DeepSHM

A deep learning approach for structural health monitoring based on guided Lamb wave technique

Ewald, Vincentius; Groves, Roger; Benedictus, Rinze

DOI
10.1117/12.2506794

Publication date
2019

Document Version
Final published version

Published in
Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2019

Citation (APA)
Ewald, V., Groves, R. M., & Benedictus, R. (2019). DeepSHM: A deep learning approach for structural health monitoring based on guided Lamb wave technique. In J. P. Lynch, H. Sohn, K-W. Wang, & H. Huang (Eds.), Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2019 (Vol. 10970). [109700H] (SENSORS AND SMART STRUCTURES TECHNOLOGIES FOR CIVIL, MECHANICAL, AND AEROSPACE SYSTEMS 2019). SPIE. https://doi.org/10.1117/12.2506794

PROCEEDINGS OF SPIE

SPIEDigitalLibrary.org/conference-proceedings-of-spie

DeepSHM: a deep learning approach for structural health monitoring based on guided Lamb wave technique

Vincentius Ewald, Roger M. Groves, Rinze Benedictus

Vincentius Ewald, Roger M. Groves, Rinze Benedictus, "DeepSHM: a deep learning approach for structural health monitoring based on guided Lamb wave technique," Proc. SPIE 10970, Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2019, 109700H (27 March 2019); doi: 10.1117/12.2506794

SPIE.

Event: SPIE Smart Structures + Nondestructive Evaluation, 2019, Denver, Colorado, United States

DeepSHM: A Deep Learning Approach for Structural Health Monitoring Based on Guided Lamb Wave Techniques

Vincentius Ewald[a*], Roger M. Groves[a], Rinze Benedictus[b]
[a]Aerospace NDT Laboratory, Delft Univ. of Technology, 2629 HS Delft, The Netherlands
[b] Structural Integrity and Composites, Delft Univ. of Technology, 2629 HS Delft, The Netherlands

ABSTRACT

In our previous work, we demonstrated how to use inductive bias to infuse a convolutional neural network (CNN) with domain knowledge from fatigue analysis for aircraft visual NDE. We extend this concept to SHM and therefore in this paper, we present a novel framework called DeepSHM which involves data augmentation of captured sensor signals and formalizes a generic method for end-to-end deep learning for SHM. The study case is limited to ultrasonic guided waves SHM. The sensor signal response from a Finite-Element-Model (FEM) is pre-processed through wavelet transform to obtain the wavelet coefficient matrix (WCM), which is then fed into the CNN to be trained to obtain the neural weights. In this paper, we present the results of our investigation on CNN complexities that is needed to model the sensor signals based on simulation and experimental testing within the framework of DeepSHM concept.

Keywords: Structural Health Monitoring (SHM), deep learning, convolutional neural network (CNN), guided Lamb wave, damage classification, Finite-Element-Modelling (FEM), signal processing

1. INTRODUCTION

Beside fuel and ground services, one of the most crucial aspect in airline operating cost is the maintenance. Alone in 2017, it was reported that 70 billion USD is spent by airlines for the maintenance, repair, and overhaul (MRO) [1] and this figure is expected to grow to 115 billion USD in 2028 due to increasing number of aircraft deliveries [2 – 3]. One of the important aspects in the maintenance is structural integrity which ensures the reliability is within its design limit, known as damage tolerance (DT) design. While the DT design offers passive protection against catastrophic failure, there is no other way to ensure reliability without involving active intervention due to the nature of aircraft operation that faces safety uncertainty and therefore structural diagnostic in aircraft maintenance must be performed regularly.

1.1 Structural Diagnostic in Aircraft Maintenance

When speaking about scheduled aircraft maintenance, one would normally follow the Maintenance Steering Group – Task 3 (MSG-3) logic [4] which serves as root methodology for modern aircraft maintenance. One of the important tasks in MSG-3 process is called Special Detail Inspection (SDI) [5] that has been assigned due to the nature of the damages that is sometimes hardly detectable by human inspector and therefore non-destructive testing (NDT) is employed. One of the classical solutions that has been established for decades is to train an NDT inspector on various inspection methods, however it is relatively expensive in matter of time and costs to train human resources until the certification is reached. To tackle this problem, two mainstreams of automated SDI has been proposed: the robotics NDT [6] and Structural Health Monitoring (SHM) [7]. While robotics NDT typically means robot-assisted non-destructive inspection (NDI) by employing NDI instruments, in SHM the NDI instruments normally become an integral part of the structure itself [8].

Both approaches have their own advantages and disadvantages, but one thing is sure for both: the data captured by the NDI instruments will require more complex procsessing as the complexity degree of the system increases. Furthermore, due to increasing complexity in the material level which did not exist in the older aircrafts which used aluminum, there are more damage classes that could be present in modern aircraft material such as glass-fiber reinforced aluminum (GLARE), carbon fiber reinforced plastic (CFRP), or even thermoplastic composites [9 – 11]. Thus, a more sophisticated data processing technique particularly in digital signal processing (DSP) is needed to be able to cope with these ever-increasing complexities.

*V.Ewald@tudelft.nl; Phone: +31-1527-88271

Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2019, edited by Jerome P. Lynch, Haiying Huang, Hoon Sohn, Kon-Well Wang, Proc. of SPIE Vol. 10970, 109700H

1.2 State of the Art: Advances of Machine Learning and its Application in Structural Diagnostic

Traditionally, the main problem of signal processing is how to analyze the captured signals, that is how to extract signal features from sensors to interpret the behavior behind some physical phenomenon. Nowadays, in the era of increasing data amount (called 'big data') and computational ability, the central question has shifted to how to model the signals rather than analyzing them and this is where pattern recognition and machine learning typically come into operation. One of the popular data classification technique in 1990's was Support Vector Machine [12]. An example of using SVM for diagnostic application can be found in the work of Virupakshappa [13] for ultrasonic flaw detection in a steel block.

In recent years, people have been increasingly talking about 'deep learning' [14], which is an extension of the artificial neural network (ANN) algorithm which was introduced in 1950's as perceptron and had its popularity grow until the 1990's [15], where at that time the computational ability was low and eventually the ANN popularity was overshadowed by SVM [16]. It was not until 2006 when Hinton [17] introduced the deep belief network (DBN), a class of deep neural network (DNN), when the neural network regains its popularity in the computer science community. Since then, many approaches within the neural network realm has been proposed, such as: Deep AutoEncoders by Baldi in 2012 [18], Inception ConvNet by Szegedy et al. in 2014 [19], VGG-16 by Simonyan and Zisserman in 2014 [20], Enhanced LSTM by Chen et al. in 2016 [21], and Energy-based Generative Adversarial Network by Zhao et al. [22].

In the diagnostic domain, particularly in NDT, several applications of DNN – largely based on convolutional neural network (CNN) for crack detection, which is mostly concentrated on the surface of structures have been proposed by: Zhang et al. [23], Cha et. al [24], Chaiyasarn et al. [25], Fan et al. [26] and many more. Beside for crack detection for surface, there are several other works involving CNN in NDT, such as for phase detection in shearography proposed by Sawaf and Groves [27], welding detection using X-Ray images by Hou et al. [28], and damaged steel and CFRP using infrared (IR) images by Yousefi et al. [29]. As pointed out by Wunderlich et al. [30], we believe that advances in machine learning will have a huge impact in several key areas of NDI.

In a similar way, deep learning has also brought some wave of excitement to diagnostic SHM, although there are less works exploiting deep learning for diagnostic SHM in comparison to NDT. Apart from the work of Ebrahimkhanlou and Salomone [31], who used deep autoencoder (deep AE) for acoustic emission (AE) source localization and the work of Choy [32] and Oliveira et al. [33] who used CNN for processing electromechanical impedance (EMI), we are not aware of any further works involving deep learning in diagnostic SHM at the time being (January 2019).

1.3 Objective and Article Outline

In our previous work [34], we demonstrated how to bias CNN with appropriate domain knowledge. Therefore, in this paper we would like to propose a hybrid model that we call DeepSHM framework. Specifically, it is a statistical signal modelling based on deep learning that is biased by a physical model. The structure of our paper is the following: the problem statement of diagnostic SHM and the DeepSHM framework are described in section 2. The study case for the DeepSHM framework involving guided Lamb waves and the corresponding data processing is described in section 3 and 4, respectively, while the experimental validation of the study can be found in section 5. Finally, in section 6 as a conclusion, we delivered the summary and the outlook of this project.

2. AIRCRAFT STRUCTURAL DIAGNOSTIC

To be more precise on the paper format, we divide section 2 into two parts: section 2.1 describes the general SHM workflow and diagnostic relevant part, while in section 2.2 we formalize the DeepSHM framework.

2.1 Diagnostic SHM in General

The SHM workflow in general is summarized in Fig. 1. The diagnostic SHM consists of 3 functional levels: level 1 is concerned with load monitoring which we would not further discuss in this article due to resource and time constraints, while level 2 is concerned with damage detection within the structure to answer the question 'does a damage exist within the structure?'. Level 2 SHM is can normally easily be answered since the only concern is a 'yes / no' question.

SHM level 3 can only proceed if the question in level 2 is answered with 'yes' and it is concerned with information that can be retrieved regarding the damage, such as damage localization, classification, and characterization to answer the questions 'where, what kind, and how severe is the concerned damage?'. The majority of diagnostic SHM works are focused in this area since it requires more complex answer rather than just the binary outcome in comparison to level 2.

SHM level 4 would go beyond these questions and try to approximate the residual lifetime given the information retrieved in level 2 and 3. However, as level 4 is part of prognostic SHM, we would not discuss it further in this paper.

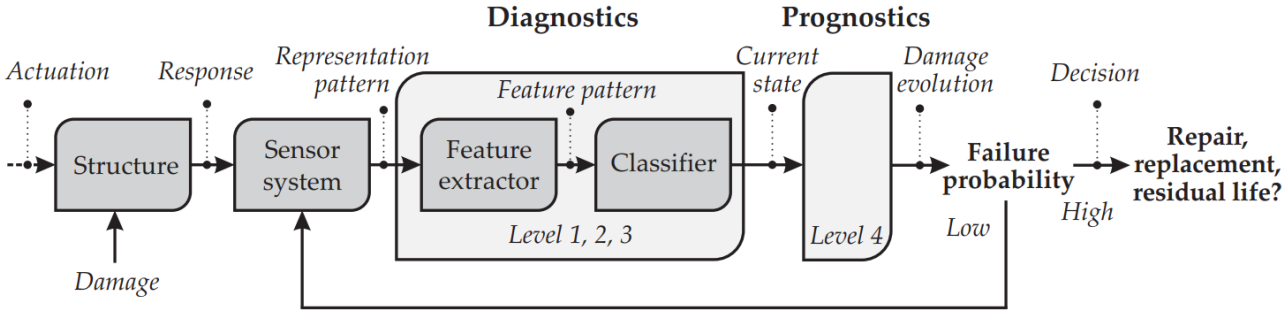

Figure 1. SHM Workflow in General [35]

Focusing our work on level 2 and 3 SHM, we can simply formulate the structural diagnostic realm depicted in Fig. 2

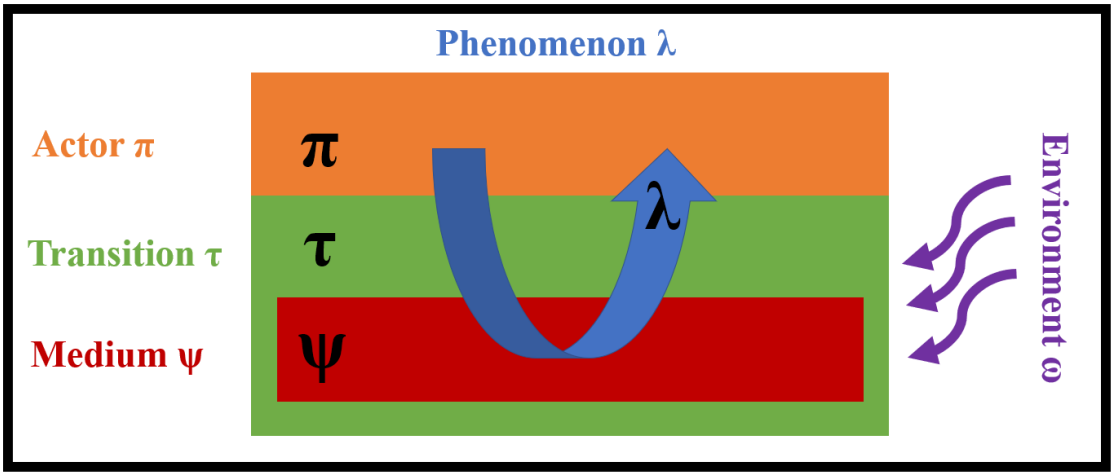

Figure 2. Diagnostic realms

The diagnostic realm depicted in Fig. 2 can be formalized in following quintuplet ***D*****:** {**π**, **ψ**, **τ**, **λ**, **ω**}:

π is the actor domain. This tuple contains the parameters that are needed to generate and measure the physical phenomenon of interest **λ**. An example of the influencing parameters in these domains include: the geometry of the actuator or sensor, the instrumentation quality including oscilloscope, measuring cable and soldering quality, the excitation parameters, and the location of the source of the phenomenon of interest.

ψ is the medium domain. This is the subject space where phenomenon of interest **λ** propagates after being excited in the actor domain. When the parameters of the medium domain changes, the behavior of **λ** changes. These changes typically serve as a diagnostic indicator to determine whether the structure is reliable or not. An example of the influencing parameters in this domain includes: the geometry and elastic properties such as Young's modulus or Poisson's ratio of the material, presence of damage inside the medium, location and severity of the damage, type of damage (e.g. delamination, crack, corrosion, fiber break), presence of mechanical fasteners, etc.

τ is the transitional domain, that separates the actor from the medium. Concretely, as in many real-world situations where the medium has a finite property, the transitional domain can be understood as the boundaries of the medium. An example for this is the boundary condition of the medium and the boundary condition between the actor and medium.

λ is the domain of the phenomenon of interest, that is the physical phenomenon which is observed in actor domain **π**. The phenomenon **λ** typically interacts with at least one parameter in all other domains. As examples, in diagnostics, it is possible to use (low or high-frequency) ultrasonic waves, electromagnetic effects such as eddy current, or (non)-visible light (ranging from X-ray to UV) to detect damage.

$\boldsymbol{\omega}$ is the environmental domain which covers actor, transition, and medium domain $\boldsymbol{\pi}$, $\boldsymbol{\psi}$, and $\boldsymbol{\tau}$, respectively. This tuple contains parameters that potentially change the observation in the actor domain $\boldsymbol{\pi}$ or even changes the behavior of $\boldsymbol{\lambda}$ completely, but is not included in the other domains. An example of the influencing parameters in these domains include measurement temperature, humidity, external vibration, electromagnetic radiation, etc.

2.2 DeepSHM Framework

The most direct formulation to describe the relations between the domains in the diagnostic realm $\boldsymbol{D}$ would be:

$$\boldsymbol{\lambda} = f(\boldsymbol{X}(\boldsymbol{\pi}, \boldsymbol{\psi}, \boldsymbol{\tau}, \boldsymbol{\omega})) \tag{1}$$

Where $\boldsymbol{X}$ is an observable variable that describes the latent variable $\boldsymbol{\lambda}$ as dependency on $\boldsymbol{\pi}$, $\boldsymbol{\psi}$, $\boldsymbol{\tau}$, $\boldsymbol{\omega}$, i.e. in reality $\boldsymbol{X}$ is the measured signal and can be written in either vector or matrix form. However, as $\boldsymbol{\lambda}$ is hidden and what we typically observe from outside is $\boldsymbol{X}(\boldsymbol{\pi}, \boldsymbol{\psi}, \boldsymbol{\tau}, \boldsymbol{\omega})$, the relation can be regarded as the inverse model:

$$\boldsymbol{X}_{\boldsymbol{\pi},\boldsymbol{\psi},\boldsymbol{\tau},\boldsymbol{\omega}} = f^{-1}(\boldsymbol{\lambda}) \tag{2}$$

For simplification, assume the null hypothesis h_θ where the existence of $\boldsymbol{\lambda}$ is caused only by a parameter in $\boldsymbol{\psi}$ and due to the stochastic nature of observable variables $\boldsymbol{X}$, the relation can be formulated via Bayes conditional probability P:

$$P(h_\theta(\boldsymbol{\lambda}) \mid \boldsymbol{X}_{\boldsymbol{\psi}}) = \frac{P(\boldsymbol{X}_{\boldsymbol{\psi}} \mid h_\theta(\boldsymbol{\lambda})) \cdot P(h_\theta(\boldsymbol{\lambda}))}{P(\boldsymbol{X}_{\boldsymbol{\psi}})} \tag{3}$$

Where in Eq. (3), $P(h_\theta(\boldsymbol{\lambda})|\boldsymbol{X}_{\boldsymbol{\psi}})$ is the posterior probability of the existence of $\boldsymbol{\lambda}$ given observable variables $\boldsymbol{X}_{\boldsymbol{\psi}}$, $P(\boldsymbol{X}_{\boldsymbol{\psi}}|h_\theta(\boldsymbol{\lambda}))$ is the prior probability where $\boldsymbol{X}_{\boldsymbol{\psi}}$ occurs given the hypothesis $h_\theta(\boldsymbol{\lambda})$, $P(\boldsymbol{X}_{\boldsymbol{\psi}})$ and $P(h_\theta(\boldsymbol{\lambda}))$ are the marginal probability of observing $\boldsymbol{X}_{\boldsymbol{\psi}}$ and $h_\theta(\boldsymbol{\lambda})$ independently, respectively.

Furthermore, in Eq. (3), θ is the black-box parameters (in machine learning, these are normally called either synaptic parameters or simply neural network weights) that have to be optimized during the learning process. According to Mitchell [36], the algorithm of machine learning is defined as: "*A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E*". In the diagnostic realm, task T is the diagnostic itself, that is retrieving the information from the observable variables $\boldsymbol{X}_{\boldsymbol{\psi}}$ regarding damage state, experience E is the iterative process for enhancing the algorithm to increase the accuracy performance of the trained algorithm that best generalizes the distribution over $\boldsymbol{X}_{\boldsymbol{\psi}}$. Specifically, the probability for the j-th class of information given the $\boldsymbol{X}_{\boldsymbol{\psi}}$ in k-dimensional Hilbert space is typically written as logit or sigmoid function [36, 37] and can be generalized in softmax function which is defined by:

$$P(h_\theta(\boldsymbol{\lambda}) = j \mid \boldsymbol{X}_{\boldsymbol{\psi}}) = \frac{\exp([\boldsymbol{X}_{\boldsymbol{\psi}}]^T \cdot \theta_j)}{\sum_{k=1}^{K} \exp([\boldsymbol{X}_{\boldsymbol{\psi}}]^T \cdot \theta_k)} \tag{4}$$

Per definition, the posterior probability $P(h_\theta(\boldsymbol{\lambda})|\boldsymbol{X}_{\boldsymbol{\psi}})$ can take any real value between 0 and 1, sometimes also written from 0% to 100%. This is normally achieved by minimizing the loss (sometimes called distance or error) between $h_\theta(\boldsymbol{\lambda})$ and $\boldsymbol{X}_{\boldsymbol{\psi}}$. The general objective of a diagnostic is to increase true positive (*TP*) rate and decrease false negative (*FN*) rate, thus maximizing the probability of detection (*POD*), or sometimes called sensitivity or recall rate, which is defined as:

$$POD = \frac{\sum TP}{\sum (TP + FN)} \tag{5}$$

The current standard of practice for diagnostic NDT according to MIL-HDBK1823 [37] is POD = 0.9 (or 90%) within 95% statistical confidence σ, although this might not be suitable for diagnostic SHM [38]. Nevertheless, to avoid further

confusion, we prefer to adapt the commonly used metrics in machine learning, which is accuracy *A* instead of *POD* and this is defined as:

$$A = \frac{\sum TP + TN}{\sum (TP + TN + FP + FN)} \tag{6}$$

Where in Eq. (6), *TN* is the true negative rate and *FP* is the false positive rate. Assuming that in diagnostic the dataset $\boldsymbol{X}_{\psi}$ always comes from the same distribution, *A* and *POD* rate would have a similar value. We would like to point out that sometimes it is useful to influence the algorithm by domain bias, as we demonstrated before in our previous work [34]. Biasing the algorithm also includes determining the data distribution which is fed into the learning algorithm to have interpretable outcome and to avoid a "garbage in – garbage out" process [39]. Therefore, it is very important to determine which task T the algorithm should perform at the beginning.

3. STUDY CASE: DEEP SHM FOR GUIDED LAMB WAVE

As a study case for the preliminary concept DeepSHM, active guided Lamb wave SHM that has been previously employed in our previous works are used. Section 3 of our paper is organized as follows: the implementation of the DeepSHM framework for guided Lamb wave is described in section 3.1, while method to generate and multiply the data in Finite Element (FE) environment are described in section 3.2. The explanation on the data pre-processing method and the deep learning algorithm by using CNN is described in section 3.3 and 3.4, respectively. In section 3.5, we briefly mention the training methods and environments for the CNN (including required hardware and software).

3.1 DeepSHM for Guided Lamb Wave

The implication of processing the observable dataset $\boldsymbol{X}_{\psi}$ that contains damage information captured by the actor $\boldsymbol{\pi}$ using CNN to give a hypothesis $h_{\theta}(\boldsymbol{\lambda})$ can be seen in Fig. 3. The central task of our framework is to find suitable parameters θ which are a good fit to describe the correlation between dataset $\boldsymbol{X}_{\psi}$ and $h_{\theta}(\boldsymbol{\lambda})$, where in guided Lamb wave SHM, $h_{\theta}(\boldsymbol{\lambda})$ is just the hypothesis of the damage information contained in medium domain $\boldsymbol{\psi}$ which is influenced by interaction between the phenomenon Lamb wave ($\boldsymbol{\lambda}$) and the damage itself.

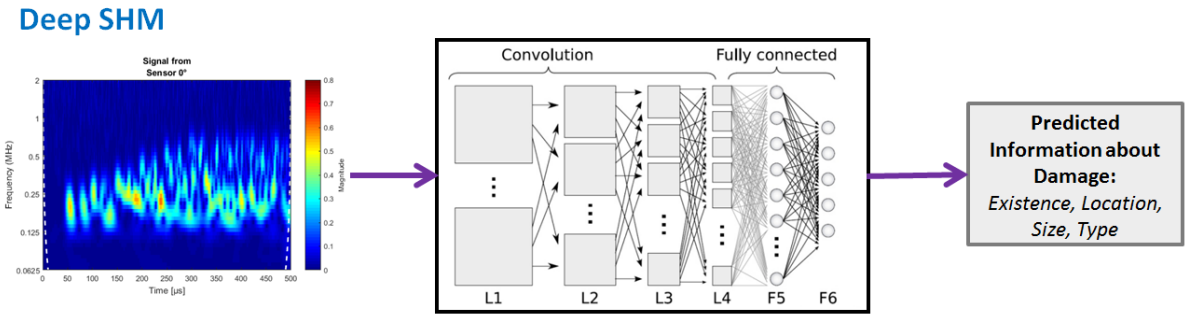

Figure 3: DeepSHM demonstrates the use of deep learning algorithm to process Lamb wave signals for damage prediction

3.2 Data Generation using FEM

Since experiments are sometimes quite expensive to make, we decided to use simulation to generate signal data in a Finite Element (FE) environment from ABAQUS software that we had from our previous work [40]. The method to determine the simulation parameters can be found there. The simulation parameters we used this time are: an aluminum plate with Young's modulus of 70 GPa, Poisson's ratio of 0.33 and density of 2700 kg/m³ which is meshed as quadratic brick elements with the global mesh size of 1.5 mm, resulting in around 160000 FE elements in total. The excitation frequency was 200 kHz. To simplify the model, no piezoelectric transducer (PZT) was modelled and only mechanical force was used, thus only displacement data in the time-domain from an FE node are taken, as depicted in Fig. 4. As an example, the displacement data in time domain from a sensing node of Fig. 4 is depicted in Fig. 5a. Alternatively, it is possible to represent signal in frequency-domain as depicted in Fig. 5b via Fast Fourier Transform (FFT).

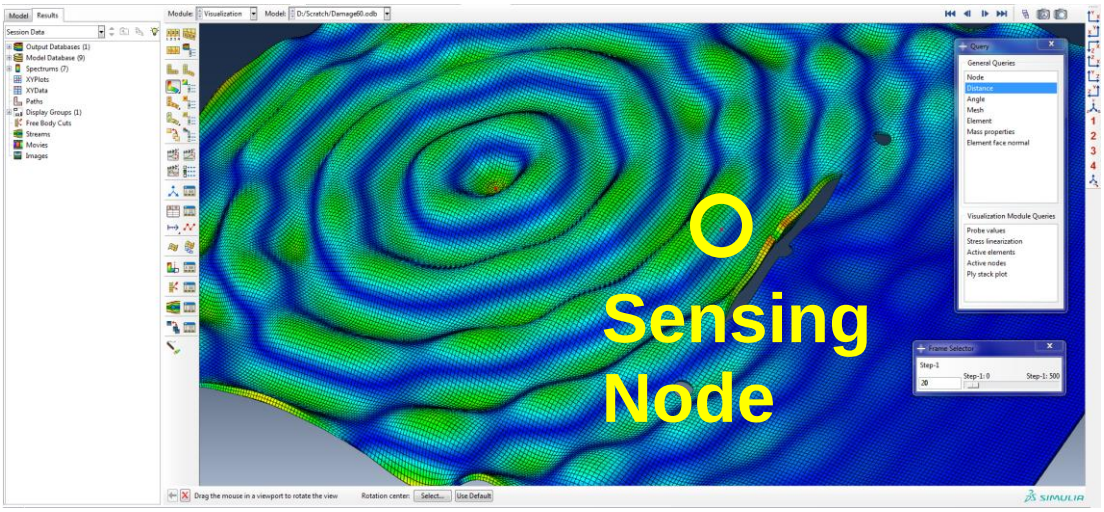

Figure 4: Illustration of propagating Lamb wave in ABAQUS with deformation magnification

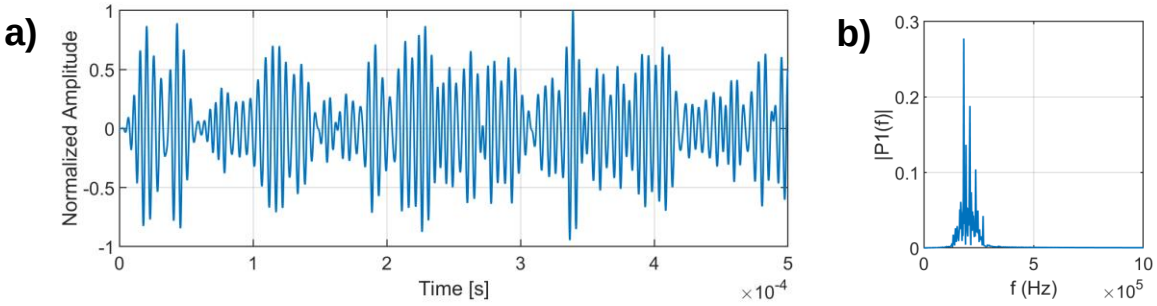

Figure 5a – b: Captured Lamb wave signal in time-domain (left) and its representation in frequency-domain (right)

3.3 Data Pre-Processing

Instead of representing time-domain (Fig. 5a) and frequency-domain (Fig. 5b) separately, one can represent the signal in the time-frequency-domain. Generally, there are two ways to represent information in both time and frequency-domain: Short-Time Fourier Transform (STFT) and Wavelet Transform (WT), as depicted in Fig. 6 and 7, respectively. There are enough studies on comparison between STFT and WT [41 – 42]. Given that 1). our dataset is not big data and 2). the wavelet function can already be determined since it follows the shape of excitation signal (e.g. 5-cycles Hann-windowed sinusoid would follow Morlet shape), wavelet analysis can still be used without sacrificing too much time. With wavelet analysis, we still conserve information in both the time and frequency-domain even that this means we slightly increase the pre-processing time. To save space, the WT matrix can be saved in PNG extension as a WCM (Wavelet Coefficient Matrix), resulting a very small file size (about 70 Kbytes per file). The normalization is then between 0 and 255 in 8-bit resolution. It is also possible to save it in bitmap form although normally the file size would be larger.

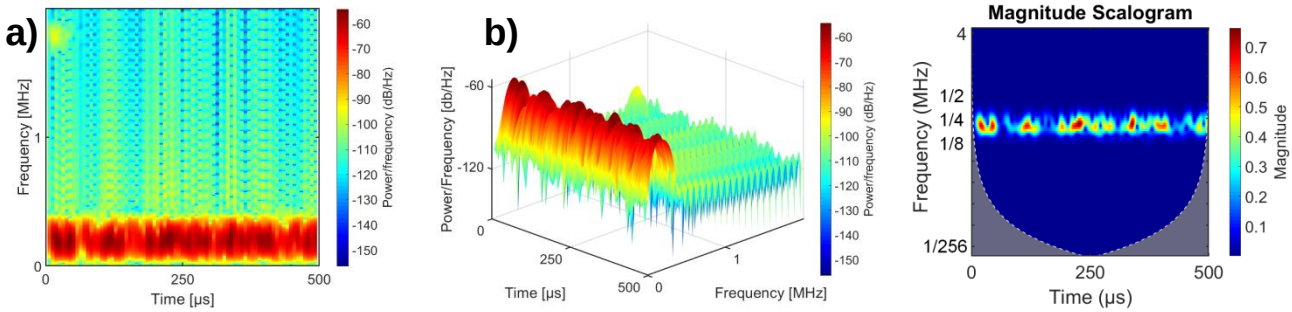

Figure 6a – b: STFT spectrogram in 2D (left) and 3D (middle) representation, Figure 7 (right): Wavelet scalogram in $\log_2$ y-axis

3.4 Convolutional Neural Network (CNN) and Domain Bias in Deep Learning

To process the observed signal $\boldsymbol{X}_{\boldsymbol{\psi}}$ in Eq. (3) and (4), which in our case is represented as a WCM, there are several deep learning architectures available, depending on how we would like to treat the WCM. The first option would be treating the WCM sequentially, thus updating the posterior probability P in Eq. (3) per given sequence, which in this case means the smallest discrete time increment. For this task, recurrent neural network [43] and its variants such as LSTM and GRU [44 – 45] would be a proper choice. While implementing sequential modelling to WCM will be elaborated in our future work, however, at the time being, we would like to see how capable our previous algorithm [34] handles the input, which in this case, is then treated as a static signal (= image). CNN is suitable to do image recognition since it learns local spatial information through a sliding convolution window (also called either filter or kernel) [27]. One example of CNN architecture is called DeepFace by Facebook AI Research [46] for face recognition.

CNN normally consists of multiple successively increasing refined data filters. Within the context of this analogy, the input data to a CNN, e.g. an image is represented by a 3D tensor with size of height × width × channels pixels. Depending on the excitation frequency, the WCM can be 2-D or 3-D. For 1 excitation frequency, the WCM is only 2-D, thus can be represented as a greyscale image, while for 3 excitation frequencies it can be represented as an RGB image. The CNN operation is as follows: the input WCM is multiplied by activation functions, and as it passes through each network layer, it is transformed into more abstract representations for the networks to produce a prediction output. The transformations performed by each layer are parametrized by its neural weights θ. Then, the difference between the predicted output and the true output is computed. This difference is called the loss function and the central task of machine learning is to minimize the loss by adjusting θ through an optimizer during the backpropagation.

In our previous work [34], we highlighted the importance of influencing machine learning algorithm with suitable bias. As previously stated in section 3.2, real experiments are sometimes quite expensive to make, we decided to use simulation to generate signal data. However, since simulation is done only once, only one time-series is generated per simulation. One of the typical bias in machine learning is called data augmentation and one appropriate technique to do this is to add random noise that normally always occurs during experimental measurement. That way, we can generate a manifold of Lamb wave signals that later can be represented as WCM. An example of data augmentation applied to simulated Lamb wave signal at 200 kHz frequency excitation recorded in sensor location 1 is depicted in Fig. 9a – b.

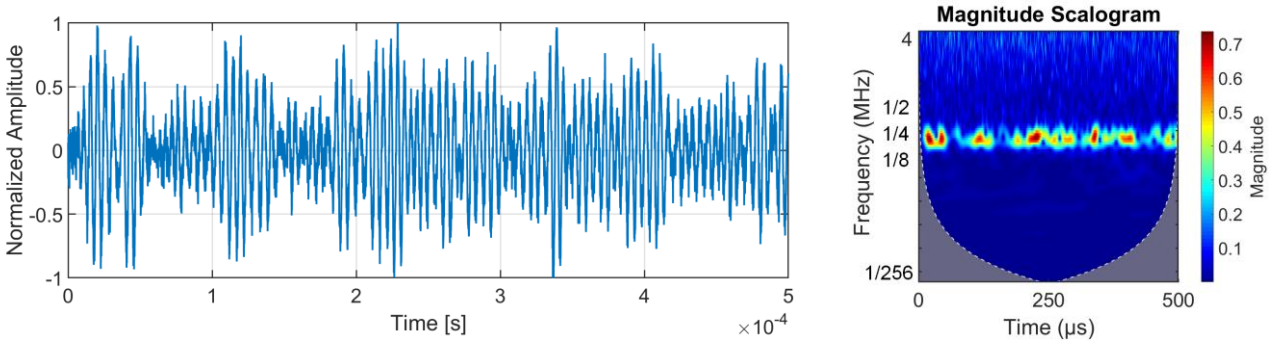

Figure 9a – b: Simulated Lamb wave signal from Fig. 6a with added white Gaussian noise (AWGN) of SNR = 15 (left) and its wavelet scalogram representation (right) with $\log_2$ y-axis. Cf. Fig. 10b with Fig. 8

3.5 Training Method for CNN

As for hardware, the standard PC specification at TU Delft is: Dell Precision T5810 running on Intel Xeon(R) E5-1620 3.5 GHz and 32 GB DDR-RAM running on Windows 7. The only exception we had is that we had our PC performance boosted with a NVidia GPU GeForce GTX1080Ti graphics card which currently has the highest performance on end-user market. Note that, more powerful GPU for large-scale GPU exploitation such as NVidia Tesla or even distributed processing center such Microsoft Azure is also available for manifold of price that we paid for GTX1080Ti.

Software-wise, we can recommend Tensorflow (developed by Google Brain) and Keras wrapper (both are available on Python) as currently these are the richest library available on market. Other libraries such as DL4J on Java or Microsoft Cognitive Toolkit (CNTK) on C++ are also available, although Python is typically preferable as it is easier to learn, especially for non-software engineers. Lightweight language such as Lua has also deep learning library (cf. Torch),

although this has less functions than Tensorflow. To enable the GPU acceleration, the API NVidia CUDA (currently we have the latest version 9.0) must be installed so that Tensorflow can interact with the graphic card.

Training a neural network is basically an optimization problem, and in general one would rather select converging iterative methods over a metaheuristics search given the high-dimensionality of parameters θ. And when talking about converging iterative methods, the natural choice for high-dimensionality of θ would be the first-order methods, and among the first-order methods, some of well-known techniques are (L)-BFGS [47], gradient descent [48], and Levenberg-Marquardt [49]. Currently only the gradient descent methods are already implemented in Keras. The native optimizer is called stochastic gradient descent (SGD) with variable batch training size which also supports neural momentum and Nesterov acceleration gradient (NAG) [50] as described in Eq. (7).

$$\begin{aligned} \delta_{t+1} &= \gamma \cdot \delta_t + \eta \cdot \left[\nabla_\theta J(\theta - \gamma \cdot \delta_t)\right] \\ \theta_{t+1} &= \theta_t - \delta_{t+1} \end{aligned} \tag{7}$$

Where in Eq. (7), δ_{t+1} is the update vector for parameter θ at iteration t+1, η is the learning rate, J is the assigned cost function (e.g. cross-entropy loss or mean-squared-loss), and γ is the momentum which is typically set to 0.9 [50].

4. RESULTS AND DISCUSSION

Due to space constraints in this paper, we restrict the results presentation to only 4 CNN architectures for each of 3 different classification scenarios. Note that, we will expand this concept into multiple sensing location and multiple excitation frequency and build a voting system from each sensing location in our future work. For this paper we present the results as follows: in section 4.1 we briefly explain the several different CNN architectures, and the training methods we employed. In section 4.2 we presented 3 different classification scenarios under which each CNN architectures were trained. The accuracy result of each trained network is presented in section 4.3.

4.1. Proposed CNN Architectures and Training Methods

As finding a suitable network has an NP-complete property since it is extendable to an infinite number of architectures, it would be logical to start the choice of architecture with the less complex series, as per *lex parsimonae*. Thus, this time we limit our choices to 4 architectures only, as described in Table 2. The abbreviations of each architectural element of the network such as **C**(*i*), **MP**, **DO**(*j*), **D**(*k*), and **CL** are given under Table 2. We are aware that there are more sophisticated CNN architectures such as InceptionNet [19] or VGG-16 [20], but for the simple demonstration purpose in this paper we do not go further beyond 8 hidden layers.

Table 2. Proposed CNN architectures

Network 1	**D**(128)-**D**(16)-**CL**
Network 2	**C**(8)-**MP**-**DO**(0.5)-**D**(128)-**DO**(0.5)-**CL**
Network 3	**C**(8)-**MP**-**DO**(0.5)-**C**(16)-**MP**-**DO**(0.5)-**D**(128)-**DO**(0.5)-**CL**
Network 4	**C**(8)-**MP**-**DO**(0.5)-**C**(16)-**MP**-**DO**(0.5)-**C**(32)-**MP**-**DO**(0.5)-**D**(128)-**DO**(0.5)-**D**(16)-**CL**

C(*i*): *i*-filter convolutional kernel; **MP**: MaxPooling layer; **DO**(*j*): dropout regularization with rate of *j*; **D**(*k*): dense (fully-connected) layer with *k*-neurons, **CL**: Classification layer, typically a *softmax* function as per Eq. (4).

Note that for the training purpose, we always normalize the data between 0 and 1 so that we can directly adapt many useful training parameters that has been proposed in computer science community. For deep neural network, it is recommended to always activate the dropout regularization and according to Hinton [51], 0.5 is the best rate found, which means 50% of the neurons at that particular layer being deactivated. As our data has been normalized, we can let the default parameter for the optimizers as described in Table 3. The explanation of each training parameters can be found in Keras optimizer documentation and TensorFlow handbook [52, 53].

Table 3. Optimizer

SGD	η = 0.01, γ = 0.0, η_{decay} = 0.0, NESTEROV = FALSE
Adam	η = 0.001, β_1 = 0.9, β_2 = 0.999, $\varepsilon = 10^{-8}$, η_{decay} = 0.0, AMSGrad = FALSE

η: Learning rate, γ: neural momentum, η_{decay}: learning rate decay, NESTEROV: Nesterov momentum parameter, β_1, β_2: exponential decay rate for 1st and 2nd moment estimates.

4.2. Classification scenarios

We will demonstrate 3 scenarios under which the CNN architectures in section 3.6.1 were trained. Assume that we have a dataset from the following half-crack length (including initial notch) measured from the center of the rivet: 0 mm (pristine plate), 15 mm, 30 mm, 45 mm, 60 mm, 75 mm, 15 mm angled crack, and 45 mm angled crack. We presented the classification scenarios in Table 4.

Table 4: Classification scenarios

Scenario 1	Binary classification between pristine (0 mm) and 75 mm crack
Scenario 2	4 classifications between crack lengths of axial 15 mm and 45 mm, angled 15 mm and 45 mm
Scenario 3	6 classifications between crack lengths of pristine plate, 15 mm, 30 mm, 45 mm, 60 mm, and 75 mm

Per class, we limit our dataset to only 1000 WCM images. With about 70 Kbytes per WCM, each class has only about 70 megabytes size. This is of course very modest in comparison to what social media produces in an hour, but this also justifies our selection of "simple" neural network architectures in Table 2. However, the curse of (high)-dimensionality we must suffer is accordingly much lower than what Facebook has with their DeepFace. As we have 8 classes, 8000 WCM images were created from the simulation data. This process took about 5 hours on full PC performance.

4.3. Results

Derived from Tables 2 – 4, we have 24 training runs in total from the combinations between 4 networks, 2 optimizers, and 3 classification scenarios. Of the 24 training runs, we obtained different classification accuracies ranging from 17% to 99.9%, depending on the network and classification scenarios. For binary classifications (scenario 1), all 4 networks can correctly classify the signals from pristine and damaged plates without any problem. For 6-classes signal (scenario 3) classifications, however, the simplest network (network 1) classify the signals only 17% correctly while the more sophisticated network (6- and 8- hidden layers) reaches 99.9% classification accuracy. This is to be expected as the simple network is too naïve to capture the complexity of WCM of various classifications.

Fig. 10 depicts the loss and accuracy during the training obtained by the network 1 (2-layers perceptron) on SGD optimizer in scenario 1. All other networks exposed similar behavior (other results are available on request, please contact the author). As scenario 1 is only a binary classification whether the output is limited to either "PRISTINE" or "DAMAGED", it is no wonder that the network can easily capture any features that belong to that signal series.

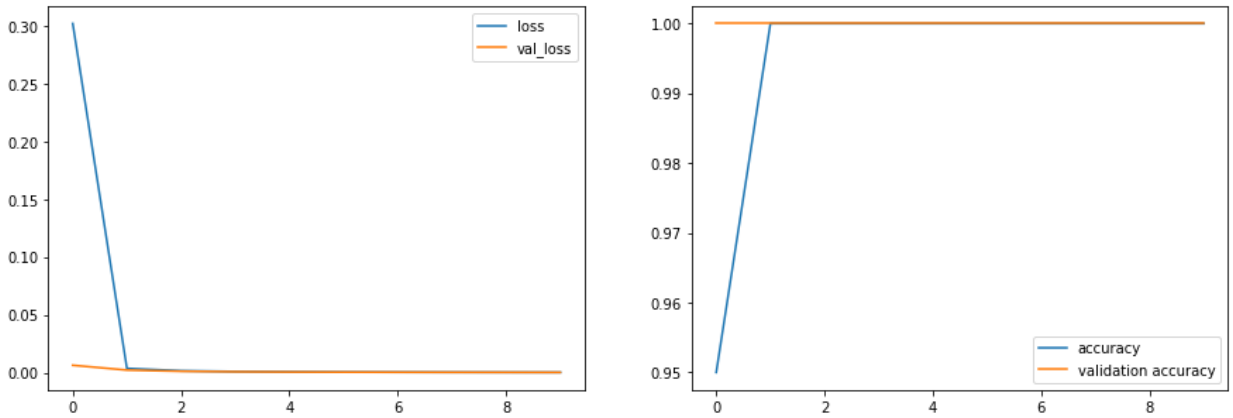

Fig. 10: Loss (left) and accuracy (right) obtained by network 1 on SGD optimizer in scenario 1

In scenario 2, however, network 1 cannot cope with the situation anymore as the classifications become more complex, as depicted in Fig. 11 top. There are 4 signal series to be classified from plate of perpendicular and angled cracks of 15 mm and 45 mm length, respectively. Network 2 (3-layers including 1-convolutional layer) has difficulties as well to get a good accuracy and to decrease the loss until the 7th epoch as shown in Fig. 11 bottom. Only from network 3 (5-layers including 2-convolutional layers), we can see that the loss declines rapidly from the 1st training epoch (Fig. 12). This is within our expectation since network 3 has more layers than network 2 that enable the network to distinguish the features of 4 signal classes.

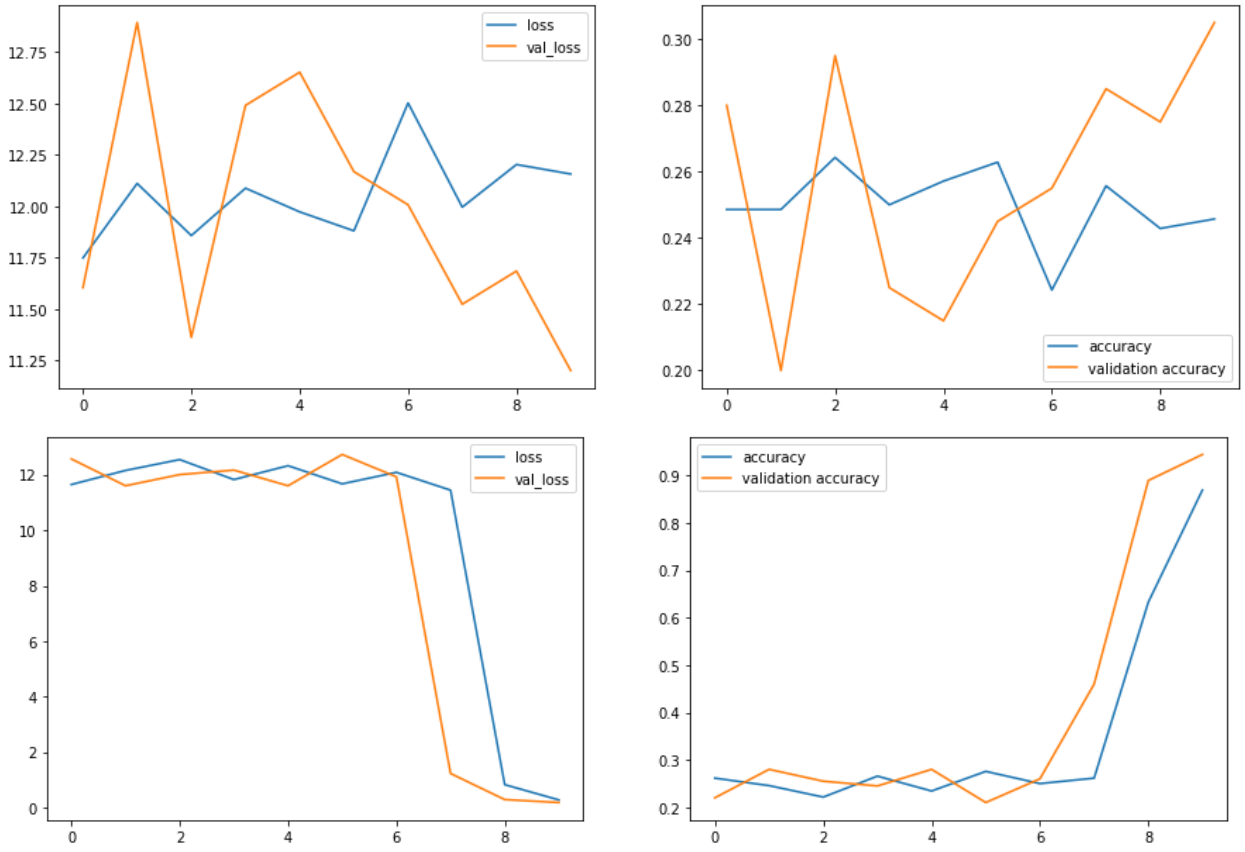

Fig. 11: Loss (left) and accuracy (right) obtained by network 1 (top) and network 2 (bottom) on Adam optimizer in scenario 2

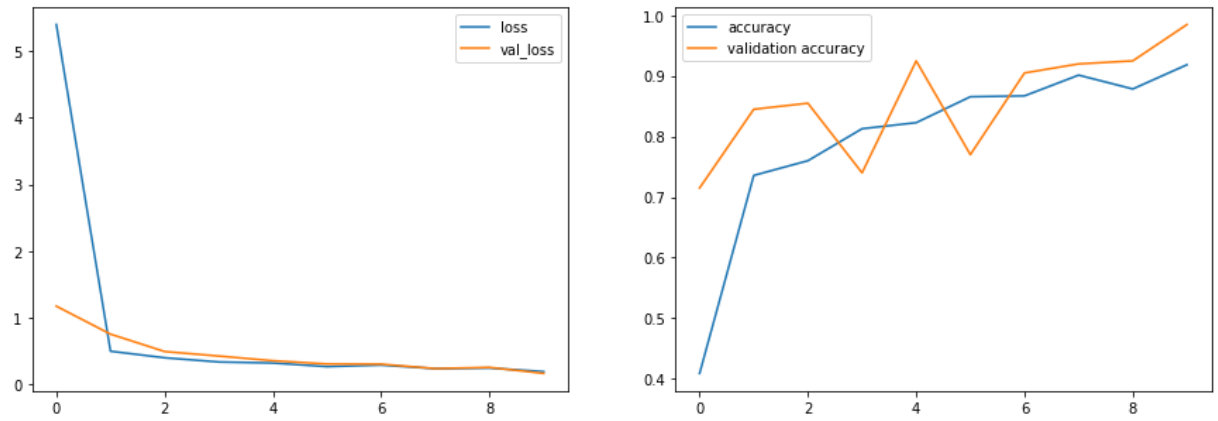

Fig. 12: Loss (left) and accuracy (right) obtained by network 3 on Adam optimizer in scenario 2

Moving to scenario 3, both network 3 (5-hidden layers including 2-convolutional layers) and network 4 (8-hidden layers including 3-convolutional layers) perform well, as can be seen in Fig. 13. Network 3, which is slightly "lighter" seems to converge faster than network 4. Obviously, for *n*-classifications of WCM signals, an approximately *n*-hidden layers would suffice and there is no need to invest on complex structures for more than necessary.

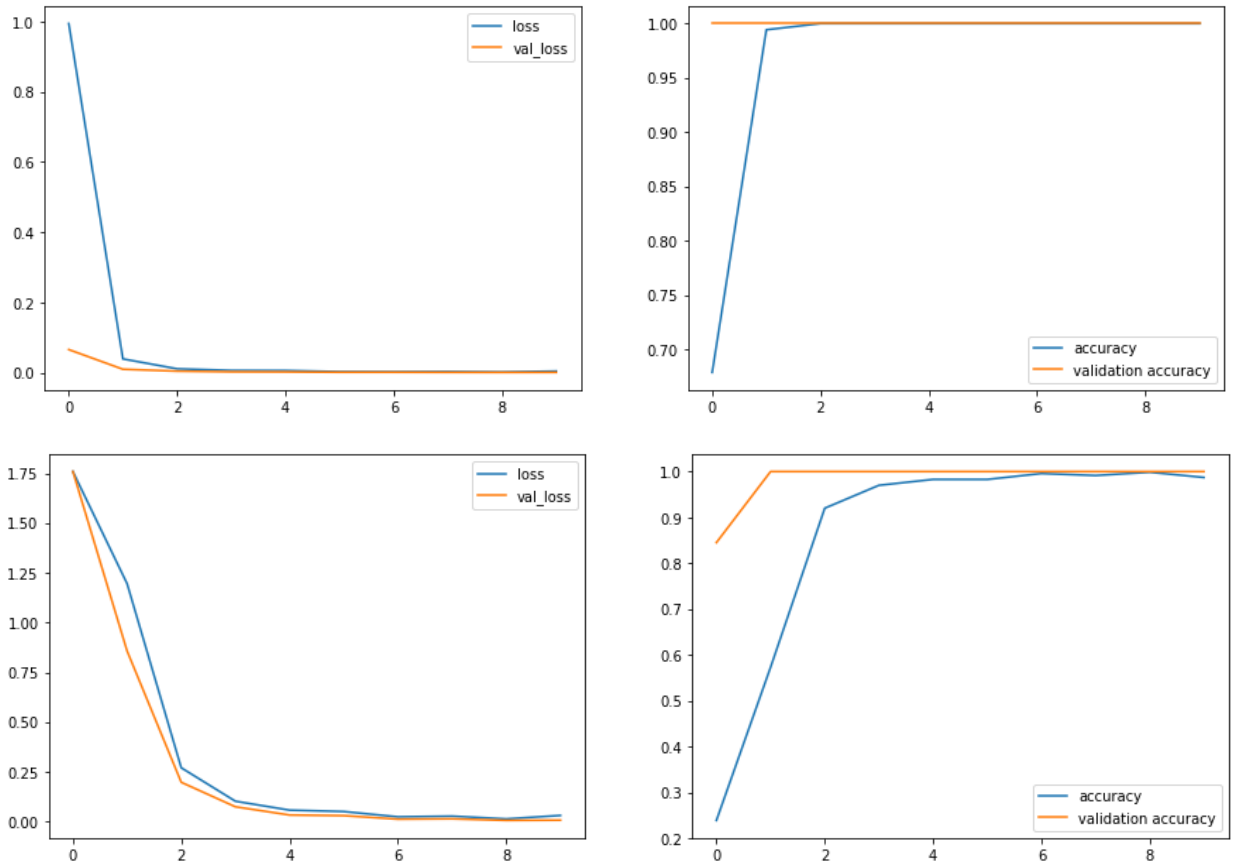

Fig. 13: Loss (left) and accuracy (right) obtained by network 3 (top) and network 4 (bottom) on SGD optimizer in scenario 3

Based on the result, it is obvious that more classes Lamb wave signal classifications generally requires a more complex DL architecture. However, a too complex network might also overfit the dataset as depicted in Fig. 13 top vs bottom in epoch 1 and 2. Not only that, a similar data distribution that is sorted into different classes (scenario 2) can significantly slowdown the accuracy, see Fig. 12 and 13 top which depict training for network 3 in scenario 2 and 3, respectively. This is due to the physical limitation that Lamb waves of a certain wavelength (200 kHz frequency = 27 mm wavelength in S_0-Mode) can well interact with damage of similar or larger size, but not that well with a damage of a smaller size. It is important to note, that while the wave well interacts with both a perpendicular and an angled crack (length a = 15 mm), it will have a hard time to cause any signal difference that is caused by angle α and crack tip shift *z* (Fig. 14).

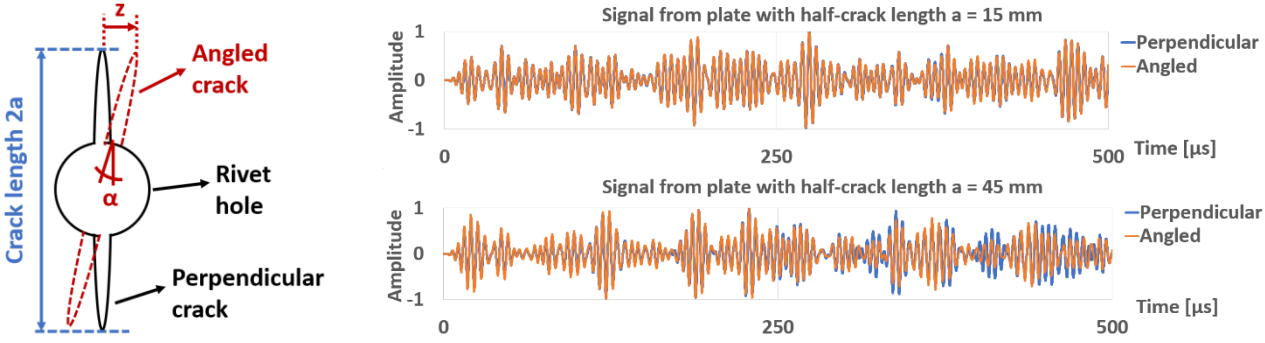

Fig. 14 left: Rivet hole with crack; right: corresponding Lamb wave signal captured by the sensor

5. EXPERIMENTAL VALIDATION

5.1 Experimental Setup and Captured Signal

The experimental setup is depicted in Fig. 15. Our NDT Lab has currently following hardware available: PicoScope 6402, Thurlby wideband amplifier WA301, Agilent waveform generator 33220A, standard cables with BNC connector, and PZT patch delivered by American Piezo APC-850 (resonance frequency: 208 kHz, Ø 9.52 mm, ρ = 7.6 g/cm³). The excitation frequency is matched to the simulation: 200 kHz 5-cycles sinusoid on a Hanning window. The captured signals that are generated by the setup are depicted in Fig. 16, while the wave source identified is shown in Table 4.

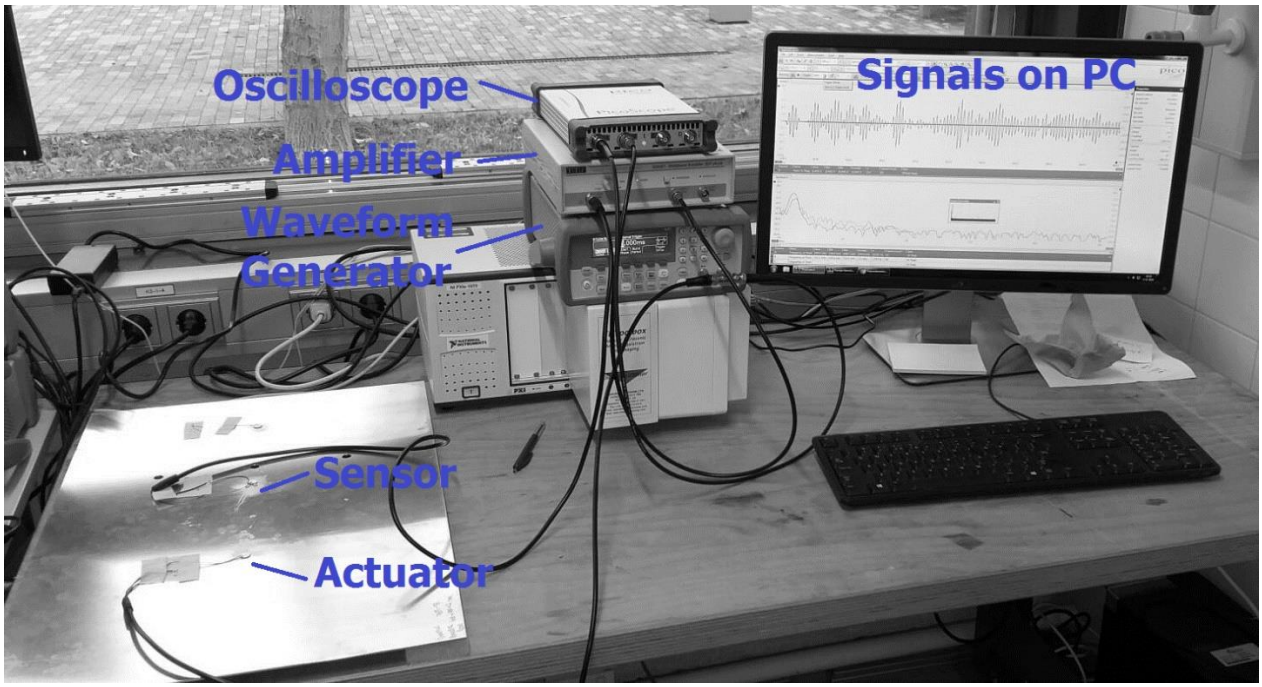

Fig. 15: Experimental setup

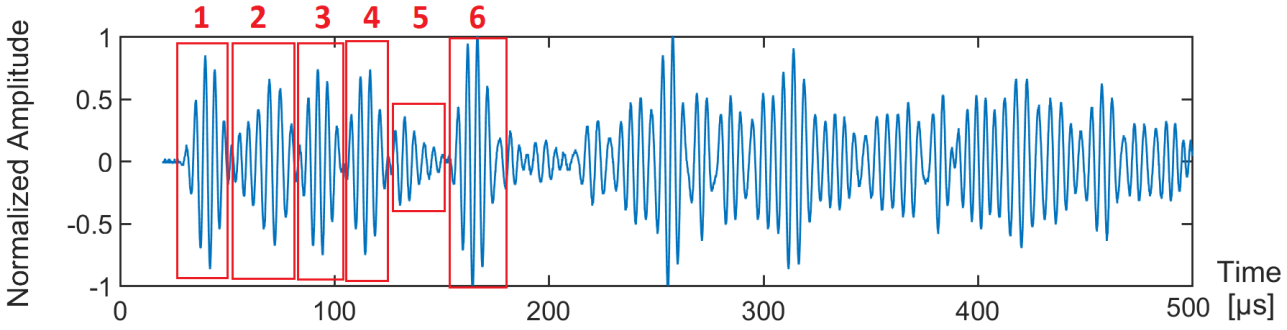

Fig. 16: Captured signal from the experiment.

Table 4. Identified wave packets. S_0- and A_0-mode speed assumption: 5300 m/s and 2900 m/s, respectively

Wave packet	Start at	Wave mode source	Covered shortest distance
1	30.88 μs	S_0 direct from source PZT	16 cm
2	45.28 μs	Reflected S_0 from rivet and A_0 from source PZT	4 cm (S0) ; 16 cm (A0)
3	81.13 μs	Reflected S_0 from top and bottom edge	43 cm
4	105.66 μs	Reflected S_0 from left edge	56 cm
5	120.75 μs	Reflected S_0 from right edge	64 cm
6	148.27 μs	Reflected A_0 from top and bottom edge	43 cm

The signals obtained from simulation are depicted in Fig. 17 top. Note that there are three displacement signals recorded: 2 in-plane displacements parallel and transverse to the wave propagation direction (commonly referred as U1 and U2 in ABAQUS), respectively and the out-of-plane displacement (referred as U3). Since U2 is typically much smaller, only U1 and U3 are presented here. The S_0-mode is dominated by U1 and only minimally influenced by U3, while the vice versa is the case of A_0-mode. Since the common PZT patch is in disk form, it is more sensitive in the radial direction, thus better in capturing S_0-mode, nevertheless it is still able to capture the signal from A_0-mode if the energy is large enough such as waveform 6 (Fig. 16). As the signals in Fig. 17 have been normalized, the energy information is lost, i.e. the maximum energy peak of U1 does not necessarily correspond to the maximum energy peak to that of U3. To compare the simulation and experiment in a less confusing way, only the envelope of the experimental and simulation signals (which can be extracted by Hilbert-Transform) in Fig. 16 and 17 top, respectively and depict the result in Fig. 17 bottom.

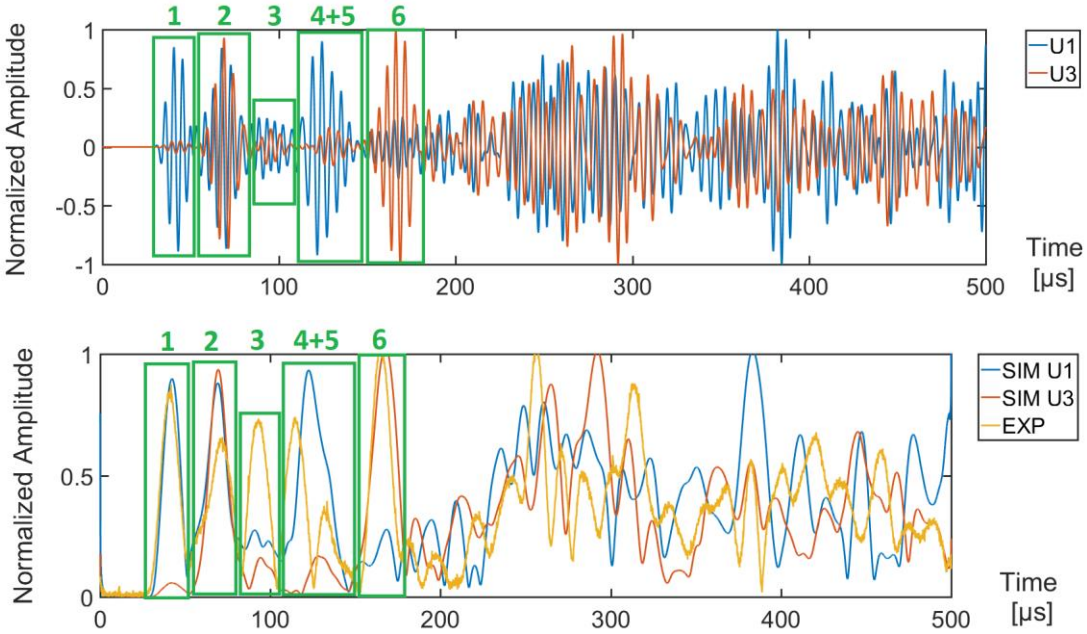

Fig. 17. Top: Normalized FE-simulated signal of in-plane displacement parallel to wave travelling direction (U1, blue) and out-of-plane displacement (U3, red). Bottom: Extracted envelope from experimental signal (Fig. 16) and simulated signal (Fig. 17 top).

5.2 Discussion

From Fig. 17 bottom, it can be seen that there is a relatively excellent match of waveform 1 and 2 (S_0-mode directly from PZT and the rivet reflected S_0-mode + A_0-mode from PZT, see Table 4) from simulation and experiment. The correlation decreases in waveform 3 – 5 and get even worse afterwards. The only exception to this is waveform 6 because the energy that is carried by A_0-mode (as we stated before: largely dominated by U3) in it is quite large in comparison to the other waveforms so that it is enough to suppress U1, and no wonder that the simulated U3 (red in Fig. 17 bottom) has a very good match to the experiment (yellow in Fig. 17 bottom).

The cross-correlation factor between simulated U1 and experiment in Fig. 17 bottom is 40.14% for the whole series (0 – 500 µs). This is relatively bad and will decrease the SHM reliability. However, when limited to waveform 1 and 2 only (0 – 85 µs), the correlation-factor increases to 89.91%. It is noteworthy that our simulation is highly idealized, particularly because rather than modelling PZT, we only took the FE node displacement. Also, there is up to +/- 2% tolerance in geometrical factors (thickness, width, rivet diameter, etc.) and material properties. Furthermore, we used the wideband amplifier to its maximum gain to increase the SNR, but as there is no free lunch, the noise is also increased. From 2 V_{pp} excitation, we were able to measure up to 200 mV_{pp} in the sensing location, which is relatively good especially because we are handling a metallic structure. All these factors contribute to error propagation, and as it can be seen in Fig. 17 bottom, the longer time series, the more complex the signal comparison would be. This is exactly the point where deep learning comes into advantage over human handpicked features selection machine learning: no matter how complex the signal is, as long as the outcome is controlled, deep learning will learn the features automatically and predict in consistent way.

6. CONCLUSION

6.1 Summary

The first section of this paper highlights the state of the art of machine learning and its application in structural diagnostics from which we derived the objective of our work. In section 2, we introduced the concept of the deep learning approach for SHM, while in section 3, we described the implementation of deep learning approach of SHM .The training result is presented in section 4, and lastly in section 5 we shortly described the experimental validation of our work. As a concluding remark, we can summarize the preliminary result of our work in two following main key points:

1. Our simulation correlates little with experiments, so it might not be good enough so if the model parameters of simulation are used to predict experiment data, as it could give low accuracy. However, this is not a problem of our CNN algorithm per se, but rather it is a question of finding a more suitable simulation model for Lamb wave propagation such as using spectral element method [54] instead of using commercial software. So, once the model has been found, our algorithm needs only to be retrained using that data. Nevertheless, finding such a simulation model is typically difficult.

2. As an alternative to overcome this problem, the model can just easily be retrained using real-world data from aircraft. In general, this huge amount of data is easily available at MRO, airlines, and OEM especially if they have been operating for quite some time. However, these labelled data are typically not publicly accessible, thus we believe 1). that our DeepSHM framework would provide little benefit for "closed-source" aircraft maintenance, and 2). given the business interests and conservativeness of the aerospace industry in general – the whole aerospace NDT/SHM community will benefit though slowly from IoT or any other Industry 4.0 related topics. Our community might be left behind in AI and big data by other more generous communities (in terms of data sharing).

6.2 Outlook

As this is the first approach using deep learning for Lamb wave signal classification, there are much room for improvement that can be addressed and as a part of ongoing work, we will continue our research work in the future in:

1. We will incorporate the simulation of PZT patch by using piezoelectric element in ABAQUS instead of just employing point force actuation as this approach is more similar to the real-world condition.

2. In the last part of section 4.3, we mentioned the crack tip shift z, which is typically small. Thus, the interaction between Lamb wave and the crack of length $2a$ would be rarely influenced by small z. However, we think that this can be "tricked" by using higher-frequency excitation, causing a smaller wavelength that might also be influencing the interaction between Lamb wave and small z. This idea can be extended into several frequencies, and thus expanding the wavelet scalogram in Fig. 8 into n-spectral images that depend on n-excitation frequencies. Consequently, there will be several wavelengths that are sensitive to different crack lengths and logically, and thus a smoother transition from "detected vs. not-detected" into pre-failure warning system that can indicate crack of different lengths.

ACKNOWLEDGEMENT AND DECLARATION OF NO CONFLICT OF INTEREST

To make more data available in the field of aircraft maintenance, we would like to make our code, including the training data, open-source. This can be downloaded at Github [55]. Also, the ABAQUS simulation file can be requested by contacting the authors. The authors would like to thank Mr. Xavier Goby who was helpful for a part of the algorithm coding.

This research is part of the TKI Smart Sensing for Aviation Project, sponsored by the Dutch Ministry of Economic Affairs under the Topsectoren policy for High Tech Systems and Materials, and industry partners Airbus Defence and Space, Fokker Technologies - GKN Aerospace, and Royal Schiphol Group.

REFERENCES

[1] Michaels K. "*Opinion: OEMs Focus on Mature Aircraft for Aftermarket Growth*". Aviation Week & Space Technology (2018). Available online <http://aviationweek.com/commercial-aviation/opinion-oems-focus-mature-aircraft-aftermarket-growth>

[2] Ann Shay L. "*Commercial Spending Will Lead MRO Field In 2018*". Aviation Week & Space Technology (2018). Available online <http://aviationweek.com/commercial-aviation/commercial-spending-will-lead-mro-field-2018>

[3] Chong A. "*Global MRO spend to reach $115 billion by 2028 – Wyman*". Flightglobal (2018). Available online <https://www.flightglobal.com/news/articles/global-mro-spend-to-reach-115-billion-by-2028-oli-445243/>
[4] ATA Maintenance Steering Group (MSG) Task Force 3. European Aviation Safety Agency. Rev. 1 (2009).
[5] Advisory circular AC 43-204: Visual Inspection for Aircraft. USDOT Federal Aviation Administration (FAA) (1997).
[6] Mineo C, Pierce SG, Nicholson PI, Cooper I. "*Robotic Path Planning for Non-Destructive Testing – A Custom MATLAB Toolbox Approach*". J Robotics and Computer-Integrated Manufacturing, Vol. 37: 1-12 (2006).
[7] Giurgiutiu V. "*Structural Health Monitoring with Piezoelectric Wafer Active Sensors*". 2nd Ed., Elsevier, Oxford & Waltham (2014).
[8] Boller C, Mofakhami MR. "*From Structural Mechanics to Inspection Processes: Getting Structural Health Monitoring into Application for Riveted Metallic Structures*". Proc. IUTAM Symp on Multi-Functional Material Structures and Systems, Bangalore, 177-185 (2008).
[9] Al Azzawi ASM. "*Analysis of Static and Fatigue Damage in Aluminium/GFRP Hybrid Composites*". PhD Diss, Cardiff University (2017).
[10] Salvetti M, Giliolia C, Sbarufatti C, Dragan K, Chalimoniuk M, Manes A, Giglio M. "*Analytical Model to Describe Damage in CFRP Specimen When Subjected to Low Velocity Impacts*". J Procedia Engineering, Vol. 167: 2-9 (2016).
[11] Mokhtari A, Ouali MO, Tala-Ighil N. "*Damage Modelling in Thermoplastic Composites Reinforced with Natural Fibres under Compressive Loading*". Intl J of Damage Mechanics, Vol. 24(8): 1239-1260 (2015).
[12] Cristianini N, Shawe-Taylor J. "*An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*". Cambridge University Press, Cambridge (2000).
[13] Virupakshappa K, Oruklu E. "*Ultrasonic Flaw Detection Using Support Vector Machine Classification*". Proc. IEEE Intl Ultrasonics Symp (IUS), Taipei, 1-4 (2015).
[14] Schmidhuber J. "*Deep Learning in Neural Networks: An Overview*". J Neural Networks, Vol. 61: 85-117 (2015).
[15] Hopfield JJ. "*Neural Networks and Physical Systems with Emergent Collective Computational Abilities*". Proc. Natl Academy of Science USA, Vol. 79(8): 2554-2558 (1982).
[16] Lee MC, To C. "*Comparison of Support Vector Machine and Back Propagation Neural Network in Evaluating the Enterprise Financial Distress*". Intl J of Artificial Intelligence & Applications, Vol. 1(3): 31-43 (2010).
[17] Hinton GE, Osindero S, Teh YW. "*A Fast Learning Algorithm for Deep Belief Nets*". J Neural Computation, Vol. 18(7): 1527-1554 (2006).
[18] Baldi P. "*Autoencoders, Unsupervised Learning, and Deep Architectures*". Proc. Intl Conf on Unsupervised and Transfer Learning Workshop, Washington, Vol. 27: 37-50 (2011).
[19] Szegedy G, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A. "*Going Deeper with Convolutions*". Proc. IEEE Conf on Computer Vision and Pattern Recognition, Boston, 1-12 (2015).
[20] Simonyan K., Zisserman A. "*Very Deep Convolutional Networks for Large-Scale Image Recognition*". Proc. Intl Conf of Learning Representations (ICLR), San Diego, 1-14 (2015)
[21] Chen Q, Zhu X, Ling Z, Wei S, Jiang H, Inkpen D. "*Enhanced LSTM for Natural Language Inference*". Proc. 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, 1-12 (2017).
[22] Zhao J, Mathieu M, LeCun Y. "*Energy-based Generative Adversarial Network*". Proc. Intl Conf of Learning Representations (ICLR), Toulon, 1-17 (2017).
[23] Zhang L, Yang F, Zhang YD, Zhu YJ. "*Road Crack Detection Using Deep Convolutional Neural Network*". Proc. IEEE Intl Conf on Image Processing, Phoenix, 1-5 (2016).
[24] Cha YJ, Choi W, Büyüköztürk O. "*Deep Learning Based Crack Damage Detection Using Convolutional Neural Networks*". J Computer-Aided Civil and Infrastructure Engineering, Vol. 32(5): 361-378 (2017).
[25] Chaiyasarn K, Sharma M, Ali L, Khan W, Poovarodom N. "*Crack Detection in Historical Structures Based on Convolutional Neural Networks*". Intl J of Geomate, Vol. 15(51): 240-251 (2018).
[26] Fan Z, Wu Y, Lu J, Li W. "*Automatic Pavement Crack Detection Based on Structured Prediction with the Convolutional Neural Network*" (2018). Available online <https://arxiv.org/abs/1802.02208>
[27] Sawaf F, Groves RM. "*Phase Discontinuity Predictions Using a Machine-Learning Trained Kernel*". Applied Optics, Vol. 53(24): 5439-5447 (2014).
[28] Hou W, Wei Y, Guo J, Jin Y, Zhu C. "*Automatic Detection of Welding Defects using Deep Neural Network*". J Physics, Vol. 933 (2018): 012006 (2018).
[29] Yousefi B, Kalhor D, Usamentiaga R, Lei L, Castanedo CI, Maldague X. "*Application of Deep Learning in Infrared Non-Destructive Testing*". Proc. 14th Quantitative InfraRed Thermography Conf, Berlin, 1-9 (2018).
[30] Wunderlich C, Tschöpe C, Duckhorn F. "*Advanced Methods in NDE using Machine Learning Approaches*". Proc. 44th Annual Review of Progress in Quantitative Nondestructive Evaluation, Provo, 1-7 (2017).
[31] Ebrahimkhanlou A, Salamone S. "*Single-Sensor Acoustic Emission Source Localization in Plate-Like Structures Using Deep Learning*". Proc. SPIE Smart Structures And NDE: Health Monitoring of Structural and Biological Systems XII, 1-9 (2018).

[32] Choy AW. "*Structural Health Monitoring with Deep Learning*". Proc. 2018 IAENG International Conf on Control and Automation, Hong Kong, 1-4 (2018).
[33] De Oliveira MA, Monteiro MA, Vieira Filho J. "*A New Structural Health Monitoring Strategy Based on PZT Sensors and Convolutional Neural Network*". J Sensors, Vol. 18: 1-21 (2018).
[34] Ewald V, Goby X, Jansen H, Groves RM, Benedictus R. "*Incorporating Inductive Bias into Deep Learning: A Perspective from Automated Visual Inspection in Aircraft Maintenance*". Proc. 10th Intl Symposium on NDT in Aerospace, Dresden, 1-9 (2018).
[35] Ooijevaar T. "*Vibration-based Structural Health Monitoring of Composite Skin-stiffener Structures*". PhD Diss, University of Twente (2014).
[36] Mitchell T. "*Machine Learning*". McGraw-Hill, Redmond & Ithaca (1997).
[37] MIL-HDBK-1823A. "*Non-Destructive Evaluation System Reliability Assessment*". US Department of Defense, Wright-Patterson (2009).
[38] Hayo T, Frankenstein B, Boller C, Bockenheimer C. "*Approach to the Technical Qualification of a SHM System in Terms of Damage Detection in Aerospace Industry*". Proc. Intl Workshop Smart Materials, Structures & NDT in Aerospace, Montreal, 1-9 (2011).
[39] Kim Y, Huang J, Emery S. "*Garbage in, Garbage Out: Data Collection, Quality Assessment and Reporting Standards for Social Media Data Use in Health Research, Infodemiology and Digital Disease Detection*". J Med Internet Res., Vol. 18: e41 (2016).
[40] Ewald V, Ochoa P, Groves RM, Benedictus RM. "*Design of a Structural Health Monitoring System for a Damage Tolerance Fuselage Component*". Proc. 7th Intl Symposium on NDT in Aerospace, Bremen, 1-9 (2015).
[41] Giurgiutiu V, Yu L. "*Comparison of Short-time Fourier Transform and Wavelet Transform of Transient and Tone Burst Wave Propagation Signals for Structural Health Monitoring*". Proc. 4th Intl Workshop for Structural Health Monitoring (IWSHM), Stanford, 1-9 (2003).
[42] Gao RX, Yan R. "*From Fourier Transform to Wavelet Transform: A Historical Perspective*". In: Wavelets. Springer, Boston (2011).
[43] Karpathy A, Johnson J, Feifei L. "*Visualizing and Understanding Recurrent Networks"*. Proc. Intl Conf on Learning Representations (ICLR), San Juan Puerto Rico, 1-12 (2016).
[44] Jozefowicz R, Zaremba W, Sutskever I. "*An Empirical Exploration of Recurrent Network Architectures*". Proc. 32nd Intl Conf on Machine Learning (ICML), Lille, 2342-2350 (2015).
[45] Shewalkar AN. "*Comparison of RNN, LSTM and GRU on Speech Recognition Data*". MS Thesis, North Dakota State University (2018).
[46] Taigman Y, Yang M, Ranzato MA, Wolf R. "*DeepFace: Closing the Gap to Human-Level Performance in Face Verification*". Proc. 27th IEEE Conf on Computer Vision and Pattern Recognition (CVPR), Columbus (2014).
[47] Morales JL. "*A Numerical Study of Limited Memory BFGS Methods*". J Applied Mathematics Letter. Vol. 15(4): 481-487 (2002).
[48] Shalev-Shwartz S, Ben-David S. "*Understanding Machine Learning: From Theory to Algorithms"*. Cambridge University Press, Cambridge (2014).
[49] Zayani R, Bouallegue R, Roviras D. "*Levenberg-Marquardt Learning Neural Network for Adaptive Predistortion for Time-Varying HPA with Memory in OFDM Systems*". 16th European Signal Processing Conf (EUSIPCO), Lausanne (2008).
[50] Ruder S. "*An Overview of Gradient Descent Optimization Algorithms*" (2016). Available online <https://arxiv.org/abs/1609.04747>
[51] Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. "*Dropout: A Simple Way to Prevent Neural Networks from Overfitting*". J Machine Learning Research, Vol. 15: 1929-1958 (2014).
[52] Keras Documentation (2019). Available online < https://keras.io/optimizers/>
[53] Géron A. "*Hands-On Machine Learning with Scikit-Learn and TensorFlow*". O'Reilly Media, Sebastopol CA (2017).
[54] Sun H, Zhang A, Qing X, Wang Y. "*Spectral Element Method for Modeling Lamb Wave Interaction with Open and Closed Crack*". J of Vibroengineering, Vol. 19(7): 4965-4976 (2017).
[55] Github repository. Available online <https://github.com/vewald/spie2019ewald>