values of $D_{\mathrm{EI}}(x)$ have to be established. This was done by following three criteria:

1. At the crack locations x_{d}, actual values of $D_{\mathrm{EI}}(x)$ were estimated by geometry as:

$$D_{\mathrm{EI}}(x_{\mathrm{d}}) = \frac{I_{\mathrm{h}}(x_{\mathrm{d}})}{I_{\mathrm{d}}(x_{\mathrm{d}})} = \left(1 - \frac{d_{\mathrm{c}}(x_{\mathrm{d}})}{h_{\mathrm{s}}}\right)^{-3} \tag{2}$$

   where: $I_{\mathrm{h}}(x_{\mathrm{d}})$ and $I_{\mathrm{d}}(x_{\mathrm{d}})$ are the moments of inertia at the crack location without and with the crack, respectively; $d_{\mathrm{c}}(x_{\mathrm{d}})$ is the crack depth; and h_{s} is the cross-section height. The moment of inertia $I_{\mathrm{d}}(x_{\mathrm{d}})$ was approximated under the assumption that the neutral axis is at the middle of the remaining cross-section height.[23,30]

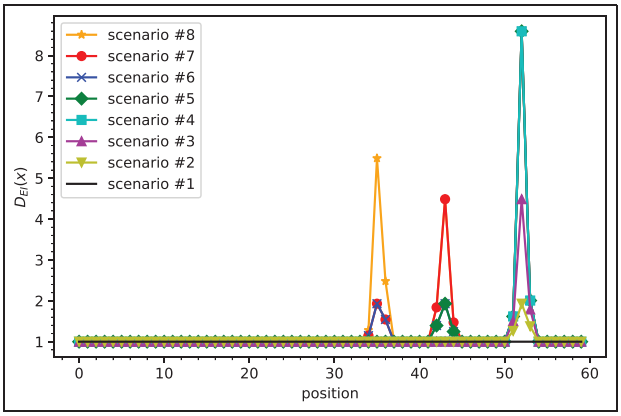


Figure 5. Actual values of $D_{\mathrm{EI}}(x)$ for the eight damage scenarios, which were used as labels for training the CNN. CNN: convolutional neural network.

2. Around each crack, actual values of $D_{\mathrm{EI}}(x)^{-1} = H_{\mathrm{EI}}(x)$ were assumed to follow a piecewise linear function in the interval $[x_{\mathrm{d}} - h_{\mathrm{d}}, x_{\mathrm{d}} + h_{\mathrm{d}}]$, as the healthy index $H_{\mathrm{EI}}(x)$ in[23,30]. The parameter $2h_{\mathrm{d}}$ is named as *apparent damage extent* and it has been shown to be dependent only on the beam height h_{s} as $2h_{\mathrm{d}} = h_{\mathrm{s}}$.[30]
3. Actual values of the damage index for all undamaged points, that is, outside the neighborhoods of the cracks $[x_{\mathrm{d}} - h_{\mathrm{d}}, x_{\mathrm{d}} + h_{\mathrm{d}}]$, are $D_{\mathrm{EI}}(x) = 1$.

The resulting actual values of $D_{\mathrm{EI}}(x)$ are shown in Figure 5 for the eight damage scenarios. These values were used as labels for training the CNN.

CNN architecture

The architecture of the CNN used in this work was defined through a previous study where various configurations were evaluated. Among the analyzed configurations, the one presented in Figure 6 was found to produce the best results, and was thus the chosen model. It consists of a neural network formed by an input layer composed of displacement amplitudes calculated for a given time instant for each of the 60 monitored points of the beam (vector **y**), followed by five convolutional layers, a flattened layer and four fully connected layers, leading to an output layer with 60 neurons, respectively associated with the values of $D_{\mathrm{EI}}(x)$ at each position x of the beam. Finally, the CNN contains a total of 874,188 learnable parameters.

Batch normalization and Rectified Linear Unit (ReLU) activation functions were applied to the outputs of each of the convolutional and fully connected layers, except for the last fully connected layer, where

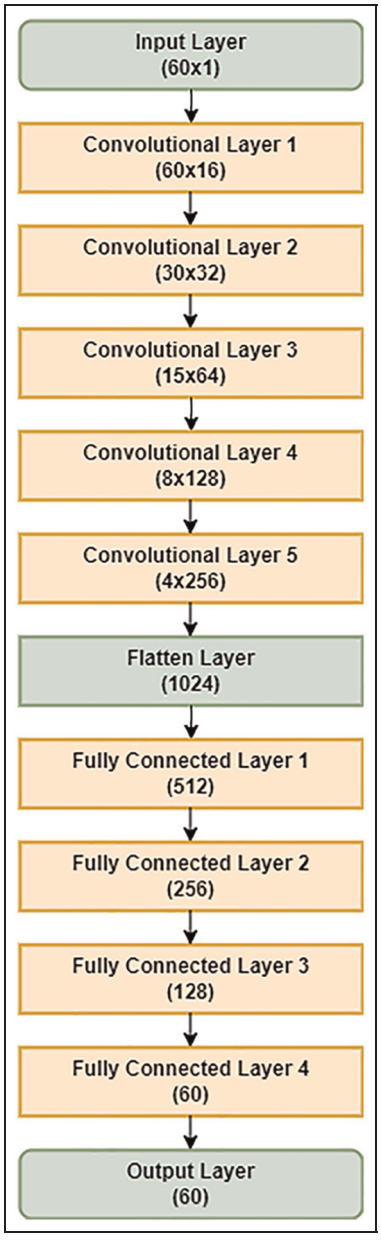


Figure 6. Architecture of the applied CNN.
CNN: convolutional neural network.

Table 2. Description of convolutional layers.

Layer	Activations	Padding	Stride	Kernel length
#1	60 × 16	Yes	1	4
#2	30 × 32	Yes	2	4
#3	15 × 64	Yes	2	4
#4	8 × 128	Yes	2	4
#5	4 × 256	Yes	2	4

batch normalization was not applied. In each convolutional layer, symmetric padding was added to both sides of the input vector. This was done to make sure that the output vectors of the layer would have the same length as the input if the kernels had a stride of one. However, to reduce dimensionality, a stride value of two was used in all layers except for the first. Figure 6 presents a scheme of the applied CNN, where the dimensions of the activations are shown. Tables 2 and 3 summarize the convolutional layers and hyperparameters, respectively. The CNN layers and hyperparameters were set after an extensive grid search analysis using Matlab® deep learning toolbox.

Table 3. Description of CNN hyperparameters.

Hyperparameter	Attribute
Gradient decay factor	0.9000
Squared gradient decay factor	0.9990
Epsilon	1.00E − 04
Initial learn rate	1.00E + 01
Learn rate schedule	Piecewise
Learn rate drop factor	0.3162
Learn rate drop period	25
L2 regularization	1.00E + 01
Gradient threshold method	l2 norm
Max epochs	100
Mini batch size	256
Shuffle	Every-epoch
Execution environment	CPU
Batch normalization statistics	Population
Output network	Last iteration

CNN: convolutional neural network.

Training/testing procedure

A 10-fold cross-validation scheme[31] was adopted in this work. It is a common technique in machine learning and statistical modeling to assess the performance of a model and estimate its accuracy. It involves dividing a dataset into 10 equally sized "folds" or subsets and then training and evaluating the model 10 times, each time using a different fold as the test dataset and the remaining nine folds as the training dataset. The 10-fold cross-validation works as follows:

1. *Dataset preparation*: The original dataset is randomly divided into 10 equal-sized folds. This procedure guarantees that each fold has equal distribution of data points and maintains the original data randomness.
2. *Model Training*: The model is then trained with nine folds (i.e., the training dataset) while holding one fold out for evaluation (i.e., the testing set). Training consists in optimizing the model parameters based on the training data.

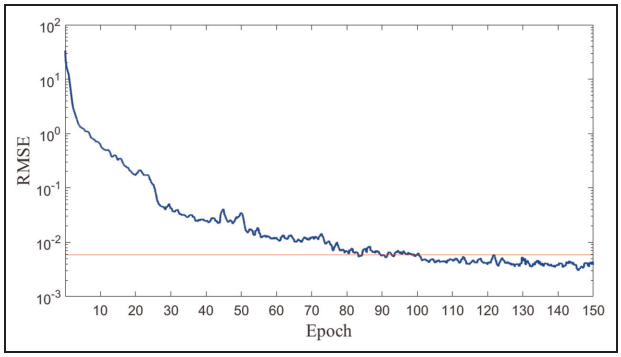


Figure 7. Convergence analysis of the root mean squared error (RMSE) through training.

3. *Model Evaluation*: The trained model is then evaluated on the held-out fold (i.e., the test dataset) to assess its performance.
4. *Fold Rotation*: Steps 2 and 3 are repeated 10 times, with each of the 10 folds used once as the test dataset and the other nine as the training dataset. This process ensures that each data point in the original dataset is used for training and testing and that the model is evaluated on different subsets of data.
5. *Performance Metrics Calculation*: After all 10 iterations, the performance metrics of the model on the test datasets are averaged to obtain an overall performance estimate for the model.

The advantage of 10-fold cross-validation is that it provides a robust estimate of the model performance using multiple train-test splits, which helps in reducing the impact of randomness and data variability. It also ensures that the entire dataset is utilized for both training and testing, making it more reliable for model evaluation when compared to other techniques.

The CNN models employed in this study were trained utilizing the Adam optimizer,[32] with the objective function (OF) being the mean squared error between the predicted $D_{\mathrm{EI}}(x)$ profile and the actual profile, that is:

$$\mathrm{OF} = \frac{1}{60}\sum_{k=1}^{60}\left(D_{\mathrm{EI}}(x_k) - D_{\mathrm{EI\,pred}}(x_k)\right)^2 \qquad (3)$$

where $[\,D_{\mathrm{EI\,pred}}(x_1) \quad \cdots \quad D_{\mathrm{EI\,pred}}(x_{60})\,]$ is the output vector of the CNN.

A convergence analysis was performed to determine the necessary number of epochs. Figure 7 presents a typical response obtained for the mean squared error as a function of the number of epochs for the first test fold (similar results are obtained for the other nine folds), showing that the mean squared error practically stabilizes after 100 epochs. In fact, the reductions of the root mean squared error from the 1st to the 100th epoch, and from the 1st to the 150th are 99,983% and 99,988%, respectively. Therefore, to avoid increasing the computational time required for the training without a significant increment in the mean squared error, a limit of 100 epochs was chosen for all the evaluations. Training data were divided in mini-batches of 256 samples that were shuffled at the end of every epoch. The initial learn rate was 0.001, but this value was programmed to be reduced by a factor of $\sqrt{0.1}$ at every 25 epochs. This is done to prevent the model from taking steps that are too large when the error is already low.

Results

The primary aim of this work is to evaluate if the proposed methodology accurately classifies the eight damage scenarios of the tested beams, pointing out the location and extent of structural damage using the CNN. In that way, out of a total of 1350 (deformed configurations) $\times$ 8 (damage scenarios) = 10,800 input data samples, 9720 were designated for training purposes, while the remaining 1080 were allocated for validation at each fold.

An assessment procedure is carried out for the position situated 100 mm from the support (i.e., monitored position 53 of the tested beam $(x_{\mathrm{d}} = 0.7\,\mathrm{m})$—refer to Figure 1), and the same analysis is similarly applied to the remaining monitored points. According to Table 1, the damage indexes for position 53 vary from 1.000 (healthy structure—scenario #1) to 8.621 (scenario #8). Figure 8 presents the results obtained using the CNN for all the validation data across all the folds and all the damage scenarios associated with position 53. These results indicate that the point clouds obtained for the validation data are in reasonably good agreement with the damage index values for position 53 across the eight damage scenarios.

By executing the same procedure for each monitored point, it is possible to summarize the results and obtain Figure 9. This figure represents the damage levels identified along the beam for each evaluated scenario. The figure shows the median predictions for each point, compared to their respective actual values. The presented average values of the medians of the damage indices for all folds at each damage state in comparison with respective actual values confirm the good classification performance of the CNN.

Table 4 presents the mean values obtained for all folds at the regions where damage was actually caused, with their respective standard deviations. The obtained values for the points located in undamaged regions

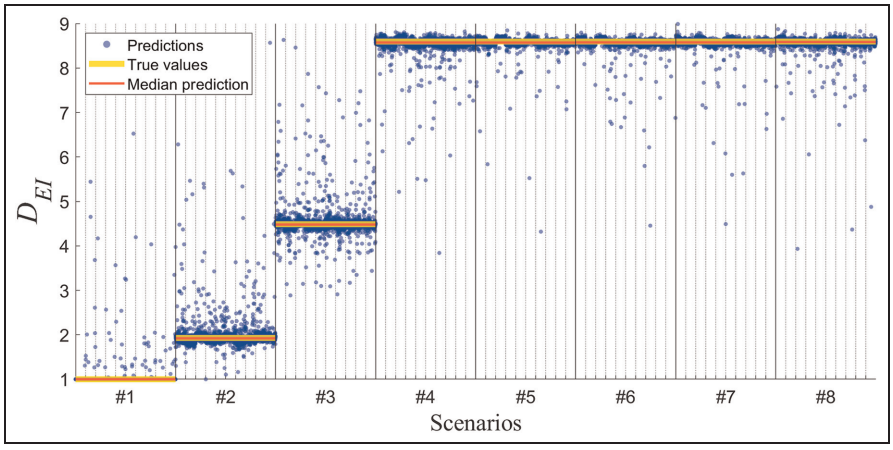


Figure 8. $D_{EI}(x)$ for position 53 ($x_d = 0.7$ m) at each validation fold and damage scenario. Thinner vertical lines for a given scenario separate validation data from each of the 10 folds.

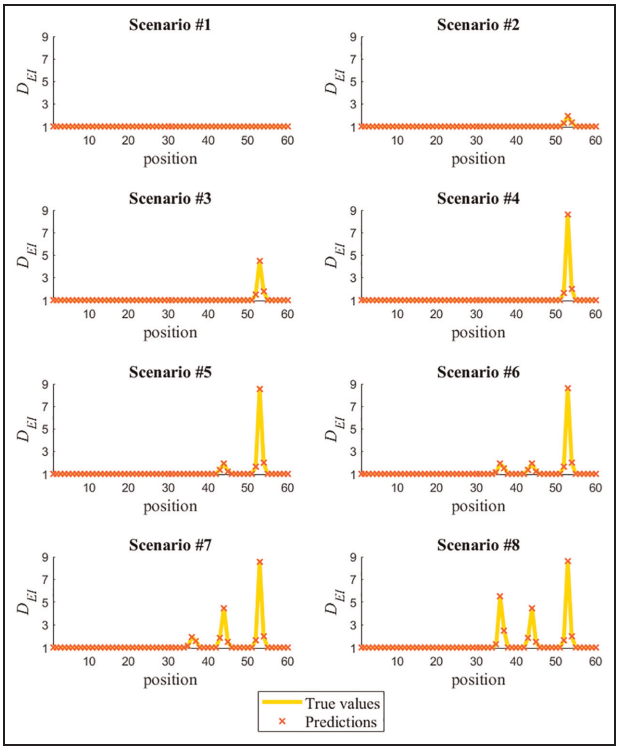


Figure 9. Comparison of actual values of $D_{EI}(x)$ and respective median predictions, constructed with validation data from all scenarios.

exhibited outstanding concurrence with the corresponding target values (unitary values). Due to the substantial number of points demonstrating these characteristics, and considering that including these results would considerably expand the size of Table 4 without contributing with pertinent information for the analyses, these findings have been omitted. Similarly, the findings for the points near the crack regions displayed remarkably similar behaviors to those observed in the corresponding points where the cracks were intentionally induced. Therefore, these results have also been excluded from Table 4. Hence, based on the results presented in Table 4, it can be affirmed that the proposed methodology not only detected the presence of damages in the analyzed beam, but also accurately located and quantified these data. The coefficient of variation (CoV), used as a measure of the relative error and calculated as $100\%\,\sigma/\mu$, resulted lower than 2% in all the cases (scenarios and positions). The largest relative error occurred, as could be expected, when only one small crack was present in the beam specimen (position 53 in scenario #2).

The training step takes around 452 s (roughly 7.5 min) on a Desktop (12th Gen Intel Core i7-1255U 1.70 GHz; 16 GB RAM; 512 GB SSD; made in Brazil). With today's CPU/GPU (Graphics Processing Unit) computational capabilities, associated with parallel computing schemes, there should be no problem in applying this methodology to more complex structures. Please note that the methodology itself does not depend on specific boundary conditions (although the training might), allowing it to be applied independently to individual beams or columns within a macrostructure. As a result, the dimensionality increases (roughly) linearly with the size of the monitored region. However, the size of the training dataset could increase faster to encompass interaction between beams or columns. From a computational perspective, the most relevant aspect is that a CNN could be assigned to each beam or column and trained in parallel.

Median of predictions as ensemble algorithm

A very simple ensemble algorithm is proposed in this work to combine a number of predictors to yield a

Table 4. Comparisons of mean values of $D_{EI}(x)$ obtained from all folds with actual values, including coefficients of variation.

Scenario	Position 36 ($x_d = 0.5$ m)	Target	Position 44 ($x_d = 0.6$ m)	Target	Position 53 ($x_d = 0.7$ m)	Target
	$\mu \pm \sigma$ (%CoV)		$\mu \pm \sigma$ (%CoV)		$\mu \pm \sigma$ (%CoV)	
#1	1.000 ± 0.000 (0.000)	1.000	1.000 ± 0.000 (0.000)	1.000	1.000 ± 0.000 (0.000)	1.000
#2	1.000 ± 0.000 (0.000)	1.000	1.000 ± 0.000 (0.000)	1.000	1.916 ± 0.033 (1.722)	1.930
#3	1.000 ± 0.000 (0.000)	1.000	1.000 ± 0.000 (0.000)	1.000	4.480 ± 0.026 (0.580)	4.487
#4	1.000 ± 0.000 (0.000)	1.000	1.000 ± 0.000 (0.000)	1.000	8.579 ± 0.035 (0.409)	8.595
#5	1.000 ± 0.000 (0.000)	1.000	1.927 ± 0.010 (0.519)	1.930	8.580 ± 0.036 (0.419)	8.595
#6	1.926 ± 0.013 (0.673)	1.930	1.927 ± 0.009 (0.466)	1.930	8.581 ± 0.032 (0.375)	8.595
#7	1.926 ± 0.015 (0.780)	1.930	4.481 ± 0.027 (0.603)	4.487	8.580 ± 0.024 (0.280)	8.595
#8	5.485 ± 0.024 (0.436)	5.488	4.484 ± 0.017 (0.379)	4.487	8.589 ± 0.022 (0.258)	8.595

CoV: coefficient of variation.

more accurate and robust global prediction. The proposal is to use predictions of the same method (the developed CNN), fed with different samples, as weak predictors.

A number of sophisticated ensemble algorithms have been proposed and implemented in practice along the years. However, the basic idea is that ensemble averaging of weak predictors can improve the performance of single predictors and add robustness. This occurs when the errors made by different predictors are independent, and thus the ensemble average reduces the variance portion of the error.[21] The sophistication of the ensemble algorithms is usually intended to increase the error independence, even when the number of weak predictors is small. This is not the case of the methodology proposed in the present paper, where the number of predictions can be huge for the reasons exposed in the introduction section.

Figure 8 shows that the predictions (blue dots) of the CNN developed in the present work have some dispersion around the actual values (yellow line). The vast majority of them are close to the actual values (see CoVs in Table 4), but some of them are certainly far. These anomalous cases can occur in practice and are likely to be particularly problematic if they trigger incorrect actions on the structure under test. An interesting feature that must be seen in Figure 8 is that the dispersion is not symmetric. For example, in scenarios with severe damage, the point clouds are biased to lower values of the damage index. This asymmetry makes the mean not the best average option as ensemble. Besides, the mean is sensitive to highly biased outliers, which certainly can occur as can be seen in Figure 8.

On the other hand, Figure 8 also shows that the median of all the predictions of each damage scenario (red line) is practically identical to the actual values (yellow line). Therefore, the median of predictions is chosen as ensemble. The remaining question is: how many predictions are needed to have a robust ensemble? This is answered by calculating the median for an increasing number of samples, each yielding a prediction, for each damage scenario and for each position on the beam specimen. The respective charts with ensemble algorithm predictions for positions 36, 45, and 53 are shown in Figure 10. For easy comparison, this figure shows the relative errors between the median of predictions and the respective actual value of $D_{EI}(x)$ for different numbers of samples. Moreover, no significant errors were detected for the remaining points of the beam

Figure 10 shows that in all scenarios, the median of predictions converged with a relative error lower than 1%, with less than 1500 samples. Excluding the most challenging scenario (#2), that is, the one with a single small crack, the convergence occurred with less than 500 samples. Note that each sample corresponds to a single video frame and it yields a single prediction, which is then averaged using the median. Therefore, 1500 samples simply means 1.5 s of *not necessarily continuous or ordered* video recording during free vibration. This can be easily achieved in many practical applications.

As a final remark, it is worth recalling that the loss of time continuity or time order of samples is problematic in classical temporal-pattern-based CNNs for SHM. In the proposed methodology, such a loss is beneficial because it is likely to increase the error independence between predictions and, therefore, to improve the ensemble performance.

Robustness assessment

A robustness test was conducted to evaluate the efficiency of the presented methodology in identifying damage in novel scenarios. This test involved training the CNN while excluding damage scenarios #5 and #6 from the training dataset. Subsequently, the trained CNN was subjected to data related to the eight analyzed scenarios, including those not considered in the

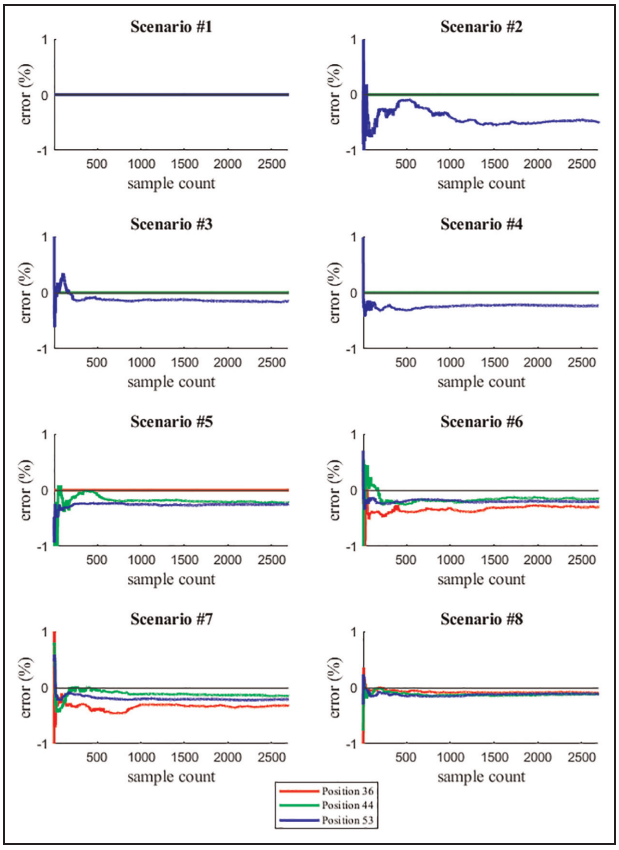


Figure 10. Performance of the median of predictions as ensemble algorithm for positions 36, 44, and 53. This is the relative error between the median of predictions and the respective actual values of $D_{EI}(x)$ for different sample counts.

training phase. This evaluation aimed to assess the ability of the methodology to generalize and accurately detect damage in unseen scenarios, thus indicating its robustness and reliability in situations closer to real-world applications.

Similarly to Figure 8, the results for position 53 ($x_d = 0.7$ m) are presented in Figure 11 but now for the robustness assessment. It can be observed that despite a noticeably higher dispersion of results for the damage scenarios not seen in the training phase, the proposed methodology accurately detected the damages at this position for the eight damage scenarios. In particular, it can be seen that the median of the predictions is slightly under the actual value of $D_{EI}(x_{53})$, that is, the model slightly underestimates the damage at position 53.

By following a similar procedure as before, Figure 12 was generated, showcasing the identified damage levels along the beam for each evaluated scenario. The figure presents the median predictions for each point compared to their actual values. As expected, the performance of the proposed methodology decreased, particularly for scenarios #5 and #6, which were not included in the training phase. Despite this, all damage scenarios were accurately detected and localized. In terms of damage quantification, more pronounced differences are observed at position 45, as well as their respective neighborhoods. At this location, damage was underestimated in scenario #5 and overestimated in scenario #6. However, the predictions are close to the actual values.

It is worth noting that scenario #5 is special in the sense that is the only one in which there are 2 cracks (the remaining have 0, 1, or 3 cracks). The prediction performance that Figure 12 shows for this scenario is very good in regards to detection and localization. This is important because it evidences that the CNN is not simply classifying the damage scenarios by learning one global spatial pattern for each scenario. On the contrary, the CNN is identifying damage by learning local spatial patterns related to local damage. If, in the prediction phase, the CNN had yielded a False Positive at position 36 or a False Negative at position 44, the

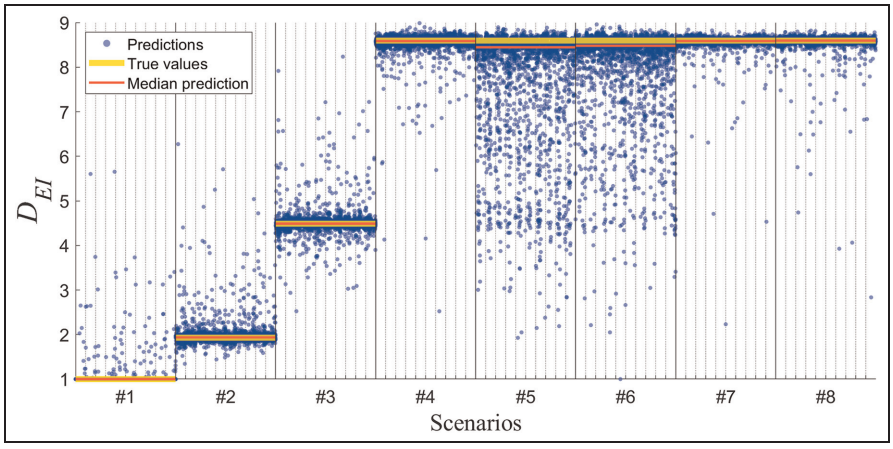


Figure 11. $D_{EI}(x)$ for position 53 ($x_d = 0.7$ m) at each validation fold and damage scenario for the robustness assessment. Thinner vertical lines for a given scenario separate validation data from each of the 10 folds.

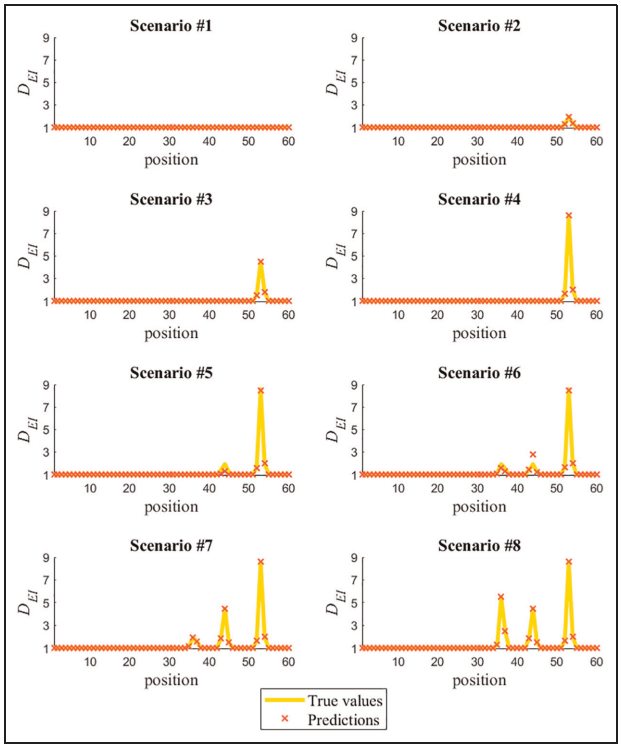


Figure 12. Comparison of actual values of $D_{EI}(x)$ and respective median predictions, constructed for the robustness assessment.

methodology would have to be limited only to damage scenarios observed in training. This would have drastically restricted the fields of application of the proposed methodology; but it was not the case.

Table 5 compares the predicted values of $D_{EI}(x)$ for the positions on the beam where cracks were induced, with their respective standard deviations, target values, and coefficients of variation, for the robustness assessment. These results confirm those shown in Figure 11, where a higher dispersion of predictions and a decrease in the performance evaluation of $D_{EI}(x)$ values can be observed, especially for the cases not used in the network training. Despite this, it can be noted that the proposed methodology exhibited promising results, as it demonstrated excellent performance for the cases utilized in the training process and reasonable performance for the remaining scenarios. The relative error in damage quantification was under the 20% even in the special case of the unseen scenario #5 at positions 36 and 44.

Conclusions

This article presented a comprehensive analysis of damage identification in a beam. Various aspects ranging from experimental tests to direct damage identification were addressed. For this purpose, a cantilever beam was tested in the laboratory and the results were used to evaluate a methodology for damage identification based on CNNs. The tested beam was subjected to eight scenarios of induced damage, and its dynamic responses were captured through image processing under free-vibration, allowing for the monitoring of transverse displacements at 60 points along its length.

Instead of using temporal histories as input data for the CNN, as commonly done in related works, this work introduced the deformed configuration of the beam obtained from the measured displacements at each of the 60 monitored positions for a single time instant. When trained with data from all the analyzed damage scenarios, the CNN was able to correctly identify, locate, and quantify the induced damage (CoV $< 2\%$).

A robustness assessment was conducted where the CNN was trained with six out of the eight damage scenarios and validated with data from all eight tested scenarios. For this situation, the proposed methodology correctly detected and located the damage, but with

Table 5. Comparisons of mean values of $D_{EI}(x)$ obtained from all folds with actual values for the robustness assessment.

Scenario	Position 36		Position 44		Position 53	
	$\mu \pm \sigma$ (%CoV)	Target	$\mu \pm \sigma$ (%CoV)	Target	$\mu \pm \sigma$ (%CoV)	Target
#1	1.000 ± 0.000 (0.000)	1.000	1.000 ± 0.000 (0.000)	1.000	1.000 ± 0.000 (0.000)	1.000
#2	1.000 ± 0.000 (0.000)	1.000	1.000 ± 0.000 (0.000)	1.000	1.934 ± 0.022 (1.136)	1.930
#3	1.000 ± 0.000 (0.000)	1.000	1.000 ± 0.000 (0.000)	1.000	4.490 ± 0.014 (0.312)	4.487
#4	1.000 ± 0.000 (0.000)	1.000	1.000 ± 0.000 (0.000)	1.000	8.586 ± 0.021 (0.245)	8.595
#5	1.010 ± 0.024 (2.376)	1.000	1.303 ± 0.222 (17.04)	1.930	8.442 ± 0.046 (0.544)	8.595
#6	1.524 ± 0.192 (12.59)	1.930	2.790 ± 0.243 (8.704)	1.930	8.482 ± 0.028 (0.329)	8.595
#7	1.934 ± 0.013 (0.671)	1.930	4.487 ± 0.012 (0.268)	4.487	8.588 ± 0.016 (0.186)	8.595
#8	5.498 ± 0.019 (0.346)	5.488	4.486 ± 0.011 (0.245)	4.487	8.593 ± 0.018 (0.209)	8.595

CoV: coefficient of variation.

less quantification precision (CoV $< 20\%$). The results obtained in this article indicate that the proposed methodology has good potential for real-life applications, where CNNs can be trained with various damage scenarios to which the structure may be subjected. Furthermore, the results demonstrate that the CNN was able to offer reasonable damage prediction even when subjected to scenarios that were not used during their training.

The main finding of the present work is that damage can be detected, located and quantified in a beam using a data-driven model that is fed with snapshots taken during free vibration. Moreover, using *instantaneous displacement field* instead of *displacement time history* as the model input, has important practical advantages:

- There is no need for data time continuity or time ordering.
- Spatial-continuity, -ordering, and -positioning are easily achieved when displacements are obtained from video.
- Free-vibration records are statistically easier to be obtained, since they can be shorter in duration.
- The number of samples corresponding to a damage scenario, and consequently the number of predictions, can be huge. This enables the use of a very simple ensemble method, such as the median of predictions.

The proposed approach could be applied to specific beams or columns of a macrostructure independently. In this way, instrumentation and CNN architecture remain invariant. However, training should be done for each beam or column under testing and, likely, hyperparameters optimized also independently. Special care should be taken in the preprocessing stage, which may involve the removal of rigid body motions. The necessity of this step depends on the camera installation, whether it is inside or outside the macrostructure.

Finally, it can be said that using a data-driven model has well-known practical advantages. For example, there is no need for physically identifying the beam parameters (E, I, length) or its boundary conditions. However, when this is combined with the extraction of structural responses from video, there is an additional advantage: the model can compensate *systematic errors*, for example, minor camera misalignment and other optical distortions. This can happen given that the training dataset is statistically rich, and the model has been trained with the same optical setup and it is complex enough. When comparing the experimental setup with the excellent results, it seems that such a compensation has occurred in the present work. However, more in-depth investigations are needed to fully elucidate this error-compensation mechanism.

The most important limitation of the proposed methodology is inherent to any data-driven method, which is that the more likely damage scenarios should be known a priori and induced in a specimen for generating a rich training dataset. This dataset could be enhanced with synthetic data through numerical simulations. However, the methodology is still based on known damage scenarios. On the contrary, physics-based methods are more general; for example, the method presented in Garrido et al.[23] only depends on damage sparsity. However, physics-based methods can be susceptible to systematic errors in instrumentation, which are easily and automatically handled by data-driven methods as the one proposed in the present paper.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: The authors would like to thank the financial support of: Argentine National Research Council (CONICET); National University of Cuyo; Universidade Federal de Juiz de Fora—Brazil; Conselho Nacional de Desenvolvimento Cientfico e Tecnológico—CNPq (Brazil), grants CNPq/FNDCT/MCTI 407256/2022-9, 303982/2022-5, 308008/2021-9; and Fundação de Amparo à Pesquisa do Estado de Minas Gerais—FAPEMIG, grants BPD 00080-22, TEC PPM 00106-17, and TEC PPM-00001-18.

ORCID iDs

Rafaelle Finotti https://orcid.org/0000-0002-8569-0041
Hernán Garrido https://orcid.org/0000-0002-8840-9390

Note

1. Weighted fitting is advantageous because image columns where the beam and the background are of similar intensities are penalized. On the contrary, zones of high contrast are emphasized.

References

1. Avci O, Abdeljaber O, Kiranyaz S, et al. A review of vibration-based damage detection in civil structures: from traditional methods to machine learning and deep learning applications. *Mech Syst Signal Process* 2021; 147: 107077.
2. de Almeida Cardoso R, Cury A and Barbosa F. A clustering-based strategy for automated structural modal identification. *Struct Health Monit* 2018; 17(2): 201–217.
3. Pepi C, Cavalagli N, Gusella V, et al. Damage detection via modal analysis of masonry structures using shaking table tests. *Earthquake Eng Struct Dyn* 2021; 50(8): 2077–2097.

4. Gillich GR, Maia NMM, Wahab MA, et al. Damage detection on a beam with multiple cracks: a simplified method based on relative frequency shifts. *Sensors* 2021; 21(15): 5215.
5. Tapeh ATG and Naser MZ. Artificial intelligence, machine learning, and deep learning in structural engineering: a scientometrics review of trends and best practices. *Arch Computat Methods Eng* 2023; 30(1): 115–159.
6. Nunes LA, Amaral RPF, de Souza Barbosa F, et al. A hybrid learning strategy for structural damage detection. *Struct Health Monit* 2021; 20(4): 2143–2160.
7. Nick H, Ashrafpoor A and Aziminejad A. Damage identification in steel frames using dual-criteria vibration-based damage detection method and artificial neural network. *Structures* 2023; 51: 1833–1851.
8. Finotti RP, de Souza Barbosa F, Cury AA, et al. Numerical and experimental evaluation of structural changes using sparse auto-encoders and SVM applied to dynamic responses. *Appl Sci* 2021; 11(24): 11965.
9. Finotti RP, Gentile C, Barbosa F, et al. Structural novelty detection based on sparse autoencoders and control charts. *Struct Eng Mech* 2022; 81(5): 647–664.
10. Chen Z, Wang Y, Wu J, et al. Sensor data-driven structural damage detection based on deep convolutional neural networks and continuous wavelet transform. *Appl Intell* 2021; 51: 5598–5609.
11. Pan Q, Bao Y and Li H. Transfer learning-based data anomaly detection for structural health monitoring. *Struct Health Monit*. Epub ahead of print 6 january 2023. DOI:10.1177/14759217221142174.
12. Gu J, Wang Z, Kuen J, et al. Recent advances in convolutional neural networks. *Pattern Recognit* 2018; 77: 354–377.
13. Sony S, Dunphy K, Sadhu A, et al. A systematic review of convolutional neural network-based structural condition assessment techniques. *Eng Struct* 2021; 226: 111347.
14. Mantawy IM and Mantawy MO. Convolutional neural network based structural health monitoring for rocking bridge system by encoding time-series into images. *Struct Control Health Monit* 2022; 29(3): e2897.
15. Pamuncak A, Zivanovic S, Adha A, et al. Correlation-based damage detection method using convolutional neural network for civil infrastructure. *Comput Struct* 2023; 282: 107034.
16. Li S, Jin L, Qiu Y, et al. Signal anomaly detection of bridge SHM system based on two-stage deep convolutional neural networks. *Struct Eng Int* 2023; 33(1): 74–83.
17. Hassani S and Dackermann U. A systematic review of optimization algorithms for structural health monitoring and optimal sensor placement. *Sensors* 2023; 23(6): 3293.
18. Feng D and Feng MQ. Computer vision for SHM of civil infrastructure: from dynamic response measurement to damage detection: a review. *Eng Struct* 2018; 156: 105–117.
19. Garrido H, Curadelli O and Ambrosini D. A heuristic approach to output-only system identification under transient excitation. *Expert Syst Appl* 2017; 68: 11–20.
20. Garrido H, Curadelli O and Ambrosini D. A new method for the identification of systems under transient excitations only from their response. In: Giusti S, Pucheta M, and Storti M (eds.) *XXII Congreso sobre Mètodos Numèricos y sus Aplicaciones—ENIEF 2016*. Córdoba, Argentina: Asociación Argentina de Mecánica Computacional, pp.1–23 (In Spanish).
21. Avnimelech R and Intrator N. Boosting regression estimators. *Neural Comput* 1999; 11(2): 499–520.
22. Sohn H. Effects of environmental and operational variability on structural health monitoring. *Philos Trans A Math Phys Eng Sci* 2007; 365(1851): 539–560.
23. Garrido H, Domizio M, Curadelli O, et al. Synthesis of healthy-structure model responses for damage quantification. *Struct Health Monit* 2023; 22(1): 689–713.
24. Deraemaeker A, Reynders E, De Roeck G, et al. Vibration-based structural health monitoring using output-only measurements under changing environment. *Mech Syst Signal Process* 2008; 22(1): 34–56.
25. Tondreau G and Deraemaeker A. Local modal filters for automated data-based damage localization using ambient vibrations. *Mech Syst Signal Process* 2013; 39(1–2): 162–180.
26. Kiranyaz S, Avci O, Abdeljaber O, et al. 1D convolutional neural networks and applications: a survey. *Mech Syst Signal Process* 2021; 151: 107398.
27. Bishop CM and Nasrabadi NM. *Pattern recognition and machine learning*, vol. 4. New York: Springer, 2006, p.738.
28. Goodfellow I, Bengio Y and Courville A. *Deep learning*. Cambridge, MA: MIT Press, 2016.
29. Bhatt D, Patel C, Talsania H, et al. CNN variants for computer vision: history, architecture, application, challenges, and future scope. *Electronics* 2021; 10(20): 2470.
30. Garrido H, Domizio M, Curadelli O, et al. Numerical, statistical, and experimental investigation on damage quantification in beams from modal curvature. *J Sound Vib* 2020; 485: 115591.
31. Kohavi R, et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: *Proceedings of the 14th international joint conference on Artificial intelligence: volume 2*. Montreal, Canada: IJCAI Organization, pp.1137–1145.
32. Kingma DP and Ba J. Adam: a method for stochastic optimization. *arXiv:1412.6980*, 2017.