

Article

Vibration-Response-Only Structural Health Monitoring for Offshore Wind Turbine Jacket Foundations via Convolutional Neural Networks

Bryan Puruncajas^{1,2} , Yolanda Vidal^{1,*} , Christian Tutivén² 

¹ Control, Modeling, Identification and Applications (CoDALab), Department of Mathematics, Escola d'Enginyeria de Barcelona Est (EEBE), Universitat Politècnica de Catalunya (UPC), Campus Diagonal-Besós (CDB), Eduard Maristany, 16, 08019 Barcelona, Spain.

² Mechatronics Engineering, Faculty of Mechanical Engineering and Production Science (FIMCP), Escuela Superior Politécnica del Litoral (ESPOL), Guayaquil, Ecuador.

* Correspondence: yolanda.vidal@upc.edu; Tel.: +34-934-137-309

Version June 15, 2020 submitted to Sensors

Abstract: This work deals with structural health monitoring for jacket-type foundations of offshore wind turbines. In particular, a vibration-response-only methodology is proposed based on accelerometer data and deep convolutional neural networks. The main contribution of this article is twofold: i) a signal-to-image conversion of the accelerometer data into gray scale multi-channel images with as many channels as the number of sensors in the condition monitoring system, and ii) a data augmentation strategy to diminish the test set error of the deep convolutional neural network used to classify the images. The performance of the proposed method is analysed using real measurements from a steel jacket-type offshore wind turbine laboratory experiment undergoing different damage scenarios. The results, with a classification accuracy over 99%, demonstrate that the stated methodology is promising to be utilised for damage detection and identification in jacket-type support structures.

Keywords: structural health monitoring; damage detection; damage identification; offshore wind turbine foundation; jacket; signal-to-image conversion; convolutional neural network

1. Introduction

Globally, wind power generation capacity has increased exponentially since the early 1990s, and as of the end of 2019, this capacity amounted to 650 GW [1]. Whereas onshore wind turbines (WTs) have dominated new wind installations during the past, the growth of offshore WTs is poised to become the new leader, because of steadier wind, in addition to vast regions where its installation is possible. In regard to the global offshore market, the cumulative installations have now reached 23 GW, representing 4% of total cumulative installations. Unfortunately, offshore WTs are placed in a harsh environment that originates from the wind and the sea conditions [2]. As a consequence, offshore WTs require rigorous safety measures because it is extremely complicated to do operation and corrective work on these huge WTs placed in remote locations. Given that approaches centered on enhancing component reliability are likely to increase capital expenditures, instead system design optimization research and development activities should focus on minimizing and, if possible, even eliminating unexpected failures. In other words, the wind industry must abandon corrective maintenance (remedy failures) and move toward predictive maintenance (repair immediately before failure) to achieve maximum availability. Thus, the development of a structural health monitoring (SHM) strategy is particularly necessary to achieve this goal.

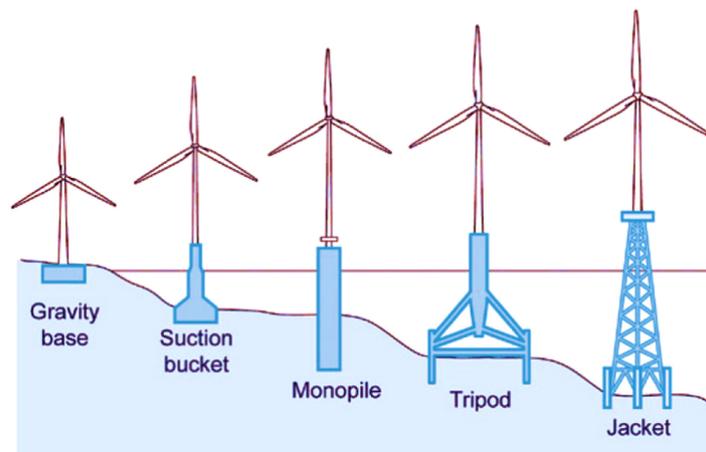


Figure 1. Fixed bottom wind turbine foundations [3].

30 Onshore and offshore fixed WTs differ mainly in the structure of their respective foundations.
 31 Several types of offshore foundations are used, with foundation choice depending on the water depth.
 32 The most common foundations are shown in Figure 1, see [3]. Note that jacket foundations, which
 33 are the object of study of this work, are preferred for greater depths (usually, between 30 to 90 m).
 34 The detection of early-stage damage in the foundation is of great importance to avoid the possible
 35 future collapse of the entire structure. As stated in “Long-term research challenges in wind energy – a
 36 research agenda by the European Academy of Wind Energy” [4]:

37 A defining marine environment main characteristic is that structures are always subject to
 38 excitations. Techniques for structural health monitoring, vibration and data analysis must
 39 be capable of coping with such ambient excitations. As the input is typically not known, a
 40 normal input-output formalism cannot be used.

41 Thus, to overcome this challenge, which is posed by the fact that the input is typically not known,
 42 in this work a structural health monitoring strategy for jacket-type foundations is developed based
 43 on a vibration response-only methodology. This is a challenge by itself as many of the works in the
 44 SHM field are based on the principle of guided elastic waves with a given (known) input excitation.
 45 See, for example, the overview of SHM systems for various WT components presented by Liu et al.
 46 [5]. In contrast, in this work, a new paradigm is introduced in which a predetermined excitation
 47 in the structure is no longer forced, but rather, the incident wind and waves serve as the exciting
 48 forces in the structure. In this way, the classic pattern recognition paradigm with identical excitation,
 49 e.g. [6], becomes a problem of pattern recognition with variable excitation. Consequently, the new
 50 paradigm implies greater complexity in the damage detection process. Furthermore, until recently,
 51 few contributions have targeted offshore WT foundations. Notably, work by Weijtjens et al. [7] was
 52 focused on a real WT foundation and contributed an SHM strategy based on the resonance frequencies
 53 of the foundation. However, the results only proved some increased stiffness of the structure but
 54 could not give a final diagnose about damage detection. Similarly, Oliveira et al. [8] introduced the
 55 main aspects in the development of a vibration-based monitoring system for an onshore 2.0 MW wind
 56 turbine based on identification of the modal properties of the most important vibration modes, in
 57 which detailed attention was given to the statistical procedure based on regression models, used to
 58 minimize the influence of operational and environmental effects over the features considered to detect
 59 structural changes in the WT. However only damage detection was pursued with a single type of
 60 damage. It is also noteworthy the work by Zugasti [9] that used damage estimators to detect damage
 61 in an experimental offshore tower similar to that employed in this work. Nevertheless, only damage
 62 detection was attained. In this work, in contrast to the aforementioned references, several types of
 63 damage are studied and not only damage detection but also its classification is achieved.

64 It is important to note that the SHM standard approach for the problem at hand is usually an
65 unsupervised one. That is, as no-one would purposely damage their assets to train a SHM tool, only
66 healthy data from the real structure is used. However is unfeasible to correctly identify different
67 damage states using solely data obtained during what is assumed to be a healthy state. In this
68 framework, detection can be accomplished by using a model of normality or unsupervised models,
69 but not classification on the type of damage. The approach proposed in this work is the opposite, that
70 is: a supervised approach. Thus data from the damaged structure is required to train the model. In
71 practice, this will be accomplished by means of computer models, as the finite element method (FEM).
72 The FEM model should be validated with a down-scaled experimental tower (as the one proposed in
73 this work). Then the full-scale finite element model would be used to generate healthy (to validate
74 with the real asset) and damage samples. Finally, the stated supervised methodology proposed in this
75 work can be used. In this work, a satisfactory experimental proof of concept has been conducted with
76 the proposed strategy and a laboratory down-scaled WT. However, future work is needed to validate
77 the technology in a full-scale and more realistic environment. Some examples of this type of approach
78 are given in [10], where bridge damage detection is accomplished by a neural network considering
79 errors in baseline finite element models, and [11] where the stated SHM method for an oil offshore
80 structure is capable to cope with several types of damage based on a finite element model.

81 On the one hand, it has been shown that traditional machine learning requires complex feature
82 extraction processes and specialized knowledge, especially for a complex problem such as WT
83 condition monitoring [12–14]. Moreover, extracting features with classic machine learning methods
84 faces the classic bias-variance dilemma from inference theory. The bias-variance trade-off implies that a
85 model should balance under-fitting and over-fitting; that is, the model should be rich enough to express
86 underlying structure in the data but simple enough to avoid fitting spurious patterns, respectively.
87 On the other hand, in the modern practice of deep learning, very rich models are trained to precisely
88 fit (i.e., interpolate) the data. Classically, such models would be considered over-fit, and yet they
89 often obtain high accuracy on test data. Thus, this paper proposes to use deep convolutional neural
90 networks (CNN) for pattern recognition (classification), avoiding the aforementioned usual problems
91 in the literature, e.g. [12–14], related to feature extraction and bias-variance trade-off. In particular,
92 we develop a novel damage diagnosis method for WT offshore foundations based on transforming
93 condition monitoring multi-vibration-signals into images (with as many channels as sensors) to be
94 processed afterward using deep CNN.

95 The paper is organized in the following manner. First, in Section 2, the experimental set-up is
96 introduced. It consists in a steel jacket-type offshore WT laboratory structure undergoing different
97 damage scenarios. Then, in Section 3, the proposed SHM strategy is described in detail. The approach
98 can be summarized by the following steps: i) accelerometer data is gathered, ii) a pre-process is
99 designed to extract the maximum amount of information and to obtain a dataset of 24 (that is, the
100 same number as accelerometer sensors) channel gray-scale images, iii) 24-channel-input deep CNN
101 is designed and trained for classification of the different structural states. In Section 4, the obtained
102 results are conferred, showing an exceptional performance, with all considered metrics giving results
103 greater than 99%. Lastly, the main conclusions are given in Section 5 as well as future work research
104 directions.

105 2. Experimental Set-Up

106 The laboratory experimental set-up is described in the following. First, a function generator (GW
107 INSTEK AF-2005 model) is employed to generate a white noise signal. Then, this signal is amplified
108 and applied to a modal shaker (GW-IV47 from Data Physics) that induces the vibration into the
109 structure. The general overview of the experimental set-up is shown in Figure 2 (left). The structure is
110 2.7 meters tall and composed of three parts:

- 111 1. The top beam (1×0.6 meters) where the modal shaker is attached to simulate a nacelle mass and
112 the effects of wind excitation,

- 113 2. The tower with three tubular sections connected with bolts,
 114 3. The jacket, **which includes** a pyramidal structure made up by 32 bars (S275JR steel) of different
 115 lengths, sheets (DC01 LFR steel), and other elements such as bolts and nuts.
 116 **It should be noted** that different wind speeds are considered by modifying the white noise signal
 amplitude (i.e., **scaling the amplitude** by 0.5, 1, 2, and 3). **To** measure vibration, eight triaxial

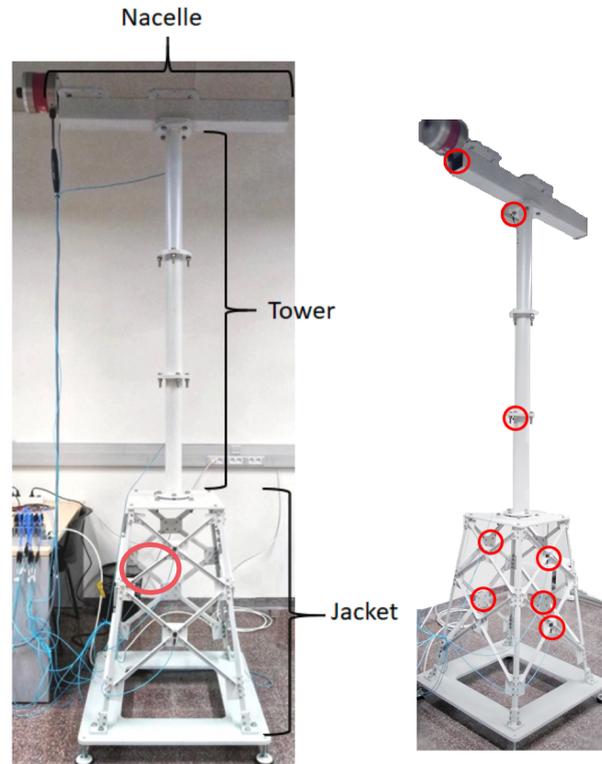


Figure 2. The experimental set-up (left) detailing the location of the damaged bar (red circle). Location of the sensors on the overall structure (right).

117 accelerometers (PCB[®] Piezotronic, model 356A17) are placed on the structure, see Figure 2 (right). The
 118 optimal number and placement of the sensors is determined according to [9]. The accelerometers are
 119 connected to six National Instruments[™] cartridges (NI 9234 model) that are inserted in the National
 120 Instruments chassis cDAQ-9188. Finally, the Data Acquisition Toolbox[™] is employed to configure the
 121 data acquisition hardware and read the data into MATLAB[®].

122 The studied damage states are related to one of the jacket bars, see Fig. 3. **The damage states**
 123 **include** a 5 mm bar crack and loosening one of the jacket bolts. Furthermore, a pristine replica bar is
 124 also **considered**.

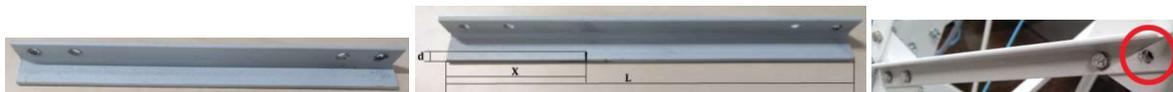


Figure 3. Different structural state scenarios studied in this work. Replica (healthy) bar (left). Crack damage where L is the length of the bar, $d = 5$ mm is the crack size, and $X = L/3$ is the location of the crack in the bar (center). Missing bolt (right).

125 Finally, note that the purpose of the paper is to verify that the conceived methodology has practical
 126 potential. The laboratory tower is a simplified model, but it is valid for **this preliminary study because**
 127 it is similar to the laboratory towers used, for example, in [9] where damage detection is accomplished
 128 (but not localization or identification) via damage indicators, in [15] and [16] where statistical time
 129 series are employed to detect damage, and in [17] and [18] where damage detection is accomplished
 130 through principal component analysis and support vector machines.
 131

132 3. Structural Health Monitoring Proposed Methodology

133 The proposed SHM strategy follows the steps detailed here. First, the raw time series data are
 134 collected. Second, the data are pre-processed to obtain a dataset of 24 channel gray-scale images. Third,
 135 a 24-channel-input CNN is designed and trained for classification of the different structural states. The
 136 following subsections describe in detail the aforementioned procedure.

137 3.1. Data gathering

The data are gathered in different experiments with a sampling rate of 275.27 Hz and a duration of 60 sec each. Table 1 shows the total number of realized experiments for the corresponding structural state (with its corresponding label) and white noise amplitude. A total of $K = 100$ experiments are conducted. Given the k -th experiment, where k is varied from 1 to $K = 100$, the raw data are then saved in the matrix $\mathbf{X}^{(k)} \in \mathcal{M}_{16517 \times 24}(\mathbb{R})$

$$\mathbf{X}^{(k)} = \begin{pmatrix} x_{1,1}^{(k)} & x_{1,2}^{(k)} & \cdots & x_{1,24}^{(k)} \\ x_{2,1}^{(k)} & x_{2,2}^{(k)} & \cdots & x_{2,24}^{(k)} \\ \vdots & \vdots & \ddots & \vdots \\ x_{16517,1}^{(k)} & x_{16517,2}^{(k)} & \cdots & x_{16517,24}^{(k)} \end{pmatrix}. \quad (1)$$

Note that there are as many rows as the number of measurements in each experiment, that is $I = 16,517$, and as many columns as the number of sensors, $J = 24$ (because each column is related to one sensor). Ultimately, the overall data matrix $\mathbf{X} \in \mathcal{M}_{1651700 \times 24}(\mathbb{R})$ is constructed by stacking the matrices that arise from each different experiment,

$$\mathbf{X} = \begin{pmatrix} \mathbf{X}^{(1)} \\ \vdots \\ \mathbf{X}^{(k)} \\ \vdots \\ \mathbf{X}^{(100)} \end{pmatrix}. \quad (2)$$

Table 1. Total number of experimental tests for the different white noise (WN) amplitudes and for each structural state.

Label	Structural state	0.5WN	1WN	2WN	3WN
1	Healthy bar	10 tests	10 tests	10 tests	10 tests
2	Replica bar	5 tests	5 tests	5 tests	5 tests
3	Crack damaged bar	5 tests	5 tests	5 tests	5 tests
4	Unlocked bolt	5 tests	5 tests	5 tests	5 tests

138

139 3.2. Data preprocessing: Scaling, reshaping, augmentation, and signal-to-image conversion

140 Data preprocessing is both the initial step and a critical step in machine learning. In this work,
 141 data reshaping is employed to guarantee that each sample includes multiple measurements from each
 142 sensor and thus has sufficient information to make a diagnosis regarding the state of the structure.
 143 Furthermore, a data augmentation strategy is proposed to improve the final test set error of the
 144 prediction model. It is clear that the signal-to-image conversion as well as the architecture and
 145 hyperparameters of the deep CNN will play a key role in the damage detection methodology. However,

146 the manner in which these data are scaled, augmented, and reshaped will significantly impact the
147 overall performance of the strategy [19].

148 3.2.1. Data scaling

The importance of preprocessing techniques for image classification by CNN is well known [20]. The main reason for data scaling is to enhance the efficiency of the neural network training process, significantly decreasing the number of epochs required for the network to learn, and thus leading to a better predictor. In particular, here, the data are scaled column-wise to fall within the specific range [0,255]. This range is selected to later allow for easy conversion into gray-scale images. In particular, the range is computed as follows. Assuming that there are K experimental tests, I samples per experiment, and J sensors,

$$M_j = \max \left(x_{ij}^{(k)} \right), i = 1, \dots, I, k = 1, \dots, K, \quad (3)$$

$$m_j = \min \left(x_{ij}^{(k)} \right), i = 1, \dots, I, k = 1, \dots, K, \quad (4)$$

where M_j and m_j are the maximum and the minimum values, respectively, of all the measures at column j , where $j = 1, \dots, J$. Accordingly, the elements of matrix \mathbf{X} are scaled

$$y_{ij}^{(k)} := \left(x_{ij}^{(k)} - m_j \right) \frac{255}{M_j - m_j}, i = 1, \dots, I, j = 1, \dots, J, k = 1, \dots, K, \quad (5)$$

to create a new matrix \mathbf{Y} as

$$\mathbf{Y} = \begin{pmatrix} y_{1,1}^{(1)} & y_{1,2}^{(1)} & \cdots & y_{1,24}^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ y_{16517,1}^{(1)} & y_{16517,2}^{(1)} & \cdots & y_{16517,24}^{(1)} \\ \hline y_{1,1}^{(2)} & y_{1,2}^{(2)} & \cdots & y_{1,24}^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ y_{16517,1}^{(2)} & y_{16517,2}^{(2)} & \cdots & y_{16517,24}^{(2)} \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline y_{1,1}^{(100)} & y_{1,2}^{(100)} & \cdots & y_{1,24}^{(100)} \\ \vdots & \vdots & \ddots & \vdots \\ y_{16517,1}^{(100)} & y_{16517,2}^{(100)} & \cdots & y_{16517,24}^{(100)} \end{pmatrix} = \begin{pmatrix} \mathbf{Y}^{(1)} \\ \hline \mathbf{Y}^{(2)} \\ \hline \vdots \\ \hline \mathbf{Y}^{(100)} \end{pmatrix}. \quad (6)$$

149 3.2.2. Data reshaping

150 In this section, data reshaping is employed to guarantee that each sample has multiple
151 measurements from each sensor and thus has sufficient information to diagnose the state of the
152 structure. In particular, matrix 6 is reshaped to matrix $\mathbf{Z} \in \mathcal{M}_{(6400) \times (256 \cdot 24)}$, as given in Table 2. It
153 should be noted that the data in the first 256 columns are related to sensor 1 and define the first
154 submatrix block, denoted as \mathbf{Z}_1 . Then, the data in columns 257 to 512 are related to sensor 2 and define

155 the second submatrix block \mathbf{Z}_2 . Next, the columns 513 to 768 are related to sensor 3 and define the third submatrix block \mathbf{Z}_3 , and so on and so forth, until the last sensor related to \mathbf{Z}_{24} has been accounted for.

$$\mathbf{Z} = \begin{pmatrix} \begin{array}{ccc|ccc} \text{Sensor 1} & & & \dots & & \text{Sensor 24} \\ \hline y_{1,1}^{(1)} & \cdots & y_{256,1}^{(1)} & & & y_{1,24}^{(1)} & \cdots & y_{256,24}^{(1)} \\ y_{257,1}^{(1)} & \cdots & y_{512,1}^{(1)} & & & y_{257,24}^{(1)} & \cdots & y_{512,24}^{(1)} \\ \vdots & \ddots & \vdots & & & \vdots & \ddots & \vdots \\ y_{16129,1}^{(1)} & \cdots & y_{16384,1}^{(1)} & & & y_{16129,24}^{(1)} & \cdots & y_{16384,24}^{(1)} \\ \hline \vdots & \ddots & \vdots & & & \vdots & \ddots & \vdots \\ \hline y_{1,1}^{(k)} & \cdots & y_{256,1}^{(k)} & & & y_{1,24}^{(k)} & \cdots & y_{256,24}^{(k)} \\ y_{257,1}^{(k)} & \cdots & y_{512,1}^{(k)} & & & y_{257,24}^{(k)} & \cdots & y_{512,24}^{(k)} \\ \vdots & \ddots & \vdots & & & \vdots & \ddots & \vdots \\ y_{16129,1}^{(k)} & \cdots & y_{16384,1}^{(k)} & & & y_{16129,24}^{(k)} & \cdots & y_{16384,24}^{(k)} \\ \hline \vdots & \ddots & \vdots & & & \vdots & \ddots & \vdots \\ \hline y_{1,1}^{(100)} & \cdots & y_{256,1}^{(100)} & & & y_{1,24}^{(100)} & \cdots & y_{256,24}^{(100)} \\ y_{257,1}^{(100)} & \cdots & y_{512,1}^{(100)} & & & y_{257,24}^{(100)} & \cdots & y_{512,24}^{(100)} \\ \vdots & \ddots & \vdots & & & \vdots & \ddots & \vdots \\ y_{16129,1}^{(100)} & \cdots & y_{16384,1}^{(100)} & & & y_{16129,24}^{(100)} & \cdots & y_{16384,24}^{(100)} \end{array} & \dots & \begin{array}{c} \mathbf{Z}^{(1)} \\ \vdots \\ \mathbf{Z}^{(k)} \\ \vdots \\ \mathbf{Z}^{(100)} \end{array} \end{pmatrix} = (\mathbf{Z}_1 \mid \cdots \mid \mathbf{Z}_{24})$$

Table 2. Data reshaping. On the one hand, this process can be viewed as the vertical stacking of $K = 100$ matrices $\mathbf{Z}^{(k)}$, $k = 1, \dots, K$, where each matrix is associated with a different experiment. On the other hand, this process can also be viewed as the horizontal concatenation of $J = 24$ matrices, \mathbf{Z}_j , $j = 1, \dots, J$, where each matrix is associated with a different sensor.

156
157 It should be noted that each row of matrix \mathbf{Z} contains the information of one sample of our SHM
158 strategy. Notice that to diagnosticate a WT the trained model requires at least one sample. Based on
159 the aforementioned reshaping process, the expected sample now contains 256 time stamps from each
160 sensor. In this manner, less than 1 second is required to gather the necessary data, when the sampling
161 frequency is 275.27 Hz. Thus, this process leads to a faster detection time (amount of time that elapses
162 between fault occurrence and detection). The intuition behind the proposed data reshape is twofold: i)
163 supplies more information to each sample; and ii) simplifies the signal-to-image conversion, as stated
164 in Section 3.2.4, because 256 is a perfect square.

165 Finally, observe that from matrices $\mathbf{Y}^{(k)}$, $k = 1, \dots, K$ in eq. (6), the last samples $y_{i,j}^{(k)}$ from
166 $i = 16385, \dots, 16517$, are discarded to reshape the data in the aforementioned new matrices $\mathbf{Z}^{(k)}$, $k =$
167 $1, \dots, K$.

168 3.2.3. Data augmentation

169 Deep convolutional neural networks rely heavily on big data to avoid overfitting, see [21].
170 Unfortunately, many application domains lack access to big data. In this work, to build a better

171 deep CNN model, a data augmentation strategy is proposed that artificially expands the size of the
172 training dataset without actually collecting new data.

173 The method consists of using each time stamp as the beginning of a new sample (and using the
174 subsequent 255 measures to complete the sample), as shown in Table 3. Accordingly, instead of the
175 previously defined matrices (see Table 2) $\mathbf{Z}^{(k)} \in \mathcal{M}_{(64) \times (256 \cdot 24)}$, $k = 1, \dots, K$, augmented matrices
176 with the same number of columns but more rows are obtained, namely, $\mathbf{D}^{(k)} \in \mathcal{M}_{(16129) \times (256 \cdot 24)}$, $k =$
177 $1, \dots, K$. Thus, from the initial 64 samples per experiment, we increased to 16,129 samples per
experiment. This is an increment of 25,200% in the total number of samples in the dataset.

	Signal 1		Signal 2		...		Signal 24		
$\mathbf{D}^{(k)} =$	$y_{1,1}^{(k)}$	\cdots	$y_{256,1}^{(k)}$	$y_{1,2}^{(k)}$	\cdots	$y_{256,2}^{(k)}$	$y_{1,24}^{(k)}$	\cdots	$y_{256,24}^{(k)}$
	$y_{2,1}^{(k)}$	\cdots	$y_{257,1}^{(k)}$	$y_{2,2}^{(k)}$	\cdots	$y_{257,2}^{(k)}$	$y_{2,24}^{(k)}$	\cdots	$y_{257,24}^{(k)}$
	$y_{3,1}^{(k)}$	\cdots	$y_{258,1}^{(k)}$	$y_{3,2}^{(k)}$	\cdots	$y_{258,2}^{(k)}$	$y_{3,24}^{(k)}$	\cdots	$y_{258,24}^{(k)}$
	$y_{4,1}^{(k)}$	\cdots	$y_{259,1}^{(k)}$	$y_{4,2}^{(k)}$	\cdots	$y_{259,2}^{(k)}$	$y_{4,24}^{(k)}$	\cdots	$y_{259,24}^{(k)}$
	\vdots	\ddots	\vdots	\vdots	\ddots	\vdots	\vdots	\ddots	\vdots
	$y_{16129,1}^{(k)}$	\cdots	$y_{16384,1}^{(k)}$	$y_{16129,2}^{(k)}$	\cdots	$y_{16384,2}^{(k)}$	$y_{16129,24}^{(k)}$	\cdots	$y_{16384,24}^{(k)}$

Table 3. Synthetic data augmentation for experiment k , $k = 1, \dots, K$.

178

Finally, the data matrix $\mathbf{D} \in \mathcal{M}_{1612900 \times (256 \cdot 24)}(\mathbb{R})$, which contains the scaled, reshaped, and augmented data from all of the experiments, is defined by stacking the data matrices derived from each different experiment (recall that $K = 100$),

$$\mathbf{D} = \begin{pmatrix} \mathbf{D}^{(1)} \\ \vdots \\ \mathbf{D}^{(k)} \\ \vdots \\ \mathbf{D}^{(100)} \end{pmatrix}. \quad (7)$$

179 3.2.4. Signal-to-image conversion

180 The fault diagnosis method converts time-domain signals from the 24 measured variables
181 into 2D gray-scale images to exploit texture information from the converted images. The data
182 conversion process was inspired based on reference [13], although the process is enhanced here
183 by using multi-channel images.

184 The image size used for signal-to-image conversion is 16×16 (256 pixels) with 24 channels,
185 constructed as follows. Each row of matrix \mathbf{D} , see eq. (7), is converted to one image of size 16×16
186 with 24 channels (one channel per sensor), similar to a standard RGB image with 3 channels. It should
187 be noted that because the sampling time is $1/257$ seconds, each image contains approximately one
188 second of data from each sensor, which is sufficient to capture all of the system dynamics. The total
189 number of images in the dataset is 1,612,900, because 16,129 images are obtained from each of the 100
190 experiments. Figure 4 shows one example of such a multi-channel image.

191 3.3. Deep convolutional neural network

192 CNNs are feed-forward artificial neural networks that use the convolution operation instead of
193 matrix multiplication. The pre-processing required in a CNN is significantly less than that required
194 by other classification algorithms because features are not hand-engineered but learned. Typically,
195 there are three kinds of layers: convolution, fully-connected, and soft-max. The convolution layer

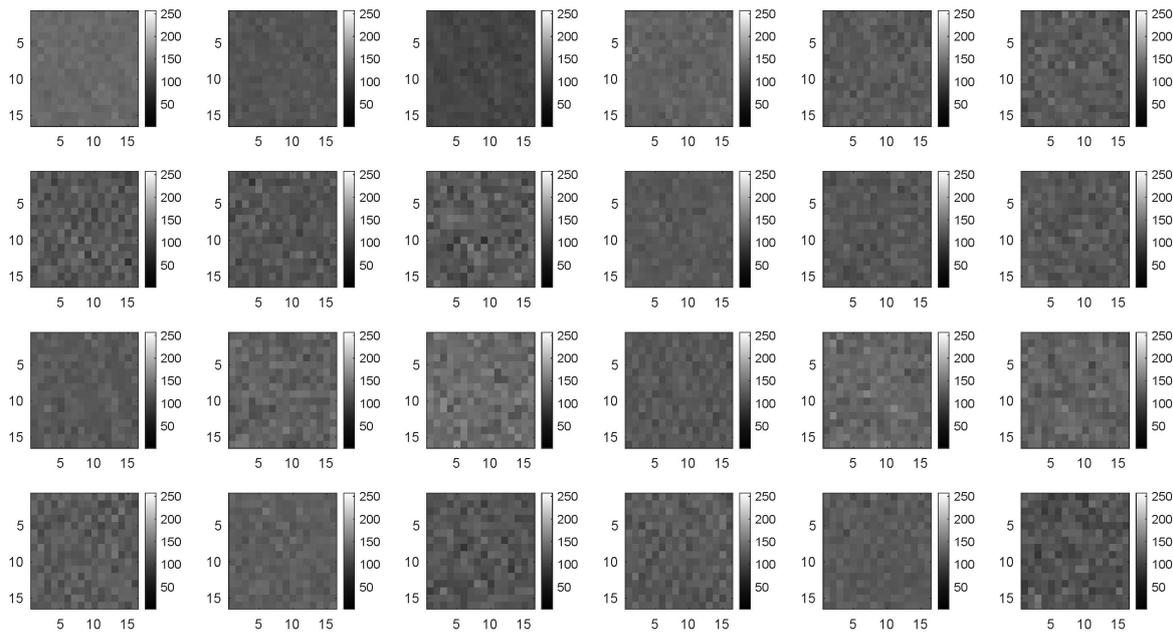


Figure 4. Multi-channel gray-scale image corresponding to the 24 sensors (size 16×16).

196 main aspects are its sparse local connectivity and filters, which significantly diminishes the number of
 197 network parameters while simultaneously increasing its performance. The convolution layers last step
 198 is to apply the so-called activation function, which is a non-linear function. Fully-connected layers are
 199 normal neural network layers in which all the outputs from the previous layer are connected to all the
 200 nodes in the next layer. Normally, these layers go towards the end of the network. Finally, a soft-max
 201 layer assigns probabilities to each class and connects to the final output layer that will have the same
 202 number of neurons as classes.

203 To construct a deep CNN for a particular application is a complex task. In comparison to the
 204 wealth of research related to color images, very little work has been carried out for gray-scale images.
 205 In this work, a CNN is designed for the detection of different structural damage states based on 24
 206 channel gray-scale images.

207 3.3.1. Data split: Training set and validation set

208 To develop the classification model, deep learning methods divide the available data into training
 209 and validation sets. The training dataset is the actual dataset used to train the model (weights and
 210 biases in a CNN). In other words, the training dataset is the sample of data used to fit the model. In
 211 contrast, the validation dataset is the sample of data used to provide an unbiased evaluation of the
 212 model fit on the training dataset while tuning the model hyperparameters.

213 In this work, the following dataset split ratio has been used: 75% of the whole dataset is assigned
 214 to the training set, and 25% is assigned to the validation set. That is, 1,209,675 images with data
 215 augmentation, or 4,800 without data augmentation, are used to train the CNN. Then, 403,225 images
 216 with data augmentation, or 1,600 without data augmentation, are used to validate the model.

217 3.3.2. Network architecture

218 The network presented in Figure 5 was designed in this work. The input is a 16×16 image with
 219 24 channels, all of which are gray-scale. Figure 6 shows an example of one image in the dataset that
 220 was obtained after the preprocess procedure stated in Section 3.2. The input is convoluted by a series
 221 of 7 convolutional layers. Each convolution layer is followed by a batch normalization step, which is
 222 used to improve the speed, performance, and stability of the CNN [22], and a ReLU (Rectified Linear

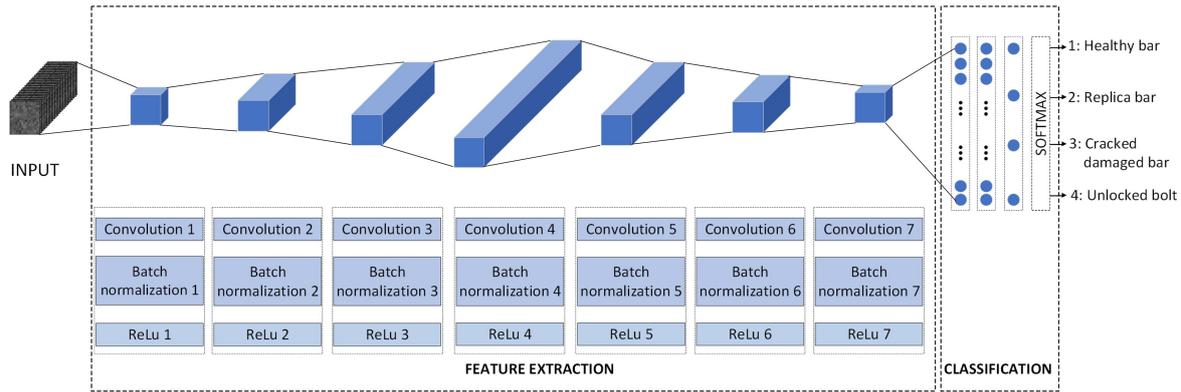


Figure 5. Architecture of the developed CNN.

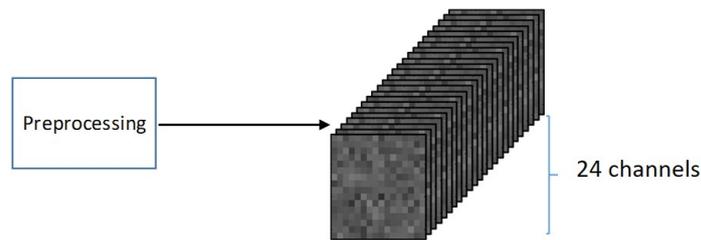


Figure 6. Example of one image in the dataset (24 channels) used as the CNN input.

Unit) activation function ($f(x) = \max(0, x)$), because this approach has been shown to speed up the training process in comparison to the classic sigmoid alternative. The final layers of the network are three fully connected layers and a soft max block, used to squash the 4-dimensional output into a categorical probability distribution: (1) original healthy bar, (2) replica bar, (3) crack damaged bar, and (4) unlocked bolt.

The most significant characteristics of the CNN architecture are summarized in Table 4.

It should be noted that the convolutions with a maximum number of parameters are the intermediate case (convolutions 4 and 5), whereas those with the minimum number of parameters correspond to the first and last convolutions. Finally, the three fully connected layers have sizes 32, 16, and 4, respectively, and are followed by the softmax function with four outputs.

It should also be noted that each convolution employs a padding of 1. The main intuition behind this selection is that, normally, the filter is applied by superimposing it on the image from the upper left edge. Then a columnar translation is applied until the filter is superimposed with its right edge on the right edge of the image. This usual way of proceeding has a problem, the edge pixels are never subjected to the central part of the filter. This is sometimes known as the border effect problem and can be solved by incorporating so-called padding [23]. That is to apply the filter beginning from outside the image frame as well as ending also outside the image, in such a manner that edge pixels reach also the center part of the filter. In this work, a padding of 1 is used to enhance the texture features extracted by the CNN for all of the data in the image, regardless of whether the data are located in the image. Table 5 compares different metrics (see Section 4.1 where a definition of these metrics is given) with and without padding (without data augmentation). It can be observed that when using padding, better results are attained.

3.3.3. Network training

The training of the CNN consists of the minimization of a loss function by means of a numerical optimization algorithm. In this work, the Adam optimizer [24] is employed to minimize the categorical cross entropy [25]. The Adam algorithm combines two versions of speeding up gradient descent: i)

Layer	Ouput size	Parameters	# of Parameters
Input 16×16×24 images	16×16×24	-	0
Convolution#1 32 filters of size 5×5×24 with padding [1 1 1 1]	14×14× 32	Weight 5×5×24×32 Bias 1×1×32 Offset 1×1×32	19232
Batch Normalization#1	14×14× 32	Scale 1×1×32	64
ReLu#1	14×14× 32	-	0
Convolution#2 64 filters of size 5×5×24 with padding [1 1 1 1]	12×12× 64	Weight 5×5×32×64 Bias 1×1×64 Offset 1×1×64	51264
Batch Normalization#2	12×12× 64	Scale 1×1×64	128
ReLu#2	12×12× 64	-	0
Convolution#3 128 filters of size 5×5×24 with padding [1 1 1 1]	10×10× 128	Weight 5×5×64×128 Bias 1×1×128 Offset 1×1×128	204928
Batch Normalization#3	10×10× 128	Scale 1×1×128	256
ReLu#3	10×10× 128	-	0
Convolution#4 256 filters of size 5×5×24 with padding [1 1 1 1]	8×8× 256	Weight 5×5×128×256 Bias 1×1×256 Offset 1×1×256	819456
Batch Normalization#4	8×8× 256	Scale 1×1×256	512
ReLu#4	8×8× 256	-	0
Convolution#5 128 filters of size 5×5×24 with padding [1 1 1 1]	6×6× 128	Weight 5×5×256×128 Bias 1×1×128 Offset 1×1×128	819456
Batch Normalization#5	6×6× 128	Scale 1×1×128	256
ReLu#5	6×6× 128	-	0
Convolution#6 64 filters of size 5×5×24 with padding [1 1 1 1]	4×4× 64	Weight 5×5×128×64 Bias 1×1×64 Offset 1×1×64	204864
Batch Normalization#6	4×4× 64	Scale 1×1×64	128
ReLu#6	4×4× 64	-	0
Convolution#7 32 filters of size 5×5×24 with padding [1 1 1 1]	2×2× 32	Weight 5×5×64×32 Bias 1×1×32 Offset 1×1×32	51232
Batch Normalization#7	2×2× 32	Scale 1×1×32	64
ReLu#7	2×2× 32	-	0
Fully connected layer#1	1×1× 32	Weight 32×128 Bias 32×1	4128
Fully connected layer#2	1×1× 16	Weight 16×32 Bias 16×1	528
Fully connected layer#3	1×1× 4	Weight 4×16 Bias 4×1	68
Softmax	-	-	0
classoutput	-	-	0

Table 4. Characteristics of the designed CNN. The neural network has a total of 2,176,308 parameters. The number of output channels in each layer is highlighted in boldface font.

249 gradient descent with momentum, where the basic idea is to compute an exponentially weighted
250 average of the gradients, and ii) root mean square propagation (RMSProp), that makes use of the
251 gradient second moments. Specifically, the Adam numerical method puts together the exponential
252 moving average of the gradient and the squared gradient (second moment), and hyperparameters
253 β_1 and β_2 handle their decrease rates, respectively. In this work, the Adam optimizer has been tuned
254 and thus employs an initial learning rate of $\alpha_0 = 0.01$, and values $\beta_1 = 0.9$, $\beta_2 = 0.992$, and $\epsilon = 10^{-7}$
255 to avoid divisions by zero. Furthermore, here, the learning rate is decreased every 2 epochs by
256 multiplying with factor 0.5.

Strategy	Accuracy	Precision	Recall	F1 score	Specificity
ReLu - Padding - L2 regularization	93.81	92.77	93.73	93.22	97.98
Relu - No padding - L2 regularization	93.69	92.73	93.44	93.07	97.92
Relu - Padding - No L2 regularization	93.63	92.73	93.82	93.25	97.89

Table 5. Metrics for different CNN architectures without data augmentation. The best metric results are highlighted in boldface font.

Convolutional layer initialization is carried out by the so-called Xavier initializer [26]. Mini-batches of size 75 in the initial dataset and 590 for the augmented dataset are used to update the weights.

Finally, L2 regularization with $\lambda = 10^{-6}$ is employed. Table 5 compares the different metrics (see Section 4.1 for a definition of these metrics) with and without L2 regularization (without data augmentation). It can be observed that when using regularization, better results are obtained because regularization reduces high variance in the validation set.

3.3.4. Network architecture and hyperparameter tuning

To select the best architecture and to tune the different hyperparameters usually require significant computational resources. Because one of the most critical aspects of computational cost is the dataset size, in this paper, following the results presented in [27] and [28], the small dataset (without augmentation) is used to define the CNN architecture and quickly (coarse) tune the hyperparameters. Next, the obtained optimal hyperparameters for the small dataset are used as initial values to finetune the hyperparameters with the large dataset (with data augmentation).

3.3.5. Network implementation

The stated methodology is coded in MATLAB[®] using its Deep Learning Toolbox[™] on a laptop running the Windows[®] 10 operating system, with an Intel Core i7-9750H processor, 16 GB of RAM, and an Nvidia GeForce RTX[™]2060 graphic card that requires 6 GB of GPU.

4. Results and Discussion

4.1. Metrics to evaluate the classification model

To measure classification performance, several metrics can be computed from a confusion matrix such as that shown in Table 6. Normally, these metrics evaluate binary classification problems. Note that, true positive (TP) is the number of positive samples that are correctly predicted as such, false positive (FP) is the number of negative samples that are incorrectly predicted, true negative (TN) is the number of negative samples that are correctly predicted, and false negative (FN) is the number of positive samples that are incorrectly predicted. The most common metrics for binary classification

		Predicted class	
		Positive	Negative
Actual class	Positive	True positive (TP)	False negative (FN)
	Negative	False positive (FP)	True negative (TN)

Table 6. Binary confusion matrix.

problems are the following.

- Accuracy: Proportion of true results (both true positives and true negatives) among the total number of cases examined.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}}$$

- Precision: Proportion of positive results that are true positive.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

- Recall: Proportion of actual positives that are correctly identified as such.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

- Specificity: Proportion of actual negatives that are correctly identified as such.

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

- F1-score: Harmonic mean of the precision and recall.

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

284 In a multi-class classification problem, [such as that considered](#) in this work, these metrics are also
 285 applicable using a one-vs.-all approach to compute [each metric](#) for each class, see [29]. [Essentially](#), that
 286 is, to compute the different metrics for each label as if the problem has been reduced to a binary 'label
 287 X' versus 'not label X' situation.

288 4.2. Results of *the* CNN classification method

289 [To](#) evaluate the developed methodology, this section presents the results obtained from the
 290 proposed SHM strategy. A flowchart of the proposed approach is given in Figure 7. When a WT [must](#)
 291 be diagnosed, the accelerometer data are scaled, reshaped, and converted into gray-scale images that
 292 are fed into the already trained CNN, and a classification is obtained to predict the structural state
 293 condition.

294 [To](#) thoroughly test the functional characteristics of the algorithm, the [datasets](#) with and without
 295 data augmentation are considered, [as well as comparison with two other methodologies given in \[17\]](#)
 296 [and \[9\], that make use of the same laboratory structure. The first methodology, given in \[17\], is based](#)
 297 [on principal component analysis and support vector machines. The second methodology, given in](#)
 298 [\[9\] \(page 67\), is based on the well-known damage indicators: covariance matrix estimate, and scalar](#)
 299 [covariance.](#)

300 Figures 8 and 9 illustrate the confusion matrices for the validation [dataset](#) without and with
 301 data augmentation, respectively. The rows [represent](#) the true class, [whereas](#) columns [represent](#) the
 302 predicted class. [The precision and false discovery rate are given in the rightmost columns. Finally, the](#)
 303 [recall and false negative rate are given at the bottom rows.](#) An examination of both confusion matrices
 304 reveals that some misclassifications come from the model confounding the healthy and replica bars
 305 (labels 1 and 2). However, this [level of misclassification](#) is acceptable [because](#) both bars are in a healthy
 306 state. [In contrast](#), some errors [are derived](#) from the model misclassifying the crack and unlocked bolt
 307 damages (labels 3 and 4), which will not detect correctly the type of damage but at least would lead
 308 to a damage alert. Finally, [it should be noted](#) that very few damaged samples (labels 3 and 4) are
 309 classified as healthy or replica bar (labels 1 and 2).

310 From the confusion matrices, the different metrics to evaluate the classification model, see Section
 311 4.1, are computed and presented in Table 7. The impact of the data augmentation strategy [can clearly](#)
 312 [be seen. Although](#) no new experimental data [were collected, nonetheless](#) the metrics [were significantly](#)

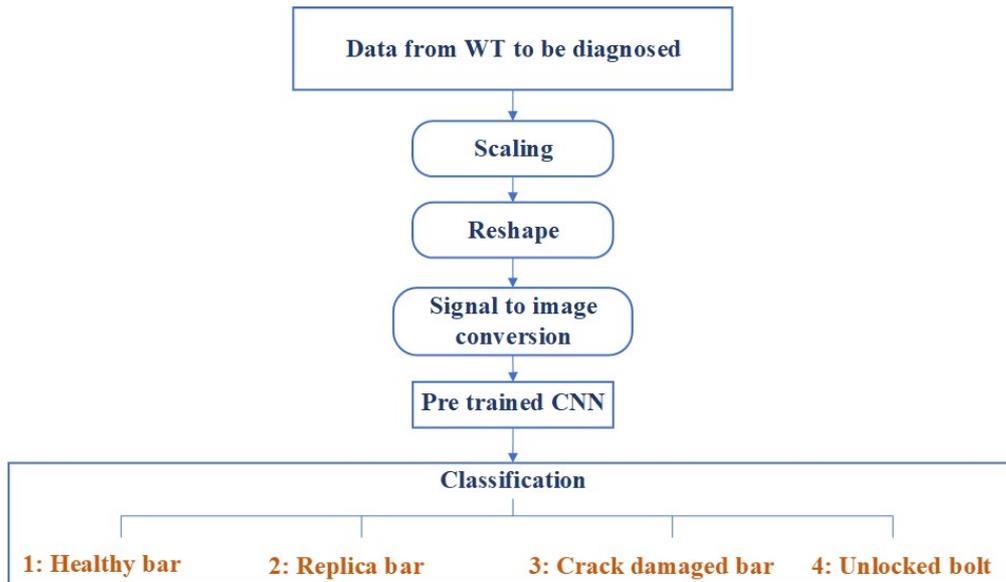


Figure 7. Flowchart to illustrate how the proposed SHM strategy is applied when a WT must be diagnosed.

	Label	Precision	Recall	F1-score	Specificity
Without data augmentation	1: Healthy bar	97.97	94.14	96.02	98.61
	2: Replica bar	90.31	94.75	92.48	97.61
	3: Crack damaged bar	90.31	92.63	91.46	97.59
	4: Unlocked bolt	92.50	93.38	92.94	98.13
With data augmentation	1: Healthy bar	99.89	99.96	99.92	99.92
	2: Replica bar	99.90	99.87	99.88	99.97
	3: Crack damaged bar	99.94	99.86	99.90	99.99
	4: Unlocked bolt	99.90	99.86	99.88	99.97

Table 7. Metrics for each label of the multi-classification problem and comparison between the datasets without and with data augmentation.

improved. It should be noted that all of the metrics (precision, recall, F1-score, and specificity) are higher than or equal to 99.86% for each label when using the augmented dataset in comparison to values between 90.31% and 98.61% for the initial dataset. Despite all metrics being relevant, considering the specific problem at hand, the most important metric is recall, which is the proportion of actual damaged cases that are correctly identified as such. It can be observed that the crack damage and the unlocked bolt, even without data augmentation, obtain recall values of 92.63% and 93.38%, respectively. When data augmentation is used, the recall values are all higher than or equal to 99.86% for all of the studied classes. The results associated with the precision metric are also satisfactory. When the initial dataset is used, precision values are between 90.31 and 97.97, but with the augmented dataset, such values are all higher than or equal to 99.89. Finally, it should be noted that the specificity metric is that which experiences less improvement when using the augmented dataset.

As already mentioned before, here, a comparison is made between our obtained results and two other methodologies. On the one hand, when using the first approach stated in [17], the crack damaged bar has a recall of 96.08%, thus inferior to the one obtained with the proposed strategy in this work which attained a value of 99.86%. Note that the crack damage is the most challenging. In fact, the second approach stated in [9] (page 82) was not capable to detect this type of incipient damage when using the scalar covariance or mean residual damage indicators. On the other hand, the first approach obtains a recall of 99.02% for the unlocked bolt damage, whereas with the proposed strategy a slightly

True Class	1	627	17	10	12	94.1%	5.9%
	2	5	289	11		94.8%	5.2%
	3		11	289	12	92.6%	7.4%
	4	8	3	10	296	93.4%	6.6%
		98.0%	90.3%	90.3%	92.5%		
		2.0%	9.7%	9.7%	7.5%		
		1	2	3	4		
		Predicted Class					

Figure 8. Confusion matrix for the validation **dataset** without data augmentation.

True Class	1	161097	31		37	100.0%	0.0%
	2	89	80557	16		99.9%	0.1%
	3	14	52	80593	45	99.9%	0.1%
	4	80		31	80558	99.9%	0.1%
		99.9%	99.9%	99.9%	99.9%		
		0.1%	0.1%	0.1%	0.1%		
		1	2	3	4		
		Predicted Class					

Figure 9. Confusion matrix for the validation **dataset** with data augmentation.

331 higher value of 99.86% is obtained. Finally, note that the unlocked bold damage is not studied in the
 332 second approach

333 The proposed CNN exhibits low bias and variance for both **datasets**, because the **training** and
 334 validation errors are small (low bias), as well as the difference between them (low variance), as shown
 335 in Table 8. In particular, when using the initial **dataset**, the training error is equal to 0.1167 and the
 336 validation error is quite close to **this same value**, being equal to 0.1692. When using the augmented
 337 **dataset**, the **training** error diminishes to 0.0026, and the validation error is **only slightly greater, at**
 338 0.0044. From this table, the **significantly increased** training time (1196 minutes) of the augmented
 339 **dataset** in comparison to **that of the initial dataset** (11 minutes) can be seen, which is **easily understood**
 340 **due to** the size of each **dataset**. That is, **there are** 1,612,800 images in the augmented **dataset** and only
 341 6,400 images in the initial.

342 Finally, Figure 10 shows the accuracy and loss curves during training and validation (black dotted
 343 lines) when using the augmented **dataset**. **It should be noted** that after 5 epochs, the CNN obtains an
 344 accuracy of 99.90% and a final validation loss of 0.0044, as **shown** in Table 8.

	Accuracy	Validation error	Training error	Training time	# of images
Whitout data augmentation	93.81	0.1692	0.1167	11 min	6400
With data augmentation	99.90	0.0044	0.0026	1196 min	1612800

Table 8. Comparison of obtained accuracy, validation error, training error, and training time when using data augmentation with respect to the original dataset.

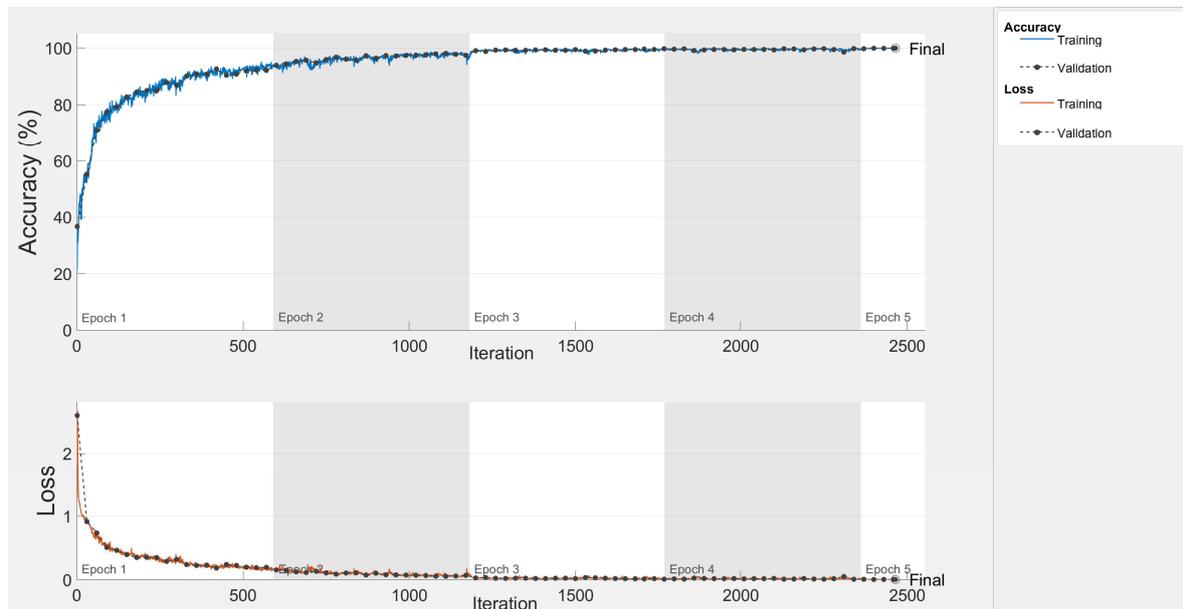


Figure 10. Accuracy and loss curve for the augmented dataset.

345 5. Conclusions and Future Work

346 In this work, a strategy based solely on vibration response was demonstrated for the structural
 347 health monitoring of offshore WT foundations. The approach was tested on a laboratory set-up, for
 348 which four different structural states for a jacket bar were studied: healthy bar, replica, crack damage,
 349 and an unlocked bolt.

350 The contribution of this work is twofold: i) how three-dimensional data (derived from different
 351 time, sensors, and experiments) are preprocessed (collected, scaled, reshaped, augmented, and
 352 converted into gray-scale images with as many channels as sensors), and ii) the design of a deep
 353 CNN, the architecture and hyperparameters of which play a key role in the specific application that
 354 concerns us, damage diagnosis. Furthermore, the proposed method does not require hand-designed
 355 features beforehand, because the CNN learns features automatically.

356 The conceived SHM methodology, with data augmentation, shows an exceptional performance,
 357 with all considered metrics (accuracy, precision, recall, F1-score, and specificity) giving results greater
 358 than 99.8%. In particular, a noteworthy overall accuracy of 99.90% is obtained with data augmentation.
 359 These results show that large (deep) CNNs are promising for the development of SHM strategies for
 360 WT offshore foundations.

361 Future work will focus in three main areas. First, based on open set domain adaptation [30],
 362 research to render capability of separating unknown damage from known targeted types of damage
 363 will be conducted. Second, not only detection and classification but also the localization of the damage
 364 will be attempted by designing an ensemble of deep CNNs, being the main idea to take advantage
 365 of individual information from each sensor signal. Last, to deal with the validation of the proposed
 366 strategy in a more realistic environment, a water tank facility will be used in which the laboratory
 367 tower will be placed and subjected to the action of regular and irregular waves.

368 **Author Contributions:** All authors contributed equally to this work.

369 **Funding:** This work was partially funded by the Spanish Agencia Estatal de Investigación (AEI) - Ministerio
370 de Economía, Industria y Competitividad (MINECO), and the Fondo Europeo de Desarrollo Regional (FEDER)
371 through research project DPI2017-82930-C2-1-R; and by the Generalitat de Catalunya through research project
372 2017 SGR 388. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan XP
373 GPU used for this research.

374 **Acknowledgments:** We thank the three anonymous reviewers for their careful reading of our manuscript and
375 their many insightful comments and suggestions.

376 **Conflicts of Interest:** The authors declare no conflict of interest. The founding sponsors had no role in the design
377 of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; nor in the
378 decision to publish the results.

379 References

- 380 1. Ohlenforst, K.; Backwell, B.; Council, G.W.E. Global Wind Report 2018. Web page, 2019.
- 381 2. Lai, W.J.; Lin, C.Y.; Huang, C.C.; Lee, R.M. Dynamic analysis of Jacket Substructure for offshore wind
382 turbine generators under extreme environmental conditions. *Applied Sciences* **2016**, *6*, 307.
- 383 3. Moulas, D.; Shafiee, M.; Mehmanparast, A. Damage analysis of ship collisions with offshore wind turbine
384 foundations. *Ocean Engineering* **2017**, *143*, 149–162.
- 385 4. Van Kuik, G.; Peinke, J. *Long-term research challenges in wind energy-a research agenda by the European Academy
386 of Wind Energy*; Vol. 6, Springer, 2016.
- 387 5. Liu, W.; Tang, B.; Han, J.; Lu, X.; Hu, N.; He, Z. The structure healthy condition monitoring and fault
388 diagnosis methods in wind turbines: A review. *Renewable and Sustainable Energy Reviews* **2015**, *44*, 466–472.
- 389 6. Qing, Xinlin and Li, Wenzhuo and Wang, Yishou and Sun, Hu. Piezoelectric transducer-based structural
390 health monitoring for aircraft applications. *Sensors* **2019**, *19*, 545.
- 391 7. Weijtjens, W.; Verbelen, T.; De Sitter, G.; Devriendt, C. Foundation structural health monitoring of an
392 offshore wind turbine: a full-scale case study. *Structural Health Monitoring* **2016**, *15*, 389–402.
- 393 8. Oliveira, G.; Magalhães, F.; Cunha, Á.; Caetano, E. Vibration-based damage detection in a wind turbine
394 using 1 year of data. *Structural Control and Health Monitoring* **2018**, *25*, e2238.
- 395 9. Zugasti Uriguen, E. Design and validation of a methodology for wind energy structures health monitoring.
396 PhD thesis, Universitat Politècnica de Catalunya, Jordi Girona, 31, Barcelona, Spain, 2014.
- 397 10. Lee, Jong Jae and Lee, Jong Won and Yi, Jin Hak and Yun, Chung Bang and Jung, Hie Young. Neural
398 networks-based damage detection for bridges considering errors in baseline finite element models. *Journal
399 of Sound and Vibration* **2005**, *280*, 555–578.
- 400 11. Kim, Byungmo and Min, Cheonhong and Kim, Hyungwoo and Cho, Sugil and Oh, Jaewon and Ha,
401 Seung-Hyun and Yi, Jin-hak. Structural health monitoring with sensor data and cosine similarity for
402 multi-damages. *Sensors* **2019**, *19*, 3047.
- 403 12. Stetco, A.; Dinmohammadi, F.; Zhao, X.; Robu, V.; Flynn, D.; Barnes, M.; Keane, J.; Nenadic, G. Machine
404 learning methods for wind turbine condition monitoring: A review. *Renewable energy* **2019**, *133*, 620–635.
- 405 13. Ruiz, M.; Mujica, L.E.; Alferez, S.; Acho, L.; Tutiven, C.; Vidal, Y.; Rodellar, J.; Pozo, F. Wind turbine fault
406 detection and classification by means of image texture analysis. *Mechanical Systems and Signal Processing*
407 **2018**, *107*, 149–167.
- 408 14. Vidal, Y.; Pozo, F.; Tutivén, C. Wind turbine multi-fault detection and classification based on SCADA data.
409 *Energies* **2018**, *11*, 3018.
- 410 15. Spanos, N.I.; Sakellariou, J.S.; Fassois, S.D. Exploring the limits of the Truncated SPRT method for
411 vibration-response-only damage diagnosis in a lab-scale wind turbine jacket foundation structure. *Procedia
412 engineering* **2017**, *199*, 2066–2071.
- 413 16. Spanos, N.A.; Sakellariou, J.S.; Fassois, S.D. Vibration-response-only statistical time series structural health
414 monitoring methods: A comprehensive assessment via a scale jacket structure. *Structural Health Monitoring*
415 **2019**, p. 1475921719862487.
- 416 17. Vidal Seguí, Y.; Rubias, J.L.; Pozo Montero, F. Wind turbine health monitoring based on accelerometer data.
417 9th ECCOMAS Thematic Conference on Smart Structures and Materials, 2019, pp. 1604–1611.
- 418 18. Vidal, Y.; Aquino, G.; Pozo, F.; Gutiérrez-Arias, J.E.M. Structural Health Monitoring for Jacket-Type
419 Offshore Wind Turbines: Experimental Proof of Concept. *Sensors* **2020**, *20*, 1835.

- 420 19. Pozo, F.; Vidal, Y.; Serrahima, J. On real-time fault detection in wind turbines: Sensor selection algorithm
421 and detection time reduction analysis. *Energies* **2016**, *9*, 520.
- 422 20. Pal, K.K.; Sudeep, K. Preprocessing for image classification by convolutional neural networks. 2016 IEEE
423 International Conference on Recent Trends in Electronics, Information & Communication Technology
424 (RTEICT). IEEE, 2016, pp. 1778–1781.
- 425 21. Chen, X.W.; Lin, X. Big data deep learning: challenges and perspectives. *IEEE access* **2014**, *2*, 514–525.
- 426 22. Santurkar, S.; Tsipras, D.; Ilyas, A.; Madry, A. How does batch normalization help optimization? *Advances*
427 *in Neural Information Processing Systems*, 2018, pp. 2483–2493.
- 428 23. Albawi, S.; Mohammed, T.A.; Al-Zawi, S. Understanding of a convolutional neural network. 2017
429 International Conference on Engineering and Technology (ICET). IEEE, 2017, pp. 1–6.
- 430 24. DP, K. Ba J. Adam: a method for stochastic optimization. *The international conference on learning*
431 *representations*, 2015.
- 432 25. Rusiecki, A. Trimmed categorical cross-entropy for deep learning with label noise. *Electronics Letters* **2019**,
433 *55*, 319–320.
- 434 26. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks.
435 *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp.
436 249–256.
- 437 27. DeCastro-García, N.; Muñoz Castañeda, Á.L.; Escudero García, D.; Carriegos, M.V. Effect of the Sampling of
438 a Dataset in the Hyperparameter Optimization Phase over the Efficiency of a Machine Learning Algorithm.
439 *Complexity* **2019**, 2019.
- 440 28. Swersky, K.; Snoek, J.; Adams, R.P. Multi-task bayesian optimization. *Advances in neural information*
441 *processing systems*, 2013, pp. 2004–2012.
- 442 29. Hossin, M.; Sulaiman, M. A review on evaluation metrics for data classification evaluations. *International*
443 *Journal of Data Mining & Knowledge Management Process (IJDKP)* **2015**, *5*, 1–11.
- 444 30. Saito, Kuniaki and Yamamoto, Shohei and Ushiku, Yoshitaka and Harada, Tatsuya. Open set domain
445 adaptation by backpropagation. *Proceedings of the European Conference on Computer Vision (ECCV)*,
446 2018, pp. 153–168.