aerospace

MDPI

Article

Single-Sensor Acoustic Emission Source Localization in Plate-Like Structures Using Deep Learning †

Arvin Ebrahimkhanlou * and Salvatore Salamone *

Smart Structures Research Group (SSRG), Department of Civil Architectural and Environmental Engineering, The University of Texas at Austin, 10100 Burnet Rd, Bldg. 177, Austin, TX 78758, USA

* Correspondence: arvinebr@utexas.edu (A.E.); salamone@utexas.edu (S.S.); Tel.: +1-(512)-232-3427 (S.S.); +1-(512)-471-3024 (A.E.)

† This paper is an extended version of the authors' conference paper published in 11th International Workshop on Structural Health Monitoring (IWSHM), Stanford, CA, USA, 12–14 September 2017, which was selected for and invited to this special issue.

Received: 27 March 2018; Accepted: 19 April 2018; Published: 1 May 2018

Abstract: This paper introduces two deep learning approaches to localize acoustic emissions (AE) sources within metallic plates with geometric features, such as rivet-connected stiffeners. In particular, a stack of autoencoders and a convolutional neural network are used. The idea is to leverage the reflection and reverberation patterns of AE waveforms as well as their dispersive and multimodal characteristics to localize their sources with only one sensor. Specifically, this paper divides the structure into multiple zones and finds the zone in which each source occurs. To train, validate, and test the deep learning networks, fatigue cracks were experimentally simulated by Hsu–Nielsen pencil lead break tests. The pencil lead breaks were carried out on the surface and at the edges of the plate. The results show that both deep learning networks can learn to map AE signals to their sources. These results demonstrate that the reverberation patterns of AE sources contain pertinent information to the location of their sources.

Keywords: acoustic emission; guided ultrasonic waves; deep learning; autoencoders; convolutional neural networks; machine learning; edge reflection; reverberation patterns; structural health monitoring

1. Introduction

Metallic plate-like structures are ubiquitous in the aerospace industry. These structures are susceptible to different types of damage, including fatigue cracks and corrosion dents. Numerous studies exist in the literature concerned with the structural health monitoring (SHM) of metallic plates [1–3]. Among these studies, SHM techniques based on acoustic emissions (AE) can detect and localize damage in metallic panels [4]. For example, Kundu et al. [5] developed an optimization-based approach for plates with known wave velocities and extended it later to localize sources in anisotropic plates with unknown properties [6,7]. The authors have also developed a dictionary-based source localization algorithm for simple isotropic plates [8]. For more interested readers, Kundu [4] has provided an in-depth review of AE source localization algorithms.

Despite the significant development of AE source localization algorithms, very few have been implemented in real structures. One reason for this lack of acceptance is the potential for these algorithms to emit false positives, which means either incorrectly identifying the location of defects or, even worse, localizing artificial defects that do not exist in the reality. One of the major sources of false positives in AE source localization is the large number of reflections and reverberations that appear in the tails (codas) of AE signals. This is because most localization algorithms rely only on the time

difference of "first arrivals" in AE waveforms and do not account for the reflections and reverberations generated by geometric features, such as boundaries, joints, stiffeners, and fasteners. In the literature, Hamstad et al. [9–11] have extensively worked on numerical simulations of edge-reflected AE and have demonstrated how such reflections affect AE waveforms. In addition, Farhangdoust et al. [12] worked on numerical simulations for stiffened rectangular plates. Alternatively, other researchers have taken an experimental approach [13,14]. In particular, Carpenter and Gonnan [14] reported AE waveform in an aluminum (7075-T651) stiffened wing panel subject to cyclic fatigue tests and eventually yielding.

A common way to overcome the sophistications imposed by reflections and reverberations is to use many sensors and limit the localization to the area covered by the sensors. However, this approach can significantly increase the complexity and cost of the AE-based SHM system. Instead, some researchers have leveraged the additional information conveyed by the reflections and reverberations to improve the localization accuracy. For example, Achdjian et al. [15] used a statistical approach to localize AE sources in a simple aluminum plate. In particular, they used the reverberation patterns of guided ultrasonic waves recorded by at least three sensors. Ernst et al. [16] took a finite element approach to find the location of AE sources by propagating backward the waveforms recorded by a laser Doppler vibrometer. This approach required six hours of computation for each source localization.

Another major source of false positives in AE source localization is the multimodal and dispersive characteristics of AE waveforms. In thin plate-like structures, AE sources excite guided ultrasonic waves, specifically the Lamb waves. In fact, one can hardly find any AE application in plate-like structures that is not based on the Lamb waves. The problem is that most AE source localization algorithms use either of the fastest propagating Lamb wave mode (first symmetric mode, S_0) or the higher amplitude mode (depending on the source type it could be either of the first symmetric mode, S_0, or the first anti-symmetric mode, A_0). As a result, they ignore the multimodal, and, in some cases, dispersive characteristics of AE waveforms. In contrast, some researchers have leveraged such characteristics to reduce the number of sensors required for localization [17–19]. For example, Holford and Carter [19] used the far-field separation of Lamb wave modes in a 50-m long I-beam to estimate the source-to-sensor distance with only one sensor.

To reduce the number of sensors and enhance the accuracy of source localization algorithms, the authors have leveraged both the reverberation patterns of AE waveforms as well as their dispersive and multimodal characteristics [20–22]. In particular, they developed an analytical model named "Multipath ray tracking" [23] to simulate the reverberation patterns of AE waveforms. Their model reconstructs AE waveforms based on experimentally recorded first arrivals. The authors used this model to localize AE sources in an isotropic plate with only a single AE sensor [20]. They later quantified the uncertainty of this single sensor AE source localization [21]. However, their work was applied to a simple plate without any stiffener or fastener.

To extend the previous work to plates with geometric features (e.g., stiffeners, rivets, etc.) and unknown material properties, this paper proposes a new data-driven approach based on deep learning. The main idea is to use deep learning to directly learn the reverberation patterns, multimodal characteristics, and dispersive properties of AE waveforms from a set of previously collected AE data. The goal here is to use only one sensor and localize AE sources within plates that have stiffeners and rivet connections [24]. In particular, this paper focuses on AEs that are due to a sudden change in the strain field around the rivet connections of plate-like structures. Such sudden changes could be due to the progression of fatigue cracks that tend to grow from rivet connections in metallic plate-like structures. Furthermore, the paper considers AE sources on the surface and at the edges of plate-like structures. To experimentally simulate such AE sources, this paper uses Hsu–Nielsen sources [25]. Then, it uses such sources to train two deep learning algorithms and map the resulted AE waveforms to the location of their sources. In this study, the localization is zonal, which means the structure is divided into multiple zones, and the zone in which an AE source occurs is detected. For example, to localize the fatigue cracks that tend to grow from the rivet connections, the area surrounding

each rivet may be defined as a zone. Beside zonal source localization, what sets the current paper fundamentally apart from the previous work by the authors [20,21] is the use of data-driven methods (i.e., deep learning) as opposed to analytical models (i.e., Multipath ray tracking).

Deep learning is a data-driven approach that eliminates the need for extracting manually designed, application-specific features from data. In the context of AE, one example feature is the time of arrival, which is traditionally used in time difference of arrival (TDOA) methods for source localization [4]. In other words, the end-to-end architecture of deep learning allows it to be directly applied to data (i.e., signals, images, etc.) rather features extracted from the data. In fact, deep learning automatically learns and extracts representative features from data. In this way, deep learning also achieves a better performance than the traditional feature-based algorithms [26,27]. In the literature, artificial neural networks, which are feature-based machine learning algorithms, have been applied to both AE source localization and characterization [28,29]. However, little-to-no research has used deep learning to localize AE sources in plate-like structures. It worth mentioning that deep learning has been recently used for AE-based fault diagnosis in gearboxes and bearings [30–32]. Nevertheless, to the best of authors' knowledge, none of such studies has used deep learning for AE source localization. To fill this gap, this paper uses two types of deep learning networks for AE source localization: (1) stacked autoencoders [33], and (2) convolutional neural networks [34]. In particular, the networks are used to identify (classify) the zone at which an AE source is generated. Since deep learning requires training data, this paper focuses on embedded and permanently attached monitoring systems that are trained once before deployment.

The organization of the subsequent sections of the paper is as follows. First, Section 2 reviews the theoretical aspects of the deep learning approaches used in this study. Then, Section 3 applies the deep learning networks to the problem of AE source localization. Sections 4 and 5 respectively include the experimental setup used to train, validate, and test the deep learning networks as well as the results obtained from them. Finally, concluding remarks are provided in Section 6.

2. Deep Learning Architectures

Deep learning uses neural networks that have multiple hidden layers [26,27]. Similar to traditional neural networks, deep learning networks consists of a series of learnable neurons that nonlinearly map inputs to outputs. However, the input to deep learning networks are raw signals and images rather features extracted from them. In other words, such networks automatically learn the most meaningful features directly from the signals and images.

Since the additional hidden layers of deep learning networks significantly increase the number of their tunable parameters, several deep learning architectures have been developed to keep the training process manageable. This paper, in particular, briefly reviews stacked autoencoders [33] and convolutional neural networks [34].

2.1. Stacked Autoencoders

Stacked autoencoders are deep neural networks that consist of multiple pre-trained layers [27,33]. Each layer of such networks is trained as a part of another neural network, which is named an autoencoder. The following subsections define the layers of a stacked network of autoencoders in details and explain their training procedure.

2.1.1. Autoencoders

Autoencoders are neural networks that reconstruct their input at their output [27]. A typical two-layer autoencoder consists of an encoder layer and a decoder layer (see Figure 1). The first layer of such networks maps (encodes) the input to a lower dimensional space, and the second layer maps (decodes) this compressed representation of the input data back to the original input space. In this way, an autoencoder automatically learns a compressed representation of its input. This compressed representation is called "features". Since the input and output of an autoencoder are

the same, this learning process is unsupervised, which means it does not need any labeled training data. In the context of AE source localization, labels could be the source coordinates/zone associated with each waveform.

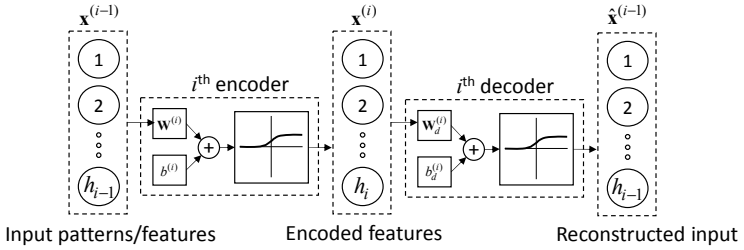

Figure 1. An autoencoder aims at reconstructing its input at its output.

Stacked autoencoders feed the encoded features of an autoencoder to another autoencoder for further compression. Let $\mathbf{x}^{(i-1)}$ be the input to the ith layer of a stacked network of autoencoders. Then, the encoder of the ith autoencoder, maps $\mathbf{x}^{(i-1)}$ to a lower dimensional space $\mathbf{x}^{(i)}$:

$$\mathbf{x}^{(i)} = f(\mathbf{W}^{(i)}\mathbf{x}^{(i-1)} + b^{(i)}). \tag{1}$$

In this equation, $\mathbf{W}$ and b are the weights and the bias of the encoder, respectively. The values of $\mathbf{W}$ and b are determined during the unsupervised training process of an autoencoder. In this notation, scalars, vectors, and matrices are indicated by a lower case italic font, a lower case bold roman font, and an uppercase roman font, respectively. In Equation (1), f is the activation function of the encoder. A typical activation function for autoencoders is a sigmoid function, which is used in this study:

$$f(x) = \frac{1}{1+e^{-x}} \tag{2}$$

The decoder of the ith autoencoder, maps $\mathbf{x}^{(i)}$ to the original feature space of $\mathbf{x}^{(i-1)}$:

$$\hat{\mathbf{x}}^{(i-1)} = f(\mathbf{W}_{\mathrm{d}}^{(i)}\mathbf{x}^{(i)} + b_{\mathrm{d}}^{(i)}), \tag{3}$$

where $\mathbf{W}_{\mathrm{d}}$ and b_{d} are the tunable weights and bias of the decoder. In this equation, $\hat{\mathbf{x}}^{(i-1)}$ is the reconstructed version of $\mathbf{x}^{(i-1)}$.

To learn the weights and the bias of an autoencoder, a loss function needs to be minimized. In this paper, a mean squared error is used as the loss function of autoencoders:

$$E = \frac{1}{N}\sum_{n=1}^{N} \|\mathbf{x}_n^{(i)} - \hat{\mathbf{x}}_n^{(i)}\|^2, \tag{4}$$

where N is the number of training samples, and the subscripts indicate the sample number. To minimize the loss function, a scaled conjugate gradient algorithm could be used [35].

2.1.2. Softmax Layer

A softmax layer is a single-layer neural network that classifies its inputs by mapping them into a finite number of output classes. This layer is typically the last layer of a stacked network of autoencoders that performs classification. For a network that consists of M autoencoders, the softmax layer can be represented mathematically as

$$\mathbf{y} = f(\mathbf{W}^{(M+1)}\mathbf{x}^{(M)} + b^{(M+1)}), \tag{5}$$

where $\mathbf{W}^{(M+1)}$ and $b^{(M+1)}$ are the weights and the bias of the softmax layer. The activation function of a softmax layer is named a "softmax function". This differentiable function is defined as

$$f(\mathbf{x}) = \frac{e^{\mathbf{x}}}{\sum e^{\mathbf{x}}}, \tag{6}$$

in which the summation is over the elements of $\mathbf{x}$.

Unlike autoencoders, the training process of a softmax layer is supervised. In other words, this process requires a set of training samples, and for each of them (i.e., for each $\mathbf{x}_n^{(M)}$), the corresponding class $\mathbf{t}_n$ needs to be known. In this notation, all elements of the vector $\mathbf{t}^{(n)}$ are equal to zero except for one of them that indicates the class membership. For example, if $\mathbf{x}_n^{(M)}$ belongs to the jth class, only the jth element of $\mathbf{t}^{(n)}$ is one. To train a soft max layer, a cross-entropy loss function is usually minimized [36]:

$$E = -\sum_{n=1}^{N} (\mathbf{t}_n \cdot \ln \mathbf{y}(\mathbf{x}_n^{(M)})), \tag{7}$$

2.1.3. Fine-Tuning

Fine-tuning is the process of updating the weights and biases of an entire deep learning network. Fine-tuning is usually performed after all layers of the network are individually trained. In this process, the pre-trained values for the weights and biases are used as initialization for minimizing the cross-entropy loss function of the entire network:

$$E = -\sum_{n=1}^{N} (\mathbf{t}_n \cdot \ln \mathbf{y}(\mathbf{x}_n^{(1)})), \tag{8}$$

2.2. Convolutional Neural Networks

Convolutional neural networks are deep learning networks that take images as input. Unlike traditional neural networks that their layers are one dimensional, each layer of a convolutional neural network has three dimensions: width, height, depth (see Figure 2a). For example, the input layer of a network that takes color images has two dimensions for the width and height of the input images and the third dimension for its color channels. In the consecutive layers of a convolutional neural network, from the input layer to the output layer, the width and height of the layers gradually decrease, but their depth increases. This decrease continues in such a way that the width and height of the output layer are equal to one.

In convolutional neural networks, unlike traditional neural networks that each neuron is fully connected to all neurons in the previous layer, the neurons are only connected to a small region in their previous layer. This region is named the "receptive field" of the neuron (see Figure 2). This architecture allows the number of tunable parameters (i.e., weights and biases) remain manageable even for large input images. This particular feature makes the training time of the convolutional neural networks less sensitive to the size of the input data than the staked autoencoders.

Figure 2c visualizes two parameters that control receptive fields: stride and zero-padding. In particular, the stride parameter defines the distance between the receptive fields of two neighbor neurons. Zero-padding is another parameter that controls the interactions of the receptive fields with the edges of the previous layer. Specifically, this parameter defines the number of added zeros to the edges of the previous layer. Both stride and zero-padding are non-tunable parameters that remain constant during the learning process.

Each layer of a convolution neural network consists of multiple channels (see Figure 2). The numbers of channels define the depth of a layer. For each channel, there is a dedicated image processing filter that its output defines the neuron values. Depending on the type of the layer,

the filter performs different tasks. The following describes the most common types of layers uses in a convolution neural network.

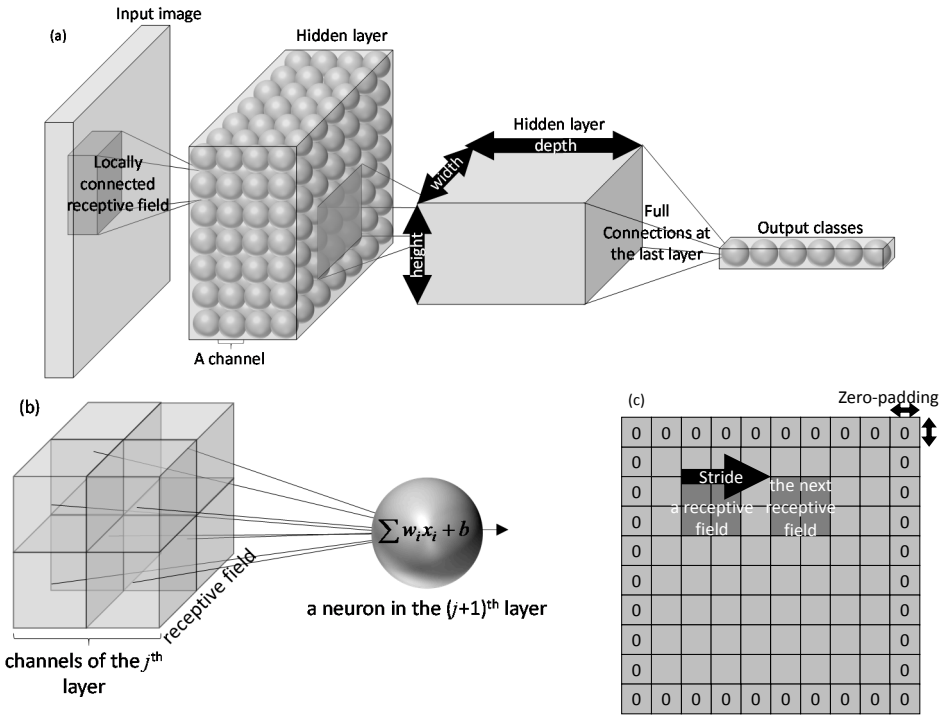

Figure 2. Convolutional neural networks: (**a**) conceptual architecture, (**b**) a neuron with a 2 × 2 receptive field on a two-channel layer, and (**c**) zero-padding and stride on a channel.

2.2.1. Convolutional Layer

Each channel of a convolutional layer applies the following filter to the receptive fields of its neurons:

$$\mathbf{x}^{(i)} = f(\mathbf{W}^{(i)}\mathbf{x}_r^{(i-1)} + b^{(i)}). \tag{9}$$

In this equation, $\mathbf{x}$, $\mathbf{W}$, b, and f are, respectively, the value of neurons as well as their weights, bias, and activation function. It is important to note that $\mathbf{x}$ here is a three-dimensional data structure, in which the first two dimensions are the height and width of the image, and the third dimension is for the number of channels. In this notation, $\mathbf{x}_r^{(i-1)}$ is the receptive field of the ith layer (see Figure 2a). A receptive field $\mathbf{x}_r$ is a subset of $\mathbf{x}$ in the first two dimensions, but as Figure 2b shows, it includes all the channels in the third (depth) dimension. The number of channels in a convolutional layer is another non-tunable parameter (other than the stride and zero-padding) that needs to be defined at the beginning and does not change during the learning process.

A commonly used activation function in most convolutional neural networks is a "rectified linear unit (ReLU)" [37]. A ReLU function nonlinearly maps each negative element x in the input vector $\mathbf{x}$ to zero:

$$f(x) = \begin{cases} x & x > 0 \\ 0 & otherwise \end{cases} \tag{10}$$

2.2.2. Max-Pooling Layer

Max-pooling layers have the same number of channels as their input layer. In this type of layer, the receptive fields are two dimensional and only apply to the height and width (not depth) of

the corresponding channel in the previous layer. The neurons of a max-pooling layer calculate the maximum value in their receptive field:

$$\mathbf{x}^{(i)} = \max(\mathbf{x}_r^{(i-1)}). \tag{11}$$

Max-pooling layers usually use a stride value equal to two to reduce the height and width of their inputs. In this way, max-pooling layers down-sample their inputs.

2.2.3. Fully Connected Layer

A fully connected layer is one of the last layers in a convolutional neural network. This layer reduces the height and width of its inputs to one. The construction of a fully connected layer is similar to a convolutional layer, but it has two main differences: (1) the receptive field of a fully connected layer has the same height and width as its input layer. (2) The neurons of a fully connected layer do not apply a ReLU function. For example, in a classification problem, which is the case in this paper, such neurons apply a "softmax" function instead (see Equation (6)).

2.2.4. Visualizing the Inception of a Convolutional Neural Network

Images that strongly activate a specific channel in a layer of a convolutional neural network represent the inception of that channel. To generate such images, which are generally named "deep dream" images, one may use stochastic optimization and find an image that maximizes the activation [38]. In particular, the optimization starts with a random noisy image and iteratively changes the image to increase the activation. In this process, a priori statistics constrain the optimization to produce images with similar statistics to natural images. For the final layer of convolutional neural networks, since each neuron corresponds to an output class, the deep dream images visualize the way that the entire network perceives that class.

Occluding a part of the input image is another way to visualize the inception of a convolutional neural network (see Figure 3) [39]. This technique tests the performance of a network on images that are partially occluded. Then, the test is repeated after slightly moving the occlusion. In this way, this technique produces a map of areas in the image that are the most sensitive to the occlusion. This map visualizes specific areas within images that the convolution neural network relies upon to classify an image.

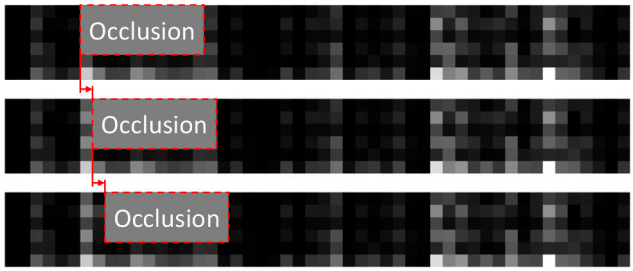

Figure 3. A moving occlusion window partially covers test images while the accuracy of the convolutional neural network is being evaluated.

2.3. Overfitting Mitigation in Training Deep Netwoprks

Almost all machine learning algorithms, including neural networks, are susceptible to overfitting. Overfitting means that the network is so specialized to the training examples that it cannot generalize its input-output map to an unseen dataset. In this case, the network achieves a minimum loss function only for the training dataset rather any unseen data.

Several approaches exists to avoid overfitting: regularization, cross-validation, and dropout [27,36,40]. In particular, regularization ensures generalization by penalizing large weights and biases. Since such

large values specialize the deep learning network to specific patterns, enforcing smaller weights and biases prevents overfitting to such patterns [36].

Cross-validation is another approach to avoid overfitting [36]. This approach withholds a subset of training dataset, which is named the "validation set", from the gradient descent algorithm. Cross-validation uses the value of the loss function on the validation set as the stopping criteria for the training procedure. Therefore, the gradient descent algorithm uses the training set to update the weights and biases of the network, but it stops when the loss function is minimized on the validation set.

Dropout is another regularization technique mostly used in convolutional neural networks. This technique adds an additional layer to the network. The added layer randomly ignores some of its input neurons during the training process [40]. In other words, this technique forces other neurons to step in and make predictions instead of the missing neurons. In this way, the network becomes less sensitive to any specific neuron, and the dropout layer makes it less likely for the network to overfit to the training data. To be more effective, the dropout layer is usually applied before the fully connected layer of a convolution neural network.

3. Acoustic Emission Source Localization with Deep Learning

This study uses deep learning to identify the zone in which AE occurs. To achieve this goal, deep learning uses a set of AE waveforms and their corresponding source locations as training examples and constructs a nonlinear map between the waveforms and the source locations. In this process, deep learning leverages the reverberation patterns as well as the multimodal and dispersive characteristics of AE waveforms to determine their source location.

Deep learning has the advantage of leaning directly from signals and images, rather features extracted from them. In the context of AE source localization, this eliminates the need to extract features, such as time of arrival. To leverage multimodal and dispersive characteristics of AE waveforms, this study applies deep learning directly to a time-frequency transform of AE waveforms. In particular, a continuous wavelet transform is used.

The wavelet transform is widely used in various structural health monitoring applications [41–47]. Let $r(t)$ be an input signal. The wavelet coefficients are defined as:

$$C(f,\tau) = \frac{1}{\sqrt{s(f)}} \int_{-\infty}^{+\infty} r(t)\Psi^*\left(\frac{t-\tau}{s(f)}\right)\mathrm{d}\tau. \tag{12}$$

In this equation, τ is the translation parameter, s is the non-dimensional scale parameter defined as $s(f) = f_c \cdot f_s / f$, and Ψ^* is the complex conjugate of the mother wavelet $\Psi(t)$. This study uses a complex Morlet mother wavelet:

$$\Psi(t) = \frac{1}{\sqrt{\pi f_\mathrm{b}}} \exp\left(2\pi f_\mathrm{c} j t - \frac{t^2}{f_\mathrm{b}}\right). \tag{13}$$

The following subsections explain how stacked autoencoders and convolutional neural networks may use the continuous wavelet coefficients of AE waveforms to localize AE sources.

3.1. Stacked Autoencoders

The input to stacked autoencoders is a one-dimensional signal. To construct a multi-frequency representation of AE waveforms, this paper converts the modulus of the wavelet coefficients to a one-dimensional pattern (see Figure 4). In particular, a few frequencies are selected from the most dispersive and high-amplitude frequency range of AE waveforms. In this study, this range approximately starts from 25 kHz and ends at 500 kHz. While selecting more frequencies will feed more information to the stacked autoencoders, it will also increase the size of the input pattern and

hence the computation time. To balance this trade-off, this study uses three frequencies. Specifically, 75 kHz, 200 kHz, and 325 kHz are selected to respectively represent the low-, mid-, and high-frequency contents in this range. Then, a fixed 500 μs-long window (starting from −10 μs to 490 μs) is used to resample the moduli of the wavelet coefficients. The size of the temporal window is selected in such a way that includes multiple reflections from the geometric features of the structure. While a longer window will include more reflections, it will increase the size of the input pattern to deep learning networks and hence the computation time. In other words, here there is another tradeoff between the accuracy and computation time, which, in this case, is balanced with a 500 μs-long window. The resampling is performed at frequencies that are as twice as the three wavelet frequencies (i.e., 37.5 kHz, 100 kHz, and 162.5 kHz, respectively). Finally, the concatenation of the resampled data points constructs the input to the deep learning network. In this study, this multi-frequency representation of AE waveforms consists of 149 data points that are normalized to have the maximum value of one (see Figure 4b).

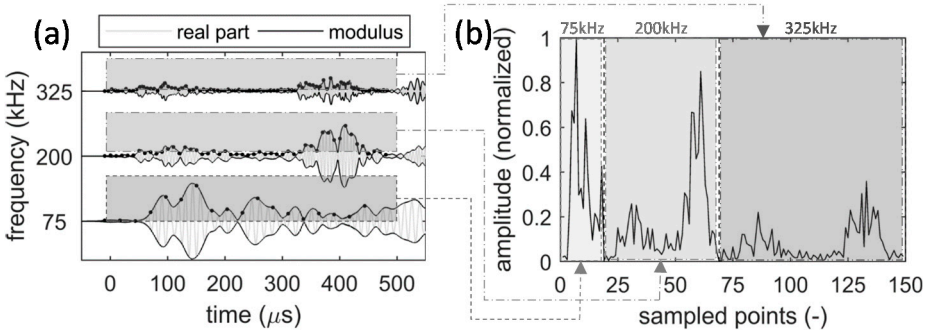

Figure 4. Across-frequency resampling of the modulus of the wavelet coefficients at 75 kHz, 200 kHz, and 325 kHz: (**a**) the real part and modulus of the wavelet coefficients for an AE waveform (resampling is indicated with dots); (**b**) the multi-frequency representation of the AE waveform.

The stacked autoencoders used in this study consists of two autoencoders and a softmax layer (see Figure 5). In this network, the autoencoders compress the input patterns first into 40 encoded features and then into 15, further compressed features. Since the goal is to identify the zone in which simulated fatigue cracks generate AE, a softmax layer is used to localize the AE source. The input to the softmax layer is the encoded features by the second autoencoder, and the output is the zone number. In other words, this layer classifies AE waveforms into a finite number of classes that each correspond to a zone of the structure. Specifically, the output is a vector that the values of its elements are negligible except for one of them, which indicates the zone number. As Figure 5 shows, the number of zones is indicated by the parameter Z.

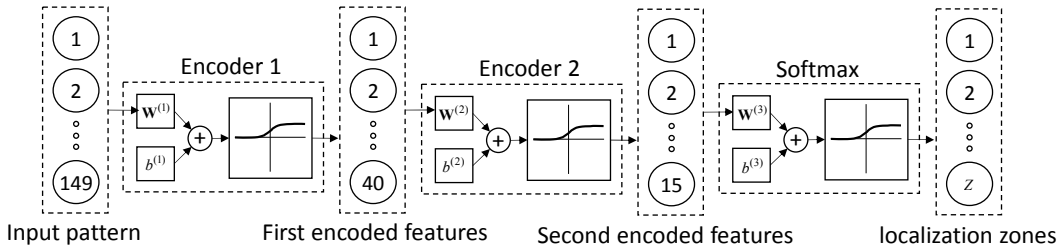

Figure 5. A stack of two autoencoders and a softmax layer. This architecture is used to find the zone in which an AE source occurs.

3.2. Convolutional Neural Networks

The input to convolutional neural networks is a multi-dimensional signal, which is usually a two-dimensional image. This allows directly using the modules of the wavelet coefficients as input to this deep learning network. As Figure 6 shows, this study normalizes the moduli of wavelet coefficients and converts it to an input image. In this process, wavelet coefficients are calculated at multiple frequencies, which are selected from the most dispersive and high-amplitude frequency range of AE waveforms (in this study, 25 kHz to 500 kHz). Since convolutional neural networks are less sensitive to large inputs than stacked autoencoders, this study uses six frequencies. As discussed in Section 2.2, this flexibility is due to the deployment of "receptive fields" in convolutional neural networks. In particular, the six frequencies start with 75 kHz and with the steps of 75 kHz end at 450 kHz. These six frequencies constitute the vertical axis of the input image. In the time domain, the horizontal axis of the input image corresponds to the same time window used for the stacked autoencoders (−10 μs to 490 μs). However, for this axis, the modules of the wavelet coefficients are calculated every 10 μs, which makes the size of the input image 6×50.

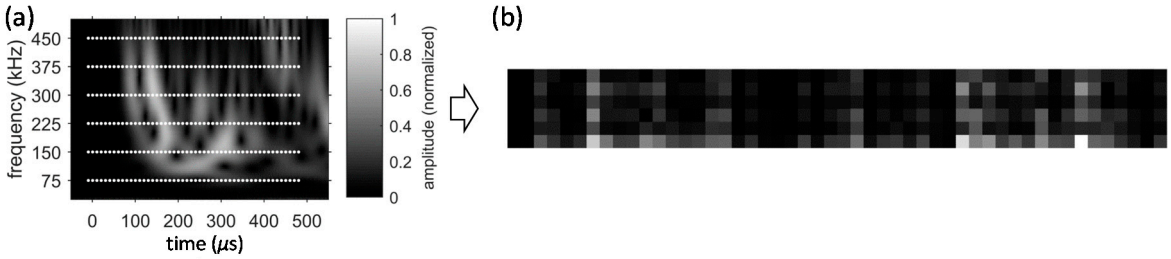

Figure 6. The process of producing the input image to the convolutional neural network: (**a**) a normalized continues wavelet transform, sampled pixels are overlaid, (**b**) the constructed image.

The convolutional neural network of this study has three convolutional layers and two max pooling layers (see Figure 7). The two types of layers are alternatively arranged to gradually reduce the height and width of the images while increasing their depth (i.e., the number of channels). The last convolutional layer is followed by a dropout layer followed by a fully connected layer and a softmax layer. The purpose of these layers is to prevent overfitting, dimensionality reduction, and classification, respectively.

All convolutional layers pad a zero pixel in all four directions (i.e., left, right, top, and bottom). Since the height and width of all convolutional filters is 3×3, this zero-padding ensures that the height and width of images remain the same before and after the convolutional layers. However, the number of the filters used in the convolutional layers gradually increase the depth of images (i.e., their number of channels). In particular, the convolutional layers increase the depth from 1 to 16, and then to 32, and eventually to 64.

Max pooling layers down-sample the images. In particular, a two-pixel stride reduces the height and width of the images by half. However, the depth of the images remains the same before and after the max pooling layers. In this study, no zero-padding is used for the max pooling layers.

Unlike convolutional and max pooling layers, as Figure 7 shows, the dropout layer does not change the size of the images. However, the fully connected layer reduces their width and height to one while matching the depth with the number of output classes. Since this study performs zonal localization, the number of output classes are the same as the number of considered zones. To find the neuron with the highest activation in the fully connected layer, the network uses a softmax layer as the last layer. This layer classified the AE waveforms into multiple classes the each correspond to a zone of the structure.

Convolutional neural networks can nonlinearly map their inputs to their outputs. The sources of nonlinearity in the network used in this study can be classified into three groups: (1) the rectified linear

units (ReLU) that are used as the activation function of the convolutional layers, (2) the down-sampling performed in the max pooling layers, and (3) the final softmax layer.

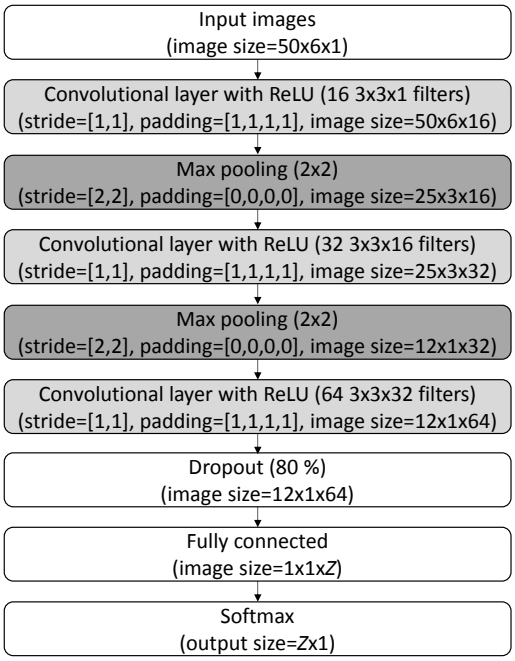

Figure 7. The architecture of the convolutional neural network used to discriminate AE sources. The parameter Z here indicates the number of zones.

4. Experiments

Figure 8 shows the experimental setup used to evaluate the effectiveness of the proposed deep learning approaches. In particular, the specimen was a 6061-T6 aluminum plate (914.4 mm × 914.4 mm × 3.2 mm). To simulate a realistic plate-like structure with a stiffener, a one-inch-wide aluminum strip (the same material and thickness) was fastened to the back of the plate with five 1/4" (6.35 mm) rivets (see Figure 8b). The rivets are numbered in Figure 8a. To collect AE waveforms, the plate was instrumented with only one AE sensor (PICO, Physical Acoustics Corporation) (see Figure 8a). The main reason to select the PICO sensor was its broad-band frequency response. Based on the recommendation of previous studies, the location of the sensor was selected away from the symmetry lines of the plate [20,21]. Specifically, it was attached at the coordinates (63.5 mm, 190.5 mm) relative to the lower left corner of the plate. It is important to mention that the sensor could have been affixed to any other location on the plate if it had been offset against symmetries. However, symmetric locations, such as the center of the plate, should be avoided. To fix the sensor in its place, hot glue was used.

In order to simulate fatigue cracks that usually initiate from rivets and fastener holes, 416 Hsu-Nielsen pencil lead break tests were performed [25]. In particular, thirteen zones were considered on the plate (see Figure 8a), and 32 Hsu–Nielsen sources were simulated in each. In the first five zones, as Figure 8c shows, 32 AE sources were simulated next to each rivet connection. Specifically, the tests included the four sides of each rivet (right, left, top, and bottom) and four distances from the edge of each rivet: 0.8 mm, 1.6 mm, 3.2 mm, and 6.4 mm. For each of these 16 combinations, the Hsu–Nielsen tests were repeated twice. Zones six to nine were square areas that had a two-inch clearance from the edges and the center lines of the plate. In each of these four zones, 32 Hsu–Nielsen AE sources were simulated. In each zone, simulations were performed at a 4 × 4 grid and at each grid intersection, the tests were repeated twice. In zones ten to thirteen, Hsu-Nielson sources were simulated at the edge of the plate. In each zone, 32 pencil lead break tests were performed at sixteen locations. Specifically, the tests were spaced by a two-inch distance, and each test was repeated

twice. In terms of data acquisition, the AE signals were first amplified by 40 dB and then filtered by a 5 kHz–1 MHz band-pass analog filter before being digitized at the sampling frequency of 5 MHz. In addition, during the post-processing in MATLAB, a digital band-pass filter (Butterworth) was used to limit the frequencies to 25 kHz–500 kHz. Finally, the AE waveforms and their corresponding zone number were randomly divided into training, validating, and testing sets. In particular, 80%, 10%, and 10% of the data was used for training, validation, and testing, respectively. In this study, two scenarios were considered: (1) AE sources only in the first five zones (at the vicinity of rivet connections) and (2) and AE sources in any of the thirteen zones. In the first experimental scenario, the data set includes 160 simulated sources: 120 for training, 16 for validation, and 16 for testing. In the second the data set includes all 416 Hsu-Nielsen sources: 332 for training, 42 for validation, and 42 for testing. To allow comparison between the stacked autoencoders and the convolutional neural network, in each experimental scenario, a similar randomization was used to divide the data into training, validation, and testing sets. However, the randomization used for the first and second scenarios are different because the size of the two databases is different.

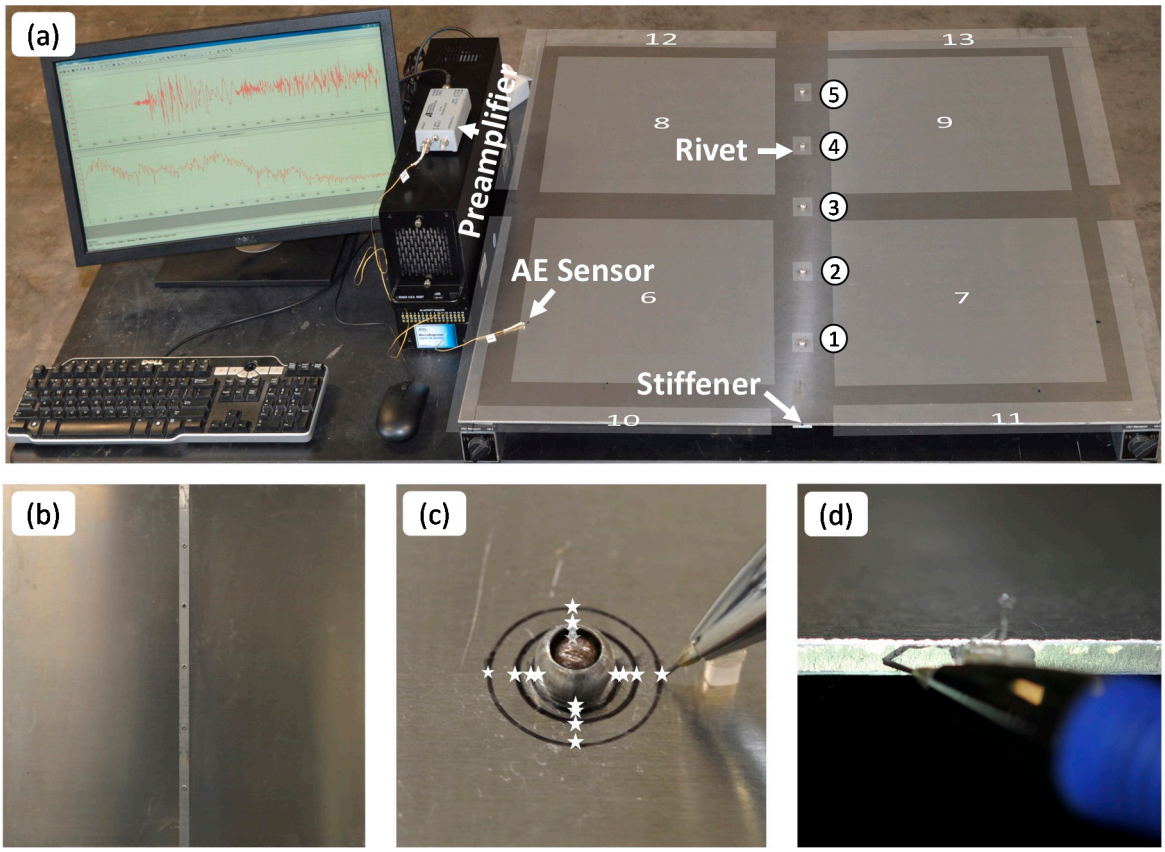

Figure 8. (**a**) Experimental setup, thirteen zones are labeled in the image; (**b**) stiffener on the back of plate; (**c**) a pencil lead break test next to a rivet (zones 1–5); (**d**) a pencil lead break test at the edge of the plate (zones 10–13).

5. Results

This section presents the results obtained from the stacked autoencoders and the convolutional neural network. For the first experimental scenario, the results obtained during the training phase of both deep learning networks as well as final zonal localization results are presented. In this experimental scenario, the focus is only on the first five zones. These zones correspond to the AE source simulated near the five rivet connections. Figure 9 shows samples of AE waveforms used to train, validate, and test the two deep learning approaches. Deep learning leverages the difference

between AE waveforms to define a map between them and their corresponding source location. For the second experimental scenario, only the final localization results and the required computational time are discussed.

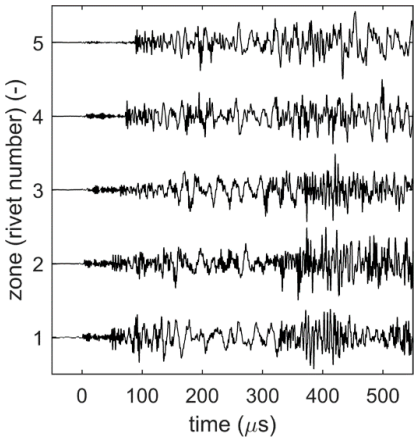

Figure 9. Samples of AE waveforms simulated at the rivets (the first five zones). To allow comparison, the waveforms were normalized.

5.1. Stacked Autoencoders

Figure 10 visualizes the learning curves of the stacked autoencoders used in the first experimental scenario of this study. In particular, Figure 10a,b correspond to the unsupervised training of the first and second autoencoders, Figure 10c belongs to the supervised training of the softmax layer, and Figure 10d corresponds to the supervised fine-tuning of the entire deep learning network. All four cases use a scaled conjugate gradient algorithm [35] for training. While the autoencoders use a mean square error as their loss function (see Equation (4)), a cross-entropy loss function was minimized for the softmax layer and the entire network (see Equations (7) and (8)). To avoid overfitting, a weight decay regularization term was added to the loss functions. In addition, in all four cases, the optimization was stopped when the global minimum was achieved on the validation data. In the graphs, the global minimum is indicated as the "best" results. For example, the global minimum for the first autoencoder was reached after 448 steps of training.

Figure 11 shows the input and output of the autoencoders for a randomly selected pattern from the testing dataset of the first experimental scenario. Since this specific pattern was never used in the training and validation processes, it is considered new data. Figure 11a,b respectively correspond to the first and second autoencoders. While the first autoencoder reconstructs the 149 input patterns, the second autoencoder reconstructs the 40 encoded features by the first autoencoder. Both figures demonstrate that the two autoencoders can successfully reconstruct their input. In addition, Figure 11c compares the input patterns with their reconstructed version by the combination of the two autoencoders. This figure, in particular, also demonstrates negligible information loss after two layers of encoding and decoding. To produce the reconstruction plot, the output of the second autoencoder was decoded by the first autoencoder.

Figure 12 contains localization results obtained from the stacked autoencoders used in the first experimental scenario. In particular, the deep learning network was tested on 16 randomly selected Hsu–Nielsen pencil lead break tests. These 16 waveforms consist of respectively five, one, three, three, and four AE sources at the first to fifth rivet connections. The confusion matrix shows that the network successfully identified the corresponding rivets to all 16 tests. For example, the first entry of the matrix reads as the testing subset included five randomly selected AE sources that were simulated at the first rivet and all five of them were correctly localized.

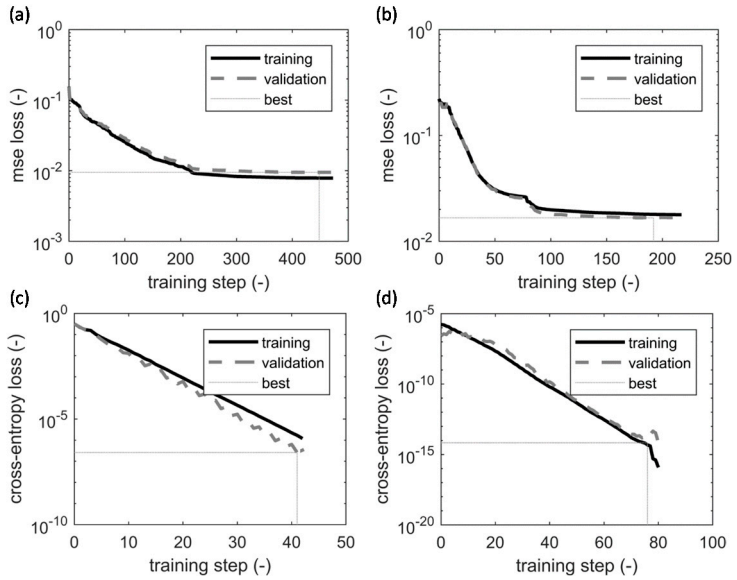

Figure 10. Learning curves: (**a**) the first autoencoder; (**b**) the second autoencoder; (**c**) the softmax layer;(**d**) the stacked deep learning network.

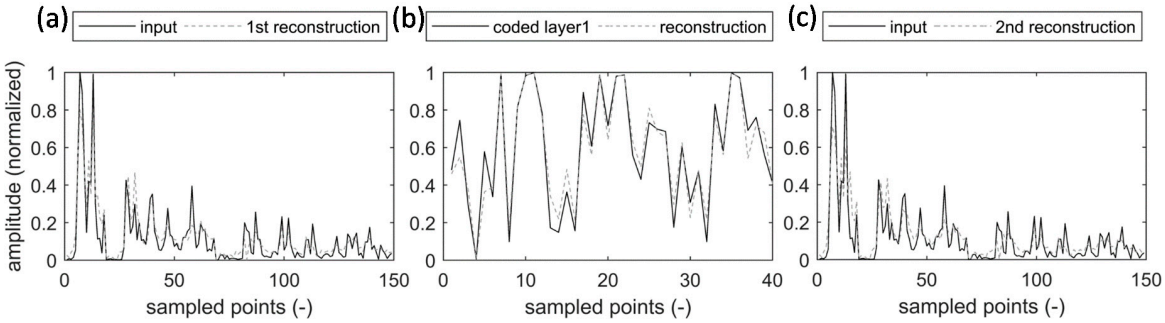

Figure 11. The input and output patterns of autoencoders: (**a**) the first autoencoder; (**b**) the second autoencoder; (**c**) reconstruction of the original input patterns after two layers of encoding and decoding.

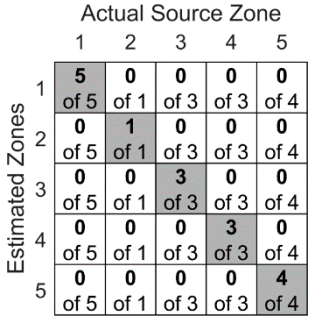

Figure 12. The confusion matrix of the stacked autoencoders in zonal localization of the first experimental scenario. Localization zone here is the closest rivet to the AE source.

Figure 13 shows the confusion matrix of the stacked autoencoders in the zonal localization of the second experimental scenario. Similar to the first experimental scenario, the stacked autoencoders were able to localize all AE sources. In this case, the testing set of the second experimental scenario included 42 AE sources that were randomly selected from a database of 416 simulated AE sources. It is important to note that the randomization used for the first and second experimental scenarios were different because each contained a different number of AE sources.

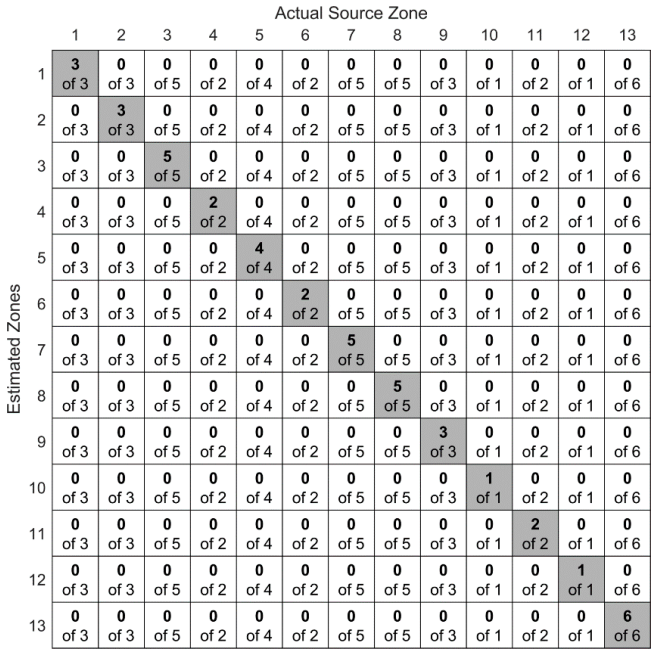

Figure 13. The confusion matrix of the stacked autoencoders in zonal localization of the second experimental scenario.

The time required to train the stacked autoencoders on a core-i5 processor was 28 s. However, it only takes less than 2 milliseconds for a trained network to localize a source.

5.2. Convolutional Neural Networks

Figure 14 visualizes the learning curves of the convolutional neural network. Since the network classifies the AE sources into five rivet locations, a cross-entropy loss was minimized. In particular, a gradient descent with momentum was used [48]. In this study, the learning rate was 0.005 and the momentum contribution was 0.9. To avoid overfitting, in addition to a dropout layer, a weight decay regularization was used. Moreover, cross-validation was used to control the training process.

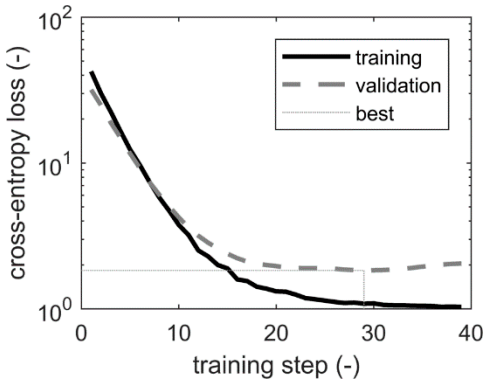

Figure 14. Learning curves of the convolutional neural network.

Figure 15a shows images that activate the final layer of the convolutional neural network the most. The five neurons in this layer correspond to the AE sources that occur near the five rivet connections. These images, which are generally called "deep dreams", represent the inception of a convolutional neural network from the wavelet image of each rivet [38].

Figure 15b shows analytically calculated arrival time for edge-reflected late arrivals that appear in the coda of AE waveforms. Specifically, the arrival times are shown for different frequencies. The time and frequency ranges are the same as the ones used for the input images of the convolutional neural network. Since in plate-like structures AE sources excite guided ultrasonic waves (Lamb waves in particular) and the propagation velocities of these waves are a function of frequency, the arrival time is not the same across frequencies. To calculate late arrivals, the Multipath ray tracking algorithm [23] was used to track the propagation paths of the Lamb waves from each rivet to the sensor (see Figure 16). Then, the dispersion curves of the first symmetric (S_0) and anti-symmetric (A_0) Lamb wave modes were used to convert the propagation distance of each path to its propagation time (i.e., its time of flight). It is assumed that the high-frequency content (425 kHz specifically) of the faster propagating mode (i.e., S_0) triggers the AE system. Accordingly, the time of flights of the higher amplitude mode (i.e., A_0) were converted to the arrival times.

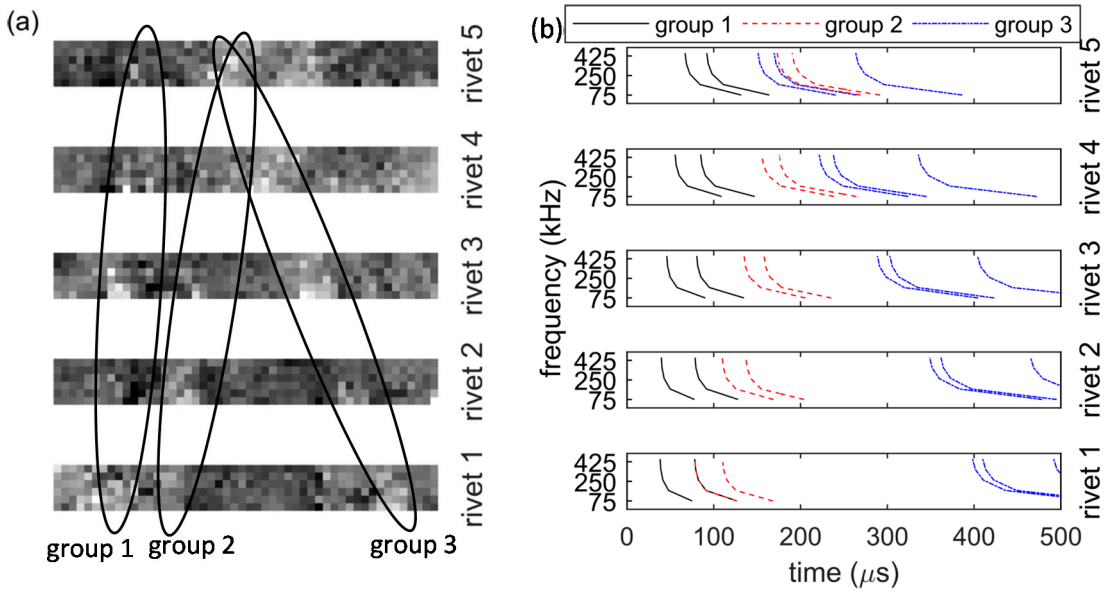

Figure 15. Three arrival groups: (**a**) deep dreams, (**b**) theoretical.

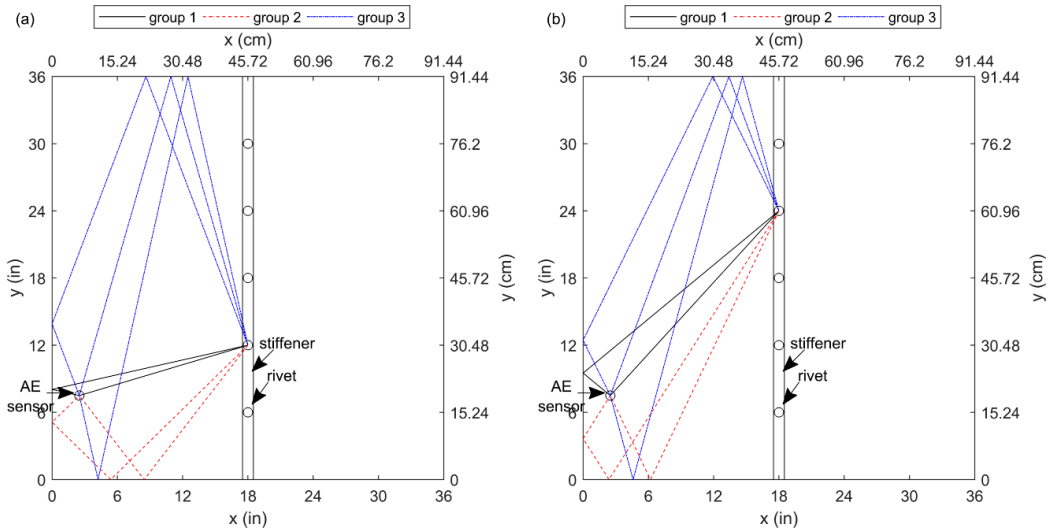

Figure 16. Propagation paths for the left side of the stiffener: (**a**) AE at the second rivet; (**b**) AE at the forth rivet.

Comparing Figure 15a,b, it could be seen that the convolutional neural network has learned the frequency-dependent reverberation patterns that appear in the coda of AE waveforms. Since the sensor was placed away from the lines of the symmetry of the plate, there is always a difference in the arrival time of the reflections that come to the sensor from different edges. Deep learning leverages such time differences and learns how to interpret them in terms of the location of AE sources. If one uses the first boundary of the plate (left, bottom, or top) that reflects the waves in each propagation path to divide the late arrivals into three groups, the arrival time of the three groups match with the high amplitude areas identified in Figure 15a. From the first to last rivet, it takes more time for the first two groups to arrive at the sensor. In addition, the higher the rivet number, the later that the second group arrives than the first group. In contrast with the first two groups, the arrival time of the third group decreases from the first to last rivet.

Figure 16 shows some possible propagation paths that connect the second and fourth rivets to the sensor. The paths are grouped based on their first reflecting boundary. For the sake of simplicity, only paths that interact with the left, bottom, or top boundaries are visualized. In addition, only paths with up to two reflections were considered. This is because the longer propagation paths, on which more reflections may occur, arrive after the 500-μs-long time window considered in this study.

Figure 17 shows the sensitivity map of the convolutional neural network. In this figure, the brighter the color map, the more sensitive the deep learning network to that particular part of the image. To produce these images a moving 10×4 occlusion window was used, and the average accuracy of the network was tested on each of the five rivets. As the figure shows, the network is the most sensitive to the arrival time of the three propagation groups identified in Figure 16. In addition, the network is more sensitive to the lower frequencies. These frequencies correspond to the most dispersive region of the higher-amplitude Lamb wave mode that dominates the AE waveform (i.e., the first anti-symmetric mode). These observations further demonstrate that the convolutional neural network leverages the reverberation of AE waveforms as well as their dispersive behavior.

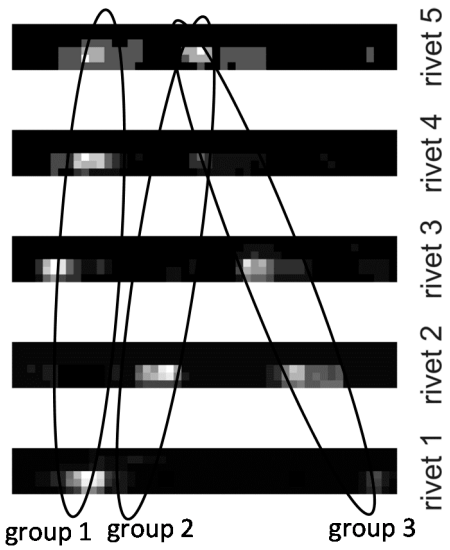

Figure 17. Sensitivity maps for a 10×4 occlusion; the three arrival groups are also indicated.

Figure 18 shows the confusion matrix of the convolutional neural network. The columns and rows of the matrix respectively represent the actual and estimated rivet numbers that are the closest to AE sources. The confusion matrix shows that the convolutional neural network, similar to the stacked autoencoders, successfully localized all AE sources in the testing dataset. For the sake of comparison, the same randomization was used for the two deep learning networks.

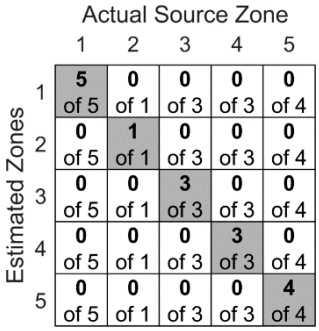

Figure 18. Confusion matrix of the convolutional neural network for the first experimental scenario. Localization zone here is the closest rivet to the AE source.

Figure 19 shows the confusion matrix of the convolutional neural network for the second experimental scenario. Overall, the accuracy of the network was 95.2%. Out of 42 pencil lead break tests, except for two, all simulated AE sources were correctly localized. One of the localization errors was for a source in zone nine (i.e., the top right surface of the plate) that was inaccurately localized in zone five (i.e., the topmost rivet). In this case, the two zones are next to each other. The other error was for a source in zone eight (i.e., the top left surface of the plate), which was confused with a source in zone eleven (i.e., the bottom right edge of the plate).

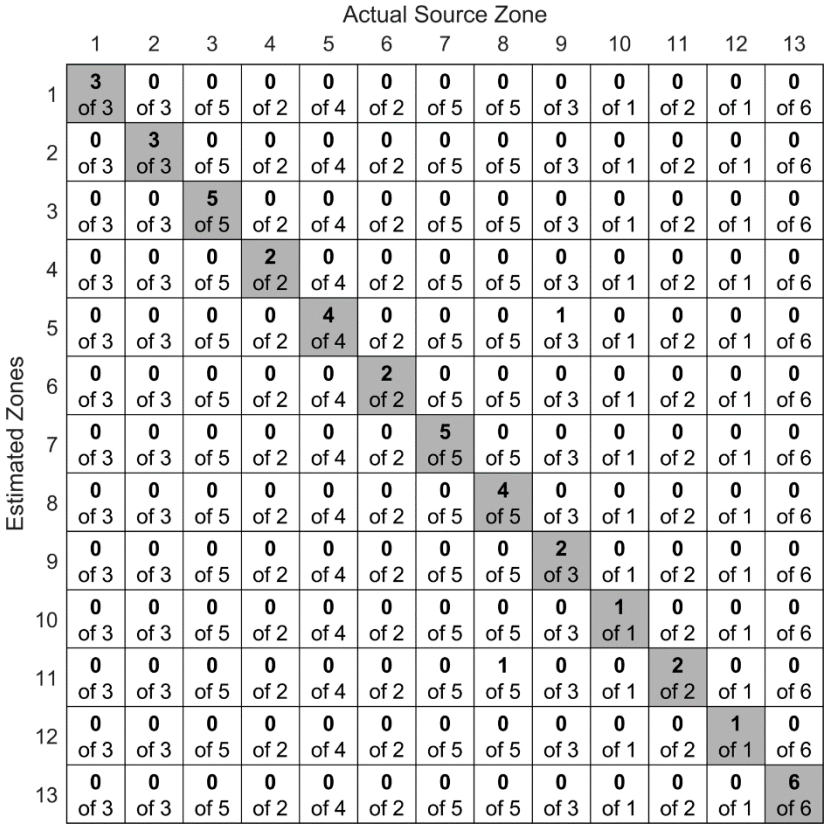

Figure 19. Confusion matrix of the convolutional neural network for the second experimental scenario.

The time required to train the convolutional neural network on a core-i5 processor was 23 s. However, it only takes less than 2 milliseconds for a trained network to localize a source.

6. Discussion and Conclusions

This paper used two deep learning approaches to localize AE sources within plates with geometric features, such as rivet-connected stiffeners. In particular, stacked autoencoders and convolutional neural networks were used. This paper leveraged the reflection and reverberation patterns of AE waveforms as well as their dispersive and multimodal characteristics to localize AE sources with only one sensor. To maximize the information attained by reflections, the sensor was attached to a corner of the plate, and symmetric locations, such as the center of the plate, were avoided. To train, validate, and test the deep learning networks, AE sources were experimentally simulated at the rivet connections of an aluminum plate that had a stiffener. In particular, Hsu-Nielsen pencil lead break tests were used for the simulations. The results showed that both deep learning networks can learn how to map AE waveforms to their sources. These results demonstrate that the reverberation patterns of AE sources contain pertinent information to the location of their sources. Overall, the performance and flexibility of the two deep learning networks were comparable. While the stacked autoencoder achieved a slightly better performance (100% accuracy versus 95.2%), the convolutional neural network was more flexible in accepting more information-rich input in the frequency domain. In particular, six frequencies were used for the convolutional neural network compared to three in the stacked autoencoders.

This paper successfully performed zonal AE source localization. In particular, AE sources that may occur near rivet connections were localized. However, the current paper does not find the coordinates of AE sources. To overcome this limitation, future research may consider replacing the softmax layer of the deep learning networks with a regression layer and using a larger training data [49]. To generate a larger training data, future research should also focus on automating the process of simulating AE sources. For example, numerical simulations and/or robotic solutions could be investigated. While this study leveraged the broad-band frequency response of a PICO sensor to localize AE sources, in future, additional tests need to be performed to evaluate how a flat response would potentially improve source localization with deep learning. In addition, this paper used Hsu–Nielsen tests to simulate fatigue cracks. To verify the performance of the proposed deep learning approaches under actual states of stress, future researchers should perform more formal tests on real propagating cracks. Another option could be Hsu–Nielsen test performed at various depths inside rivet connections. Moreover, the scope of this paper was limited to embedded and permanently attached monitoring systems that require one-time training before deployment. Since deep learning can also learn to generalize over the differences between different sensors and structures, future studies may investigate the idea of training a deep learning network on one structure and deploying the network on another similar structure.

Author Contributions: Arvin Ebrahimkhanlou and Salvatore Salamone conceived and designed the experiments; Arvin Ebrahimkhanlou performed the experiments and analyzed the data; Arvin Ebrahimkhanlou wrote the paper and Salvatore Salamone revised it.

Acknowledgments: This work was supported by the Office of Naval Research under grant number N00014-17-1-2367, Program Director Ignacio Perez.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wilson, H.A.; Goldfein, D.L. *USAF Posture Statement Fiscal Year 2018*; Department of the Air Force: Washington, DC, USA, 2017.
2. *Encyclopedia of Structural Health Monitoring*; Boller, C.; Chang, F.; Fujino, Y. (Eds.) John Wiley and Sons, Ltd.: Chichester, UK, 2009.
3. Nicolas, M.; Sullivan, R.; Richards, W. Large Scale Applications Using FBG Sensors: Determination of In-Flight Loads and Shape of a Composite Aircraft Wing. *Aerospace* **2016**, *3*, 18. [CrossRef]
4. Kundu, T. Acoustic source localization. *Ultrasonics* **2014**, *54*, 25–38. [CrossRef] [PubMed]
5. Kundu, T.; Das, S.; Jata, K.V. Point of impact prediction in isotropic and anisotropic plates from the acoustic emission data. *J. Acoust. Soc. Am.* **2007**, *122*, 2057–2066. [CrossRef] [PubMed]

6. Kundu, T.; Yang, X.; Nakatani, H.; Takeda, N. A two-step hybrid technique for accurately localizing acoustic source in anisotropic structures without knowing their material properties. *Ultrasonics* **2015**, *56*, 271–278. [CrossRef] [PubMed]
7. Sen, N.; Kundu, T. A new wave front shape-based approach for acoustic source localization in an anisotropic plate without knowing its material properties. *Ultrasonics* **2018**. [CrossRef] [PubMed]
8. Dubuc, B.; Ebrahimkhanlou, A.; Salamone, S. Simultaneous Localization and Classification of Acoustic Emission Sources in Plates Using a Guided Wave-Based Sparse Reconstruction. In Proceedings of the 11th International Workshop on Structural Health Monitoring, Real-Time Material State Awareness and Data-Driven Safety Assurance, Stanford, CA, USA, 12–14 September 2017; Chang, F.-K., Fotis, K., Eds.; Destech Publications: Stanford, CA, USA, 2017; Volume 1, pp. 1779–1787.
9. Prosser, W.H.; Hamstad, M.A.; Gary, J.; O'Gallagher, A. Reflections of AE Waves in Finite Plates: Finite Element Modeling and Experimental Measurements. *J. Acoust. Emiss.* **1999**, *17*, 37–47.
10. Hamstad, M.A.; Gallagher, A.O.; Gary, J. Effects of lateral plate dimensions on acoustic emission signals from dipole sources. *J. Acoust. Emiss.* **2001**, *19*, 258–274.
11. Hamstad, M.A.; Downs, K.S.; O'Gallagher, A. Practical aspects of acoustic emission source location by a wavelet transform. *J. Acoust. Emiss.* **2003**, *21*, 70–94.
12. Farhangdoust, S.; Younesian, D.; Esmailzadeh, E. Interaction of Higher Modes in Nonlinear Free Vibration of Stiffened Rectangular Plates. In Proceedings of the ASME 29th Conference on Mechanical Vibration and Noise, Cleveland, OH, USA, 6–9 August 2017; ASME: Cleveland, OH, USA, 2017; Volume 8, p. V008T12A043.
13. Bhuiyan, M.Y.; Haider, M.F.; Poddar, B.; Giurgiutiu, V. Guided wave crack detection and size estimation in stiffened structures. In Proceedings of the SPIE Nondestructive Characterization and Monitoring of Advanced Materials, Aerospace, Civil Infrastructure, and Transportation XII, Denver, CO, USA, 4–8 March 2018; Shull, P.J., Ed.; SPIE: Denver, CO, USA, 2018; p. 91.
14. Carpenter, S.H.; Gonnan, M.R. A Waveform Investigation of the Acoustic Emission Generated during the Deformation and Cracking of 7075 Aluminum. *J. Acoust. Emiss.* **1995**, S01–S07.
15. Achdjian, H.; Moulin, E.; Benmeddour, F.; Assaad, J.; Chehami, L. Source Localisation in a Reverberant Plate Using Average Coda Properties and Early Signal Strength. *Acta Acust. United Acust.* **2014**, *100*, 834–841. [CrossRef]
16. Ernst, R.; Zwimpfer, F.; Dual, J. One sensor acoustic emission localization in plates. *Ultrasonics* **2016**, *64*, 139–150. [CrossRef] [PubMed]
17. Toyama, N.; Koo, J.-H.; Oishi, R.; Enoki, M.; Kishi, T. Two-dimensional AE source location with two sensors in thin CFRP plates. *J. Mater. Sci. Lett.* **2001**, *20*, 1823–1825. [CrossRef]
18. Jiao, J.; Wu, B.; He, C. Acoustic emission source location methods using mode and frequency analysis. *Struct. Control Health Monit.* **2008**, *15*, 642–651. [CrossRef]
19. Holford, K.M.; Carter, D. Acoustic Emission Source Location. *Key Eng. Mater.* **1999**, *167–168*, 162–171. [CrossRef]
20. Ebrahimkhanlou, A.; Salamone, S. Acoustic emission source localization in thin metallic plates: A single-sensor approach based on based on multimodal edge reflections. *Ultrasonics* **2017**, *78*, 134–145. [CrossRef] [PubMed]
21. Ebrahimkhanlou, A.; Salamone, S. A probabilistic framework for single-sensor acoustic emission source localization in thin metallic plates. *Smart Mater. Struct.* **2017**, *26*, 95026. [CrossRef]
22. Ebrahimkhanlou, A.; Salamone, S. Probabilistic location estimation of acoustic emission sources in isotropic plates with one sensor. In Proceedings of the SPIE, Health Monitoring of Structural and Biological Systems, Portland, OR, USA, 25–29 March 2017; Kundu, T., Ed.; SPIE: Portland, OR, USA, 2017; Volume 10170, p. 1017029.
23. Ebrahimkhanlou, A.; Dubuc, B.; Salamone, S. Damage localization in metallic plate structures using edge-reflected lamb waves. *Smart Mater. Struct.* **2016**, *25*, 85035. [CrossRef]
24. Ebrahimkhanlou, A.; Salamone, S. A Deep Learning Approach for Single-sensor Acoustic Emission Source Localization in Plate-like Structures. In Proceedings of the 11th International Workshop on Structural Health Monitoring: Real-Time Material State Awareness and Data-Driven Safety Assurance, Stanford, CA, USA, 12–14 September 2017; Chang, F.-K., Fotis, K., Eds.; Destech Publications: Stanford, CA, USA, 2017; Volume 2, pp. 2139–2146.
25. Hsu, N.N. Acoustic Emissions Simulator. US Patent 4018084 A, 19 April 1977.

26. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
27. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016; ISBN 0262035618.
28. Sharif-Khodaei, Z.; Ghajari, M.; Aliabadi, M.H. Determination of impact location on composite stiffened panels. *Smart Mater. Struct.* **2012**, *21*, 105026. [CrossRef]
29. Al-Jumaili, S.K.; Pearson, M.R.; Holford, K.M.; Eaton, M.J.; Pullin, R. Acoustic emission source location in complex structures using full automatic delta T mapping technique. *Mech. Syst. Signal Process.* **2016**, *72–73*, 513–524. [CrossRef]
30. Li, C.; Sanchez, R.-V.; Zurita, G.; Cerrada, M.; Cabrera, D.; Vásquez, R.E. Gearbox fault diagnosis based on based on deep random forest fusion of acoustic and vibratory signals. *Mech. Syst. Signal Process.* **2016**, *76–77*, 283–293. [CrossRef]
31. Jia, F.; Lei, Y.; Lin, J.; Zhou, X.; Lu, N. Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data. *Mech. Syst. Signal Process.* **2016**, *72–73*, 303–315. [CrossRef]
32. He, M.; He, D. Deep Learning Based Approach for Bearing Fault Diagnosis. *IEEE Trans. Ind. Appl.* **2017**, *53*, 3057–3065. [CrossRef]
33. Vincent, P.; Larochelle, H.; Bengio, Y.; Manzagol, P.-A. Extracting and composing robust features with denoising autoencoders. In Proceedings of the 25th international conference on Machine learning (ICML '08), Helsinki, Finland, 5–9 July 2008; ACM Press: New York, NY, USA, 2008; pp. 1096–1103.
34. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Adv. Neural Inf. Process. Syst.* **2012**, *15*, 1–9. [CrossRef]
35. Møller, M.F. A scaled conjugate gradient algorithm for fast supervised learning. *Neural Netw.* **1993**, *6*, 525–533. [CrossRef]
36. Bishop, M.C. *Pattern Recognition and Machine Learning*; Springer: Berlin, Germany, 2006; ISBN 9780387310732.
37. Glorot, X.; Bordes, A.; Bengio, Y. Deep sparse rectifier neural networks. In Proceedings of the 14th International Conference on Artifcial Intelligence and Statistics (AISTATS) 2011, Fort Lauderdale, FL, USA, 11–13 April 2011; Volume 15, pp. 315–323.
38. Alexander, M.; Olah, C.; Tyka, M. Inceptionism: Going Deeper into Neural Networks. Available online: https://research.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html (accessed on 21 April 2018).
39. Zeiler, M.D.; Fergus, R. Visualizing and understanding convolutional networks. In Proceedings of the Computer Vision ECCV 2014, Zurich, Switzerland, 6–12 September 2014; Lecture Notes in Computer Science. Springer: Berlin, Germany, 2014; Volume 8689, pp. 818–833. [CrossRef]
40. Agarwal, A.; Negahban, S.N.; Wainwright, M.J. Noisy matrix decomposition via convex relaxation: Optimal rates in high dimensions. *J. Mach. Learn. Res.* **2011**, *15*, 1929–1958. [CrossRef]
41. Sarrafi, A.; Poozesh, P.; Niezrecki, C.; Mao, Z. Mode extraction on wind turbine blades via phase-based video motion estimation. In Proceedings of the SPIE, Smart Materials and Nondestructive Evaluation for Energy Systems, Portland, OR, USA, 25–29 March 2017; Meyendorf, N.G., Ed.; SPIE: Portland, OR, USA, 2017; p. 101710E.
42. Mostavi, A.; Kamali, N.; Tehrani, N.; Chi, S.-W.; Ozevin, D.; Indacochea, J.E. Wavelet-based-based harmonics decomposition of ultrasonic signal in assessment of plastic strain in aluminum. *Measurement* **2017**, *106*, 66–78. [CrossRef]
43. Ebrahimkhanlou, A.; Dubuc, B.; Salamone, S. A guided ultrasonic imaging approach in isotropic plate structures using edge reflections. In Proceedings of the Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems, Vegas, NV, USA, 20–24 March 2016; Lynch, J.P., Ed.; SPIE: Vegas, NV, USA, 2016; Volume 9803.
44. Sarrafi, A.; Mao, Z.; Niezrecki, C.; Poozesh, P. Vibration-based damage detection in wind turbine blades using Phase-based Motion Estimation and motion magnification. *J. Sound Vib.* **2018**, *421*, 300–318. [CrossRef]
45. Sarrafi, A.; Mao, Z. Structural operating deflection shape estimation via a hybrid computer-vision algorithm. In Proceedings of the SPIE, Health Monitoring of Structural and Biological Systems XII, Devor, CO, USA, 5 March 2018; Kundu, T., Ed.; SPIE: Devor, CO, USA, 2018; p. 94.
46. Mohammadi-Ghazi, R.; Marzouk, Y.M.; Büyüköztürk, O. Conditional classifiers and boosted conditional Gaussian mixture model for novelty detection. *Pattern Recognit.* **2018**. [CrossRef]

47. Sarrafi, A.; Niezrecki, C.; Poozesh, P.; Mao, Z. Applying video magnification for vision-based operating deflection shape evaluation on a wind turbine blade cross-section. In Proceedings of the SPIE, Health Monitoring of Structural and Biological Systems XII, Devor, CO, USA, 5 March 2018; Kundu, T., Ed.; SPIE: Devor, CO, USA, 2018; p. 21.
48. Murphy, K.P. *Machine Learning: A Probabilistic Perspective*; The MIT Press: Cambridge, MA, USA, 2012.
49. Ebrahimkhanlou, A.; Salamone, S. Single-sensor acoustic emission source localization in plate-like structures: A deep learning approach. In Proceedings of the SPIE, Health Monitoring of Structural and Biological Systems, Devor, CO, USA, 5 March 2018.