Saeed Ullah, M.Sc. Eng.

GUIDED WAVE-BASED METHODS FOR DELAMINATION IDENTIFICATION ENHANCED BY DEEP LEARNING

A dissertation submitted to the Scientific Board of the Szewalski Institute of Fluid-Flow Machinery, Polish Academy of Sciences in partial fulfillment of the requirements for the Degree of Doctor of Philosophy

Supervisor:
Paweł Kudela, D.Sc. Ph.D. Eng.

October, 2023

ACKNOWLEDGMENTS

The gratification that accompanies the successful accomplishment of this thesis would not be completed without the mention of those people who made it possible and whose motivation, encouragement, and guidance have been a vital source of inspiration throughout the course of my PhD studies.

I would like to express my sincere gratitude to my supervisor Dr. Paweł Kudela for his valuable guidance, inspiration, and productive suggestions throughout the course of this work. I thank him for his patience, motivation, and support that helped me to make my stay at The Szewalski Institute of Fluid-Flow Machinery, Polish Academy of Sciences (IMP PAN), Poland. It has been an honor for me to be his student.

I would also like to thank all of my colleagues and friends at IMP PAN and in Poland, who made my time enjoyable and memorable. I thank all of them for accepting me into their team with an open heart and keeping me involved in different team and cultural-based activities, for helping me to get accustomed to Polish life, for encouraging me, and always being there for me to help. Thank you all for being my family away from home.

I would like to thank my parents for their endless love, trust, encouragement, and support throughout my life.

I would like to thank my wife, Sedra Atta, and my son, Ehan Saeed, for their consistent support and encouragement.

ABSTRACT

The methods of non-destructive testing (NDT) and structural health monitoring (SHM) play a pivotal role in predicting the remaining functional life of structures through precise diagnostic and prognostic techniques. Their primary objective is to identify and characterize potential defects that could jeopardize the structural integrity and performance. However, it is noteworthy that advancements in SHM and NDT, particularly those reliant on guided wave-based methodologies, have experienced a slowdown in recent years. This deceleration can be attributed to the limitations associated with traditional signal processing techniques when applied to intricate and the formidable challenge of extracting damage-related features from propagating wave signals.

Conversely, there has been a remarkable acceleration in the domain of artificial intelligence (AI) methods, particularly in deep learning and computer vision, in recent times. This progress has unveiled new avenues for problem-solving and presents opportunities for seamless integration with NDT and, subsequently, SHM methodologies.

The primary aim of this dissertation is to create an innovative AI-driven diagnostic system tailored for the identification of delamination in composite laminates, specifically for carbon fiber-reinforced polymers (CFRP). This endeavor involves exploring the potential of leveraging artificial neural networks (ANNs)-based methods to enhance damage identification through the analysis of Lamb wave propagation. The ANNs-based systems developed in this context employ an end-to-end approach, enabling the direct transformation of propagating Lamb wave animations into comprehensive damage maps.

Moreover, the dissertation addresses the challenge of slow data acquisition inherent in high-resolution full wavefield imaging techniques. To surmount this hurdle, I introduced a deep learning solution designed to reconstruct high-resolution frames depicting Lamb wave propagation, as well as their interactions with delaminations and structural boundaries from low-resolution measurements. This innovative approach promises to expedite the data acquisition process significantly.

Furthermore, another approach of deep learning-based surrogate modeling for solving the inverse problems of delamination identification is also proposed in this study. This surrogate model is able to predict the full wavefield of Lamb waves interacting with delamination in much shorter time than by using traditional finite element method.

PUBLICATIONS

Journal papers

1. Abdalraheem A. Ijjeh, **Saeed Ullah**, and Pawel Kudela. “Full wavefield processing by using FCN for delamination detection.” Mechanical Systems and Signal Processing 153 (2021): 107537.

2. **Saeed Ullah**, Abdalraheem A. Ijjeh, Pawel Kudela, “Deep learning approach for delamination identification using animation of Lamb waves.” Engineering Applications of Artificial Intelligence, Volume 117, 2023, 105520.

3. Abdalraheem A. Ijjeh, **Saeed Ullah**, Maciej Radzienski, Pawel Kudela, “Deep learning super-resolution for the reconstruction of full wavefield of Lamb waves.” Mechanical Systems and Signal Processing, Volume 186, 2023, 109878.

4. **Saeed Ullah**, Pawel Kudela, Abdalraheem A. Ijjeh, Eleni Chatzi, Wieslaw Ostachowicz “Simulation of full wavefield data with deep learning approach for delamination identification.” Mechanical Systems and Signal Processing (Under review).

Conference Papers

1. **Saeed Ullah**, Pawel Kudela, “Delamination detection by using Pyramid Scene Parsing Network on Guided waves,” 10th International Conference on Structural Health Monitoring of Intelligent Infrastructure, 30th June – 2nd July 2021, Porto, Portugal.

2. **Saeed Ullah**, Pawel Kudela, “ConvLSTM Based Approach for Delamination Identification Using Sequences of Lamb Waves,” In: Rizzo, P., Milazzo, A. (eds) European Workshop on Structural Health Monitoring. EWSHM 2022. Lecture Notes in Civil Engineering, vol 270. Springer, Cham. `https://doi.org/10.1007/978-3-031-07322-9_55`

3. **Saeed Ullah**, Pawel Kudela, Wieslaw Ostachowicz, “A deep learning based super-resolution approach for the reconstruction of full wavefields of Lamb waves,”

50th Annual Review of Progress in Quantitative Nondestructive Evaluation, July 24–27, 2023, Austin, TX.

Scientific monographs

1. **Saeed Ullah**, Guided Wave Based Structural Health Monitoring, chapter in: Wybrane zagadnienia inzynierii mechanicznej, Praca zbiorowa pod redakcja M. Mieloszyk, T. Ochrymiuka, Wydawnictwo Instytutu Maszyn Przep ływowych PAN, Gdansk, 2020, ISBN 978-83-88237-97-3.

2. **Saeed Ullah**, Artificial Neural Networks and its applications in Guided Wave-based Damage Detection, chapter in: Wybrane zagadnienia inzynierii mechanicznej 2021, Praca zbiorowa pod redakcja M. Mieloszyk, T. Ochrymiuka, Wydawnictwo Instytutu Maszyn Przepływowych PAN, Gdansk, 2021, ISBN 978-83-66928-00-8.

3. **Saeed Ullah**, Deep learning-based approach for delamination identification using animation of Lamb waves propagation, chapter in: Wybrane zagadnienia inzynierii mechanicznej 2022, Praca zbiorowa pod redakcja M. Mieloszyk, T. Ochrymiuka, Wydawnictwo Instytutu Maszyn Przepływowych PAN, Gdansk, 2022 (Accepted).

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

The presented work in this dissertation is the outcome of 4 years research work on two projects, both of which were funded by Polish National Science Center:

- Feasibility studies of artificial intelligence-driven diagnostics, grant agreement no. 2018/31/B/ST8/00454 (3 years).
- Fusion of models and data for enriched evaluation of structural health, grant agreement no. 2019/01/Y/ST8/00060 (1 year).

1.1 Problem Statement

Composite materials find extensive applications across various industries, primarily owing to their exceptional qualities, including high tensile strength, low density, and resistance to fatigue and corrosion. In composite structures, a perplexing form of damage that emerges is known as inter-laminar delamination. This type of delamination typically initiates from micro-cracks within the matrix material and follows a non-linear progression, especially when subjected to cyclic loading conditions [1–3]. The presence of delamination can lead to a reduction in the compression strength of composite laminates over time, ultimately rendering the composite structure susceptible to failure, particularly through buckling mechanisms. As a result, the overall performance of composite structures can be severely compromised. Thus, the early detection and identification of delamination when it is still in its incipient stages can play a crucial role in preventing catastrophic structural failures.

Different methodologies for non-destructive evaluation (NDE) and structural health monitoring (SHM) have been employed in the context of damage detection within composite structures. These approaches can broadly be categorized into two main groups: model-based methods and data-driven techniques.

Model-based approaches aim to replicate the process of damage development in composite materials by employing physics-based numerical models and incorporating relevant variables to tailor the model to specific application scenarios. Nevertheless, these model-based methods come with practical limitations that constrain their applicability to relatively simple structures in controlled environments.

Conversely, data-driven approaches for NDT/SHM rely on signal processing of data recorded by sensors in response to the structure's exposure to various external conditions. Among these data-driven strategies, those leveraging artificial neural networks have gained prominence thanks to advancements in hardware as well as computer vision methods. Consequently, deep learning-based approaches have opened up new avenues for problem-solving and integration with NDE/SHM methods. This application of deep learning (DL) techniques addresses challenges related to data preprocessing and feature extraction. Contemporary approaches adopt an end-to-end paradigm, where unprocessed data is directly inputted into the model to generate a damage map as output, allowing the model to autonomously recognize patterns and detect damage.

1.2 Purpose of the Study

In recent years, artificial neural networks (ANNs) have found applications across various scientific disciplines. In alignment with this trend, I intend to leverage ANNs-based methods within the domain of non-destructive testing and structural health monitoring (NDT/SHM) to craft an ANNs-based diagnostic system tailored for identifying delamination in composite laminates. This adoption of ANNs promises to enrich several facets of NDT/SHM, spanning elastic wave behavior, image processing, and animation analysis.

Hence, I propose an end-to-end DL approach that autonomously conducts feature extraction of delamination characteristics. This implies that the supervised DL model will independently learn to extract damage-related features and subsequently identify and localize damage in the composite laminates. The chosen approach for this endeavor is supervised learning. To facilitate this, a substantial dataset representing the full wavefield of propagating elastic waves was generated numerically. It is worth noting that the creation of such a dataset is a time-intensive process, taking approximately three months. Consequently, I anticipate that implementing this system will serve as an additional safety measure for operational structures, consequently reducing maintenance costs.

1.3 Objectives and Motivation

The primary aim of this research is to establish an ANN-based diagnostic system geared towards identifying delamination within composite laminates. Additionally, it seeks to investigate the possibilities of utilizing ANN-based methodologies to delve into damage detection and identification, leveraging the propagation of Lamb waves.

The data related to elastic wave propagation encompasses complex patterns of wave reflections. Explicitly programming instructions to yield a damage map for a structural element based on anomalies in propagating elastic waves, such as reflections from structural discontinuities, proves challenging. Consequently, this study endeavors to explore potential solutions through the adoption of deep neural networks (DNNs), given their promising capabilities. Recent advances in machine learning, coupled with increased computational power, provide an opportune moment for delving into potential applications of DNNs.

DNNs represent an emerging tool that has demonstrated successful applications in fields like computer vision and speech recognition, surpassing human-level accuracy in image classification [4–6]. A key advantage of DNNs, compared to other machine learning approaches, is their scalability. Performance improves in tandem with both the size of the neural network and the dataset employed for supervised learning. This scalability paves the way for an end-to-end approach in which DNN processes the animation of propagating waves (input) directly into a damage map (output).

Main thesis:

Therefore, it is possible to develop DL methods that surpass conventional full wavefield processing methods for delamination identification in composite laminates.

Another objective of this research centers on addressing the challenge of slow data acquisition in obtaining high-resolution full wavefields of Lamb wave propagation. As a solution, the aim is to develop a DL system proficient in restoring high-resolution frames of Lamb wave propagation, along with their interactions concerning delamination and structural boundaries, using low-resolution measurements while maintaining high accuracy.

Sub-thesis 1:

Therefore, it is possible to develop a DL-based super-resolution method that significantly reduces the data acquisition time by the SLDV.

The final objective of this research work is the development of DL-based surrogate model for solving the inverse problems of delamination identification. The surrogate model developed in this research work takes the full wavefield of Lamb waves without delamination (delamination information is provided in binary form) as inputs and is able to produce full wavefield with delamination.

Sub-thesis 2:

Therefore, it is possible to develop a DL-based surrogate method that can be efficiently utilized in the inverse problems of delamination identification.

This research seeks to provide insights into the utility of ANN-based techniques by analyzing the full wavefield of propagating elastic waves for delamination identification in composite laminates:

- Can the proposed ANN-based diagnostic model outperform conventional signal processing techniques?
- Given that experimental signals exhibit noise, is it sufficient to employ a numerical model for dataset generation?
- To what extent can DL models generalize to previously unseen data, including experimental data acquired via SLDV?
- Is it computationally practical to utilize all frames of propagating waves, or can the utilization of specific frames prove efficient?
- Does the DL technique employed for delamination identification in this thesis harbor long-term potential for practical applications?
- Can the DL technique designed for super-resolution image reconstruction be repurposed to recover high-resolution full wavefield frames from low-resolution measurements with sufficient accuracy for damage detection?
- Is it possible to eliminate the dataset computation by just developing a DL-based surrogate model?
- Will the inverse problem in damage detection be solved with a DL-based surrogate model?

1.4 Thesis Contribution

This research work is composed of three different tasks:

i. **Delamination identification:** This work introduces a novel approach by employing a full wavefield dataset of elastic wave propagation, generated numerically, as input into the developed model to generate a comprehensive damage map. For this purpose, ConvLSTM model was specially designed.

ii. **Reconstruction of full wavefields of Lamb waves:** A DL model has been developed to facilitate an end-to-end super-resolution technique concerning elastic wave propagation and its interaction with damage and structural boundaries. The process begins with acquiring a low-resolution input frame utilizing a uniform mesh grid with a minimal number of scanning points. These low-resolution frames are then fed into a DL model, which transforms them into high-resolution frames. Subsequently, the predicted high-resolution frames can be input into the many-to-one model to identify and characterize damage.

iii. **Simulation of full wavefield for delamination identification:** This research embraces an innovative approach centered on DL to explore the dynamics of guided wave propagation within composite structures containing different delamination scenarios. At its core, this approach harnesses the power of a sophisticated deep ConvLSTM autoencoder-based surrogate model, specially designed for generating comprehensive full wavefield data that mimics the behavior of propagating guided waves within these composite structures. Moreover, this cutting-edge DL model, employed for predicting full wavefield patterns, marks its pioneering application within the domain of the inverse problem associated with delamination identification.

1.5 Thesis Organization

Chapter 2 serves as an introduction to Structural Health Monitoring (SHM) and its applications, focusing on SHM using Guided Waves (GWs) for composite materials. Additionally, this chapter provides an overview of damage detection and localization techniques employing GWs. Some literature review of GW-based damage detection in composite structures (without the use of AI methods) is also provided.

In chapter 3, the focus shifts to Artificial Neural Networks (ANNs) approaches applied within the domain of GW-based NDT/SHM. The chapter delves into the comparison of artificial intelligence, machine learning, and deep learning, working procedure of ANNs, and best practices for training a neural network are elaborated. Different variants of ANNs algorithms are explained. Furthermore, it offers an overview of ANNs-based approaches using GW-based NDT/SHM in composite structures.

Chapter 4 is dedicated to the acquisition of a synthetic dataset that resembles the full wavefield of Lamb waves propagating in a Carbon Fiber Reinforced Polymer (CFRP) plate. The dataset encompasses interactions between these waves, structural boundaries, and discontinuities, such as damage. This chapter also introduces data preprocessing, and dataset splitting process performed in the upcoming chapters.

Chapter 5 describes the first task of this research work (delamination identification) by using DL-based approach. The ConvLSTM model is explained. The assessment of the proposed approach is elaborated, which includes numerical test cases to demonstrate their efficacy in predicting previously unseen data. Furthermore, the proposed DL model undergoes evaluation using experimentally acquired data, featuring single and multiple Teflon inserts simulating delamination scenarios, to assess the generalization capabilities of the developed DL model.

Chapter 6 describes the second task of this research work (reconstruction of full wavefields of Lamb waves) by using a DL-based approach. The need for the super resolution based approach for the reconstruction of full wave fields from low-resolution into high-resolution is elaborated. The assessment of the proposed super-resolution approach is elaborated, which includes numerical test cases to demonstrate its efficacy in predicting previously unseen data. Furthermore, the results of the proposed super-resolution-based approach are compared with the conventional compressive sensing technique.

Chapter 7 describes the third task of this research work (simulation of full wavefield for delamination identification) by using a DL-based approach. The construction of full wavefield data by using a DL-based surrogate model is explained. The assessment of the proposed approach is elaborated, which includes numerical test cases to demonstrate their efficacy in predicting previously unseen data. Furthermore, the results of the proposed DL-based surrogate model are compared with the results of the delamination identification task (of Chapter 5).

Chapter 8 provides a summary of the findings and insights derived from this dissertation. It also outlines potential future research directions stemming from the work presented here.

Chapter 2

Guided Wave-based SHM/NDT

2.1 Introduction

Numerous civil engineering and aerospace structures are approaching or exceeding their design lives. Therefore, assessing the condition of these structures is essential to determine their serviceability, safety, and load-carrying capacity [1–3]. It is crucial to monitor the health of structural elements in mechanical, civil, and aerospace industries where the presence of small defects may result in a very catastrophic failure [3]. A defect or damage can be defined as any degradation in the structural properties that alters the dynamic behaviour of the structure [7, 8]. These changes can occur at either micro-scale level such as matrix anomalies, or in the macro-scale level such as cracks. Damage negatively influences the current or future performance of a structure. Recently, damage detection methods have been widely studied for the purpose of locating and quantifying structural defects. There are many ways and indicators for detecting damage in a structure, such as variations in strain, stiffness reduction, and natural frequencies, etc. [9].

Damage detection is usually part of one or more related approaches, including Structural Health Monitoring (SHM), Nondestructive Evaluation (NDE), also known as Nondestructive Testing (NDT), Condition Monitoring (CM), Health and Usage Monitoring System (HUMS), Statistical Process Control (SPC) and Damage Prognosis (DP) [1, 8]. SHM can be defined as the process of implementing a damage detection and health assessment system for civil, mechanical, or aerospace infrastructure [1, 8]. The SHM process includes continuous monitoring of a mechanical structure or system using dynamic response measurements. Damage-sensitive features acquired from these measurements are employed to determine the current status of the system's health [1, 8]. The output of these measurements can be periodically updated for long-term SHM, which is particularly useful in the case of extreme events. SHM provides reliable information and rapid condition assessment about the performance of the system in near real-time [8].

NDT/E is commonly carried out offline in a local manner by the operator [3, 10]. NDT/E is primarily used for severity checks and damage characterisation when the damage location is already known [1, 8]. NDT/E systems are often limited to single-point measurements [3]. Most NDT/E techniques require component disassembly, especially for those components that are inaccessible which results in interruption of the daily operation of the structural systems leading to downtime which consequently results in an increase in operational costs. NDT/E systems are typically costly, labor-intensive, and time-consuming, relying heavily on the expertise and skills of the operator [11].

CM is analogous to SHM but typically concerns damage identification in rotating machinery, such as those used in power generation and manufacturing [12].

HUMS has been adopted for damage detection in rotorcraft drive trains [13]. In this context, the health monitoring section helps distinguish damage, while the usage monitoring reports the estimate of load cycles that the system encounters for the purposes of assessing fatigue life consumption.

SPC is a process-based technique, and its final aim is to process diagnostics. It employs various sensors for monitoring changes in a process [3, 14].

DP is employed for predicting the remaining useful life of a system once the damage has been detected [1, 7, 8]. DP systems use the knowledge about the location and size of damage, as well as expected operational loads, to predict the remaining life of a structure [3].

2.2 Structural Health Monitoring

SHM aims to detect, identify, and characterise damage and degradation in engineering structures [15]. SHM serves as a critical technical foundation for the development of condition-based maintenance strategies. These strategies hinge on real-time assessments of a structure's health and performance, ultimately enhancing safety while concurrently lowering operational and maintenance expenses. SHM is a more reliable and modified variant of NDT/E [11]. SHM can reduce costs and maintain a decent level of safety regarding performance conditions, particularly for complex structures [16]. SHM is extensively adopted by the aerospace community, especially for aging aircraft situations [17]. Sensors are used in the SHM system to monitor physical quantities such as acceleration, strain, tensile and compressive stress, and so on [18]. SHM techniques should exhibit the following characteristics:

i. Low possibility of missing the damage.

ii. Suitability for continuous online monitoring.

iii. Handling of large datasets applicable for large engineering structures [19].

iv. Rapid calculation.

The SHM system tasks can be categorised as a process composed of four operations that make four primary elements or levels, as shown in Fig. 2.1. These levels are: (i) damage detection, (ii) damage localization, (iii) damage assessment, and (iv) life prognosis [3, 20].

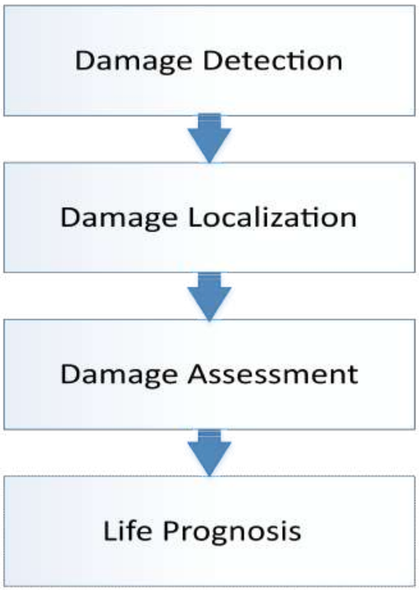


Figure 2.1: Four major levels of SHM.

According to these levels, damage detection level provides a qualitative explanation that damage may exist, the localization level gives an indication about the feasible region of the damage, the assessment level indicates the estimation of the severity of damage by providing information regarding the size and type of damage, and finally, the evaluation of remaining structural life is provided at the prognosis stage. The prognosis level also predicts possible failures or breakdowns. The initial three levels (i.e. detection, localization, and assessment) are usually related to damage identification, signal processing, and modelling features. The last level (i.e. life prognosis) falls into statistical analysis, reliability, fracture mechanics, fatigue analysis, and design assessment fields. Many researchers have comprehensively investigated the prognosis level but currently, there are no commercially available solutions [3, 20].

2.3 Basic Elements of an SHM System

A typical SHM system is composed of three basic elements:

i. A network of actuators and sensors, most likely smart materials, which are permanently connected to the structure. This perspective of SHM makes it different from conventional NDT/E techniques and it is mandatory for executing automated inspections. This step involves observation of the structure from arrays of sensors using regular sampled response measurements, storing the measured data, and transmitting data to the control center.

ii. On-board information administration and computing equipment. A large number of sensors are continuously producing a huge volume of data for processing in real-time. Computational power and data transmission are employed within a structure for the purpose of detecting, localizing, assessing, and predicting defects that can produce structural impairment immediately or later in the future.

iii. Algorithms that examine collected data from the structure with the recently obtained data. Algorithms also determine a damage index and then notify about the existence, position, type, and size of the damage [3, 19, 21, 22].

For monitoring of possible changes in the structures, all SHM systems need a proper sensor network. The sensitivity of the SHM techniques is usually linked to better interaction between the sensors and the structure. Therefore, it is essential to choose suitable sensors to be installed. During the implementation of a network of sensors in an SHM system, the information for obtaining defect identification, the material of the structure to examine, and the variables to measure are considered. The next stage is data acquisition. At this stage, the signals produced by each sensor are obtained. The characteristics of SHM systems such as mobility, scalability, and costs are considered. The data collected by the SHM can be influenced by sensor configuration, environmental and operational noise, or any other event. Many of these obstacles need to be resolved before executing any analysis on the produced information for generalising the methods applied for recognition, identification, or classification. This level is associated with preprocessing or signal conditioning. It can be accomplished with the use of hardware accessories, software algorithms, or both. Finally, data analysis tools are used to determine the presence of defects in the instrumented structure and to characterise the possible source of damage [20, 22].

2.4 Classification of SHM Methods

Damage detection techniques for SHM can be categorised into two major classes based on the type of information to be used: Local and Global methods [3, 19].

Local techniques monitor a tiny part of the structure surrounded by sensors using measurements of structural response [3, 19]. Comparative Vacuum Monitoring (CVM), ultrasonic waves, eddy currents, acoustic emission, and Electromechanical Impedance (EMI) are normally considered as local SHM techniques [21, 23, 24].
Global techniques are executed when global motion is induced in the structure during its operation. Vibration-based methods and thermography systems are the examples of this class. Global methods are helpful when local damage has an influence on the behaviour of the global structure in terms of space and time [3, 19]. As opposed to local methods, global methods have many benefits:

i. The whole structure can be measured with these methods by using a sparse sensor network.

ii. It is not necessary to locate sensors near the damage.

iii. Only a little information about the critical region is enough.

Global SHM techniques also have some limitations:

i. The wavelength of vibrations is nearly equivalent to the dimension of the component or structure, therefore, these methods have relatively low sensitivity to small defects.

ii. Usually these methods are quite expensive [3].

Although the damage size and location are roughly estimated with global methods, it can be successfully utilised for damage detection. The relationship between defects in structures and structural vibration is used in health assessment [3, 25].

Recently, three types of SHM techniques have been successively developed: (1) SHM based on vibrations [26]; (2) Electromechanical Impedance (EMI) based SHM [23, 24]; and (3) Guided wave-based SHM [27–34]. Vibration-based SHM techniques detect damage by measuring vibration signals on structures. These techniques identify local defects by detecting changes in modal shapes, natural frequencies, or dynamic responses [26]. EMI-based techniques identify changes in structures by measuring the EMI of a PZT transducer, which is connected to (or embedded into) the monitored structure [23, 24]. Guided Waves (GWs) are elastic waves that propagate along a path determined by the structure boundaries. GWs and the applications of GWs in SHM/NDT are explained in the next section.

2.5 Guided Wave-based SHM/NDT

2.5.1 Guided Waves

GWs are of particular interest for NDT and SHM of engineering structures such as rails, aircraft components, and oil and gas pipelines [35–38]. Developing a GW-based technique needs attentive understanding obtained over-analysis and modelling of mode-damage interaction and wave propagation because of the dispersion and multimodal character of these waves [39]. These techniques have several benefits:

i. A short wavelength provides plentiful interaction of the GWs and relatively small defects.

ii. These waves have very high excitation frequency, therefore, low-frequency ambient influence can be filtered out.

iii. These waves can propagate for long distance inside the structure under investigation.

iv. GWs provide high sensitivity to various types of defects and the extent of the monitored area [40].

In these techniques, an actuator is used for generating the waves. Signals are received by the transducers at one or multiple locations which are then analyzed for the identification of defects [40]. Using GW-based SHM/NDT technique, damage can be detected and monitored with a very small number of transducers [41–43]. In many GW-based SHM/NDT techniques, collections of actuators/sensors are adjusted on a plate which not only identifies the existence of defects but also localize the defects [8].

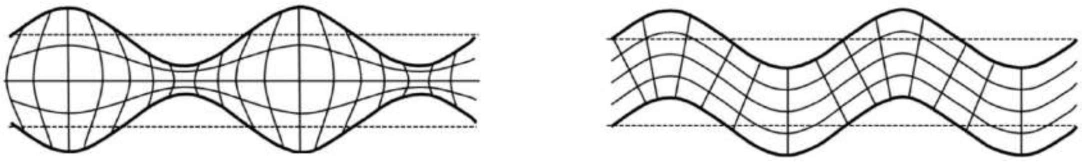
In the GW-based SHM/NDT techniques, usually, a dense network of Lead Zirconate Titanate (PZTs) is utilised for sensing and excitation of signals. It provides the ability of online monitoring and the permanent integration of modifications in GWs propagation. However, the problem of applying an array composed of a few PZTs for the localization of defects is that the resolution of the imaging of defects may be very low. Whereas, applying a highly dense arrangement of PZTs is also not feasible in most situations. This problem can be solved by adopting Scanning Laser Doppler Vibrometry (SLDV). SLDV is capable of measuring GWs under a very dense grid of points over the surface of a large specimen. This combination of signals is known as a full wavefield [44, 45].

2.5.2 Lamb Waves

Various types of GWs have been studied for application in SHM/NDT. Lamb waves, named after the British applied mathematician Horace Lamb [46], are one of the most popular types of GWs utilised for damage detection [41]. Lamb waves are a form of ultrasonic GWs that propagate in plate-like structures [3, 41, 46]. Lamb wave interrogation of structures is an appealing SHM/NDT technique for damage detection and monitoring [34, 43, 47, 48]. Lamb waves can travel long distances with minimum dispersion [49, 50]. The high-frequency and low attenuation nature of these waves allow it to be used for the detection of small defects [51]. By using Lamb waves, damage in structures can be evaluated by analyzing anomalies in the propagation of Lamb waves, considering backward and forward scattering [47, 48]. These waves provide more enhanced information about the presence, size, type, location, and severity of damage than frequency response techniques [52]. Both internal and external defects in large monitoring structures can be inspected with Lamb waves. Due to these reasons, Lamb waves have been established as highly suitable for damage detection in plate-like structures [53]. Both contact and non-contact type of transducers, such as air-coupled ultrasonic transducers, can be employed for the generation of Lamb waves. Piezoelectric transducers made of PZT ceramics are mostly used for the generation and detection of Lamb waves [54–57]. Sophisticated signal-processing techniques are used for processing of the dynamic response signals received by PZT transducers [58].

These waves are guided by the free surfaces of plates. The wavelength of Lamb waves is similar or larger than the thickness of the plate. These waves couple shear and longitudinal waves within a plate [8]. To use Lamb waves for SHM/NDT techniques, it is beneficial to possess a waveform which is efficiently recognisable before and after the propagation through the plate. The frequency dependency of wave speed in Lamb waves makes these waves dispersive. A narrow-band frequency helps reduce the unwanted dispersive nature of Lamb waves [52]. By choosing a proper driving frequency, the response of the input signal is recorded at receiving sensors with minimal interference [8].

The propagation of Lamb waves can be categorised into either Symmetric (S_0) or Antisymmetric (A_0) modes with respect to the neutral plane of the plate. A graphical representation of Symmetric (S_0) and Antisymmetric (A_0) modes of Lamb waves in thin plate-like structures is shown in Fig. 2.2. At lower ultrasonic frequencies only these two fundamental modes (the S_0 mode and the A_0 mode) of Lamb waves exist. In the range of low frequency-thickness product values, the S_0 mode's motion involves stretching and compressing the plate, with a predominant in-plane displacement, as depicted in Fig. 2.2a, and it tends to propagate over long distance. In contrast, the A_0 mode predominantly induces an out-of-plane motion, causing the plate to bend, as illustrated in Fig. 2.2b. The A_0 mode of Lamb waves is quite valuable for damage identification. The wavelength and wave velocity of the A_0 mode is relatively smaller than the S_0 mode. The smaller wavelength of the A_0 mode is very helpful for interacting with the damage, as the half wavelength of a chosen wave mode must be smaller or similar to the size of the damage. A_0 modes can efficiently be generated by the surface mounted transducers [59].



(a) Symmetric (S_0) mode

(b) Antisymmetric (A_0) mode

Figure 2.2: Representation of Symmetric (S_0) and Antisymmetric (A_0) modes of Lamb waves.

The main problem of Lamb waves is that it is an active technique. It requires a regular supply of voltage and function-generating signals [52]. The resulting data of Lamb waves is more complicated than many other techniques. Therefore, the interpretation of these signals is also very difficult [52]. Despite extensive research efforts, Lamb wave-based real engineering applications are still limited. This is mainly due to the complexity of the propagation of these waves, i.e. dispersion, possible reflections from boundaries, multimodal nature, and other structural features that generate a wave field which is laborious to analyze [3].

2.6 Sensors and Transducers used in GW-based SHM/NDT

In the last few years, the structural engineering community has increasingly implemented sensor networks for the SHM/NDT purposes of monitoring structures. For an SHM/NDT system, it is essential to acquire a proper assessment of a system's dynamic response. There are several different types of sensors and data retrieval methods that can be implemented in the SHM/NDT. The sensors used in an SHM/NDT system are application-specific [8].

GW-based SHM/NDT sensing and data acquisition system consists of some or all of the below-mentioned elements:

i. **Transducers** are responsible for converting changes in the domain variable of interest, i.e., temperature, strain, or acceleration, to changes in an electrical signal, i.e., voltage, resistance, or impedance.

ii. **Actuators** are responsible for applying a prescribed input to the system, i.e., a piezoelectric transducer attached to the surface of a structure.

iii. **Analogue-to-Digital (A/D) converters** are responsible for converting analogue electrical signals into digital signals, which can be processed later on digital hardware. When a SHM system is using actuators, a Digital-to-Analogue (D/A) converter is also required for converting a designated digital excitation signal to an analogue voltage, which is useful for controlling the actuator [8, 20].

Damage detection for a SHM/NDT system requires the employment of an arrangement of sensors with the primary function of capturing data which can be employed for determining the state of the monitored structure. Many SHM/NDT techniques employ elastic wave signals which are produced by an actuator. The inspection methods depend on the type of transducer used for inspection and the type of propagating signals over the structure. The transducers employed in the SHM/NDT system should be light and small in size for the purpose of integration into the structure without any considerable impact on its behaviour [3]. Due to the improved implementation of SHM/NDT methods, many new sensors have been developed. These new sensors are very helpful in improving the ability to detect, characterize, and localize defects in an SHM/NDT system [60]. These advancements in sensors strive to reduce the weight and power consumption of the system. In addition, it also helps in resolving installation problems, and improve data analysis and operation facilities [20]. Sensors can be categorised based on the physical variable which they sense or on the principle of transduction on which they are based. The subsequent sections illustrate some of the various types of sensors used in GW-based SHM/NDT systems. These sensors can be adapted for the inspection of both composite and metallic structures [8, 20, 41].

2.6.1 Piezoelectric Sensors

Piezoelectric materials are made of polymers and ceramic. These transducers are mostly used for the sensing and generation of GWs [54–57]. Therefore, these devices are very suitable for GW-based SHM/NDT [3]. These materials can also act as actuator and electricity harvester [61]. Piezoelectric transducers are lightweight, small in size, consume a low amount of power, and are able to generate a broad frequency response [3, 62]. Some other advantages of piezoelectric sensors are: these sensors can be assembled in various forms, such as circular and rectangular [3, 20]. These transducers can easily be arranged as arrays of sensors in order to record multi-point measurements on or under the surface of the monitored structure [63]. Lead Zirconate Titanate ($Pb[Zr_xTi_{1-x}]O_3$, ($0 \leq$ x ≤ 1), PZT), barium titanate ($BaTiO_3$), and ferroelectric polymers, such as polyvinylidene difluoride (PVDF), are the most commonly used materials in these sensors [64].

PZT transducers can be used for rapid and in situ SHM. With piezoelectric transducers, it is easy to measure wave propagation and acquire information about various damage types, such as corrosion or delamination [20, 41]. However, the methods utilising these transducers sometimes require a high amount of data points and the baseline signal for the purpose of comparison with the damaged signal for accurate damage estimation [8, 65].

Pressure-based piezoelectric sensors can measure the changes in strain, pressure, temperature, acceleration, and force by applying an electrical charge. These piezoresistive pressure sensors are also called strain sensors or strain gauges [60, 66]. These kinds of sensors have widely been applied in medical, nuclear, and aerospace instrumentation [66]. Piezoresistors (essentially piezoelectric ceramic) are frequently used in EMI techniques [11, 20]. Strain gauges are sensitive but provide only localized measurement. Changes in vibrations, frequency, displacement, and mode shapes can be measured with accelerometers [67]. Typical accelerometers are quite bulky in size and do not provide good sensitivity. Whereas, piezoelectric accelerometer, Surface Micromachined Piezoelectric Accelerometers (PiXLs), and piezoresistive Microelectromechanical Systems (MEMS) accelerometers are light in size, less expensive, provide better sensitivity, and better resolution [60, 68–70].

2.6.2 Fiber Optics

Fiber Optic Sensors (FOS) are employed in situations where high precision and immunity to electromagnetic interference are required [71]. In FOS, an optical fiber is operated throughout the structure, and then optical reflectometry is conducted for the detection/localization of defects using distributed strain through scattering processes [72]. FOS generally consist of:

i. Core, a thin glass fiber.

ii. Cladding, which confines the dispersion of light within the fiber core.

iii. Protective coating, which absorbs moisture and also provides mechanical strength [73].

FOS have proved very efficient for SHM/NDT and are good competitors with other sensors. These sensors provide better resistance to electromagnetic interferences and noise [60]. FOS offer significant advantages in the realm of structural state monitoring. They can effectively monitor various state parameters, including temperature, strain, and aerodynamic pressure. The advantages of FOS over traditional sensors are that they are lightweight, corrosion-resistant, and durable. FOS are passive in nature because of the dielectric construction and are capable of withstanding harsh environments and high temperatures. Also, these sensors can be integrated with the structure reliably for long-term monitoring and can also be monitored from a remote location [74, 75]. Expensive instruments such as interferometers are employed in FOS [72]. FOS usually contain multimode fibers and auto-compensate for temperature changes.

There are three technologies of FOS: point sensors such as Fiber Bragg Grating (FBG) sensors [76], long gauge sensor or quasi-distributed sensors such as multiplexed FBG sensors [77], and fully-distributed sensors such as Brillouin Optical Time Domain Reflectometry (BOTDR) and Brillouin Optical Time Domain Analysis (BOTDA) [78]. Out of these sensors, FBG sensors are commonly used for the practical monitoring of structures [75]. FBG is applicable for the strain measurement under static and dynamic environments [11]. Advanced FBG-based accelerometers have been introduced to overcome the drawbacks of conventional or wireless accelerometers [75].

FBG sensors can be used for the registration of Lamb waves. The grating pitch of the sensor is changed due to the propagating Lamb waves, which then generates the Bragg wavelength shift. Lamb waves could be reconstructed by monitoring the Bragg wavelength shifts. The structural information, such as the presence, location, size, and growth of damage, can be acquired by analyzing the received Lamb waves [79]. Various researchers have employed FBG sensors for damage detection and localization using Lamb waves in aircraft structures [80–83]. Moreover, recent progress in laboratory-oriented investigations has shown that FBG sensors are significantly mature for civil applications [79, 84]. However, FBG sensor technology still needs further development for efficient use in the aerospace market [79].

2.6.3 Microelectromechanical Systems (MEMS)

Miniaturization techniques are used in the construction of this type of sensor, combining several kinds of transducers [85]. These sensors are very beneficial in terms of maintenance and implementation costs [86]. These sensors are capable of integrating with the onboard computing, which makes these sensor self-diagnosing and self-calibrating. MEMS sensors have some additional attractive characteristics, such as their small size and easy connectivity with a wireless sensor network [20]. It is also desirable to find MEMS sensors that utilise inductive, optical effects, capacitive, or piezoelectric [87, 88]. In addition, actuators can also be included [89]. Generally, MEMS sensors are composed of a mixture of multiple varieties of sensors [90]. These sensors are adopted for measuring the magnitude of distinct variables i.e. angular velocity (gyroscopes), strain, displacement, and acceleration [8, 20]. This type of sensor provides high sensitivity, integration with communication systems, the determination of multiple variables, and responses at lower frequencies. In the last few years, MEMS sensors are increasingly being used for GW-based SHM and NDT due to the mentioned factors [91–93]. Currently in-use MEMS have some issues such as traceable calibration, mounting schemes, and long-term survival capabilities.

2.6.4 Scanning Laser Doppler Vibrometer (SLDV)

The Laser Doppler Vibrometer (LDV) works on the principles of the Doppler shift and is highly applicable for measuring out-of-plane surface velocities [41]. In-plane velocities can also be measured by the configuration of three laser beams [94]. LDV measures these velocities with high precision, including non-contact mode during GW propagation [41, 95]. LDV sensing systems have been widely employed for the sensing of GWs, and more specifically for Lamb waves [41, 95].

Generally, an LDV-based sensing system contains a laser head that drives a laser beam and then records the reflected beam from the vibrating surface. A demodulator is used for transform the information of the reflected beam into velocity measurements, and then a controller is used to deflect the optical mirrors for the purpose of precise adjustment of the laser beam. A simpler LDV sensing system contains a single head that can offer only a single-point measurement [41]. The Scanning LDV (SLDV) is capable of recording and measuring vibrations at various points on a predetermined grid. SLDV provides high resolution and sensitivity, making it more suitable for high-frequency analysis, rapid, and precise imaging of the wave response of a structure in a non-contact way [41]. SLDV enables measurement of full wavefield of a structure rather than single-point measurements typically obtained through conventional sensors. SLDV is useful for measuring the velocity of the inspected area at points associated with a predefined grid. SLDV-based measurements can be combined with adequate signal processing and imaging algorithms for the purpose of damage identification. High-resolution SLDV measurements provide very detailed visualization of different types of multiple defects [96].

A layout of mirrors is equipped with SLDV, which is helpful for changing the angle of measurement beams. The SLDV is also equipped with a camera which helps define a measurement grid precisely on the selected area of the object of interest. Furthermore, a screen of monitor is used for the visualization of the grid [97]. 3D vibrometers are used to provide information about the monitored object in three dimensions, which is very helpful in many cases. However, acquiring 3D measurements is more challenging than obtaining 1D measurements because 3D measurements require additional alignment of the three laser beams [97].

Recently, notable progress has been made in measurement techniques such as shearographic interferometry and SLDV. These techniques allow the analysis of the full wavefield of elastic waves. The advancement in these techniques has initiated new opportunities for various types of damage detection problems in structures [41].

SLDV has some limitations too, which are:

i. The inspected object surface has to be characterised by proper reflectivity; otherwise, the acquired signal-to-noise ratio is low.

ii. For a specified time, the measurements can only be performed at one point in space.

iii. Measurements need to be repeated many times for full field Lamb wave registration at a dense grid of points [97].

iv. High-resolution SLDV measurements take much time and require a large amount of hard drive space.

v. SLDV is very expensive.

2.7 GW-based SHM/NDT in Composite Structures

Composites are polymer matrices reinforced with man-made or natural fibers or any other engineering reinforcing material [11, 98]. In recent years, various industries, more specifically aerospace, are progressively employing composite structures to achieve the desired performance in a wide range of applications [11]. The use of composite structures is growing rapidly due to its simple structure, design flexibility, high specific strength, stiffness, lightweight, convenient integral molding, and many other desired mechanical properties [11, 45]. Incorporating composite materials has proven to be a highly efficient approach for reducing the weight of substantial structures, enhancing overall efficiency, and diminishing operational expenditures.

Maintenance is required for the composite materials because of less frequent but often abrupt and complex material failures in them. The state of a structure in which the structure fails to provide adequate output is termed as the failure of the composites [98]. This failure in composite structures depends on many quantitative and qualitative factors such as the structural stiffness of the composite, the strength of the structures, resistance to corrosion, resistance to impact, fatigue due to loading and unloading cycles, resistance to lightning and thunderstorms, and yield capacity of the composite [98]. Different forms of damage can occur in composite materials, such as corrosion, matrix cracking, debonding, delamination, and fiber breakage [3, 98–100]. Composite structures are very sensitive to impact loads. Even low-intensity and the low-velocity impact can commence matrix cracks in composites. Matrix cracks in composites then lead to delamination, which is one of the most common susceptible defects in composite structures. Delamination can grow and badly affect the structural integrity and mechanical properties of the composites, leading to very catastrophic failures if not detected at an early stage [43]. Therefore, the detection, localization, and quantification of delamination in composite structures are very crucial for the reliable and safe implementation of these structures in real-world applications.

In recent decades, many researchers have developed numerous intelligent and computational-based NDT/E and SHM techniques for composite structures [101]. Eddy currents (electromagnetic testing), acoustic emissions, optical methods, ultrasonic inspection, vibration analysis, thermography, radiography, and GWs are extensively used damage detection techniques in composite structures [98, 101]. GWs are broadly recognised as one of the most promising tools for significant identification of defects in composite structures [3, 52, 59]. GWs-based SHM and NDT techniques have been broadly applied for detecting numerous types of defects in composite structures, including debonding [31], delamination [29, 30], and impact damage [32–34].

GW-based detection, identification, and localization of damage for composite structures have extensively been studied in the literature. Yang et al. [102] evaluated the size, location, and shape of damage with Lamb waves by using PZT wafer transducers in composite materials. Hameed et al. [48] developed an efficient damage detection technique by introducing quantitative size estimation and transverse damage localization for composite materials with the use of Lamb waves. Li et al. [103] developed a technique based on the second harmonic Lamb waves propagation for assessing the thermal fatigue defects in composite materials. Zak et al. [104] used scanning laser vibrometry for the experimental measurements of defects detection in metal and thin-walled composite materials using Lamb waves. They applied the spectral finite element method for numerical investigations.

Fakih et al. [100] proposed a Lamb wave-based technique for the detection, localization, and assessment of the severity of barely visible indentation damage in composite structures. Toyama and Takatsubo [105] used an Acoustic Emission (AE) transducer and Angle Beam Transducer (ABT) for the detection as well as evaluation of the size of impact-induced delamination in composite laminates using S_0 mode of Lamb waves. Rauter et al. [106] developed an impact damage detection method for composite structures based on nonlinear Lamb waves. Park et al. [30] developed a complete non-contact ultrasonic wavefield imaging technique for automatic detection and visualization of hidden delamination and debonding in composite structures. Xu et al. [107] proposed a weighted sparse reconstruction-based anomaly imaging method for damage detection in composite plates based on Lamb waves by utilizing 3D laser vibrometry.

2.8 Conclusions

In this chapter, SHM and NDT techniques, major methods of damage detection, and the main levels of SHM along with the functionalities of every level are elaborated. Firstly, the damage is detected, then the location of the damage is identified. After that, the size and type of damage are estimated, and in the end, the remaining life of the structure is evaluated. Major components of an SHM technique and various types of SHM techniques are described. The use of GWs, and Lamb waves in SHM and NDT are also explained. Lamb waves are usually preferred over vibrations and EMI for implementing SHM and NDT techniques due to their longer propagation distances and suitability for distinguishing smaller defects. Various kinds of sensors applied for implementing GW-based SHM and NDT techniques are explained. PZT-based transducers are commonly used for the generation of Lamb waves in GW-based SHM and NDT. However, SLDV is mostly employed for the analysis of the full wavefield in GWs-based NDT techniques. Furthermore, various types of GW-based damage detection, localization, and assessment techniques implemented in composite materials are described.

There is a knowledge gap in accurate estimation of damage size because of the multimodal complex character of Lamb waves. Therefore, DL methods can serve as an alternative to conventional signal processing methods.

Chapter 3

Artificial Neural Networks for GW-based SHM/NDT

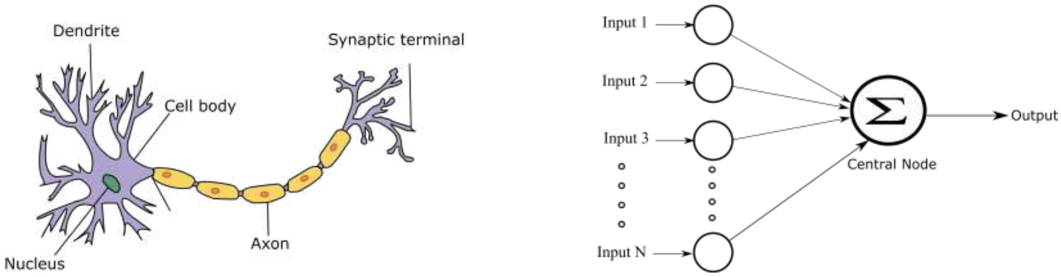
3.1 Artificial Neural Networks

Artificial Neural Networks (ANNs) are simply referred to as neural networks. ANNs are computational models inspired by the nervous system of animals and human beings, attempting to mimic their behaviour [108, 109]. The mammalian nervous system is composed of cells referred to as neurons. These neurons in the mammalian nervous system are interconnected using dendrites and axons, and the regions connecting these dendrites and axons are known as synapses. The strengths of the connections in these synapses often change in response to external stimuli. This change in synaptic connections represents how learning is performed in living organisms [110].

This biological mechanism is simulated in ANNs, which are composed of computational units are known as artificial neurons. The neurons in ANNs are interconnected through weights, which perform the same role as the strength of synaptic connection in biological organisms [110]. These artificial neurons in ANNs are a set of computational processing units responsible for acquiring and maintaining knowledge (information-based) and generating desired outputs [109–111]. These neurons communicate with each other by sending information in the form of activation signals along directed connections. Multiple artificial neurons in ANNs are connected and form a network. A simple structure of a biological and an artificial neuron is illustrated in Fig. 3.1.

Generally, ANNs have two main components: nodes and layers. Nodes are responsible for specific operations, and layers are composed of groups of nodes operating together at a specific depth within a neural network.

A typical neural network is composed of an input layer, an output layer connected by one or more hidden layers. Each successive layer accepts the output from the previous



(a) A biological neuron

(b) An artificial neuron

Figure 3.1: Graphical illustration of a biological and artificial neuron.

layer as input. These three layers are described below:

i. **Input layer:** The input layer is responsible for receiving data (information), features, signals, or measurements from the external environment.

ii. **Hidden layer(s):** Hidden layers are also known as intermediate or invisible layers. These layers are sandwiched between the input and the output layers, therefore, they are invisible from the outside. Hidden layers extract features associated with the process or system being analysed. The ANNs can include any desired number of hidden layers, each comprising the necessary number of neurons or units. The initial hidden layers handle simpler patterns, while the later layers detect more complex ones. Most of the internal processing in a network occurs in these layers.

iii. **Output layer:** This is the last layer of a neural network and is responsible for producing and presenting the final output of the network, acquired from processing in the hidden layers. The number of output nodes is usually equal to the number of classes in the target class [109].

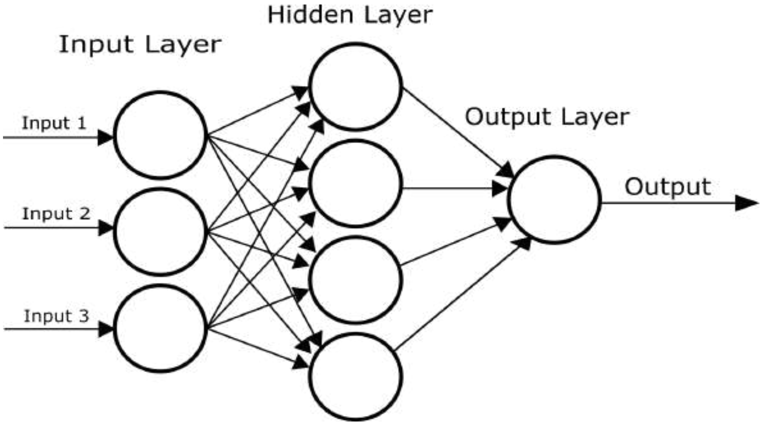


Figure 3.2: A simple three-layered neural network.

A simple three-layered neural network, composed of an input layer, a hidden layer, and an output layer, is shown in Fig. 3.2. When there is more than one hidden layer in a neural network, the neural network is then referred to as a deep neural network (deep learning).

Based on the information propagation, a neural network can be grouped into two generic categories [109, 112, 113]:

- **Feed-forward networks:** In these types of networks, information flows only in one direction. The connection between the nodes has no cycles or loops in the graph when these networks are considered as a graph with neurons as its nodes. These network architectures can also be referred to as Directed Acyclic Graphs (DAG). Examples of these types of networks are Multilayer Perceptrons (MLPs) and Convolutional Neural Networks (CNNs).
- **Feed-back networks:** These networks have connections that form directed loops or cycles. These architectures can operate and generate sequences of arbitrary sizes, possess memorization ability, and can store information and sequence relationships in their internal memory. Examples of these types of networks are Recurrent Neural Networks (RNNs) and Long-Short Term Memory (LSTM) networks.

3.2 A Brief History of ANNs

McCulloch and Pitts [114] proposed ANNs for the first time in 1943 as electric circuits, and a learning algorithm was proposed by Donald Hebb in 1949 [115]. In 1952, Hodgkin and Huxley [116] incorporated neural phenomena such as action potential propagation and neuronal firing into a set of evolution equations capable of yielding quantitatively accurate spikes and thresholds.

Between 1957 and 1958, Rosenblatt and Wightman [117] developed the first neurocomputer named Mark I perceptron, establishing the basic model of the perceptron. Their model was demonstrated by recognizing different simple characters. Widrwo and Hoff [118] developed the Adaline (adaptive linear element) model in 1960, capable of recognizing simple patterns. They trained Adaline using the least mean squares (LMS) method.

In 1969, Minsky and Papert [119] mathematically proved that the Perceptron cannot handle complex logical function, which significantly reduced the interest of many researchers in neural networks. During that period, the Adaline model and its multilayer version known as Madaline was successfully applied in many tasks but struggled with solving linearly inseparable problems due to the use of linear activation functions.

In 1980, Fukushima proposed a multilayered neural network known as NeoCognitron [120]. The architecture of NeoCognitron was inspired by the seminal studies of the mammalian visual system by Hubel and Wiesel [121]. NeoCognitron consists of a sequence of two-dimensional layers of neurons, with each neuron connected to nearby neurons from the previous layers. In NeoCognitron, the weights for the neurons are not computed in a fully automatic way.

In the late 1980s, LeCun et al. [122–124] developed the first Convolutional Neural Networks (CNNs) take inspiration from the work of Fukushima. CNN included the concept of feature maps produced by the layers of the neurons with local and heavily restricted connections, which is similar to the receptive fields of biological neurons, and whose weights are adapted to the use of backpropagation algorithm [125].

In 1988, George Cybenko made important contributions by proving the universal functional approximation ability of neural networks. In 1989, Hornik, Funahashi, and Stinchcombe presented their work, demonstrating the multilayer perceptron network as a universal approximator.

Subsequently, ANNs have widely been used in various engineering and science- related areas. Recently, ANNs have been extended from a simple pattern recognition problems to more complex tasks, including DNA and gene recognition, object detection, face detection, classification, and many more. The applications of ANNs have been extended from engineering and physical sciences to economics, finance, and social sciences [126].

3.3 Artificial Intelligence, Machine Learning and Deep Learning

Artificial Intelligence (AI), Machine Learning (ML), and Deep Learning (DL) are often used interchangeably but these are not exactly the same terminologies. The diagram in Fig. 3.3 represents the relationship between AI, ML, and DL.

There are many similarities and differences among these techniques, which are elaborated below.

3.3.1 Artificial Intelligence

Artificial means man-made, which is not natural and intelligence is the ability of acquiring and applying knowledge and skills. As the name indicates that AI is the field

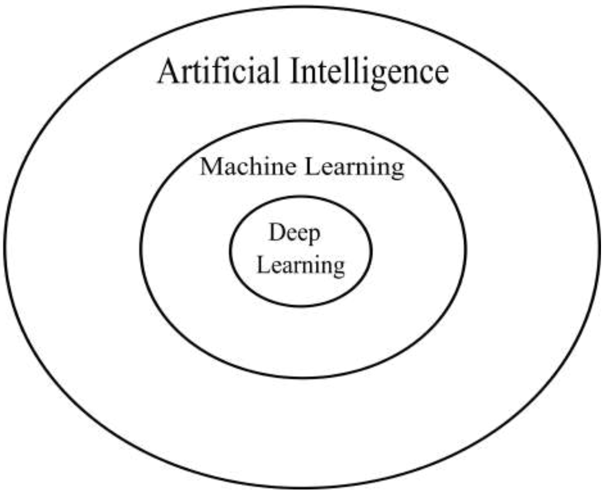


Figure 3.3: Relationship between AI, ML and DL

of science that makes the machine able to mimic or replicate the behaviour of human beings. AI is concerned with building smart machines, which are capable of performing tasks that usually need human intelligence [127, 128]. AI techniques are frequently being applied for the tasks of developing systems endowed with intellectual characteristics of humans, such as the ability to reason, generalize, problem-solving, discover meaning, learning from past experiences, and many more.

3.3.2 Machine Learning

ML is a subset of AI, which makes the computer able to learn from data, without being explicitly programmed for a task. ML is concerned with constructing computer programs that are automatically improving with experience [129, 130]. The real power of ML comes from making future predictions based on the received data in the form of observations of real-world events. ML algorithms are capable of learning patterns from the input data and these learned patterns are then used for making informed predictions in the future. Every ML technique can be an AI technique but every AI technique may not be an ML technique, non-ML techniques such as rule-based systems and alpha-beta pruning are also widely used in AI. Some examples of ML algorithms are linear regression [129, 130], logistic regression [129, 130], decision trees [130, 131], random forest [130, 132, 133], support vector machines [130, 134], and boosting algorithms [135, 136].

Traditional ML algorithms are useful in many situations, however, they are largely dependent on the quality of features for getting superior performance. The creation of features is also a time-consuming task and needs a lot of domain expertise. Furthermore, with the increasing complexity of the problems, more specifically with the advent of unstructured data such as voice, text, images, and so on, it can be almost impossible to create features for such tasks that represent the complex functions. Therefore, there is often a need to find a different approach for solving such complex problems; that is where end-to-end ML approaches come into play.

3.3.3 Deep Learning

Deep Learning is a sub-field of ML, DL is just an extension of traditional ANNs. DL is an end-to-end ML architecture that could be applied directly to the data. The main difference between DL networks and ANNs is the depth and complexity of the network. Traditional ANNs have only one hidden layer, while DL networks have more than that [137–139]. In DL, neural networks may consist of thousands of interconnected neurons (nodes), mostly arranged in multiple layers, where one node is connected to many nodes in the previous layer from where it accepts its input data, as well as being connected to neural nodes in the following layer, to which it sends the output data once it has been processed [4, 137]. One defining characteristic of DL models is the ability to learn features automatically from the input data. Unlike traditional ML, where there is a need to create features manually, DL excels in learning different hierarchies of features across multiple layers. DL can solve more complex problems by modeling complex patterns than traditional ANNs. Therefore, DL is more widely used nowadays in computer vision and natural language processing applications such as object detection, image recognition, face detection, chatbots, and text generation [4, 137, 140].

DL techniques have made great progress in the past decade. There are many factors that led to this significant rise of DL techniques such as the availability of large quantities of data, improved accuracy, scaling effectively with the data, and more powerful hardware. However, while comparing to the traditional ML techniques, DL needs more training data, more computational power, and more time to train. Moreover, DL methods are also difficult to interpret.

DL is a field of undergoing intense research activities. Researchers are devoted to inventing new neural network architectures that either increase the performance of the previously implemented architectures or tackle new problems. Some of the popular DL techniques are Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Generative Adversarial Networks (GANs).

3.4 Bayesian Neural Networks

Bayesian neural networks (BNNs) represent a fascinating and innovative approach to DL and neural network modeling. What sets them apart from traditional neural networks is their ability to capture uncertainty and provide probabilistic outputs. Unlike conventional neural networks that provide point estimates, BNNs incorporate Bayesian principles to model weight parameters as probability distributions. This enables them to express not only what they have learned but also their level of uncertainty about their predictions. This aspect makes BNNs particularly valuable in scenarios where uncertainty assessment is crucial, such as in medical diagnoses, autonomous driving, or financial predictions. By propagating uncertainty throughout the network, BNNs offer more robust and reliable predictions, making them a powerful tool in machine learning and artificial intelligence. While they may require more computational resources for training and inference, their capacity to handle uncertainty and provide richer insights into data makes them an area of growing interest and research within the AI community [141].

3.5 Types of Learning

ML or neural networks algorithms are composed of four major types of learning, which are explained one by one in the subsequent sections.

3.5.1 Supervised Learning

Supervised learning refers to the type of learning where an algorithm is trained to learn patterns based on prior knowledge. This is one of the simplest form of learning. The data in this type of learning consist of targets or labels, where the algorithm seeks patterns that explain the relationship between the features and the target. Trained models can then predict labels for new data [129, 130, 137, 138]. In supervised learning, the algorithms iterate during the learning process to minimize the difference between the predicted value and the target value [137].

Supervised learning is further composed of two types: classification and regression [129, 130]. Classification accepts data (observations) and labels them with a certain category or class. For example, fraud detection, where the category is either fraud or legitimate, classifying email as spam or non-spam, and classifying voters as Democrat, Republican, and Independent. On the other hand, regression takes data observations and outputs a real number value. Examples of regression tasks include predicting the sale price of a house based on different factors such as region and national unemployment rate and predicting the number of points a basketball team will score in a match.

Some of the popular supervised learning techniques are K-nearest neighbors, decision trees, random forest, support vector machines (SVM), linear regression, logistic regressions, and ANNs (ANNs can also be used in unsupervised learning).

3.5.2 Unsupervised Learning

In contrast to supervised learning, unsupervised learning consists of training a model with unlabeled data, which means that there are no target values available. In this type of learning, the algorithm automatically learns patterns from the input data by itself [129, 130, 138]. The purpose of this type of learning is to come up with a better understanding of the input data. These types of algorithms generally detect similarities or differences between records and variables and then try to group them [129, 130, 138].

Two main types of unsupervised learning are dimensionality reduction and clustering [129, 130]. In dimensionality reduction, the algorithms try to reduce the number of variables or features associated with a sample of data. Dimensionality reduction is very useful in preprocessing, feature engineering, noise reduction, and generating plausible artificial datasets. Whereas, clustering is used for grouping records and is very popular for performing customer segmentation tasks, where algorithms group those customers together who have similar shopping behaviour.

3.5.3 Semi-supervised Learning

Semi-supervised learning is in-between supervised and unsupervised learning. This type of learning is employed in situations when only some of the labeled data is available. The main goal of this type of learning is to use a large collection of unlabeled data with a few labeled samples for improving the performance of the generalization [142]. An example of semi-supervised learning is a photo archive application where only some of the images are labeled, and many of the images are not labeled. Semi-supervised learning has wider use in bioinformatics, text and web mining, database marketing, and video indexing where only a portion of input data is labeled [142].

3.5.4 Reinforcement Learning

In reinforcement learning, the algorithm learns how to act in a specific environment on the basis of feedback it receives. In this type of learning, the algorithm learns directly from the input data as it comes in, rather than being trained over static data as in the case of supervised learning. However, decisions are not being made on the basis of immediate rewards but are taken on the accumulation of awards in the entire learning process [110, 143, 144].

All of the above four types of learning techniques are simplified in Fig. 3.4.

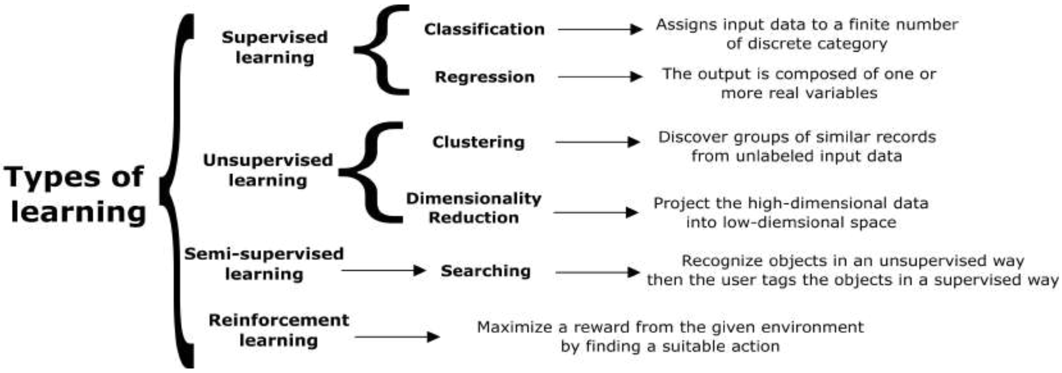

Types of learning
Supervised learning
Classification
Assigns input data to a finite number of discrete category
Regression
The output is composed of one or more real variables
Unsupervised learning
Clustering
Discover groups of similar records from unlabeled input data
Dimensionality Reduction
Project the high-dimensional data into low-diemsional space
Semi-supervised learning
Searching
Recognize objects in an unsupervised way then the user tags the objects in a supervised way
Reinforcement learning
Maximize a reward from the given environment by finding a suitable action

Figure 3.4: Types of learning techniques.

3.6 Learning Process of Neural Networks

A neural network accepts inputs, X_1, X_2, X_3, ..., X_n, a specific weight W_1, W_2, W_3, ..., W_n is multiplied to each element of the input. These weighted inputs are then summed together along with a bias term, and then the summation is passed through a function f to generate the output $y = f(Z)$, as illustrated in Fig. 3.5. The result can then be transmitted to other neurons [143].

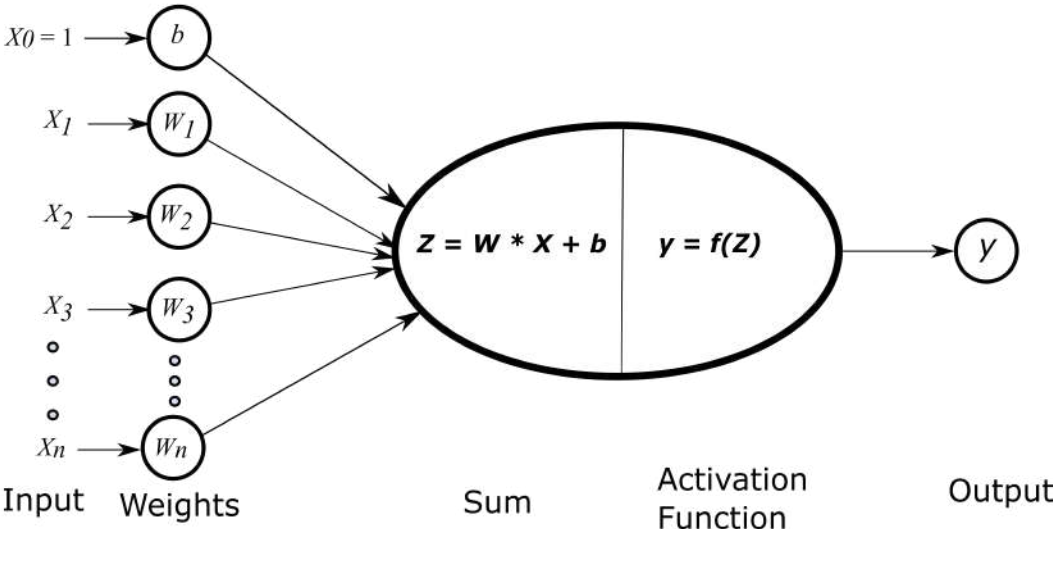


Figure 3.5: Learning mechanism of ANN.

In a more detailed mathematical form, given X_n, real parameters $W_i \in \mathbb{R}$, where $i =$

1, 2, 3, ..., X_n, and a bias term ($b \in \mathbb{R}$), the calculation of the first step is:

$$Z = W_1X_1 + W_2X_2 + ... + W_nX_n + b, \tag{3.1}$$

now, the next step is to apply a function f to Z, predicting the output y.

$$y = f(Z) = f(W_1X_1 + W_2X_2 + ... + W_nX_n + b), \tag{3.2}$$

weights are real numbers. The weights on a connection from one neuron into another neuron indicate how strongly one neuron affects the other one. It also indicates whether the effect is positive or negative, i.e., a direct or inverse relationship. In the initial iteration, the values of weights are randomly initialized within the range of 0 to 1. After the first iteration is executed, these values get updated, allowing the iterative process to start again. The bias term assists the neural network in producing better results by allowing each layer to model a true linear combination of the input values. The value of the bias is a constant and is also referred to as the intercept.

In neural networks, inside each node or neuron, there is an activation function, typically a non-linear function, which helps to make the model non-linear. If this non-linear activation function is removed from a neural network, then the several hidden layers will collapse due to a series of matrix multiplications, resulting in an extremely simple linear model. The activation function is employed for transforming an unbounded input into an output that has a nice, predictable form. Various types of activation functions are used in neural networks, as discussed in section 3.6.2.

The values of weights and biases in a neural network can be optimized through the training process to produce highly accurate predictions based on the input data by adjusting for error feedback. Neural networks typically require many runs (or epochs) to tune the weights and biases of nodes. To adjust the optimal weights and biases of a neural network model, a function is used to measure error. This function is called the loss function, also known as the cost function, and it measures the difference between the network-predicted outputs and the real outputs in the dataset. The loss function is computed in each iteration to assess the progress of the neural network model throughout the iterative procedure. The primary objective is to determine the weight and bias values that minimize this cost function.

3.6.1 Training a Neural Network

For training a neural network model, the input dataset is commonly split into three other datasets: training, validation, and test. Both the train and validation sets are used for training the network. The network uses the training set as input, while the validation set is employed by the loss function to compare the output of the neural network to the real data. The test set is used at the final stage once the network has been trained. This set evaluates how the network can perform on the data which the network has not seen during the training process. Training a neural network involves setting up a loss function, which indicates how accurate the output of the neural network model is. The main aim of a neural network model is to predict output that minimizes the loss function.

While building and training neural network models, two important issues can arise: underfitting and overfitting [138]. Underfitting and overfitting are similar to the concepts of variance and bias for a model. Generally, if a model is not capable enough to learn the patterns and relations in a dataset, there will be a high training error, and such a model is referred to as a model with high bias. This situation is referred to as underfitting. On the other hand, if a neural network model is too flexible for a given dataset and able to learn patterns, relations, and even noise in the training data, it may cause a large increase in the test error compared to the training error. The larger the gap between the test error and the training error, the more it indicates that the model has high variance. This situation is called overfitting.

Neither high variance nor high bias is desirable in practical scenarios. The final objective of neural networks is to find a model with the lowest possible values of variance and bias at the same time. Underfitting is a less problematic issue than overfitting, and it can be solved by making the network larger or more complex. This means altering the architecture of the network, increasing the number of units in the layers, or adding more layers to the network. The overfitting problem can be fixed by making the network smaller or providing more training data to it.

There are many other methods for preventing the problems of underfitting and overfitting in practice, some of which are explained in section 3.6.2.

3.6.2 Parameters and Hyperparameters of Neural Networks

Parameters are properties that affect how a model makes predictions from a particular dataset. Parameters can be learned from the data and modified dynamically. Hyperparameters show that how a model learns from the data. Hyperparameters are higher-level properties, not typically learned from the data and are defined before the beginning of the training process. Hyperparameters can be fine-tuned to improve the accuracy of a neural network model. An essential part of building a neural network is the process of fine-tuning the hyperparameters by playing around with the hyperparameters and performing error analysis to address issues affecting the network.

To achieve accurate predictions from a neural network, it needs tweaking a lot of hyperparameters such as the number of hidden layers, the number of neurons in each layer, the number of epochs, the learning rate, the batch size, activation functions, regularization strategies, and optimizers [145, 146]. Each of these hyperparameters is briefly explained below:

The number of hidden layers: Neural networks with a single hidden layer can also perform well on many problems. However, complex problems may require more layers. Extra hidden layers enable the model to learn more complex decision boundaries. To determine whether adding new hidden layers is beneficial, understanding their role in a neural network is crucial. Each hidden layer form a model representation of its input data, with earlier layers creating lower-level representations and later layers creating higher-level representations.

The number of neurons in each layer: The number of neurons or nodes in a layer of a neural network relates to the structure of both the input and output data.

There is no clear way to know in advance the right number of layers or the right number of neurons per layer. This has to be figured out by the trial and error approach. Increasing the number of layers and neurons per layer extends training time, requires more computational power, and necessitates additional hyperparameter tuning.

Epochs: Epochs represent the number of times a neural network adjusts its weights in response to the data passing through and its resulting loss function. A single epoch constitutes one training iteration in which the neural network processes all training instances. More epochs allow a network to learn more from the data, but there is a risk of overfitting, particularly with larger datasets.

The learning rate: The learning rate determines the size of the steps the model takes to reach the local or global minima in each iteration. Lower learning rates result in slower learning but may yield better models, while higher learning rates speed up learning but can lead to convergence issues.

The batch size: Batch size specifies the number of instances fed to the neural network during a single iteration, which will be used to perform a backward or forward pass through a network. A new set of instances is used for the next iteration. Batch size also helps in improving the ability of the model, and generalize well to the training data, because in every iteration, it is fed with new combinations of instances, which is very useful when dealing with an overfitted model.

There is no specific rule to find the right values for the number of epochs, the learning rate, and the batch size in advance, these values have to be figured out through trial and error.

The activation functions: Activation functions determine which individual neurons are activated in a neural network. The activation functions also add non-linearity to neural network models, enabling them to learn complex patterns [138].

The most common activation functions include sigmoid, softmax, tanh (the hyperbolic tangent function), and ReLU (Rectified Linear Unit). Each activation function has its uses in neural networks, choosing an activation function for a neural network depends on the problem. However, the ReLU activation function has been proved to outperform other activation functions in most situations.

i. **Sigmoid:** The sigmoid function is also known as the logistic function. This function converts the output of the model into a probability. The output values of a sigmoid function ranges from 0 to 1. This function is commonly employed at the final layer of an ANN for binary classification problems. The mathematical formula of the sigmoid function is as follows:

$$f(x) = \sigma(x) = \frac{1}{1+e^{-x}} \tag{3.3}$$

ii. **Softmax:** The softmax function shrinks the values of a list to be between 0 and 1, ultimately makes the sum of all the elements to 1. This function is also commonly used at the last layer of a neural network but for multi-class classification problems, as it can generate the probabilities for each of the different output classes. However, it can also be applied for binary classification problems. While using the softmax function, the class with the highest probability is chosen for the final prediction. The mathematical formula of the softmax function is as follows:

$$f(x) = \frac{\mathrm{e}^{\mathrm{x}_\mathrm{i}}}{\sum\limits_{\mathrm{j}=1}^{\mathrm{K}} e^{x_j}}, \tag{3.4}$$

where K is the number of classes in a multi-class classifier.

iii. **Tanh:** The tanh activation function is quite similar to the sigmoid function and it is usually used in the hidden layers of a neural network. Its values range between -1 and 1. The formula for tanh function is:

$$f(x) = \tanh(x) = \frac{\mathrm{e}^{\mathrm{x}} - \mathrm{e}^{-\mathrm{x}}}{\mathrm{e}^{\mathrm{x}} + e^{-x}}, \tag{3.5}$$

tanh functions are computationally expensive, therefore, ReLUs are often used instead [147].

iv. **ReLU:** ReLU is currently the most widely used activation function for hidden layers. The equation for ReLU is very simple:

$$ReLU(x) = \max(0, x), \tag{3.6}$$

the ReLU function seems pretty linear, it is just *f(x)* $=$ *0* for $x <$ *0* and *f(x)* $= x$ for $x \geq$ *0.* Furthermore, its mapping functionality of all negative values to 0 helps in avoiding overfitting and it also assists in training the model faster. Recently, different variants of ReLU functions have been proposed, such as leaky ReLU [148] and Parametric ReLU (PReLU) [149].

Regularization: Overfitting occurs when a model relies too heavily on certain features,performing well on training data but failing to generalize to unseen or test dataset. In order to avoid this problem in neural networks, regularization is used. Regularization techniques attempt to modify the learning algorithm in a way that reduces the variance of the model. Regularization methods intend to reduce the generalization error while not increasing the training error. These techniques provide some kind of restriction that assists with the stability of the model [138, 150].

There are different types of regularization techniques, among them $L1$ and $L2$ regularization, dropout, early stopping, data augmentation, and adding noise are the most commonly used. All of the mentioned types of regularization techniques are briefly described below:

i. **$L1$ and $L2$ regularization:** Both of these techniques add a regularization term to the loss function as a way of penalizing high weights that may affect the performance of the model. This type of regularization is also called weight regularization. $L1$ is generally used for feature extraction tasks as it creates sparse models, while $L2$ has proven to work better for regular data problems. The main difference between these two techniques is that $L1$ uses the absolute values of the magnitude of the weights, while $L2$ uses the squared magnitude of the weights [138].

ii. **Dropout:** Another common regularization technique is a dropout, which is used for preventing overfitting in ANNs [151]. It randomly selects nodes at each iteration during the training process and removes them along with their connections. The dropout is very useful in minimizing co-adaptation, which is when multiple neurons in a layer extract the same or very similar features from the input data. Co-adaptation among neurons can arise when the connection weights for different neurons are nearly identical. Co-adaptation commonly occurs in fully connected layers which has a larger number of neurons. Co-adaptation can lead to redundant neurons computing and overfitting.

iii. **Early stopping:** Early stopping technique is to stop training the neural network once the error for the validation set starts increasing while the training set error is decreasing. This technique can easily be added to a neural network without significantly altering the training procedure and it is extensively being used in practice [112, 113, 138].

iv. **Data augmentation:** Data augmentation is a regularization technique, it tries to prevent overfitting by training the model on more training data. In this technique, the available data is transformed in different ways and then fed into the neural networks as new training data [138]. Data augmentation has proved to be very effective in many domains especially for object detection and object recognition in computer vision and speech processing.

v. **Adding noise:** The underlying idea behind this technique is the same as that of data augmentation. In order to minimize overfitting, new data examples are generated by injecting some noise into the available data [152].

Optimizers: The process of altering the values of the parameters in order to minimize the loss function and to get the optimal values of the parameters is called optimization. Different types of optimization algorithms also called optimizers are being used in ANNs. The most common one is Gradient Descent (GD), which tries to find some local or global minimum of a cost function [153]. GD does so by determining the direction in which the model should move to minimize the error. It updates the weights of a model based on a gradient of the loss function until it reaches the minimum loss. GD utilizes backpropagation [154] also called backward propagation for finding the optimal gradient for a neural network model.

Backpropagation involves calculating the gradient of the cost function with respect to the biases and weights in each layer along with the network. As the biases and weights are not directly contained in the loss function, therefore, the chain rule of differential calculus is leveraged for propagating the error from the loss function in the reverse direction until it reaches the input layer of the network. Next, a weighted average of the derivatives is calculated, which is used for updating the values of biases and weights before running a new iteration. Backpropagation is used for determining the direction in which the biases and weights should be updated for the purpose of reducing the error until it reaches a minimum point. It should be noted that backpropagation does not always find the global minima, because it stops updating after reaching the lowest point in a slope, regardless of any other regions.

The size of the gradient in GD depends on the learning rate, which defines how quickly a neural network will update its weights. Smaller learning rates are more likely to reach the minimum, but it may take longer. On the other hand, a model with a larger learning rate will reach the minimum loss quicker, but could also overshoot the minimum.

An improved version of GD is Stochastic Gradient Descent (SGD), which basically follows the same process as of GD, the only change is that SGD takes the input in random batches instead of in one chunk. This functionality of SGD improves the training times while reaching outstanding performance and also reduces the use of heavy computational resources.

Some of the other popular optimizers used in neural networks are the Levenberg-Marquardt algorithm, Adam, momentum, Nesterov accelerated gradient, Adagrad, Adadelta, and RMSprop.

3.7 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are said to mimic the visual cortex of mammalian [138, 155]. CNNs are the most widely used type of neural networks for computer vision and image processing tasks [123, 156–159]. CNNs have also been applied in natural language processing and speech recognition tasks. Similar to ANNs, CNNs are composed of neurons that accept inputs, which are processed using weighted sums and activation functions. However, in contrast to ANNs, which accept vectors as inputs, CNNs typically employ images as their input. CNNs include the feature extraction within its training process, and the weights are also determined through the training process. A typical CNN accepts image(s) as input, and passes through a series of **convolutional (feature extraction) layers**, **pooling (dimensionality reduction) layers**, and **fully connected (FC) layers** [113, 160, 161]. These three main components of CNNs are elaborated in the following subsections.

3.7.1 Convolutional Layers

This is the first step in extracting features from an image. A convolutional layer in CNNs is employed for detecting patterns in an image (represented as a matrix of pixels) using filter(s), also known as kernels [113, 138, 147]. Filters play a significant in image recognition; they are used to transform inputs and extract features that enable CNNs to recognize specific images. A filter is just a matrix, applied to a subsection of the input image through a convolutional operation, and the output of this operation is stored in another image known as feature map or activation map, which ultimately serves as input for the next layer. The feature map stores the highlighted patterns found by the filter. These filters are similar to the weights in ANNs; their values are not set in advance but are learned by CNNs automatically [113, 147, 160, 161]. An element-wise multiplication is performed on the selected region (the same shape as the filter) from an input data and a filter of size $n * n$. The element-wise product is just the multiplication of an element from the selected region from the input image to the corresponding value from the filter. Afterward, these values are summed together and the summation of all these values is inserted into the feature map, as shown in Fig. 3.6.

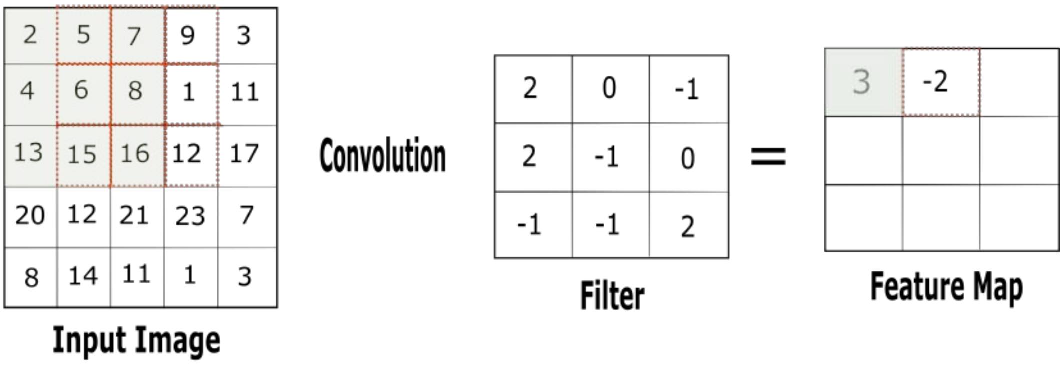


Figure 3.6: Convolution operation between the image and filter.

In this example, a filter of size $3 * 3$ is applied to the top-left region of the image for the first iteration. For the next iterations, the same operation is applied by sliding the filter to the right by one column (horizontally) or one row (vertically) in a downwards direction from the input image. This sliding operation is performed step-by-step until the entire image is covered; the parameter defining the length of this sliding operation is known as the stride [113, 160, 161]. The size of stride is the horizontal or vertical offset of a filter matrix as it moves along the input data. In the above case, stride size is set to 1, but it can also be set to 2 or 3 or any other appropriate value. With a larger stride size, there will be fewer overlapping pixels, but the resulting feature maps will have smaller dimensions, potentially causing the CNNs to lose some information. In addition to the above process, a convolution also uses a trainable bias term. The bias term is added to the output of each resultant matrix from element-wise multiplication.

In a convolution operation, the number of element-wise product matrix depends on the dimension of the input data, kernel matrix, and the size of the stride. The resultant feature map from the convolution operation has smaller dimensions than the input image. In the above example of a convolution operation, if a filter of size 2 x 2 and the stride size of 2 is applied, the last row and the last column of the input data matrix would not have been used in the convolution. In such situations, a technique known as padding is used [160, 161]. Padding consists of adding a layer of pixels with a value of zeros to the edges. As 0 multiplied by any number results in 0, padding does not affect the matrix dot products. Padding is crucial for removing distortions and controlling the size of the feature maps.

In an actual CNN, multiple filters are used, and several feature maps are generated. As the network grows in terms of the number of layers, the filters perform more complex operations that make use of previously detected features. Once the input has been passed through all the filters, the output is fed into an activation function. ReLU is mostly used in CNNs, as it has shown outstanding performance. The output generated from the convolutional layer becomes the input of the subsequent layer, which is mostly a pooling layer [160, 161].

3.7.2 Pooling Layers

Typically, pooling layers are the final step in the feature selection process. Pooling layers are used for dimensionality reduction and thresholding, and they are also referred to as downsampling [113, 160, 161]. The convolutional layer finds numerous features in an image, many of which may not be necessary. Pooling layers extract the most relevant features while disregarding less common features found only in a few examples, as well as potential distortions, thereby preventing overfitting. Pooling layers mostly reduce the height and width of the input data by half, resulting in fewer computations and faster neural networks training [160, 161]. The depth or number of channels in the input remains unchanged since pooling layers apply the same operation to all channels of the image. Pooling layers not only reduce the output from the convolutional layers but also effectively eliminate any noise present in the extracted features, leading to improved model accuracy.

Similar to a convolution operation, a filter matrix is used in pooling. However, the pooling filter does not contain any weights and also does not perform any matrix element-wise matrix multiplication. It only applies a reduction operation to subsections of the input data. Furthermore, the same concept of strides and padding as in convolution layers are also used here.

Pooling layers select the relevant features from subsections of the image by either taking the maximum value or averaging the values in the subsection of the image, which are termed as max pooling and average pooling, respectively. Max pooling is commonly used in CNNs as compared to average pooling [162]. The reason for wider usage of max pooling is due to preserving the most relevant features, while average pooling has proven to work better in smoothing related tasks in images. An example of max pooling is shown in Fig. 3.7.

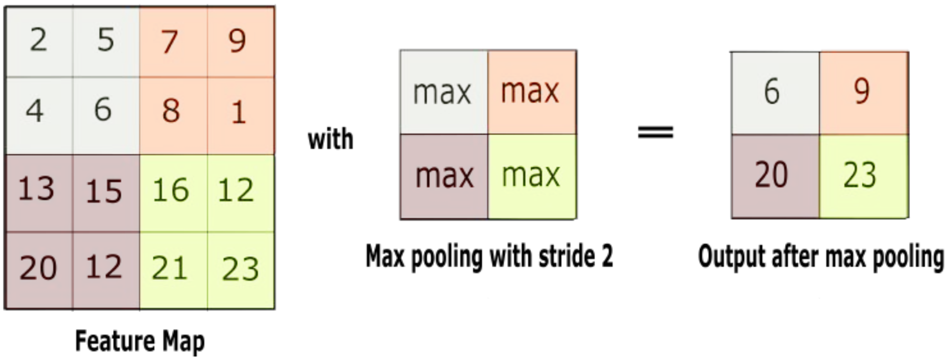


Figure 3.7: Max pooling with a 2 x 2 filter and stride size of 2.

The depth of the CNN is adjusted as needed by repeating convolution-pooling sequences, extracting high-dimensional features from input data, and employing fully-connected layers for classification.

3.7.3 Fully Connected Layers

The fully connected (FC) layers are defined at the end of the CNNs after the input has passed through a set of convolutional and pooling layers [113, 160, 161]. The data resulting from the layer preceding the first FC layer is flattened into a vector, which is then fed into the FC layer. The primary purpose of the FC layers is to consider all the features detected by the previous layers and obtain the expected output form [160, 161].

In FC layers, a relevant activation function (usually a softmax function) is applied for the final prediction. Similar to ANNs, classification is performed by calculating the probability of the image belonging to each class label. The class label with the highest probability is selected as the final prediction for that image. The complete architecture of a CNN is illustrated in Fig. 3.8.
CNNs are primarily trained in a supervised manner, commonly employing the stochastic gradient descent technique, also known as the backpropagation (BP) algorithm. In each BP iteration, the gradient magnitude (or sensitivity) of every network parameter, including weights in convolutional and fully-connected layers, is calculated. These pa-

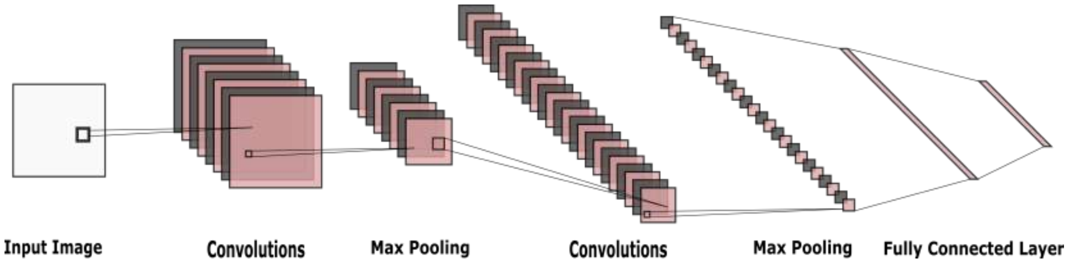


Figure 3.8: The overall architecture of a CNN.

rameter sensitivities are then used to incrementally adjust the CNN parameters until a certain stopping criterion is met.

Many researchers have conducted exploratory work and proposed various CNN architectures. Currently, the most famouse ones include Resnet [140], AlexNet [157], VGG [163], and Inception [164].

3.8 Recurrent Neural Networks

Typically, the input data fed into the neural networks have a fixed length. Due to this reason, most neural networks have a feed-forward structure, which means they utilize multiple layers of fixed sizes to compute the output. However, sometimes, a sequence of inputs with varying sizes, such as in the case of text data, needs to be handled by the neural networks. In such situations, Recurrent Neural Networks (RNNs) are used, specifically developed to work with sequential data of varying lengths [110, 137, 138]. Recently, RNNs have become very popular in various applications, including text translation, speech recognition, chatbots, stock market forecasting, and many more.

RNNs process inputs (X) and generate outputs (z), with the output being influenced by both the current input and the entire past input history, often referred to as the model's internal state or memory. This history comprises ordered and interconnected data sequences like time series data. RNNs contain loops, allowing information to remain in memory for longer periods, even when a subsequent set of input is being processed, which can be useful for analysing the next bit of information [110, 137, 138, 165]. This memory-keeping ability makes RNNs very accurate in predicting what is coming next. The main component of RNNs is their cell. A basic representation of an RNN process for a single input is shown in Fig. 3.9.

The rolled form of RNN represents the true depiction of the RNNs network. It is composed of a single cell (although a multi-layer RNN will have multiple stacked cells) and three types of connections: input, recurrent, and output. The unrolled diagram

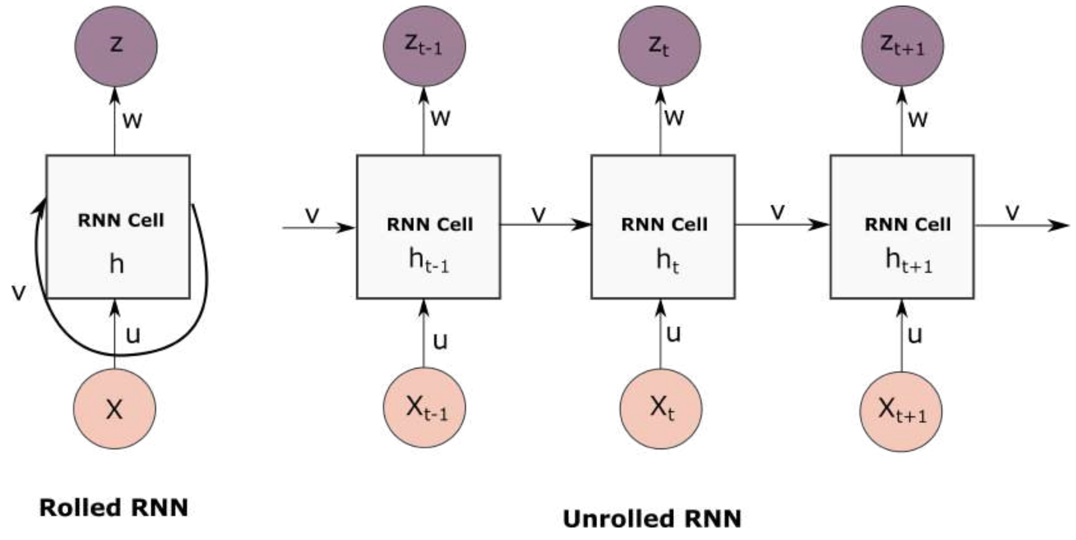


Figure 3.9: Basic RNN architecture for a single input.

depicts a better representation of each of the connections in an RNN. In the unrolled diagram above, the RNN consists of three times steps, which means that the length of the input sequence of the RNN is 3. In this example, the RNN also outputs a sequence of length 3.

At each time step in the above unrolled diagram, the arrow going into the cell shows the token at that particular index of the input sequence. The arrows coming outside in an upward direction from the cell show the output of the cell. The left-to-right arrows connecting the cell at each time step are the recurrent connections, which are the foundation of an RNN and represent the transmission of the state of each time step. This state in RNN provides information about the cell input and output from previous time steps to the cell at the current time step. This recurrent connection is very useful for capturing dependencies that make it easier to calculate probabilities and then perform predictions.

A default RNN cell is composed of two fully connected layers. The first layer is used for computing the state of the cell at a particular time step based on the previous state and the time step of the input. The second fully connected layer is used for computing the output of the cell at the time step. Furthermore, RNNs employ a special type of backpropagation technique for training, known as Backpropagation Through Time (BPTT) [110, 166]. As the loss here is defined as the sum of losses at each time step. In BPTT, the usual chain rule, along with the sum of the gradients at each time step over time, is applied.

Traditional RNN models sequential data by mapping the input sequences to hidden states, and then hidden states to outputs by employing the following recurrence equations:

$$\begin{aligned} h_t &= f(W_X X_t + W_h h_{t-1} + b_h) \\ z_t &= f(W_z h_t + b_z), \end{aligned} \tag{3.7}$$

where X_t is the input data, $h_t \in \mathbb{R}^N$ is the hidden state with N hidden units, f is an element-wise non-linear function, such as hyperbolic tangent or sigmoid, and z_t is the output at time t. W_X, W_h, and W_z represent the weights at the input, hidden and output layer, respectively. For a length T input sequence (X_1, X_2, X_3,...,X_T), the updates according to the equation 3.7 are computed sequentially as h_1 (letting h_0 = 0), z_1, h_2, z_2, ..., h_T, z_T.

Cell layers in RNNs can also be stacked. A basic representation of stacked RNN for two cell layers is shown in Fig. 3.10.

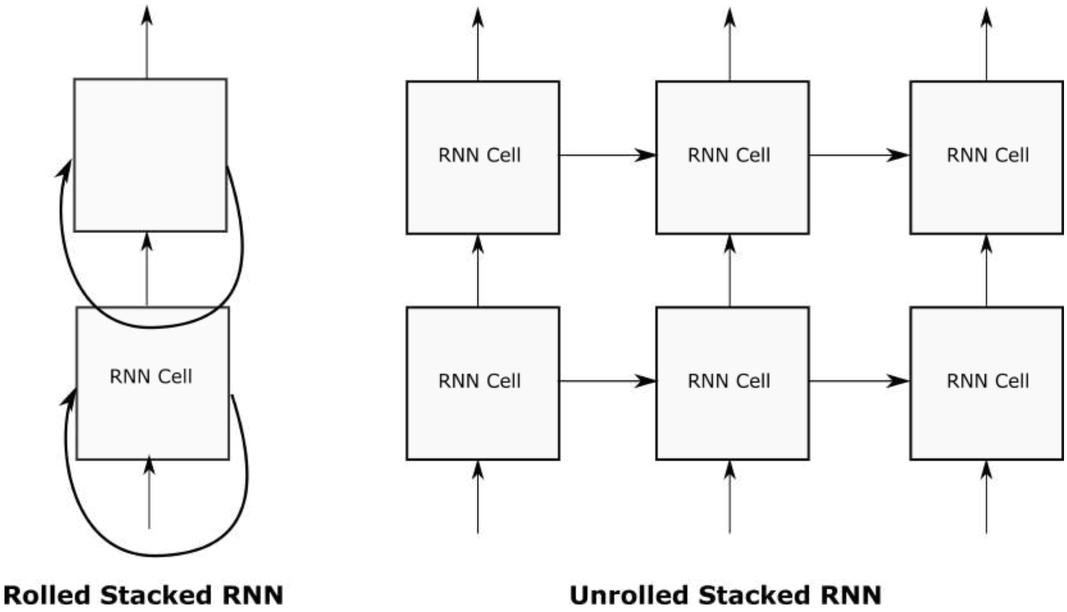


Figure 3.10: Diagram of an RNN with 2 cell layers.

At each time step, the output of the first cell becomes the input for the second cell, and the output of the second cell is the output of the overall RNN for that particular time step. Stacking cell layers allows the model to capture more complex features from the input data, leading to improved performance when trained on a large dataset. However, it can also increase the risk of overfitting. This makes regularization techniques, such as dropout, more essential as the size of the network is increases.

RNNs also allow sequences of input data to be processed in the form of a sequence of inputs, a sequence of outputs, or even both at the same time, as shown in the following diagram.

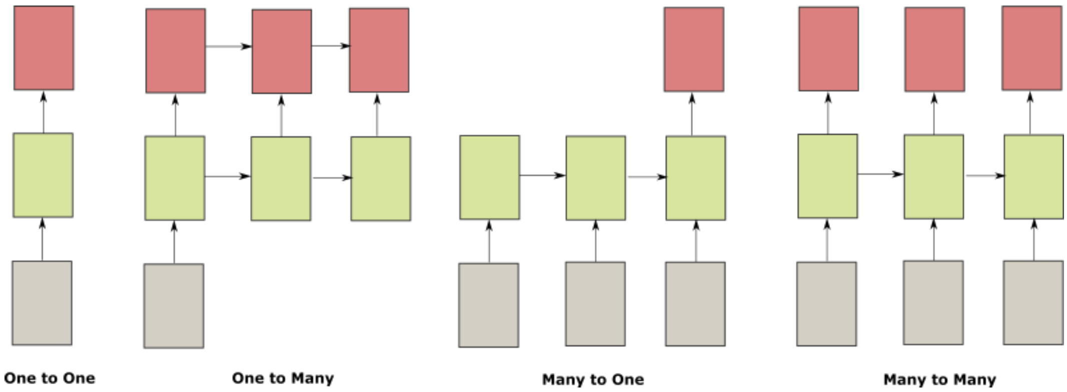


Figure 3.11: Sequence of data handled by RNNs.

Here, the bottom boxes represent the inputs, the middle boxes show the state of the RNN at that particular input, and the top boxes are the outputs.

While regular RNNs work well for most tasks, they are not always the best option. RNNs can also process sequences in both forward and backward directions, known as bidirectional RNNs [110, 167]. In the preceding examples, the direction of recurrent connections was only from left-to-right (only forward). In bidirectional RNNs, the recurrent connections also have a backward loop that looks at information from right to left. By processing a sequence in both directions, RNNs can take into account both past and future predictions, leading to improved performance in understanding sequences.

A regular RNN can capture all the necessary dependencies in an input sequence, but it struggles with handling long-term dependencies, such as dependencies between words that are far apart in a text sequence. To handle long-term dependencies, two variants of RNNs are commonly used: Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) [168].

3.8.1 Long Short-Term Memory (LSTM)

LSTM is specifically developed to keep track of all the relevant dependencies in a sequence. LSTMs were developed to tackle vanishing and exploding gradient problems [137, 138, 169]. An LSTM adds a few additional layers to the default RNNs. These additional layers are referred to as gates, which are helpful in regulating the information that is added or removed from the cell state. The additional gates enable the LSTM to handle long-term dependencies by providing a mechanism for selectively permitting or blocking information flow. They consist of a sigmoid neural network layer combined with a pointwise multiplication operation. Furthermore, the hidden states in LSTM hold the short-term memory, while the cells state holds the long-term memory.

An LSTM has three gates (forget gate, input gate, and output gate) to control and protect the cell state, as shown in Fig. 3.12.

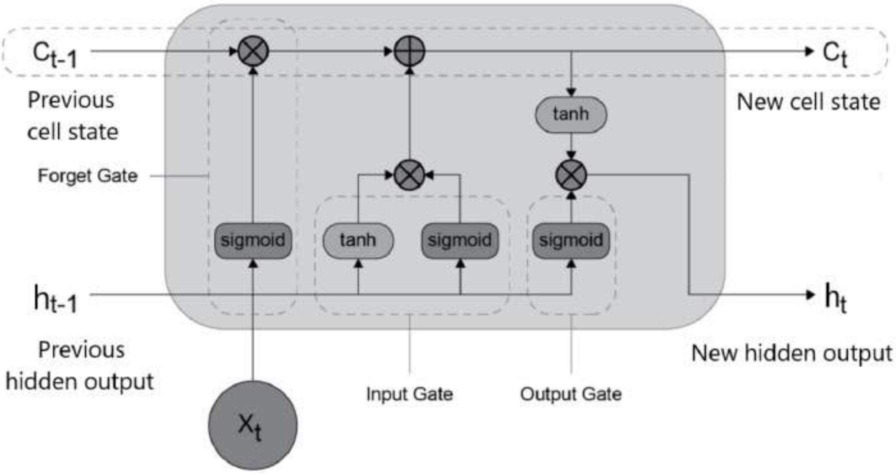


Figure 3.12: Architecture of the LSTM cell.

The first step in LSTM is to determine which information needs to be discarded from the cell state. This determination is carried out by a sigmoid layer known as the forget gate layer. It assesses the input from h_{t-1} and X_t and produces an output value between 0 and 1 for each element within the cell state C_{t-1}. A value of 0 signifies the previous cell state is completely forgotten, while a value of 1 represents the cell state is completely retained. Mathematically:

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, X_t] + b_f\right), \tag{3.8}$$

the subsequent phase involves determining which new information is going to be stored into the cell state, and this operation comprises two components. Initially, the input gate layer, a sigmoid layer, identifies which values should be modified. Following this, a tanh layer generates a vector of new potential values, denoted as $\tilde{C}_t$, which could be appended to the current state. In the subsequent step, these two components are combined to formulate an update for the state.

$$\begin{aligned} i_t &= \sigma\left(W_i \cdot [h_{t-1}, X_t] + b_i\right) \\ \tilde{C}_t &= \tanh\left(W_C \cdot [h_{t-1}, X_t] + b_C\right), \end{aligned} \tag{3.9}$$

Now, in this step, the old cell state (C_{t-1}) needs to be updated to the new cell state

(C_t). To achieve this, the old state is multiplied by f_t, effectively eliminating the elements designated for forgetting. Subsequently, the product of i_t and $\tilde{C}_t$ is added.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t, \tag{3.10}$$

in the final step, the output is derived from the cell state but in a refined form. Initially, a sigmoid layer is employed to determine which aspects of the cell state should be presented as output. Following this, the cell state undergoes a tanh transformation to confine the values within the range of -1 to 1, after which it is multiplied by the output from the sigmoid gate. This multiplication ensures that only the selected portions are included in the final output.

$$\begin{aligned} o_t &= \sigma\left(W_o \cdot [h_{t-1}, x_t] + b_o\right) \\ h_t &= o_t * \tanh\left(C_t\right), \end{aligned} \tag{3.11}$$

LSTMs are more powerful than basic RNNs because LSTMs use more parameters and have an explicit cell state. However, training an LSTM takes more time, is more computationally expensive, and is also more prone to overfitting [137, 138, 170].

3.8.2 Gated Recurrent Units (GRUs)

GRU is an excellent alternative to LSTM, and it works very well on smaller datasets [110, 171]. Since LSTMs have many parameters and are much more complex than regular RNNs, researchers developed GRU to reduce the complexity of LSTM and simplify it a little without losing its significant predictive capability. GRU merges the input and forget gates into a single gate, the “update gate”. It also combines the cell state and hidden state. GRUs provide the same performance as LSTMs in many situations, especially with smaller datasets, but with fewer parameters and faster training time.

Both variations of RNNs (LSTM and GRUs) can be stacked together, and both have bidirectional functionality [165, 171].

3.9 ANNs in GW-based SHM/NDT

GW-based damage identification techniques produce vital information regarding the existence, location, type, and size of defects in various kinds of structures, such as civil, mechanical, and composite structures [172–176].

Recently, Scanning Laser Doppler Vibrometry (SLDV) has been widely employed for measuring GWs on a very dense grid of points over the surface of a large specimen (full wavefield) [176–179]. Such wavefields contain rich information about the interaction of GWs with potential defects. However, due to the complex nature of these full wavefields, analysing them is very difficult for conventional physics or classical machine learning-based models. On the other hand, ANN-based approaches have shown much better performance in handling such nonlinear and complex data in various domains, such as speech recognition, computer vision, object detection, remote sensing, medical sciences, and many more [4–6, 180].

In the last two decades, many researchers have applied ANNs and their different variants for damage detection using GWs. Mardanshahi et al. [181] applied support vector machines (SVM), multilayer perceptron (MLP), and linear vector quantization (LVQ) neural networks for the purpose of detection and classification of matrix cracking in composites with the use of GWs. They first reduced their dataset with a dimensionality reduction technique called linear discriminant analysis (LDA) to discriminate between the classes and find a linear combination of features. They reported that the SVM outperformed other basic neural networks on their data.

Fenza et al. [182] applied probability-based methods and ANNs together to determine the degree and location of damage in composite and metallic plates by utilising Lamb waves. Results from both of their methods shown quite good performance in the detection and localization of micro defects in plate-like structures. Su and Ye [183] used Lamb waves for the identification of delamination in composite structures by employing multi-layer feedforward ANN architecture, wavelet transform, and Intelligent Signal Processing and Pattern Recognition (ISPPR) technique.

Perfetto et al. [184] proposed a GW-based ANN approach, which was crafted using the Finite Element Method (FEM), to identify the location of structural damage. The initial portion of the study delves into the creation and evaluation of this modeling approach. The precision of the FEM was validated by comparing its predicted outcomes against both experimental and analytical data. Subsequently, the ANN was designed, trained using an aluminum plate, and then tested on a composite plate, as well as various damage configurations. The findings presented in their study demonstrate that the ANN exhibits a remarkable ability to accurately detect and localize damage across all investigated scenarios.

Feng et al. [185] presented two time of flight (ToF)-based damage localization techniques using scattered Lamb waves in carbon fiber reinforced polymer (CFRP) plates. In their method, a probability matrix is constructed first by the probabilistic approach, which is used for the localization of damage, and then ANN is applied to improve the accuracy of damage localization.

Chetwynd et al. [186] applied MLP classification and regression neural networks on the acquired GW responses from healthy and damaged CFRP. They used the MLP classification technique to classify damaged and undamaged regions of CFRP, while the MLP regression technique was applied to estimate the exact location of the damage on the panel.

Melville et al. [187] used SVM and DL methods for detecting defects in metal plates (aluminum and steel) by employing full wavefield signals of ultrasonic GW images. They showed that DL methods performed better than the SVM.

Tabian et al. [188] presented a CNN-based technique for impact detection and localization in composites. They transformed the acquired GW-based data into 2D images and then applied CNN to perform health state classification. Ewald et al. [189] presented a DL technique for SHM with the use of Lamb waves. They preprocessed the sensor signal responses by applying wavelet transform to acquire the wavelet coefficient matrix (WCM). The WCM was then fed into the CNN to be trained to acquire the neural weights.

Song and Yang [190] proposed a multi-scale DL-based approach for the detection of subwavelength defect imaging using GWs. They combined two distinct fully convolutional networks (FCN). The first FCN acted as a global detection network, detecting subwavelength defects globally in a raw low-resolution GW beamforming image. The second FCN worked as a local super-resolution network, resolving subwavelength-scale details locally by finding the structural details of the detected defects.

Rautela et al. [191] presented a model-assisted DL approach for the detection and localization of structural defects using ultrasonic GWs. They applied their combined damage detection and localization approach on two different datasets: a time-history dataset and a time-frequency dataset. Detection of defects was performed on both the datasets using CNNs, and localization of defects was performed using regression-based CNN and LSTM models. They showed that the DL-based predictions surpassed the conventional machine learning approaches.

3.10 Conclusions

In this chapter, basic ANNs, biological neurons, and artificial neurons are discussed. A brief history of ANNs is mentioned. Then the relationship and differences between AI, ML, and DL techniques are elaborated. After that different types of learning tasks, which are carried out by machine learning and ANNs techniques are described. The learning and training process of a basic neural network is demonstrated. The parameters and hyperparameters of ANNs and their tuning techniques were described, which is very essential for the optimal output of a neural network. Then the working mechanism and applications of the two very popular architectures of ANNs: CNNs and RNNs (particularly LSTMs) are explained. The applications of ANNs and their variants (MLPs, CNNs, and RNNs) in GW-based SHM and NDT techniques are discussed.

Despite the extensive use of DL, the integration of DL techniques in the domain of GW-based damage identification has been rather limited. The exploration of DL capabilities within this context is still in its nascent stages, with only a few researchers having applied DL in GW-based damage identification. This, in turn, implies that there is substantial room for further investigation and innovation in this domain. As the potential benefits of leveraging DL for GW-based damage identification become increasingly evident, it becomes evident that there is still much work to be done in order to fully harness its capabilities and unlock its transformative potential.

Chapter 4

Dataset Computation and Preprocessing

4.1 Dataset Computation

In this research work, a synthetic dataset was generated by simulating propagating waves within carbon fiber reinforced composite (CFRP) plates. This was accomplished by using a parallel implementation of the time-domain spectral element method based on Mindlin-Reisner plate theory. Fifth order spectral elements with 36 nodes in each element were used. This gave a reasonable computation time in comparison to the use of solid elements and good accuracy, as detailed in [192]. Essentially, this dataset resembles the particle velocity measurements observed at the lower surface of the plate, acquired through Scanning Laser Doppler Vibrometry (SLDV) in the transverse direction in response to piezoelectric (PZT) excitation at the center of the plate. The input signal employed was a five-cycle Hann window-modulated sinusoidal tone burst with a carrier frequency of 50 kHz and the modulation frequency of 10 kHz. Opting for a relatively low carrier frequency offered several advantages, including the ability to employ a less dense mesh and a considerable reduction in computational time compared to simulations involving higher frequencies. Furthermore, the excitation signal was selected so that interaction of generated A_0 Lamb wave mode with the smallest delamination can still be used as a feature for damage identification. To ensure that the GWs had sufficient time to propagate to the edges of plate and return to the actuator, the total wave propagation time was set at 0.75 ms. Further, 150000 time integration steps were used to maintain the stability of the central difference scheme.

The material was a typical cross-ply CFRP laminate with the stacking sequence $[0/90]_4$ was employed in the model. The properties of a single ply were as follows [GPa]: $C_{11} = 52.55$, $C_{12} = 6.51$, $C_{22} = 51.83, C_{44} = 2.93, C_{55} = 2.92, C_{66} = 3.81$. These property choices were made to replicate wavefront patterns and wavelengths closely resembling those observed in wavefield measurements conducted with SLDV on CFRP specimens. These CFRP specimens were subsequently employed to evaluate the effectiveness of the developed approach for identifying delamination. Specifically, for numerical simulations, the shortest wavelength of the propagating A_0 Lamb wave mode was 21.2 mm, whereas for experimental measurements, it was slightly shorter at 19.5 mm.

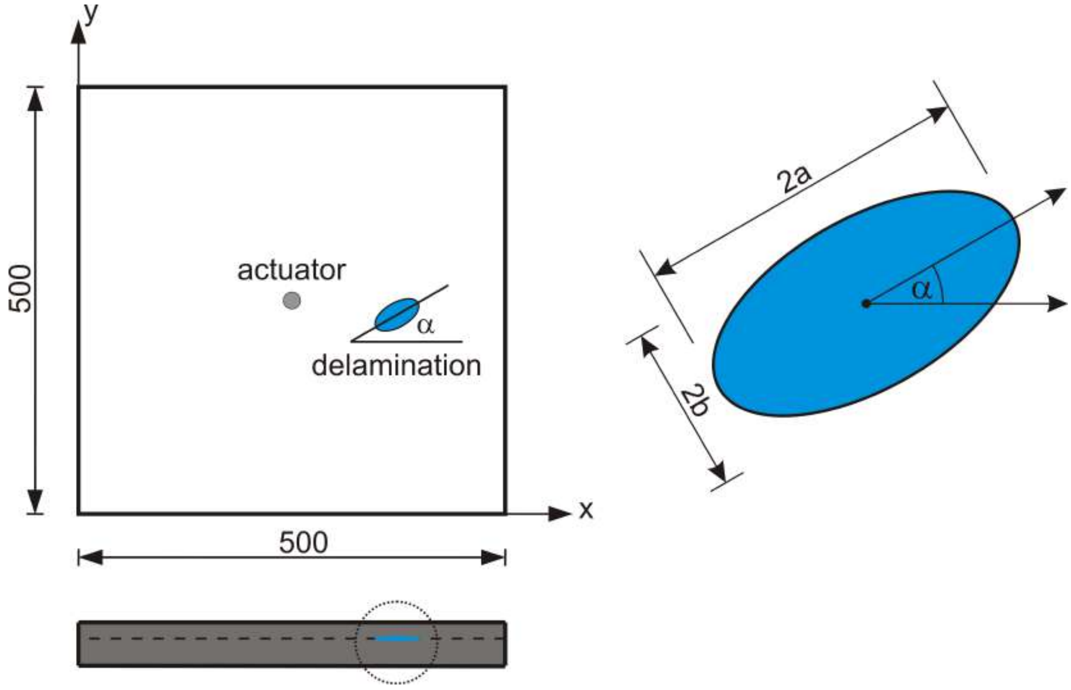


Figure 4.1: Setup for computing Lamb wave interactions with delamination.

A total of 475 scenarios were simulated to represent the propagation and interaction of Lamb waves with single delamination in each case. In each scenario, a single delamination was incorporated into the model using the technique of node splitting between appropriate spectral elements. The composite laminate was assumed to consist of eight layers, with a total thickness of 3.9 mm. The delamination was specifically modeled between the third and fourth layers, as illustrated in Fig. 4.1. It is important to note that Fig. 4.1 presents an exaggerated cross-sectional view of the delamination.

In the dataset computation process, zero-volume delamination were considered, and their spatial locations were randomly determined to introduce variability in the interaction between GWs and delamination. This variation included cases where the delamination was positioned at the corner of the plate, which is particularly challenging for signal processing methods to detect. Additionally, the size and shape of the delamination were randomly generated, involving the selection of the minor and major axes of an elliptical shape. Furthermore, the angle between the major axis of the delamination and the horizontal axis was chosen randomly. The simulated delamination scenarios incorporated these various random factors:

- the geometrical size of the delamination, represented by $2b$ and $2a$, namely ellipse minor and major axis, was randomly selected from the interval $[10\,\text{mm}, 40\,\text{mm}]$,
- the angle of the delamination (α) was also randomly chosen within the range of $[0°, 180°]$,
- the coordinates of the centre of delamination (x_c, y_c) were randomly determined within two intervals $[0\,\text{mm}, 250\,\text{mm} - \delta]$ and $[250\,\text{mm} + \delta, 500\,\text{mm}]$, where $\delta = 10\,\text{mm}$). The parameter δ was selected to avoid situation in which the delamination is in proximity to an actuator or directly below it.

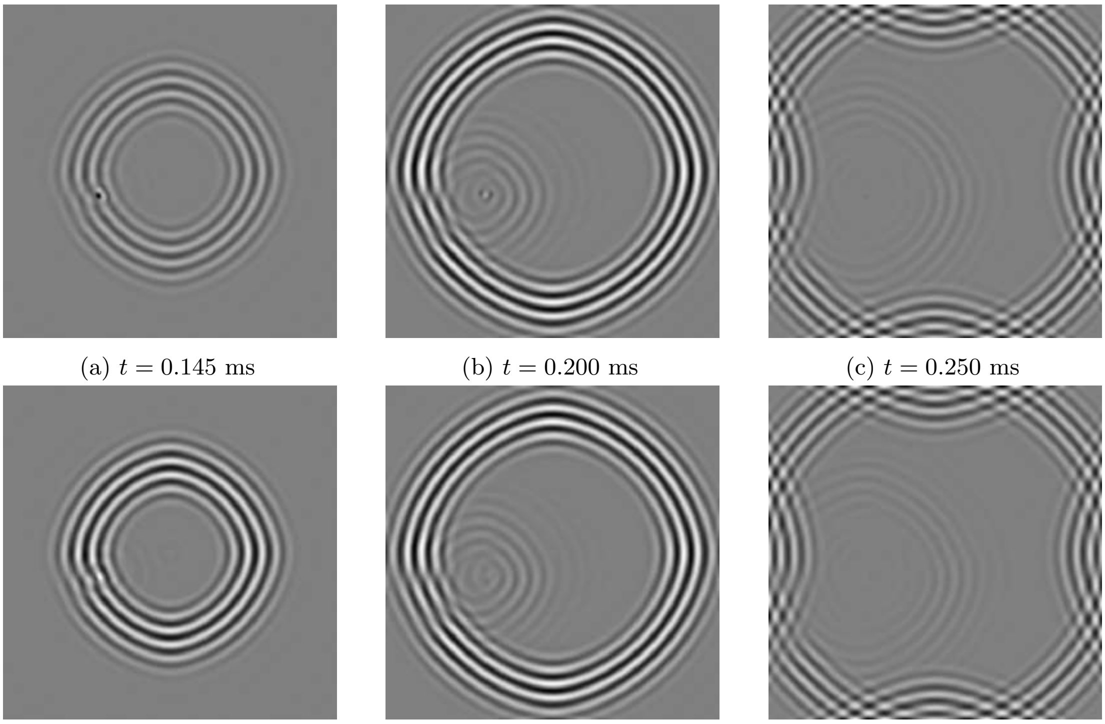
The output, consisting of particle velocities at the nodes of spectral elements on both the upper and lower surfaces of the plate, was interpolated onto a uniform grid comprising (500×500) points. This interpolation was accomplished using the shape functions of the elements (further details can be found in [193]). Essentially, this grid representation closely resembles measurements obtained by SLDV in the transverse direction, which is perpendicular to the plate surface.

An example of the simulated full wavefield data obtained from both the top and bottom surfaces is shown in Fig. 4.2. It is worth noting that a stronger wave entrapment phenomenon at the delamination region is observed in the wavefield at the top surface, primarily because the delamination within the cross-section is situated closer to the top surface. This proximity makes delamination detection more feasible when processing wavefields from the top surface. Additionally, it is important to mention that the output from the wave propagation model is in the form of a 3D matrix, containing wave amplitudes at location (x, y) and time t_j. Consequently, this can be viewed as a series of frames representing propagating waves at discrete time intervals.

Detecting delamination near the top surface is facilitated through the use of signal energy, which is represented by root mean square (RMS), as defined in Eq. 4.1, applied to the wavefield. The RMS is defined as:

$$\hat{s}(x,y) = \sqrt{\frac{1}{N}\sum_{j-1}^{N} s\left(x,y,t_j\right)^2}, \tag{4.1}$$

in this research study, a total of 512 sampling points ($N = 512$) were utilized. Consequently, the dataset was reduced to 475 two-dimensional matrices, with amplitudes represented as double-precision values. The outcome of this transformation is visually depicted in Fig. 4.2. Based on image analysis, it becomes apparent that the shape of the delamination is more readily distinguishable in the top case. However, it should also be noted that the dataset used in the research work is of the bottom case, which was more challenging for the DL models than the top case.


(a) $t = 0.145$ ms (b) $t = 0.200$ ms (c) $t = 0.250$ ms

(d) $t = 0.145$ ms (e) $t = 0.200$ ms (f) $t = 0.250$ ms

Figure 4.2: Full wavefield at the top (a)−(c) and bottom surfaces (d)−(f), respectively, at selected time instances representing the interaction of GWs with delamination.

4.2 Data Preprocessing

The dataset, consisting of frames capturing propagating waves (512 frames per delamination scenario), was further converted to 8-bit greyscale images, and is accessible online [194]. It encompasses a total of 475 distinct delamination cases, with each case composed of 512 frames. This results in a total of $243,200$ frames or images, where each frame is sized at (500×500) pixels, representing the dimensions of the specimen (500×500) mm^2.

However, utilizing all frames for each case incurs substantial computational and memory expenses. Notably, frames illustrating the propagation of GWs prior to encountering the delamination lack discernible features (as illustrated in Fig. 4.3). Consequently, for the training process of all the developed models in chapters 5, 6, and 7, I opted to select a specific number of frames starting from the initial interaction with the delamination (f_1) until a specific number of frames window (f_w), considering the available computational memory.

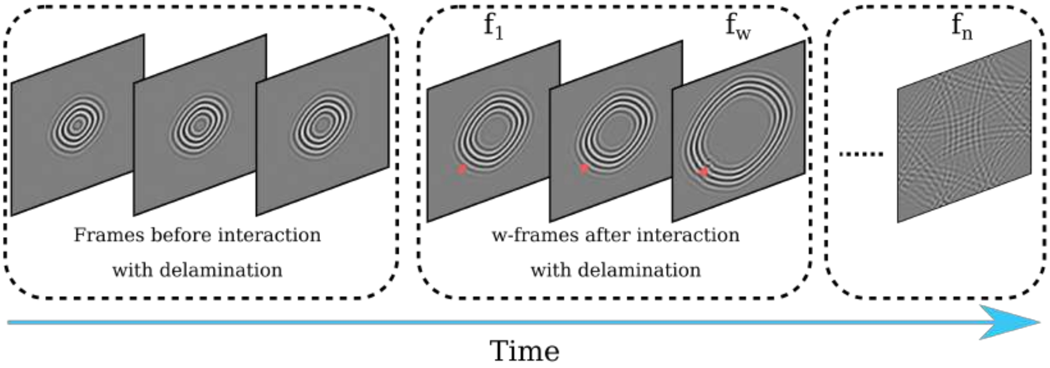

f1
fw
fn
Frames before interaction with delamination
w-frames after interaction with delamination
......
Time

Figure 4.3: A sample of frames from the full wave propagation dataset.

Figure 4.3 illustrates the selected frames at different time-steps of the propagating Lamb waves before and after the interaction with the damage. The initial frame, denoted as f_1, illustrates the initial interactions of Lamb waves with the delamination. This frame was computed based on the delamination's location and the velocity of the A_0 mode of Lamb waves. On the other hand, f_w represents the final frame within the training sequence window, and f_n shows the last frame in a delamination case; the value f_n is 512 for all of the delamination cases, as each delamination case is composed of 512 frames.

To enhance the performance of optimizer throughout the training phase for all the developed models in this research work, the colour scale values were normalised to a range of $(0-1)$ instead of the initial scale values, which were in the range of $(0-255)$.

4.3 Dataset Division

It is a common practice in machine learning/DL to split a dataset into two distinct subsets: a training subset and a testing subset. A third subset, named the validation set, is also generated, but mostly, this subset is taken from the training subset during the training process of an algorithm. The primary reason for dividing the dataset into these three distinct subsets is to ensure the appropriate training, fine-tuning, and evaluation of the model while avoiding the introduction of bias. Here is an overview of each subset:

- **Training set:** As its name implies, this set is utilized for training the neural network. In the context of supervised learning, it encompasses both the features and the corresponding target values. Typically, this subset is the largest among the three, as neural networks demand a substantial amount of data for effective training.

- **Validation set:** The primary function of this set is to gauge the performance of the model, facilitating adjustments to hyperparameters to enhance its effectiveness. This fine-tuning process enables machine learning engineers to configure hyperparameters that yield optimal results. Although the model is not directly trained on this data, it indirectly influences it, which is why it is not used for the final performance evaluation to avoid potential bias.
- **Testing set:** This set plays no role in the development or training of the model, making it invaluable for conducting a final evaluation on unseen data. This evaluation serves as a benchmark for the prospective performance of the model on new datasets.

While there is no precise technique for determining the ideal split ratio for these three sets, as data problems vary widely, developing DL solutions often involves a trial-and-error approach. However, it is generally accepted that for larger datasets (comprising hundreds of thousands or millions of instances), a split ratio of approximately 98% for the training set, 1% for the validation set, and 1% for the testing set is effective, emphasizing the need to allocate as much data as possible to the training set. For datasets of more typical sizes, a common split ratio is around 60% for training, 20% for validation, and 20% for testing. This data splitting process for a normal size data is elaborated in Fig. 4.4.

To facilitate the training and evaluation of all the developed DL models in this research work, elaborated in chapters 5, 6, and 7, I partitioned the dataset into two distinct subsets: the training set, encompassing 80% of the data, and the testing set, which comprises the remaining 20%. Furthermore, within the training set, 20% of the data was set aside as a validation set. This validation set played a crucial role in verifying the performance of model during the training phase.

4.4 Conclusions

In this chapter, the computation of the synthetic dataset, the materials used for generating it, and the specification of those materials are explained. The dataset employed in this research work for developing all the DL models is elaborated. A sample of the top and bottom cases of the full wavefield is shown. Furthermore, the data preprocessing steps implemented all of the developed models in this research work are explained.

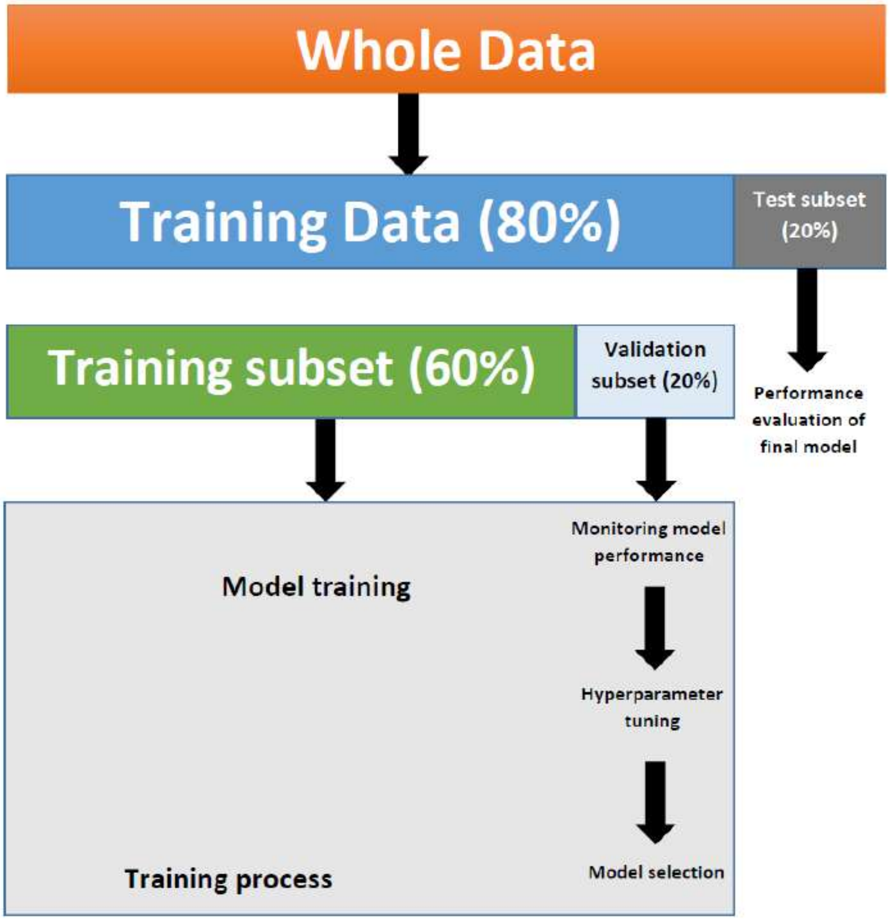


Figure 4.4: The dataset splitting process.

Finally, the data splitting process for training a DL algorithm and the data division process used in this research work are described.

Upon analysing the dataset, it was discovered that delaminations positioned closer to the top surface are more easily detectable through RMS analysis. Consequently, the decision was made to opt for the more challenging scenario of delaminations closer to the bottom surface for training the DL models. Moreover, due to computational constraints, it is impractical to input all frames into a DL model, necessitating the use of a specific number of frames window that fits within the available computational memory.

Chapter 5

DL-based delamination identification by using animation of guided wave propagation

5.1 Introduction

In this chapter, I describe the approach developed for the 1st task of this research work, which is delamination identification using DL approach. The developed DL model is able to process the full wavefield frames depicting the propagation of Lamb waves, as detailed in the dataset described in the previous chapter (chapter 4). Consequently, there was no need for additional signal post-processing techniques, such as Root Mean Square (RMS). The employed DL model operated on a many-to-one prediction scheme, signifying the utilization of multiple input frames to generate a damage map. In simpler terms, a sequence of full wavefield frames, effectively forming an animation, served as the input for the proposed DL model. The proposed model was inspired by the Convolutional Long Short-Term Memory (ConvLSTM) [195] architecture and was fine-tuned for the specific task of delamination identification. In the context of pixel-wise segmentation, two classes were defined: “damaged” and “undamaged.”

Notably, this is among one of the first implementation of deep neural networks employing Lamb wave propagation animations through semantic segmentation for delamination identification. The proposed approach exhibited remarkable proficiency in identifying delamination in numerically generated datasets. Furthermore, the proposed approach demonstrated a capacity for generalization, implying its potential applicability in real-world scenarios. This assertion finds support in the experiments conducted on Carbon Fiber Reinforced Polymer (CFRP) plates featuring single and multiple delaminations.

The image segmentation, ConvLSTM, the developed DL model architecture, and results obtained from numerical and experimental data are elaborated in the subsequent sections.

5.2 Image Segmentation

Processing an entire image as a single entity may not be a good solution for a deep learning model, as there are often regions within an image that lack pertinent information. Image segmentation, therefore, entails dividing an image into segments, allowing for the selective processing of essential segments. This concept encapsulates the essence of image segmentation. Image segmentation constitutes a foundational element in various visual recognition systems. Semantic segmentation, in particular, holds significant relevance in the realm of image comprehension and plays a pivotal role in image analysis tasks. Researchers in computer vision and machine learning are increasingly drawn to the field of image semantic segmentation, given its growing importance. Many emerging applications demand precise and efficient segmentation techniques. In recent years, image segmentation has found extensive application in diverse domains, including medical applications [196], autonomous driving [197], augmented reality [198], agricultural sciences [199], and more. Across these domains, semantic segmentation often surpasses alternative approaches by delivering superior accuracy and efficiency.

The objective of image segmentation is to partition images or video frames into multiple objects or segments [200]. This process can be framed as a pixel-level classification problem with semantic labels, known as semantic segmentation, or as the partitioning of images into individual objects, termed as instance segmentation [200, 201]. Consequently, semantic segmentation is generally more complex than image classification, which assigns a single label to the entire image [201]. Moreover, the effectiveness of semantic image segmentation hinges not only on the semantics in question but also on the specific problem being addressed [202].

To comprehend how modern DL architectures tackle semantic segmentation effectively, it is essential to recognize that this field is not isolated but rather constitutes a natural progression from coarse to fine inference. It originates from classification, where predictions pertain to the entire input, such as identifying objects within an image or providing a ranked list for multiple objects. Localization or detection represents the subsequent step, furnishing not only class labels but also spatial information, such as centroids or bounding boxes. Semantic segmentation naturally follows as the next stride toward achieving fine-grained inference, aiming to make dense predictions by assigning labels to every pixel. Consequently, each pixel is associated with the class of the object or region it belongs to. Further refinements can include instance segmentation (assigning distinct labels to different instances of the same class) and even part-based segmentation (low-level decomposition of already segmented classes into their components).

5.3 ConvLSTM

The dataset for this research task is composed of the animation of full wavefield frames, which contains sequences of images. Each image or frame within the same delamination case relies on its preceding frame, effectively creating a video-like scenario. While Convolutional Neural Networks (CNNs), elaborated in section 3.7, have demonstrated promising performance in various image analysis and computer vision tasks, they face challenges when it comes to learning temporal features from sequences of images and maintaining continuity with previous observations.

On the other hand, Long Short-Term Memory networks (LSTMs), described in section 3.8.1, have excelled in processing time series data but exhibit limitations in capturing spatial information when the input consists of consecutive images. To address this issue, the ConvLSTM architecture was introduced by Shi et al. [195], offering a solution to such problems.

Within ConvLSTM, convolution operations are applied in both the input-to-state transition and state-to-state transitions. This concept aligns with the idea of traversing a series of images, one slice at a time, much like how an LSTM progresses through a series of data points individually. This approach proves valuable when dealing with image sets where the order of frames is crucial.

The ConvLSTM cell architecture, depicted in Fig. 5.1, represents a variation of the LSTM, incorporating convolutional operations within the LSTM cell itself.

Essentially, ConvLSTM serves as a combination of convolutional operations and LSTM cells, enabling the capture of both time-correlated and spatial features within a series of consecutive images. Equation (5.1) depicts the ConvLSTM operations as the inputs $x_1, \ldots, x_t$, hidden states $h_1, \ldots, h_t$, cell states $c_1, \ldots, c_t$, and input, forget, and output gates are represented as i_t, f_t, and o_t, respectively:

$$\begin{aligned}
i_t &= \sigma\left(W_{x_t} * x_t + W_{h_{t-1}} * h_{t-1} + W_{ci} \cdot c_{t-1} + b_i\right) \\
f_t &= \sigma\left(W_{xf} * x_t + W_{hf} * h_{t-1} + W_{cf} \cdot c_{t-1} + b_f\right) \\
c_t &= f_t \cdot c_{t-1} + i_t \cdot \tanh\left(W_{xc} * x_t + W_{hc} * h_{t-1} + b_c\right) \\
o_t &= \sigma\left(W_{xo} * x_t + W_{ho} * h_{t-1} + W_{co} \cdot c_t + b_o\right) \\
h_t &= o_t \cdot \tanh\left(c_t\right),
\end{aligned} \tag{5.1}$$

where $(*)$ indicates an element-wise multiplication operation.

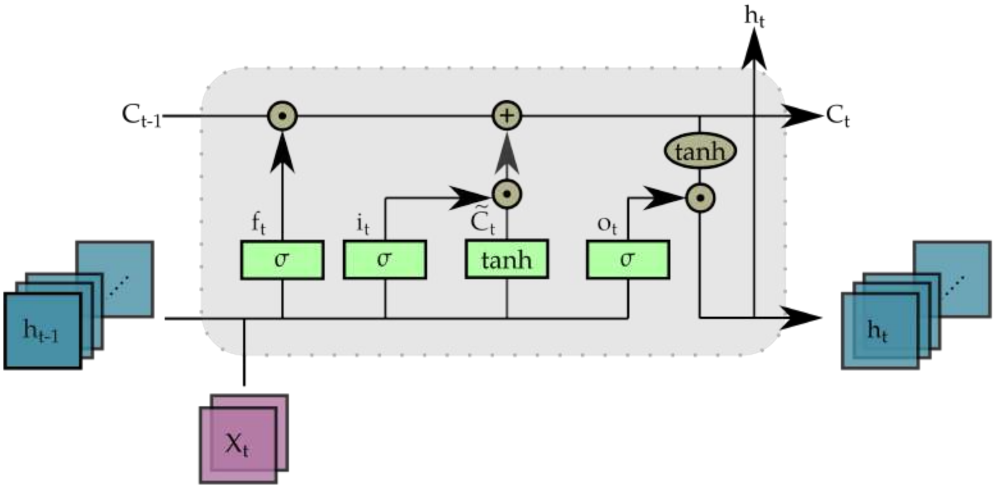


Figure 5.1: A single ConvLSTM cell at time t.

Recently, ConvLSTM has gained substantial popularity and is increasingly applied in various image processing and computer vision applications.

5.4 The proposed approach

For the purpose of identifying delamination in CFRP materials, I developed a DL model based on ConvLSTM approach. The proposed model leverages the full wavefield frames depicting the propagation of Lamb waves to discern the presence of delamination. The developed model follows a many-to-one sequence prediction approach. It operates by considering a sequence of w frames, each representing the full wavefields progression over time and its interaction with the delamination. These frames are used to extract features indicative of damage and ultimately predict the location, shape, and size of delamination in a single output image.

The proposed DL model takes 64 frames as input, i.e., the size of f_w is set to 64 here, and the dataset splitting is being performed according to section 4.3.

5.4.1 The proposed ConvLSTM model

The proposed ConvLSTM model, as illustrated in Fig. 5.2, comprises three ConvLSTM layers. These layers are configured with varying filter counts: the initial ConvLSTM layer employs 12 filters, followed by the second layer with 6 filters, and finally, the third layer with 12 filters.

In all of the ConvLSTM layers a kernel size of (3×3), and a stride of size (1) are used.

To maintain output equivalence with the input under a stride of (1), the padding is set to "same".

Within these ConvLSTM layers, a tanh (hyperbolic tangent) activation function is employed, ensuring that output values fall within the range of (-1 to 1). Additionally, after the initial two ConvLSTM layers, a batch normalization technique [203] is applied to enhance the performance of the model by providing the normalized output to the next layer.

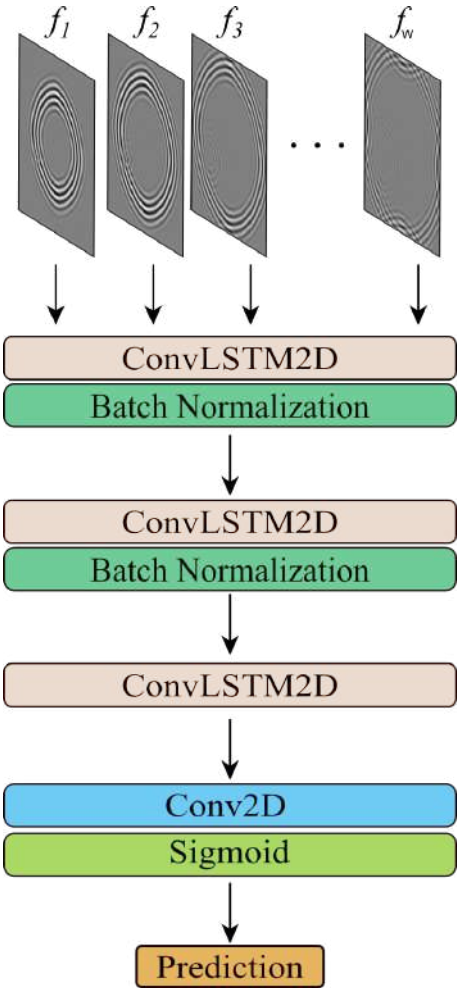


Figure 5.2: The architecture of the proposed DL model.

A 2D convolutional layer is employed at the last output layer, followed by a sigmoid activation function. This arrangement yields output values within the range of $(0, 1)$, effectively indicating the likelihood of delamination.

To classify the output, a threshold value must be established, distinguishing between damaged (represented as 1) and undamaged (represented as 0 in the case of employing sigmoid activation function. Consequently, I set the threshold value at (0.5), designating all values below this threshold as undamaged and considering only those values surpassing the threshold as indicative of damage.

5.4.2 The Development Environment

For the development of the proposed DL model, Pycharm IDE (Integrated Development Environment) Community Edition was used along with the Python version 3.8. The Keras API [204] version (2.0.5) running on top of TensorFlow version (2.5.0) on two Tesla V100 GPUs from NVIDIA, with each GPU having 32 GB of memory, resulting in a total of 64 GB of GPU memory was utilized.

5.4.3 Evaluation Metrics

Assessing the performance of a DL model is essential to ensure its effectiveness, not only on the training data but also on unseen data. Evaluation becomes relatively straightforward in supervised learning scenarios, where ground truth serves as a benchmark for comparing the model's predictions.

To evaluate the performance of the proposed model, the mean Intersection over Union IoU, also known as the Jaccard index, was employed as the accuracy metric. IoU has wide applications in image segmentation-based tasks. This metric calculates IoU by assessing the intersection area between the ground truth and the predictions. A graphical illustration of the IoU metric is shown in Fig. 5.3.

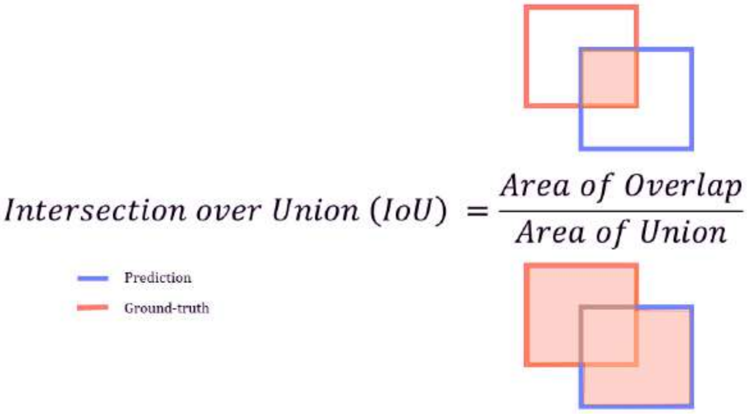

Intersection over Union (IoU) = Area of Overlap / Area of Union
Prediction
Ground-truth

Figure 5.3: A graphical representation of the IoU.

As this task has two output classes (damaged and undamaged), IoU was specifically calculated for the damaged class. Equation (5.2) illustrates the IoU metric:

$$IoU = \frac{Intersection}{Union} = \frac{\hat{Y} \cap Y}{\hat{Y} \cup Y}, \tag{5.2}$$

in the equation provided, $\hat{Y}$ represents the predicted output, and Y corresponds to the ground truth. Additionally, the percentage area error (ϵ) depicted in Eq. 5.3 was utilised to evaluate the performance of the proposed model:

$$\epsilon = \frac{|A - \hat{A}|}{A} \times 100\%, \tag{5.3}$$

here, A and $\hat{A}$ denote the area in mm^2 attributed to the damage class in both the ground truth and the predicted output, respectively. This metric serves as an indicator of how closely the predicted delamination area aligns with the ground truth. A lower (ϵ) value signifies greater accuracy in identifying the damage.

Moreover, it is worth noting that for all predicted outputs, the delamination localization error (i.e., the distance between the delamination centers of the ground truth and the predicted output) was found to be less than (0.001%) thus, it is not discussed further.

5.4.4 Hyperparameters tuning and DL model training

To enhance the IoU value and minimize the loss of proposed DL model, the Adam optimizer [205] was employed. The Adam optimizer integrates a momentum term and adapts the learning rate during training, yielding superior performance compared to other optimizers. The detailed training configurations and values of the hyperparameters are provided in Table 5.1.

Table 5.1: Configuration of the training process of the proposed model.

Hyperparameter	Value
Loss function	Binary crossy-entropy
Learning rate	3e-4
Batch size	2
Number of epochs	249
Early stopping	30 epochs

Since there is no simple approach to select the best parameters for training a DL model, a trial-and-error approach was employed. The value of the regularization technique, early stopping, was set into 30 epochs and was evaluated based on IoU of validation data. During the model training process, if the IoU value on the validation data does not increase for consecutive 30 epochs, the model stops training and saves the model with the highest IoU value on the validation data. The training process of the proposed ConvLSTM model is shown in Fig. 5.4.

During the training process of the proposed model, it achieved highest values of the IoU at 219th epoch on training and validation data are 0.96 and 0.93, respectively.

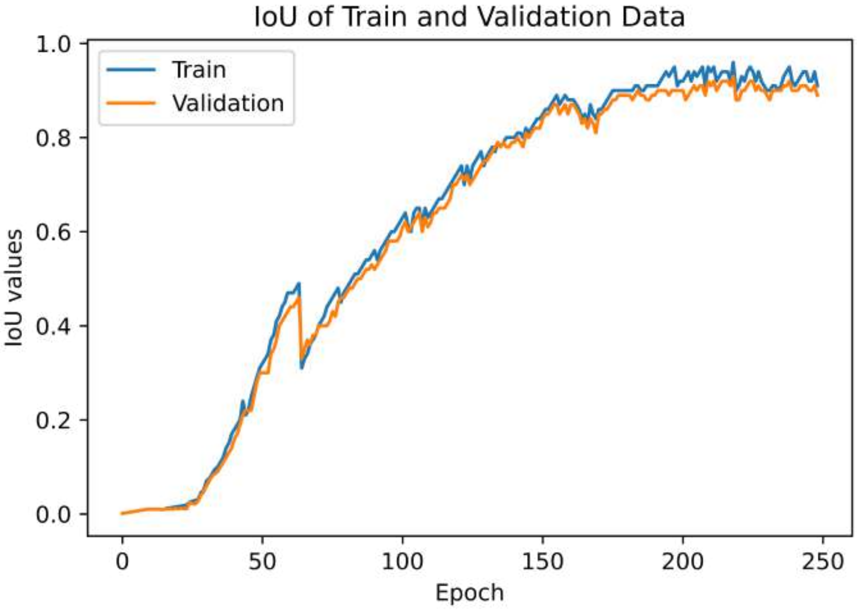


Figure 5.4: Illustration of the proposed ConvLSTM model training.

5.5 Results and discussions

In this section, I present the evaluation of the proposed model on the numerical test data of 95 different cases and experimental data representing the frames of the full wavefield propagation acquired by SLDV. The proposed model was evaluated using both numerical and experimental data to demonstrate the capability to predict the delamination shape, size, and location.

From the numerical cases, I selected four representative cases to illustrate the performance of the model. In these numerical scenarios, predictions were derived solely from the initial frame window following the damage interaction, given the availability of delamination ground truths.

To assess the generalization ability of the proposed model, experimental data encompassing single and multiple delaminations were considered. Furthermore, the IoU metric was employed to evaluate the performance of the proposed model.

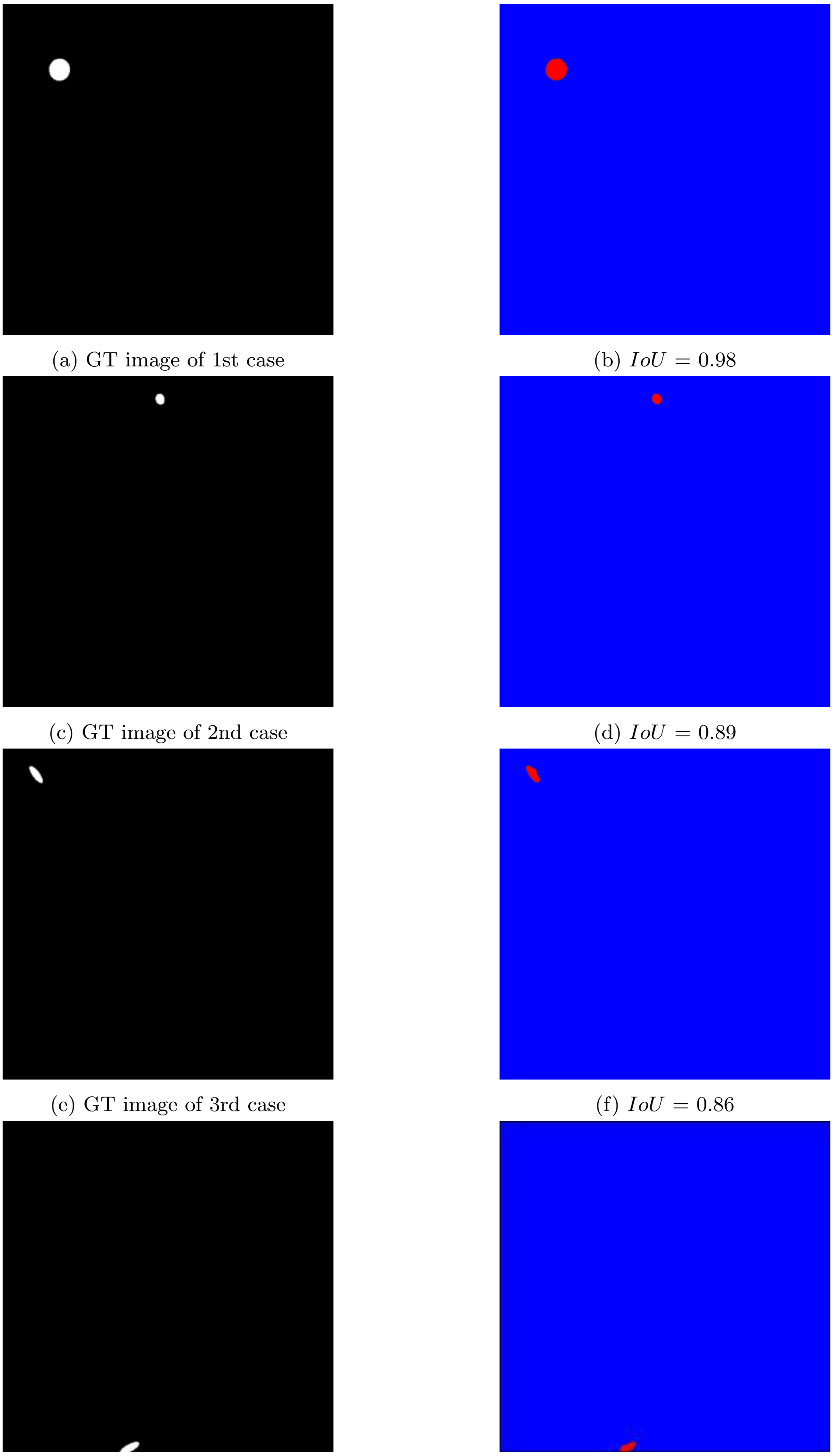
5.5.1 Numerical cases

In the first numerical scenario, the delamination is located at the upper left corner, located slightly away from the edges, as evident in Fig. 5.5a, representing the ground truth (GT). This delamination case is considered to be easy for the proposed model and has gained the highest IoU value (0.98) among all of the test cases. The predicted output of the model is illustrated in Fig. 5.5b.

In the second numerical case, the delamination is centrally located at the upper section of the plate, as depicted in Fig. 5.5c, representing the GT. This case presents a challenging scenario as wave reflections from the edges of plate exhibit patterns similar to those arising from the delamination itself. Consequently, Fig. 5.5d displays the model's prediction for this delamination case.

In the third case, the delamination is on the upper left corner, as showcased in Fig. 5.5e, representing its GT. Similar to the second case, this scenario poses difficulty due to edge wave reflections sharing resemblances with delamination reflections. This can be observed from the predicted output of model for this particular case in Fig. 5.5f.

In the fourth test case, the delamination is located at the left bottom of the plate and is touching the edges of the plate, with its GT shown in Fig. 5.5g. The prediction of this case is shown in Fig. 5.5h. This case is considered the most challenging case for the DL model to predict because wave reflections from the edge overshadows reflections from delamination, and the size of the delamination is also tiny as compared to other delamination cases. Due to these reasons, the proposed model has achieved the lowest IoU value (0.63) among selected test cases.



(a) GT image of 1st case

(b) $IoU = 0.98$

(c) GT image of 2nd case

(d) $IoU = 0.89$

(e) GT image of 3rd case

(f) $IoU = 0.86$

(g) GT image of 4th case

(h) $IoU = 0.63$

Figure 5.5: Delamination cases on numerical data (Figures: (a), (c), (e), and (g) correspond to the GT of each numerical case. Figures: (b), (d), (f), and (h) represent the predictions of the proposed model).

As can be seen in all predicted outputs, the proposed model is able to identify the delamination with high accuracy and without any noise.

Table 5.2 provides an overview of the evaluation metrics applied to assess the performance of the proposed model, regarding the numerical cases shown in Fig. 5.5. As shown in Table 5.2, the actual (A) and predicted areas ($\hat{A}$) of delaminations were computed in mm^2 with respect to each case. The percentage area error (ϵ) was also calculated for the proposed model.

Table 5.2: Evaluation metric of the four numerical cases.

case number	A [mm^2]	IoU	$\hat{A}$ [mm^2]	ϵ
1	842	0.98	871	3.4%
2	186	0.89	196	5.4%
3	272	0.86	318	16.9%
4	174	0.63	182	4.6%

The achieved mean IoU with respect to all numerical data of 95 cases was (0.90), indicating a high level of predictive accuracy. Furthermore, the mean percentage area error (ϵ) calculated for 95 numerical test cases was equal to 4.57%.

5.5.2 Experimental cases

In this section, I present an investigation into the performance of the proposed model using experimentally acquired data. This experimental setup is similar to that of the synthetic dataset, where a frequency of 50 kHz is applied to excite a signal in a transducer placed at the centre of the plate. The A_0 mode wavelength for this particular CFRP material at such frequency is 19.5 mm.

The measurements involved employing the Polytec PSV-400 Scanning Laser Doppler Vibrometer (SLDV) on the bottom surface of the plate, with dimensions of 500 × 500 mm point grid. The measurements were conducted on a regular grid of 333 × 333 points at a sampling frequency of 512 kHz, and aligned with the edges of the plate. To improve the signal-to-noise ratio, 10 averages were used. The scanning process took approximately 1 hour and 40 minutes for each specimen. Subsequently, a median filter with a window size of three was applied to every frame, and cubic interpolation was employed to upsample all frames to (500 × 500) points.

During the testing phase with the synthetic dataset, the model was provided with a consecutive sequence of identified frames (referred to as a “window of frames”) that encapsulated the interactions of Lamb waves with the delamination, facilitating the identification process.

The same size of "window of frames" was used here as in numerical data, namely (64), but in the case of experimental data, the delimitation initiation information is unknown. Therefore, I slide the window over all input frames by shifting the window one frame at a time and noted the IoU value for each sliding operation.

5.5.3 Single delamination

The first experimental case involved a CFRP specimen with a single delamination, created artificially by a Teflon insert of a thickness 250 μm. The complete specifications of this CFRP specimen are the same as in the numerical simulations, as described in section 4.1. The Teflon insert was square in shape and inserted during the specimen's manufacturing, providing knowledge about its shape and location. Consequently, the ground truth was prepared manually to accurately depict the location, shape, and size of the artificial delamination. The configuration of the experimental case for the single delamination arrangement is shown in Fig. 5.6.

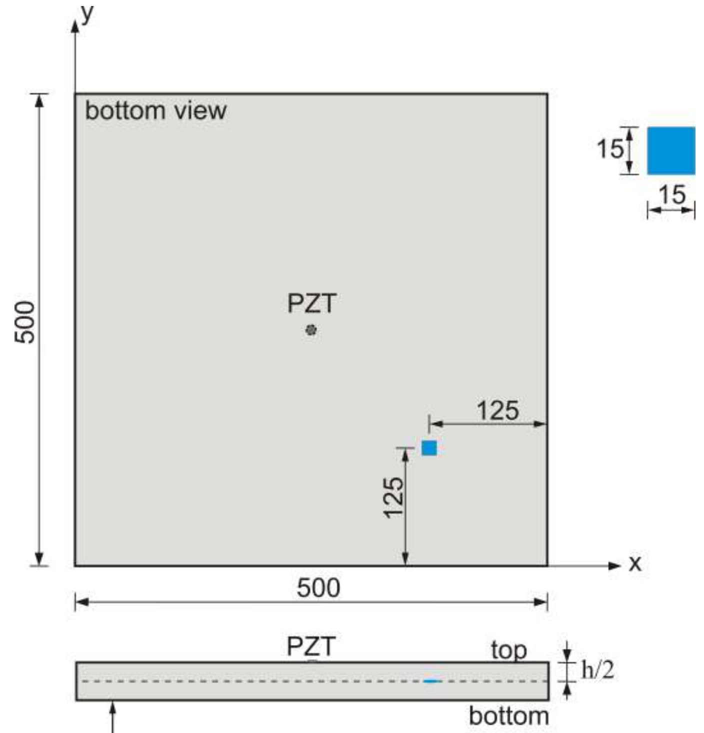
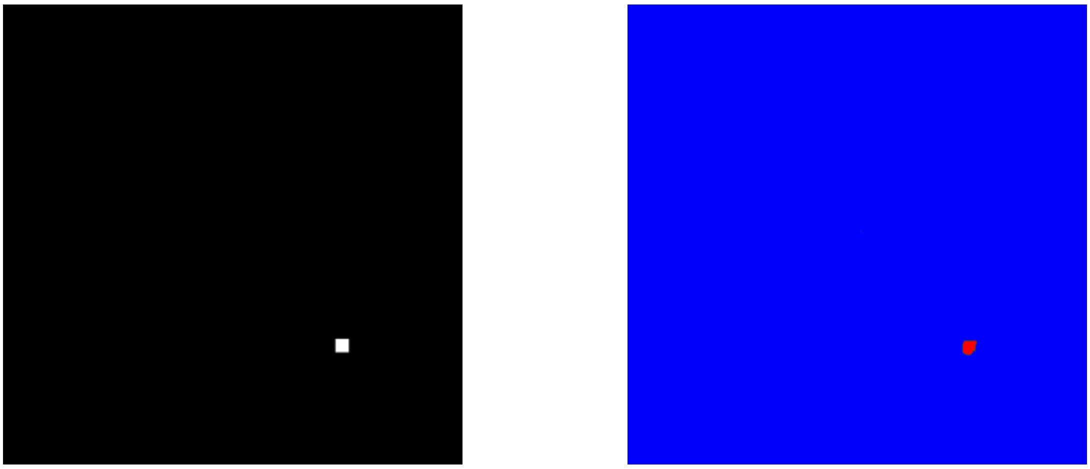


Figure 5.6: Experimental case of single delamination arrangement.

Figure 5.7a shows the GT image corresponding to the artificial delamination's location, shape and size. The number of full wavefield frames in this case is 256 frames. Figure 5.7b shows the delamination prediction for the proposed model, and the highest IoU achieved is (0.53) for a window of frames from $(35 - 99)$.

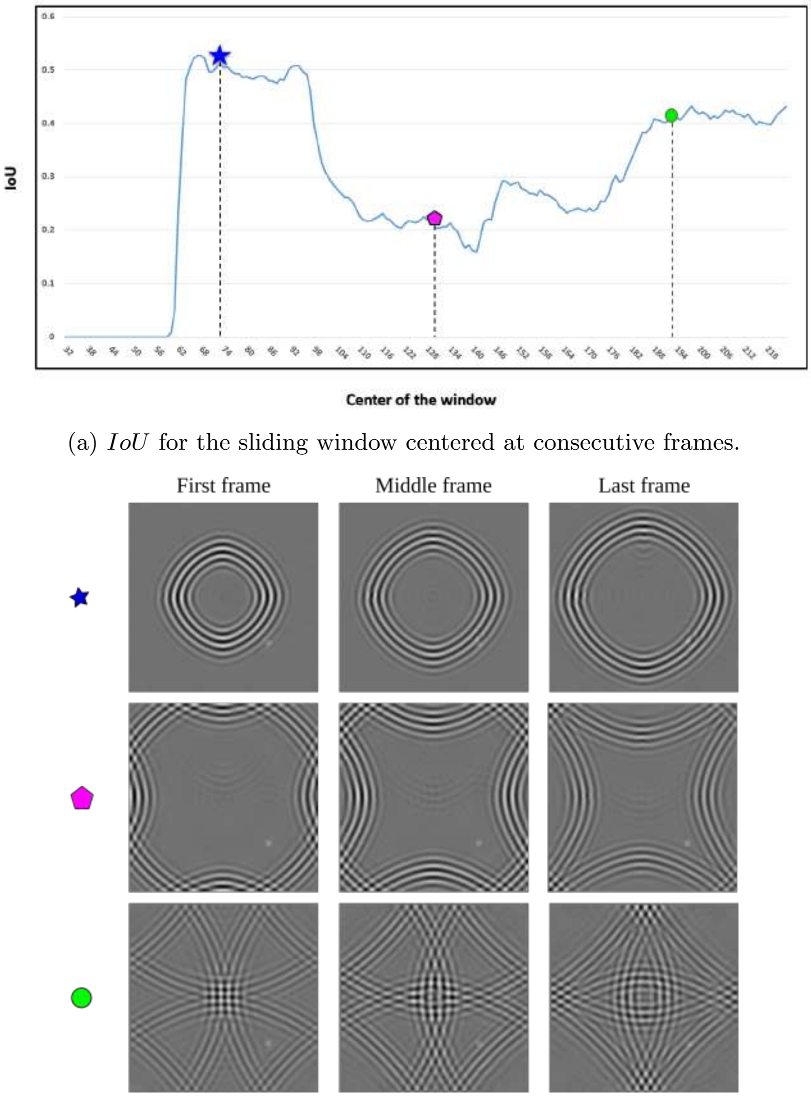
Furthermore, the percentage area error metric (ϵ) was equal to 41.78%. Therefore, the delamination was detected and located, but its size was was not identified correctly, with the mentioned error. The predictions were highest for the group of frames corresponding to the first interaction of the GWs with the delamination. Accordingly, such frames contain the most valuable feature patterns regarding delamination. Furthermore, this behaviour can be depicted in Fig. 5.8, which shows the IoU values with respect to the predicted outputs as I slide the window over all input frames from the starting frame till the end. Since there are 256 frames of full wavefield in this damage case, there are 192 frames of windows and has 192 consecutive predictions. Furthermore, in Fig. 5.8a, three places for the sliding window was selected. The first place depicted in a dark blue star shown in Fig. 5.8b represents a frames of window $(72-136)$, which correspond to the initial interaction of GWs with the delamination. The second place, depicted in the pink pentagon shape in Fig. 5.8b, represents a frames of window $(129-193)$ that correspond to the GWs reflected from the edges, where a drop in the IoU values can be noticed, as these frames have fewer damage features. The third place, depicted by a green circle in Fig. 5.8b represents a frames of window $(192-256)$ corresponding to the interaction of the GWs reflected from the edges with the delamination. As it can be seen, the value of IoU increases again as the valuable feature patterns regarding delamination start to appear. The predicted outputs of the proposed model regarding the dark blue star, pink pentagon, and the green circle are shown in Fig. 5.9.


(a) GT of Teflon insert

(b) IoU = 0.53

Figure 5.7: Experimental case: single delamination of Teflon insert. (a) GT, (b) predictions of the proposed model.

Additionally, for the experimental cases, I applied the root mean square (RMS) according to Eq. 5.4 for all N predicted outputs $\hat{Y}$ regarding all frames of windows to



(a) IoU for the sliding window centered at consecutive frames.

(b) Corresponding frames of guided waves.

Figure 5.8: IoU corresponding to a sliding window of frames (Teflon insert-single delamination).

create the damage map.

$$RMS = \sqrt{\frac{1}{N}\sum_{k=1}^{N}\hat{Y}^2}, \tag{5.4}$$

to separate damaged and undamaged classes from the RMS images, a binary threshold with a value 0.5 is applied, as shown in Fig. 5.10a. The threshold level was selected to limit the influence of noise while highlighting the damage. Figure 5.10b displays the RMS image for the experimental case of a single delamination predicted by the proposed model. The calculated IoU value for the case of the single delamination was (0.46).

Table 5.3 presents the evaluation metric for the proposed model, regarding the experimental case of single delamination shown in Fig. 5.10. As shown in Table 5.3, the

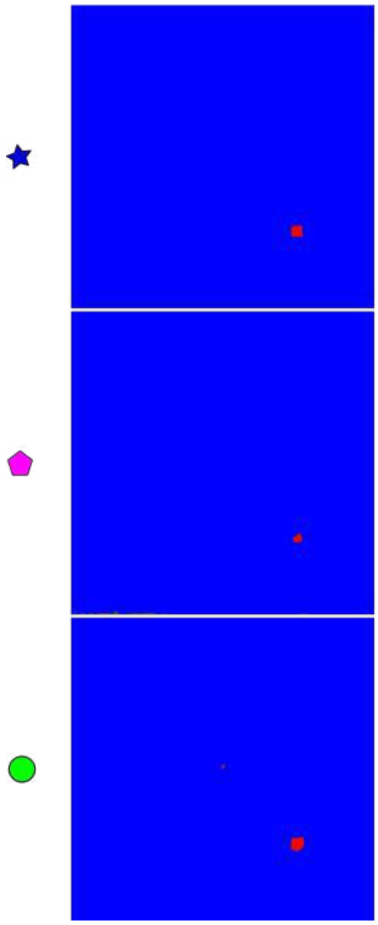


Figure 5.9: Predictions of the proposed model at different window places (Teflon insert-single delamination).

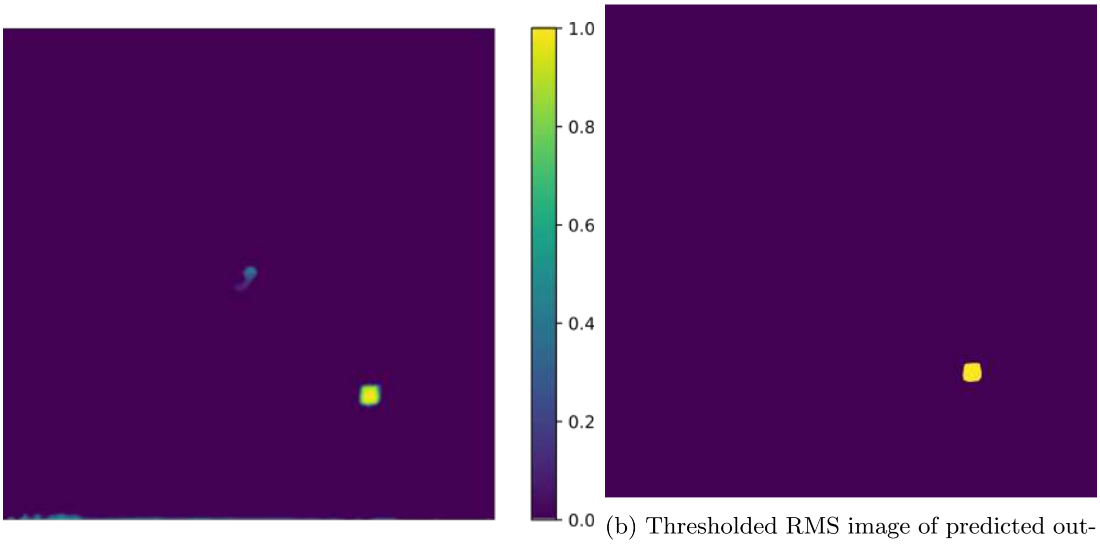
actual (A) and predicted areas ($\hat{A}$) of delaminations were computed in [mm^2] with respect to each case, and the percentage area error (ϵ) was calculated.

Table 5.3: Evaluation metric for experimental case of single delamination.

Experimental case	A [mm^2]	IoU	$\hat{A}$ [mm^2]	ϵ
Single delamination	255	0.46	319	41.78%

5.5.4 Multiple delaminations

In the second experimental case, I investigated three specimens of carbon/epoxy laminate reinforced by stacking sequence of 16 layers of plain weave fabric as shown in Fig. 5.11. Teflon inserts with a thickness of 250 μm were used to simulate the delaminations. The prepregs GG 205 P (fibres Toray FT 300–3K 200 tex) by G. Angeloni and epoxy resin IMP503Z-HT by Impregnatex Compositi were used for the fabrication of the specimen in the autoclave. The average thickness of the specimen was 3.9 mm.



(a) RMS image of predicted output

(b) Thresholded RMS image of predicted output (IoU = 0.46)

Figure 5.10: RMS and thresholded RMS images of predicted outputs; Teflon insert (single delamination).

In Specimen II, three large artificial delaminations of elliptic shape were inserted in the upper thickness quarter of the plate between the 4^{th} and the 5^{th} layer. The delaminations were located at the same distance, equal to 150 mm from the centre of the plate. For Specimen III, delaminations were inserted in the neutral plane of the plate between 8^{th} layer and 9^{th} layer. For Specimen IV, three small delaminations were inserted in the middle of the thickness of the plate, and three large delaminations were inserted at the lower quarter of the thickness of the plate between the 12^{th} layer and 13^{th} layer. The details of Specimen II, III and IV are presented in Fig. 5.11.

Furthermore, the SLDV measurements were conducted from the bottom surface of the plate. Consequently, Specimen II is the most difficult case, as the wave reflections from delaminations were barely visible. For Specimens (II, III, and IV), 512 consecutive frames were generated representing the full wavefield measurements in the plate. The measurement parameters were the same as in the experiment with a single delamination.

Figure 5.12a shows the GT image of Specimen II. The predicted output of the proposed model is shown in Fig. 5.12b in which the highest calculated IoU value of 0.15 is achieved for the frames of window $(167 - 231)$. Figure 5.12c shows the GT image of Specimen III. The predicted output is shown in Fig. 5.12d in which the highest calculated IoU value of 0.18 is achieved for frames of window $(279 - 343)$.

Figure 5.12e shows the GT image of Specimen IV. This is assumed to be the largest delaminations in the cross-sections because the full wavefield was acquired from the bottom surface of the specimen. It is also to be noted that such a case with stacked delam-

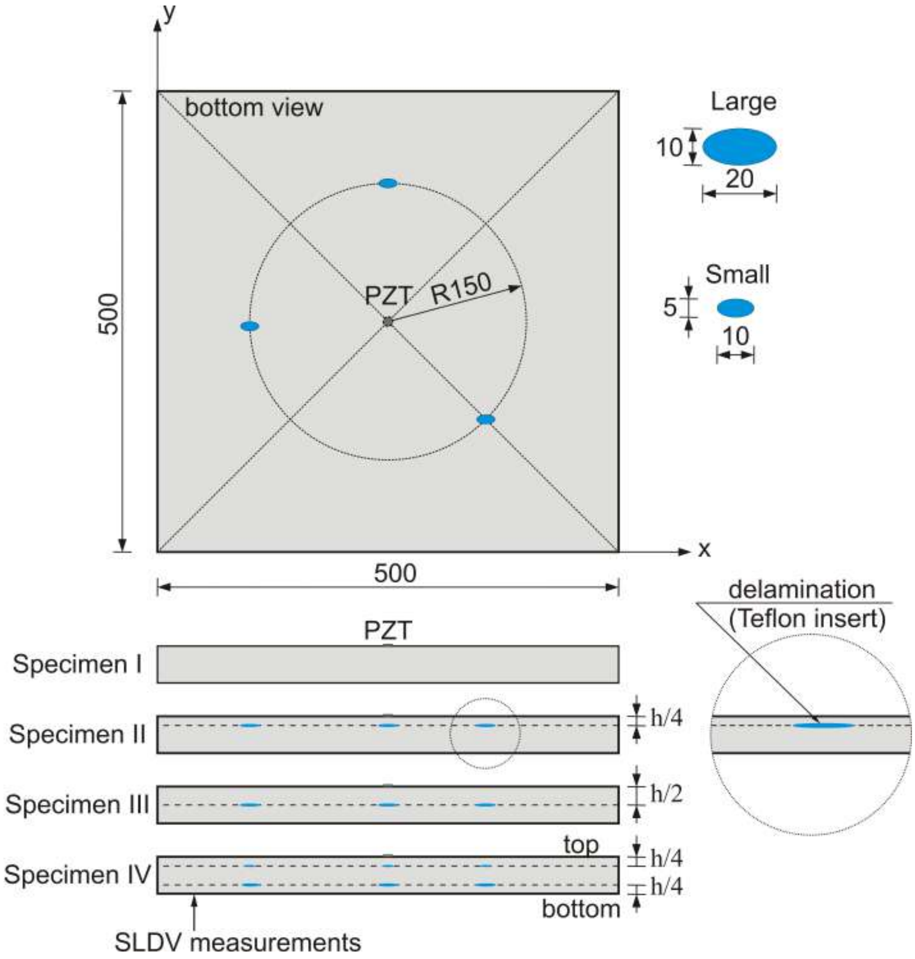
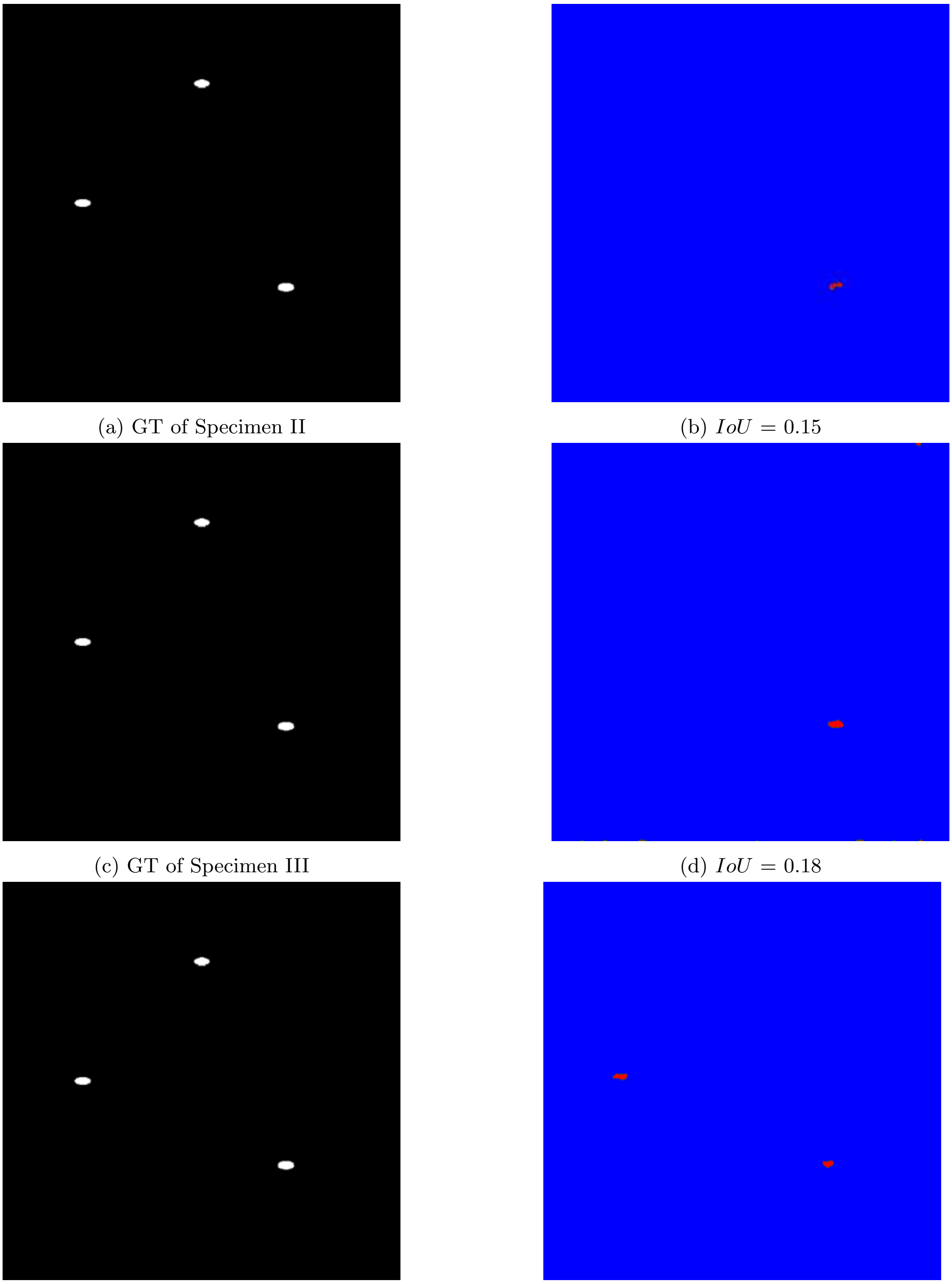


Figure 5.11: Experimental case of delamination arrangement.

inations in cross-sections was not modeled numerically (see Specimen IV in Fig. 5.11). Although the model was not trained on such a scenario, the predictions were satisfactory. The predicted output of the proposed model is illustrated in Fig. 5.12f in which the highest calculated IoU value of 0.18 achieved for frames of window $(235 - 299)$.



(a) GT of Specimen II

(b) IoU = 0.15

(c) GT of Specimen III

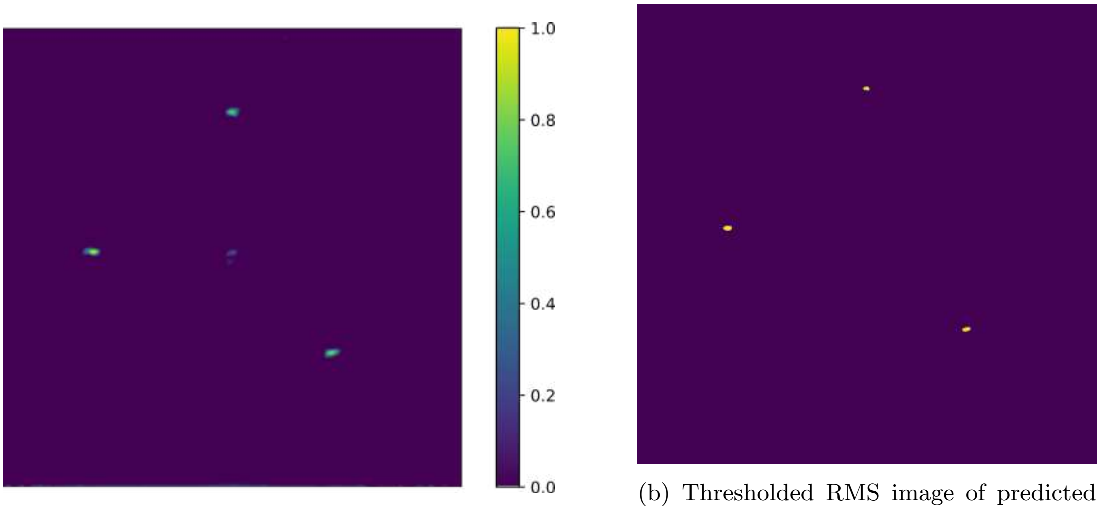
(d) IoU = 0.18

(e) GT of Specimen IV

(f) IoU = 0.18

Figure 5.12: Experimental cases of Specimens II, III, and IV. (Figures: (a), (c), and (e) correspond to the GT of each Specimen. Figures: (b), (d) and (f) correspond to the predictions of the proposed model).

The RMS images depicting the damage maps of Specimen IV are shown in Fig. 5.13a for the proposed model. Figure 5.13b shows the thresholded RMS image for the proposed model, and the calculated value of IoU is (0.07). Furthermore, the mean percentage area error (ϵ) with respect to the three delaminations (Specimen IV) for the proposed model was equal to 79.41%.



(a) RMS image of the predicted output

(b) Thresholded RMS image of predicted output (IoU = 0.07)

Figure 5.13: RMS and thresholded RMS images of predicted outputs - Specimen IV.

5.6 Conclusions

In this chapter, image segmentation and its applications in DL is briefly described. Then the ConvLSTM architecture is elaborated. After that a novel DL pixel-wise semantic segmentation-based methodology for delamination identification in composite laminates was presented. This approach implements an end-to-end framework that specializes in many-to-one sequence prediction to discern the location, size, and shape of delaminations.

The proposed model is systematically trained on consecutive frames depicting the full wavefield of Lamb wave propagation within CFRP plates. These frames capture the interactions of Lamb waves with both delaminations and plate edges. The proposed model is able to extract crucial damage-related features from these frames, enabling accurate predictions.

To evaluate the performance of the proposed model, it was tested on a numerical test set that had never been seen before. The results substantiated remarkable accuracy in identifying delaminations. Furthermore, to evaluate its generalization capabilities, the model was applied to various experimentally measured scenarios involving single and multiple delaminations simulated by Teflon inserts. The predictions of the model in these cases were highly promising, especially in the challenging scenario of multiple delaminations, despite having been trained solely on single delamination cases. This underlines the potential of the proposed model for accurately identifying multiple delaminations in real-world applications.

A major limitation pertains to the developed DL model is that the material properties of the inspected structure have to be approximately known to simulate the dataset for training.

Other limitations are associated with the SLDV measurement technique, which is employed for full wavefield acquisition. SLDV measurements are stationary and time-intensive, making them more suitable for non-destructive testing (NDT) rather than structural health monitoring (SHM). Additionally, SLDV measurements necessitate direct access to the inspected structure's surface, which could require partial disassembly of the structure. Nevertheless, advancements in laser technology may lead to faster data acquisition in the future, possibly by scanning an array of points instead of a single point or by reducing the number of points in the spatial grid while employing compressive sensing techniques.

Chapter 6

DL-based super-resolution approach for the reconstruction of full wavefields of Lamb waves

6.1 Introduction

Recently, the use of SLDV in GW-based NDT/SHM has received much attention, as elaborated in chapter 2 and section 3.8. The utilization of SLDV in GW-based NDT/SHM typically involves the use of either a piezoelectric transducer or a pulse laser for GW excitation, and SLDV for collecting measurements at various points on the surface of the inspected structure in a scanning manner. While full wavefield measurements obtained through SLDV offer valuable data, they are more time-consuming than sparse measurements taken by an array of transducers. Consequently, they are less suitable for continuous monitoring in SHM and are better suited for offline NDT applications. In future, it can be anticipated that the development of laser arrays that replace the single laser head used today will be able to reduce SLDV measurement time. Further advancement in SLDV technology could enable SLDV measurements to be collected on a low-resolution grid of points, and then the low-resolution full wavefield data can be reconstructed to high-resolution.

Generally, two techniques are commonly used to reconstruct a high-resolution wavefield from SLDV measurements taken on a low-resolution grid. These techniques are compressive sensing (CS), and deep learning super resolution (DLSR).

CS was initially introduced in the field of statistics [206, 207], and is employed for efficient signal and image acquisition, and reconstruction. It operates on the fundamental assumption that a signal or image can be represented sparsely in another domain using suitable bases like Fourier, wavelet, or cosine. In such bases, many coefficients are close to or equal to zero, enabling the signal or image to be reconstructed from fewer samples than required by the Nyquist-Shannon sampling theorem. However, there is no unique solution for estimating unmeasured data. Therefore, optimization methods that promote sparsity are employed to solve under-determined systems of linear equations [208–210]. Additionally, a suitable sampling strategy is required. Due to its ability of efficiently reconstruct signals and images from limited data, CS has widely been applied in diverse fields like medical imaging, communication systems, seismology, ultrasonic signal processing, and many more [211–217].

Mesnil and Ruzzene [214] focused on the reconstruction of a wavefield that includes the interaction between Lamb waves and delamination. Their approach involved generating a compressive sensing matrix using analytical solutions. Nonetheless, this method necessitated prior knowledge of the dispersion curves of Lamb waves propagating within the analyzed plate. Perelli et al. [215] employed the warped frequency transform within a compressive sensing framework to enhance damage localization. They employed both wavelet transform and frequency warping techniques to create a sparse decomposition of the dispersive signal. Di Ianni et al. [216] investigated various bases within the realm of compressive sensing with the aim of minimizing the acquisition time required for SLDV measurements. Similarly, in [217], a damage detection and localization method based on compressive sensing was introduced. The authors provided evidence that substantial reductions in acquisition time were achievable without compromising detection precision.

DLSR techniques are primarily utilized in the domain of images and videos, finding significant applications in fields such as astronomy, medicine, satellite imaging, surveillance, and security [218–220]. Advanced DL methods, including improved convolutional neural networks and extensions of PixelCNN to generative adversarial networks, have been extensively employed for tasks like image super-resolution (SR), and video SR. However, these methods have not been widely explored for wavefield data depicting propagating Lamb waves, except for certain instances where wavefield enhancement served as a secondary step in SR, following traditional compressive sensing (CS) methods [221, 222].

This research work introduces a novel technique for reconstructing the full wavefield of Lamb waves using spatially sparse SLDV measurements with resolutions lower than the Nyquist wavelength (λ_N). The Nyquist wavelength represents the shortest spatial wavelength accurately recoverable from a wavefield by acquiring sequential observations spaced at intervals of Δx, where $\lambda_N = 2\Delta x$. The proposed approach provides an end-to-end solution to the super-resolution (SR) challenge by employing a deep neural network trained and evaluated on a synthetic dataset obtained through SLDV. In contrast to other methods described in the literature that rely on CS theory [217] or combine CS theory with super-resolution convolutional neural networks to enhance wavefield images [221, 222], the proposed approach exclusively utilizes DLSR.

This chapter demonstrates the efficacy of the proposed framework and compares it with the traditional CS approach. In the next section, the basic methodology of the proposed approach and the low-resolution dataset acquisition for this task is elaborated.

6.2 Methodology

The approach outlined in this chapter comprises three main stages: dataset preparation (modelling), training a deep neural network model, and prediction, as depicted in Fig. 6.1. In the first stage, the time-domain spectral element method is employed to simulate wave propagation within a plate containing a single delamination. This simulation is repeated across various random sizes, shapes, and locations of delaminations, resulting in high-resolution (HR) animations depicting wave propagation for each scenario. In the subsequent step, frames with delamination interactions are selected and down-sampled to create low-resolution (LR) frames. In the second stage, the DLSR model is trained to learn the mapping from LR frames to HR frames. Finally, in the last stage, the model takes LR measurements acquired via SLDV as input and generates HR frames as output. This process can be iterated for all available frames.

6.2.1 Low-resolution Dataset Acquisition

The original high-resolution dataset, elaborated in chapter 4 was resized to achieve a frame shape of (512×512) pixels, aligning with the desired output dimensions for low-to-high-resolution image reconstruction. Following the Nyquist theorem, which imposed a maximum allowable distance between grid points of 19.5 mm, determined that in 2D space, 73 Nyquist sampling points were needed along the edges of a (500 × 500) mm^2 plate.

To construct the low-resolution training set, frames with a size of (32 × 32) pixels were generated by subsampling the high-resolution dataset. This subsampling was carried out using bi-cubic interpolation, resulting in a compression rate of 19.2%. This rate corresponds to a frame size that falls below the Nyquist sampling rate for a 2D frame. More details regarding the calculation of Nyquist sampling points, and the application of compressive sensing theory can be found in [223]. It is essential to emphasize that the subsampling was performed solely in the spatial domain, as the time domain sampling frequency remains unaltered in SLDV measurements.

Figure 6.2 illustrates three low-resolution frames along with their corresponding high-

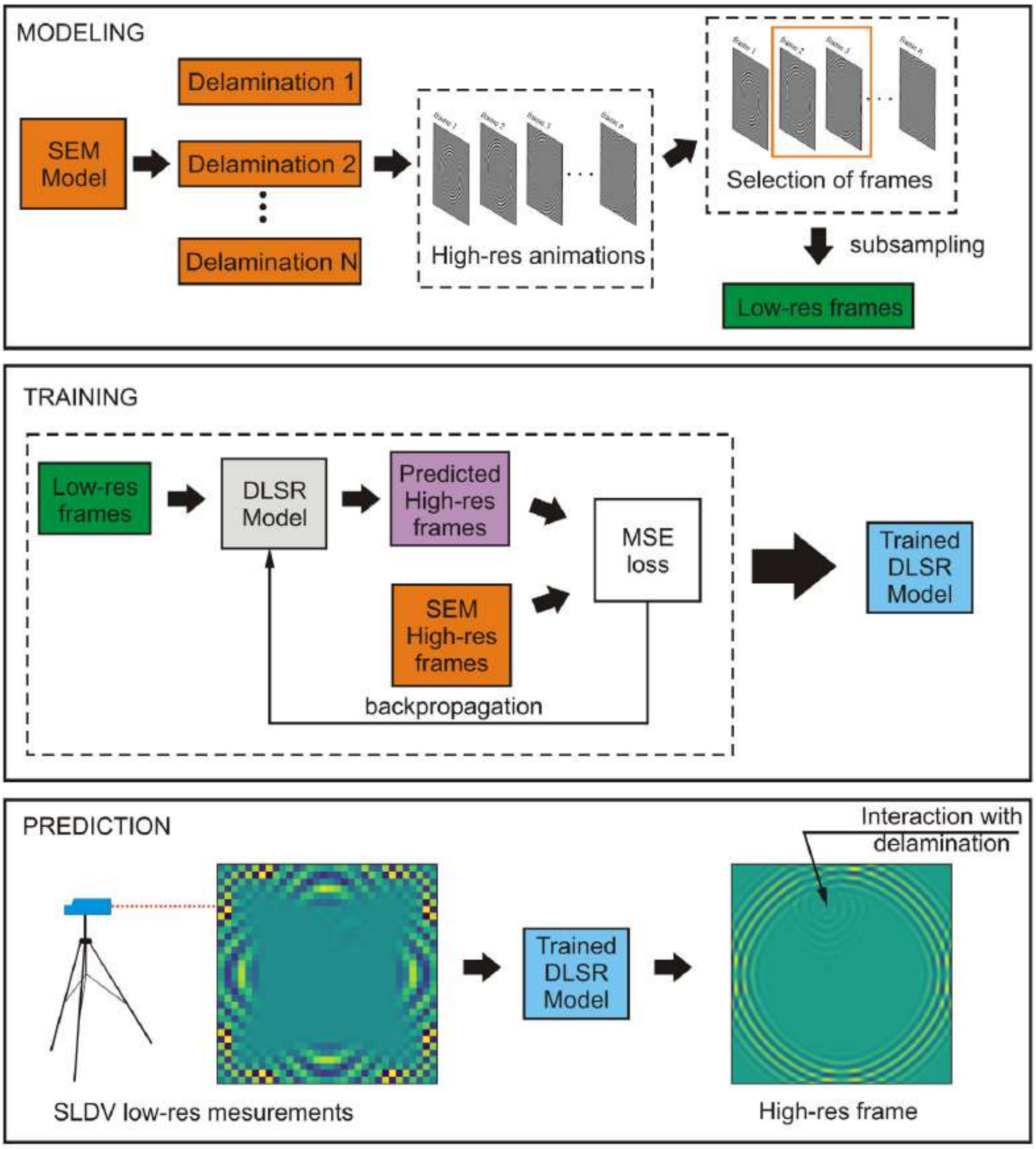


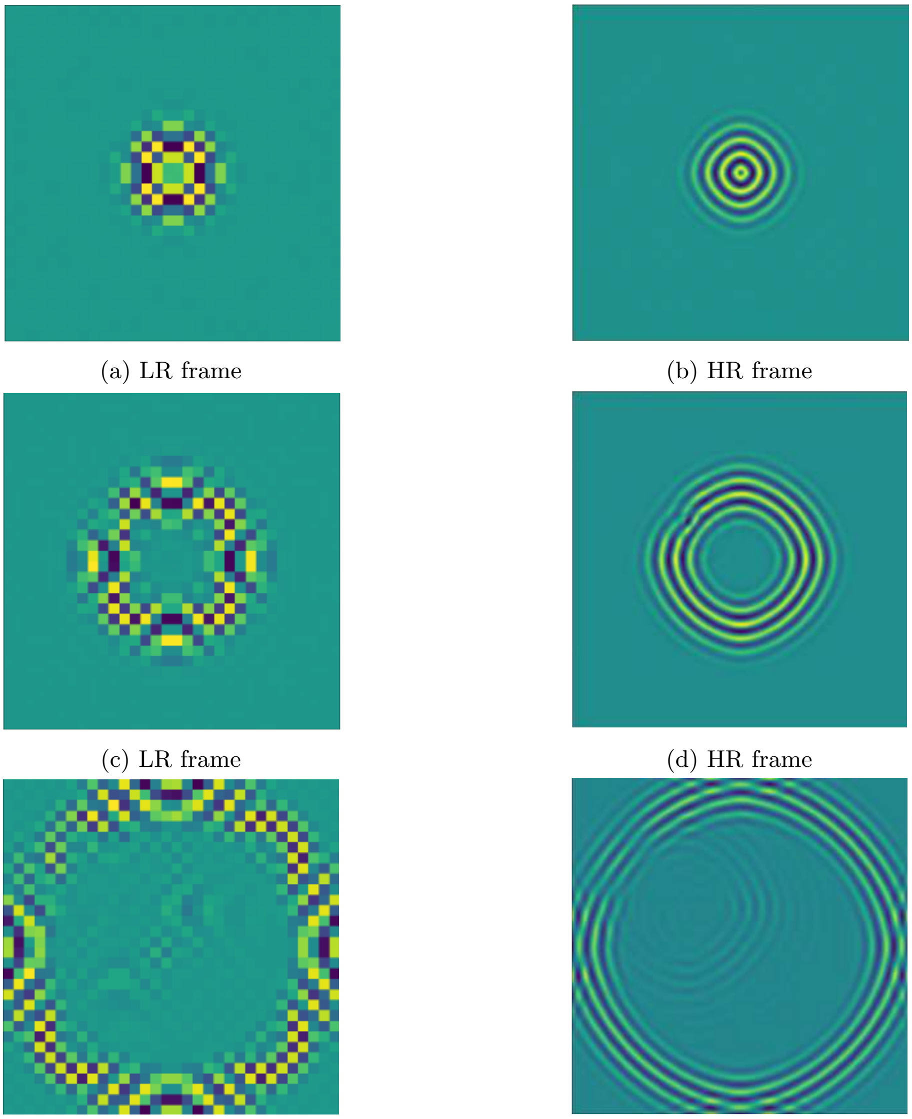
Figure 6.1: Flowchart of the proposed DLSR approach.

resolution counterparts at frame numbers 54, 86, and 150.

Furthermore, the SPGL1 compressive sensing algorithm [210], and Fourier basis were employed for the comparison purpose with the predictions of the proposed DLSR model.

6.3 DLSR approach for image super-resolution reconstruction

For the purpose of reconstruction of full wavefields of Lamb waves, I developed a comprehensive DL-based single image super-resolution (SISR) model.



(a) LR frame

(b) HR frame

(c) LR frame

(d) HR frame

(e) LR frame

(f) HR frame

Figure 6.2: Low-resolution and High-resolution frames at different time steps: (a) and (b) at $N_f = 54$, (c) and (d) at $N_f = 86$, and (e) and (f) at $N_f = 150$.

The goal of SISR is to produce a high-resolution (HR) image that is visually appealing, using its degraded low-resolution (LR) measurement. It is worth noting that I selected 128 consecutive frames for each simulated case, commencing from the initial interaction between the propagating wave and the delamination, i.e. the size of f_w is set into 128 here. This approach helped reduce computational complexity during training. Further, the dataset splitting is being performed according to section 4.3.

6.3.1 The proposed DLSR model

Figure 6.3 illustrates the architecture of the proposed DLSR model. It begins with an initial input in low-resolution. Subsequently, a convolutional layer is employed to extract shallow-level features from the original LR input. Following this, it uses a Residual Dense Network (RDN), which was specifically designed as a solution for SISR challenges, as discussed in [219]. RDN aims to address the lack of utilization of hierarchical features that are extracted from the original LR images. To tackle this issue, the authors introduced the concept of Residual Dense Blocks (RDBs), which effectively utilize all hierarchical features derived from various convolutional layers.

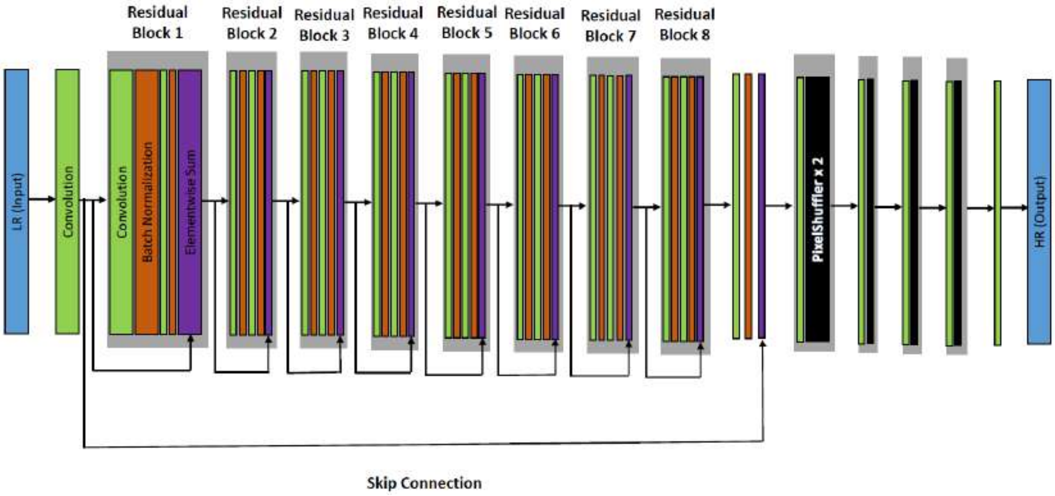


Figure 6.3: The architecture of the proposed DLSR model.

As shown in Fig. 6.3, 8 residual dense blocks right after the first convolutional layer are applied. In each of the RDBs, two convolution layers followed by batch normalizations are applied and at the end, in each RDB an elementwise summation is applied. After the RDBs, a convolution layer, batch normalization, and an elementwise summation are applied consequently. A skip connection is also used before the RDBs to the elementwise summation and after the RDBs. Once the features are learned by the model in lower resolution form, now it was needed to Up-Sample the input from the low resolution (32 × 32) to high resolution (512 × 512) frames, I applied four Up-Sampling Net (UPNet) layers. Each UPNet layer is composed of a convolution layer followed by a pixel shuffle layer [224]. The pixel shuffle technique involves a sub-pixel convolution operation that reshapes the input tensor by rearranging the elements from $(h \times w \times c.r^2)$ to $(rh \times rw \times c)$, where h represents the height, w is width, c is the total number of channels, and r is the up-scaling factor. The dimension at $(c.r^2)$ represents the number of channels (or feature maps) in the output HR image. It is r^2 times larger than the number of channels in the LR image. Here, the up-scaling factor is set to 2, which means that the size of input image is doubled at each pixel shuffle layer. The sub-pixel convolution layer comprises of a general convolutional operation and pixel rearrangement, and combines each pixel on multiple-channel feature maps into one

$(r \times r)$ square area in the output image. This implies that each pixel on feature maps is equal to the sub-pixel on the generated output image. Finally, the last convolutional layer has one filter of size (1×1) which produces one output channel, resulting in an HR grey image. It should also be noted that at each of the convolutional layer, a nonlinear activation function rectified linear unit (ReLU) is applied except the last convolutional layer on which a sigmoid activation function is applied.

6.3.2 The Development Environment

For the development of the proposed DLSR model, Python Anaconda Jupyter Notebook, with Python version 3.7 was used. Further, the Keras API [204] version (2.0.0) running on top of TensorFlow version (2.4.0) was utilized. The GPU used for this task was GeForce RTX 2080 Ti composed of 16 GB of memory.

6.3.3 Evaluation Metrics

For evaluating the performance of the proposed model, two evaluation metrics, peak signal-to-noise ratio (PSNR), and Pearson correlation coefficient (Pearson CC) were utilized. The PSNR quantifies the maximum potential power of a signal and the power of the noise that affects the quality of its representation and is expressed mathematically in Eq. 6.1:

$$\mathrm{PSNR} = 20 \log_{10} \frac{L}{\sqrt{\mathrm{MSE}}}, \tag{6.1}$$

where L denotes the highest degree of variation present in the input image. The MSE used in Eq. 6.1 stands for mean squared error, which represents the discrepancy between the predicted output and the relevant ground truth. The calculation of the MSE is shown in Eq. 6.2:

$$MSE = \frac{1}{m * n} \sum_{m,n} \left(Y_{(m,n)} - \hat{Y}_{(m,n)} \right)^2, \tag{6.2}$$

where m and n represent the number of rows and columns in the input images, $Y_{(m,n)}$ is the ground truth value, and $\hat{Y}_{(m,n)}$ is the predicted value.

Pearson CC is a metric that calculates the linear relationship between two sets or of variables, x (which represents the ground truth values) and y (which represents the predicted values). The mathematical formula for computing Pearson CC is shown in Eq. 6.3:

$$r_{xy} = \frac{\sum_{k=1}^{n} (x_k - \bar{x})(y_k - \bar{y})}{\sqrt{\sum_{k=1}^{n} (x_k - \bar{x})^2} \sqrt{\sum_{k=1}^{n} (y_k - \bar{y})^2}}, \tag{6.3}$$

where r_{xy} represents the Pearson CC, n represents the number of data points in a sample, and x_k and y_k denote the values corresponding to the ground truth and predicted values, respectively, for each data point. Moreover, $\bar{x}$ denotes the mean value of the sample, while $\bar{y}$ represents the mean value of the predicted values. The values of r_{xy} fall within the range of -1 to $+1$. A value of 0 indicates that there is no relation between the samples and the predicted values. A value greater than 0 indicates that there is a positive relationship between the samples and the predicted data, whereas, a value less than 0 represents a negative relationship between them.

6.3.4 Hyperparameters tuning and DL model training

To enhance the PSNR value and minimize the loss of the proposed DLSR model, the Adam optimizer [205] was employed. The detailed training configurations and values of the hyperparameters are provided in Table 6.1.

Table 6.1: Configuration of the training process of the proposed DLSR model.

Hyperparameter	Value
Loss function	MSE (L2 norm)
Learning rate	3e-4
Batch size	4
Number of epochs	131
Early stopping	30 epochs

Since there is no simple approach to select the best parameters for training a DL model, a trial-and-error approach was employed. The value of the regularization technique, early stopping was set to 30 epochs, and was evaluated on validation PSNR, which means during the DL model training process, if the value of PSNR on validation data is not increasing for consecutive 30 epochs, the model will stop training, and will save the model with the highest value of PSNR on the validation data. The training process of the proposed DLSR model is shown in Fig. 6.4.

During the training process of the proposed model, it achieved the highest PSNR values at 101^{th} epoch on training and validation data as 48 dB and 47.7 dB, respectively. The Pearson CC values of the proposed model at the same epoch on training and the validation data were noted as 0.999, and 0.998, respectively.

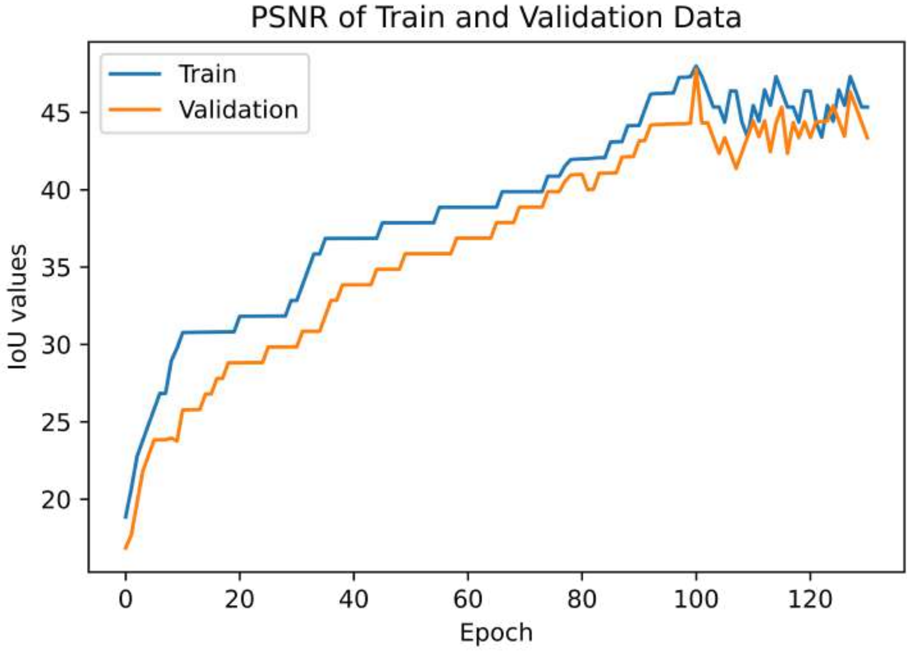


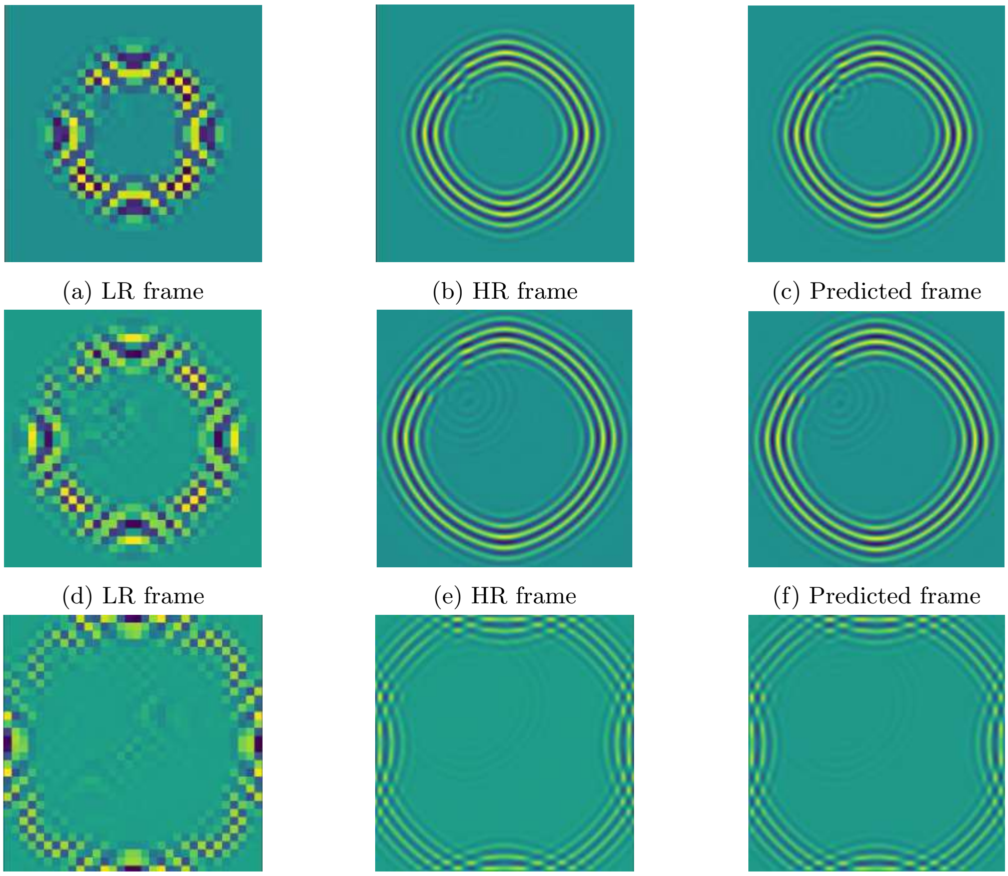
Figure 6.4: Illustration of the proposed DLSR model training.

6.4 Results and discussions

In this section, the retrieval of high-resolution frames from low-resolution ones using numerical test data from 95 different cases are demonstrated. Further, the predictions of the DLSR and CS approach are also compared.

6.4.1 Evaluation of DLSR model on test data

The mean PSNR value was 45.5 dB, and the mean Pearson CC value was 0.991 on all of the test data. Figure 6.5 illustrates the results of the reconstruction of high-resolution (HR) frames for a numerical test case using the developed DLSR model. It represents the recovery of three different HR frames at different time steps with frame numbers $N_f = 111, 133,$ and 168, respectively, from the test data, which were not shown to the model during training. Figures 6.5a, 6.5d, and 6.5g depict the input to the DLSR model in the LR form. Figures 6.5b, 6.5e, and 6.5h show the reference HR frame, which serve as the ground truth for the predicted frames shown in Fig. 6.5c, 6.5f, and 6.5i, respectively. The PSNR value for the predicted frame in Fig. 6.5c is 47.5 dB, while for Fig. 6.5f, the PSNR value is 46.8 dB, and for Fig. 6.5i, the PSNR value is 47.1 dB. The Pearson CC values for all of the mentioned frames are 0.998, 0.996, and 0.998, respectively.



(a) LR frame

(b) HR frame

(c) Predicted frame

(d) LR frame

(e) HR frame

(f) Predicted frame

(g) LR frame

(h) HR frame

(i) Predicted frame

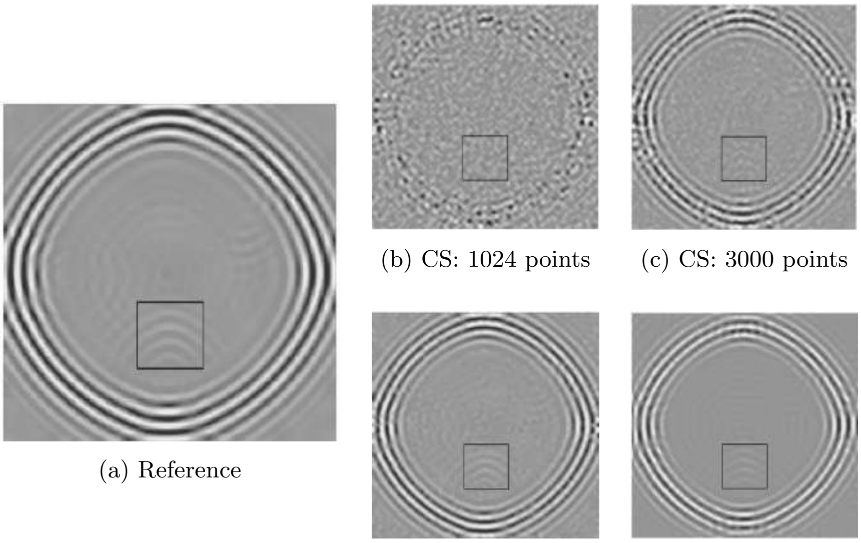
Figure 6.5: Reconstruction of three different frames with DLSR model for frames $N_f = 111, 133,$ and 168, respectively.

6.4.2 Comparison of DLSR model with compressive sensing

A traditional compressive sensing (CS) approach was employed as a reference to reconstruct the wavefield, using different numbers of points. The CS approach was applied with varying points along with the reference frame and the prediction from the DLSR model, as shown in Fig. 6.6.

In particular, the comparison between the reference wavefield reconstructed using both the CS and DLSR model for frame number 110 is elaborated. The rectangular box in all of the frames in Fig. 6.6 denotes the region with the most prominent delamination reflection. Figure 6.6a presents the reference frame, which serves as input for both the CS approach and the DLSR model reconstruction process. Figure 6.6b, 6.6c, and 6.6d illustrate the CS approach reconstructions with 1024, 3000, and 4000 points, respectively.

As expected, the accuracy of HR frame recovery diminishes as the number of points decreases in the CS method. Conversely, Fig. 6.5e showcases the reconstructed HR



(a) Reference

(b) CS: 1024 points

(c) CS: 3000 points

(d) CS: 4000 points

(e) DLSR prediction

Figure 6.6: Comparison of reference wavefield with reconstructed one by CS and DLSR model.

frame achieved by the DLSR model. The proposed DLSR model successfully recovers the high-resolution frame using only 1024 points, equivalent to 19.2% of the Nyquist sampling rate. It is crucial to highlight that when the point count is 1024, the CS algorithm struggles in the delamination region, while the DLSR model excels in accurately reconstructing the high-resolution frame.

6.5 Conclusions

In this chapter, a DL approach for the reconstruction of full wavefields of GWs is proposed. The generated large synthetic dataset resembling full wavefields of GWs was used. The process of creating a low-resolution dataset from the high-resolution synthetic dataset is explained. The compression rate used in the DLSR model was 19.2% of the Nyquist interval. The evaluation of the proposed DLSR model on test data is elaborated, and then the DLSR model is compared with the conventional compressive sensing (CS). It was found out that the DL model outperformed CS in reconstructing high-resolution of full wavefield frames from low-resolution data. The DLSR model provides better reconstruction of the full wavefields, especially in areas of delamination identification. Additionally, DLSR approaches can significantly improve the speed of data acquisition by the SLDV.

Chapter 7

DL-based approach for the simulation of full wavefield for delamination identification

7.1 Introduction

One possible solution to reduce the acquisition time of full wavefields of Lamb waves is to obtain Lamb waves in a low-resolution format and subsequently apply compressive sensing (CS), potentially enhanced by deep learning (DL) methods [222], such as the end-to-end DL-based super-resolution technique described in chapter 6.

However, when dealing with scenarios where only spatially sparse data is accessible, for instance when an array of sensors is deployed on the structure, applying damage identification methods originally designed for full wavefields is not straightforward. In such cases, inverse procedures are required, demanding efficient techniques for solving wave equations. Addressing numerical modeling of ultrasonic GW propagation in solid materials with discontinuities like damage is a challenging endeavor, requiring fine discretization and being computationally intensive. Even methods like the p-version of the finite element method (p-FEM)[225], iso-geometric analysis (IGA)[226], spectral cell method (SCM)[227], or the time-domain spectral element method (SEM)[228] eventually prove inadequate in terms of efficiency. Executing the objective function for each damage case scenario utilizing the forward solver becomes unfeasible.

An alternative approach is leveraging a DL-based surrogate model. Such models serves as a means to generate full wavefield data or time series that emulate the signals captured by an array of sensors. Essentially, a surrogate model mimics the behavior of the simulation model, replacing time-consuming forward simulations with approximated solutions.

Typically, when training DL models, streamlined neural network (NN) architectures have been favored, coupled with curated input data. This approach aims to compel the DL model to autonomously acquire relevant features from the input, a process often referred to as feature engineering. However, delving into feature engineering is a meticulous task, demanding domain-specific knowledge and a substantial investment of time. Furthermore, the nuances of feature engineering tend to vary across different data types, complicating the establishment of universally applicable techniques.

An effective alternative to traditional feature extraction lies in the form of autoencoders-neural networks designed to autonomously learn features from unlabeled data through unsupervised learning techniques. This intrinsic capability obviates the need for extensive feature engineering efforts [229, 230]. An autoencoder comprises two key components: an encoder that maps inputs to a desired latent space and a decoder that adeptly translates the latent space back to the original input domain. By leveraging carefully curated training data, autoencoders can generate a latent representation, eliminating the requirement for labor-intensive feature engineering.

It is worth noting that conventional autoencoders may not effectively capture spatial features, such as those found in images, or sequential information when dealing with dynamic data, such as time-series forecasting. To address this limitation in capturing spatial features, CNN-based autoencoders are recommended, while RNN-based autoencoders are typically employed for learning features from time-series data. Deep CNN-based autoencoders (DCAEs), in particular, excel at extracting spatial features from image-based inputs. However, they might fall short when handling sequences of images, especially in scenarios involving full wavefield data with numerous sequential frames or images for each delamination scenario. In such situations, ConvLSTM [195] is employed, for more details about the working procedure and architecture of ConvLSTM (refer to section 5.3).

DCAE-based surrogate modeling approaches have been applied in several studies [231–233]. In the work by Jo et al. [231], a DCAE framework was developed to extract latent features from spatial properties. They employed their framework for adaptive surrogate estimation, specifically focusing on carbon dioxide (CO_2) sequestration in heterogeneous deep saline aquifers. To reduce computational costs and retain spatial characteristics, they combined DCAE with a fully-convolutional network. Nikolopoulos et al. [232] introduced a non-intrusive DL-based surrogate modelling scheme for predictive modelling of complex systems described by parametrized time-dependent partial differential equations. Sharma et al. [233] proposed a DCAE-based surrogate predictive model tailored for wave propagation studies. Their model was designed to generate data for a one-dimensional rod of isotropic material, considering different crack locations and depths.

Zargar and Yuan [234] introduced a hybrid CNN-RNN framework to address challenges related to spatio-temporal information extraction in impact damage detection problems. Recently, Peng et al. [235] proposed an encoding convolution long short-term memory (encoding ConvLSTM) framework. Their approach aimed to create surrogate structural models capable of handling spatio-temporal evolution. It was employed for estimating structural spatio-temporal states and predicting dynamic responses under future dynamic load conditions.

In this study, an innovative approach is employed to investigate the propagation of GWs in composite structures with varying delamination instances. The key element of this approach is the utilization of a deep ConvLSTM autoencoder-based surrogate model. The primary objective of this model is to capture and comprehend the full wavefield representation embedded within frames that depict delamination scenarios, subsequently condensing this information into a compressed domain known as the latent space.

During the training process, the encoder is presented with inputs encompassing both reference frames (without delaminations) and explicit data detailing the characteristics of the delaminations, such as their shape, size, and location. This process eliminates the necessity for repetitive solving of the governing equations of the system, resulting in significant time and computational cost savings compared to conventional forward modeling techniques.

The novelty of this research lies in the implementation of ConvLSTM-based autoencoders for generating full wavefield data related to propagating GWs in composite structures. Additionally, this DL model for full wavefield prediction is applied for the first time in the context of the inverse problem of damage identification. To facilitate this task, particle swarm optimization (PSO) is employed [236].

7.2 Methodology

The framework of the proposed inverse approach for damage identification is illustrated in Fig. 7.1, consisting of three fundamental components: (i) dataset computation, (ii) supervised learning, and (iii) the inverse method.

Acquiring a substantial experimental dataset for GW-based damage identification poses significant challenges due to the need for multiple samples with various damage scenarios, which can be expensive and impractical. To address this, an alternative approach involves generating a numerical dataset using tools like a finite element solver. In this

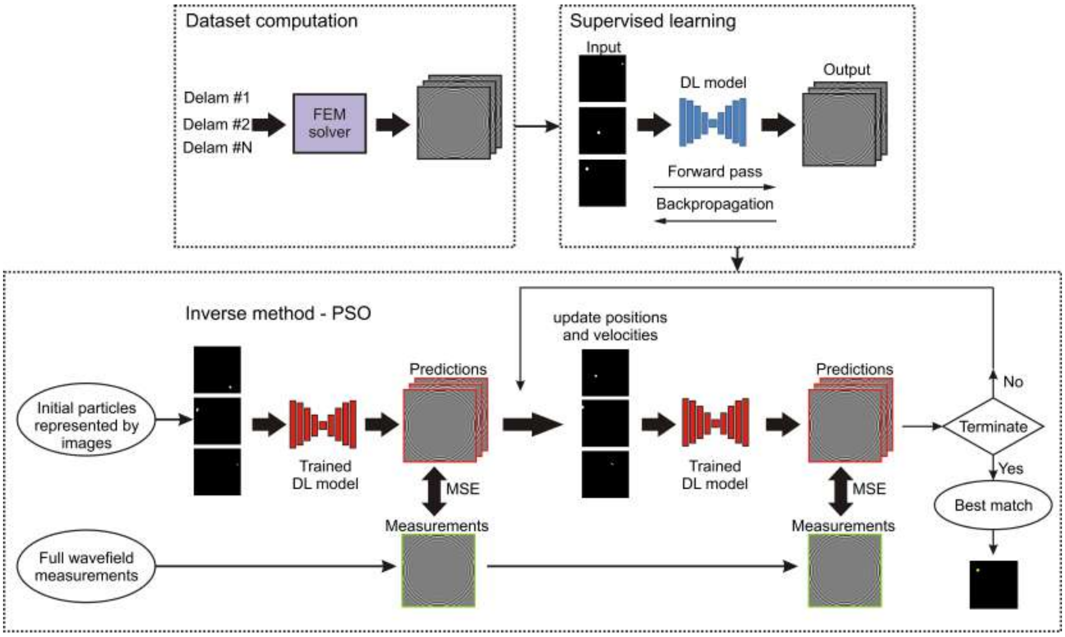


Figure 7.1: Flowchart of the proposed inverse method for damage identification.

research, the dataset, as detailed in chapter 4, was created using the time-domain spectral element method [192], focusing primarily on delamination defects, which are particularly critical in composite laminates like carbon fiber-reinforced polymers (CFRP).

This dataset served as the foundation for training the DL model. The concept involved feeding the DL model with binary images, where white pixels represented the delamination region, and black pixels represented the healthy region for specific delamination cases. Once the model was trained, it could replace computationally intensive forward solvers like p-FEM or SEM in the inverse procedure as an efficient surrogate model.

It is important to highlight that the PSO algorithm was chosen for the inverse method due to its effectiveness in handling objective functions that are more versatile, accommodating various formulations beyond strict algebraic constructs. However, alternative algorithms can also be applied. In this process, the initial particles, represented by binary images in Fig. 7.1, serve as inputs. These are input into the trained DL model, which subsequently predicts the full wavefield of propagating waves for corresponding delamination scenarios. Simultaneously, full wavefield measurements are collected. An objective function is constructed based on the mean square error (MSE) calculation between the predicted and measured wavefields. The positions and velocities of particles are then iteratively updated until the termination criterion is satisfied. Eventually, the identified delamination is visualized based on the best match.

It is also important to emphasize that the proposed method was validated only on synthetic data. The subsequent sections provide a comprehensive breakdown of each constituent element within the proposed method.

7.3 Dataset computation and preprocessing

The spatial size of the full wavefield data, elaborated in chapter 4, was further downsampled to (256×256) for the purpose of reducing the computational complexity. This (256×256) size ensures the preservation of a symmetrical shape throughout the encoding and decoding phases of the model. Further, the data augmentation technique (described in section 3.5.2) is applied to increase the generalization capabilities of the proposed DL model. In this research work, the dataset is composed of 475 delamination cases, which is not enough for a targeted DL model to perform well. Therefore, all the images in the 475 delamination cases are flipped diagonally, horizontally, and vertically in order to enhance the performance of the proposed DL model. Consequently, the total dataset after data augmentation is now composed of 1900 $(475 \times 4 = 1900)$ delamination cases. The input frame size (f_w) for the proposed DL model is chosen to be 32 frames. This less size for the (f_w) was opted due to memory limitations. Furthermore, the dataset splitting is being performed according to section 4.3.

7.4 The proposed DL-based Surrogate Model

In this research work, I developed a novel deep ConvLSTM autoencoder-based surrogate model utilising full wavefield frames of Lamb wave propagation for the purpose of data generation for delamination identification in thin plates made of composite materials. The developed DL model takes as an input 32 frames without delamination (reference frames) representing the full wavefield and the delamination information of the respective delamination case in the form of binary image for the purpose of producing a full wavefield propagation of Lamb waves through space and time (3D matrix). The most important aspect of the DL model is the prediction of the interaction of Lamb waves with the delamination so that the delamination location, shape, and size can be estimated. The complete flowchart of the proposed DL model is presented in Fig. 7.2.

The training and evaluation process of the proposed model can be summarized in the following three steps:

1. **Feature extraction:** Since there are no labels available for the dataset, the available dataset is composed of delamination cases. So the first task was to extract features from all of the delamination cases, and then use these features

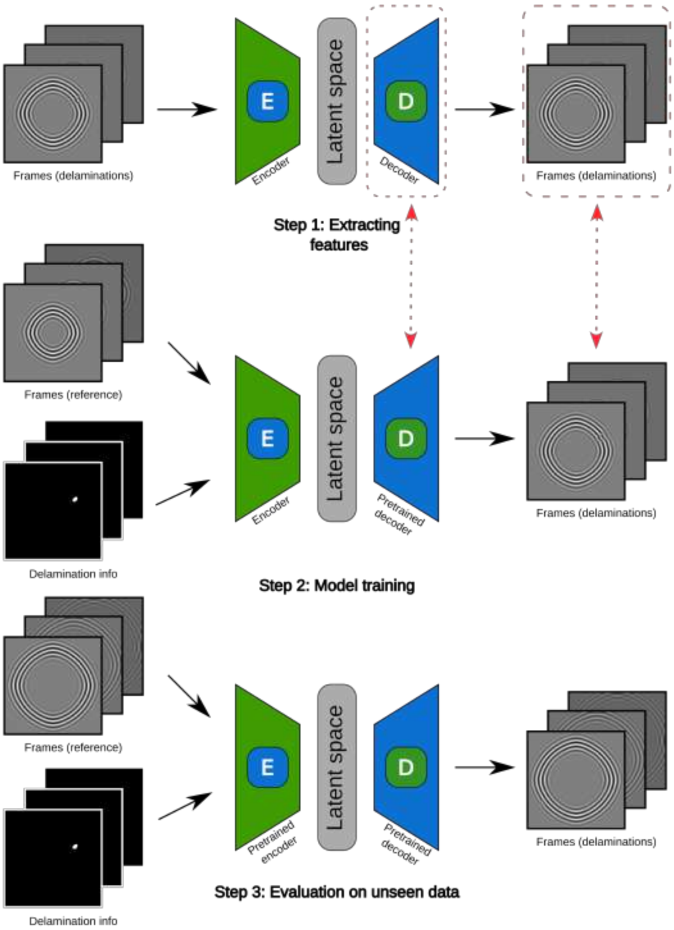


Figure 7.2: The flowchart of the proposed DL model.

as labels in the second step during model training. Therefore, in this step, the encoder and decoder parts of the proposed model are jointly trained, allowing the decoder part to be used separately for full wavefield predictions. During this step, the features are extracted with very minimal reconstruction error in a compressed form, matching the dimensions of the latent space.

2. **Model training:** In this step, the actual model training is carried out. Full wavefield frames in a plate without delamination, along with the binary image of the respective delamination case, are fed into the encoder part of the DL model for training. The features extracted in the first step from encoder part are used as labels in this step, as shown in Fig. 7.2.

3. **Evaluation of the proposed DL model on unseen data:** In this stage, both of the pre-trained models (the pre-trained decoder from step 1 and pre-trained encoder from step 2) are utilised for the prediction of full wavefield frames on unseen data. During this step, the model takes reference frames with delamina-

tion information and produces the output as the full wavefield frames containing interaction of Lamb waves with delamination.

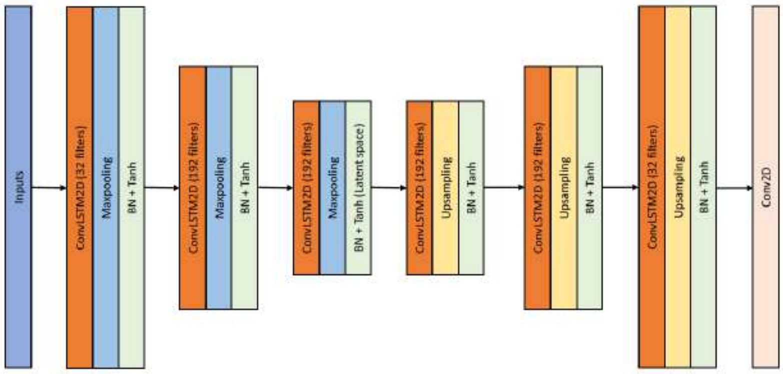


Figure 7.3: The architecture of the proposed ConvLSTM-based surrogate model.

The proposed ConvLSTM autoencoder model takes 32 frames as input, concatenated with a binary image that is replicated 32 times (see Fig. 7.2). The DL model consists of six ConvLSTM layers. The first ConvLSTM layer has 32 filters, the second and third layer has 192 filters, the fourth layer has 32 filters, and the last two ConvLSTM layers has 192 filters. The kernel size of the ConvLSTM layers is set to (3×3) with a stride of (1). Padding is set to "same", which ensures the output size matches the input size in the case of stride 1. Furthermore, a tanh (the hyperbolic tangent) activation function is used within the ConvLSTM layers, which outputs values in a range between (-1 and 1). Maxpooling and upsampling were applied at each ConvLSTM layer to reduce the size of features and for reconstruction purposes, respectively. Moreover, a batch normalization technique is applied at each of the ConvLSTM layers. At the final output layer, a 2D convolutional layer is followed by a sigmoid activation function.

It should also be noted that no skip-connections were used so that it is easier to train the encoder and decoder separately.

7.4.1 The Development Environment

For the development of the proposed DL-based surrogate model, Python Anaconda Jupyter Notebook with Python version 3.8 was used. Additionally, the Keras API [204] version (2.0.5), running on top of TensorFlow version (2.5.0), was utilized. The GPU used for this task was Tesla V100 GPU from NVIDIA, composed of 32 GB of memory.

7.4.2 Evaluation Metrics

For evaluating the performance of the proposed models, two evaluation metrics, peak signal-to-noise ratio (PSNR), and Pearson correlation coefficient (Pearson CC) were utilized. The PSNR quantifies the maximum potential power of a signal and the power of the noise that affects the quality of its representation. Both the PSRN and Pearson CC evaluation metrics are explained in section 6.3.3.

7.4.3 Hyperparameters tuning and DL model training

The Adam optimizerr [205] was employed for back-propagation, and MSE was used as a loss function for both training steps. The detailed training configurations and values of the hyperparameters are provided in Table 7.1.

Table 7.1: Configuration of the training process of the ConvLSTM-based surrogate model.

Hyperparameter	Value
Loss function	MSE (L2 norm)
Learning rate	3e-4
Batch size	2
Number of epochs	142
Early stopping	30 epochs

Since there is no simple approach to select the best parameters for training a DL model, a trial-and-error approach was employed. The value of the regularization technique, early stopping, was set into 30. It was evaluated based on validation PSNR, which means that during the DL model training process, if the PSNR value on the validation data does not increase for consecutive 30 epochs, the model will stop training and save the model with the highest PSNR value on the validation data.

The training process of the proposed ConvLSTM-based surrogate model is illustrated in Fig. 7.4. The model achieved the highest PSRN value at the 112^{th} epoch on training and validation data, reaching 24.6 dB and 23.7 dB, respectively. The Pearson CC values of the proposed model for the same epoch on training and the validation data were recorded as 0.99, and 0.97, respectively.

7.5 Inverse method for damage identification

A global-best Particle Swarm Optimization (PSO) algorithm implemented in Python was used in the optimization process [237]. It takes a set of candidate solutions and aims to find the best solution using a position-velocity update method. It uses a star-topology where each particle is attracted to the best-performing particle. The

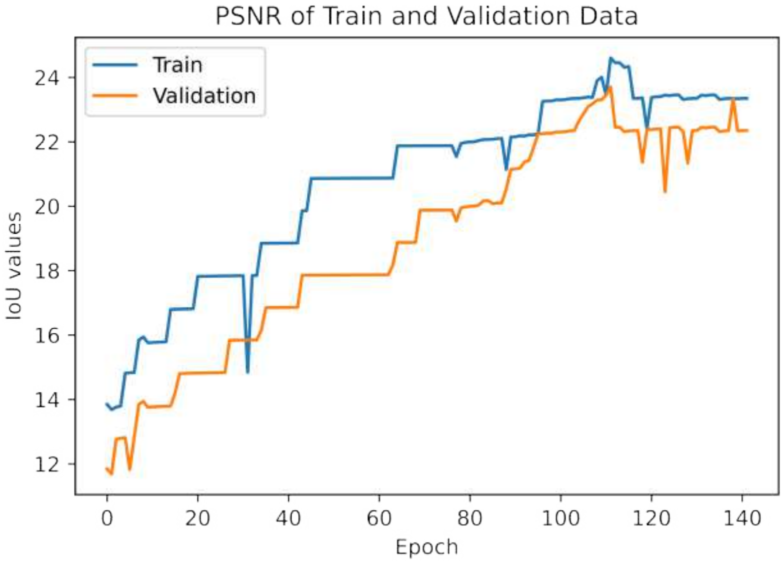


Figure 7.4: Illustration of the proposed ConvLSTM-based surrogate model training.

algorithm follows two basic steps:

- the position update:
$$x_i(t+1) = x_i(t) + v_i(t+1), \tag{7.1}$$
- and the velocity update:
$$v_{ij}(t+1) = w\,v_{ij}(t) + c_1\,r_{1j}(t)\,[y_{ij}(t) - x_{ij}(t)] + c_2\,r_{2j}(t)\,[\hat{y}_j(t) - x_{ij}(t)], \tag{7.2}$$

where r are random numbers, y_{ij} is the particle's best-known position, $\hat{y}_j$ is the swarm's best known position, c_1 is the cognitive parameter, c_2 is the social parameter, and w is the inertia parameter that controls the swarm's movement. Cognitive and social parameters control the particle's behaviour given two choices: (i) to follow its personal best or (ii) to follow the swarm's global best position. Overall, this dictates if the swarm is explorative or exploitative in nature. In this research work, the following parameters were used: $c_1 = c_2 = 0.3$ and $w = 0.8$. Good convergence was achieved for these set of parameters settings, rendering further parameter tuning was unnecessary.

The following decision variables were used in the PSO:

- delamination coordinates (x_c, y_c) with bounds [0 mm, 500 m],
- delamination elliptic shape represented by semi-major and semi-minor axis a, b with bounds [5 mm, 20 mm],
- delamination rotation angle α with bounds [0°, 180°].

Based on decision variables, binary images of (256×256) pixels are generated (one image per particle - see Fig. 7.1). In these images, ones (white pixels) represent delamination, whereas zeros (black pixels) represent healthy area.

The most important component of the proposed inverse method is the surrogate DL model described in section 7.4. The trained DL model is used for ultrafast full wavefield prediction, as illustrated in Fig. 7.1. For a single particle and its respective binary image, the DL model is evaluated 7 times for 32 consecutive frames giving an output of 224 frames. These predicted frames are compared to “measured” frames using the MSE metric, which is utilised in the objective function. However, for the sake of result reproducibility and compatibility with the available dataset, the synthetic data was used instead of measured data (acquired by SLDV).

For each PSO iteration, particles are updated according to Eqs. (7.1)-(7.2). The termination criterion was assumed as 100 iterations but it was observed that the objective function value converges much faster. In the final step, the best matching wavefields indicate coordinates, semi-major, semi-minor axis and the rotation angle of the elliptic-shaped delamination. These parameters are used for a visual representation of the best-matched delamination in the form of binary image, compared against the ground truth (see also Fig. 7.1).

As an evaluation metric for assessing the accuracy of the identified delamination, the Intersection over Union (IoU) was used, which is explained in section 5.4.3.

7.6 Results and discussions

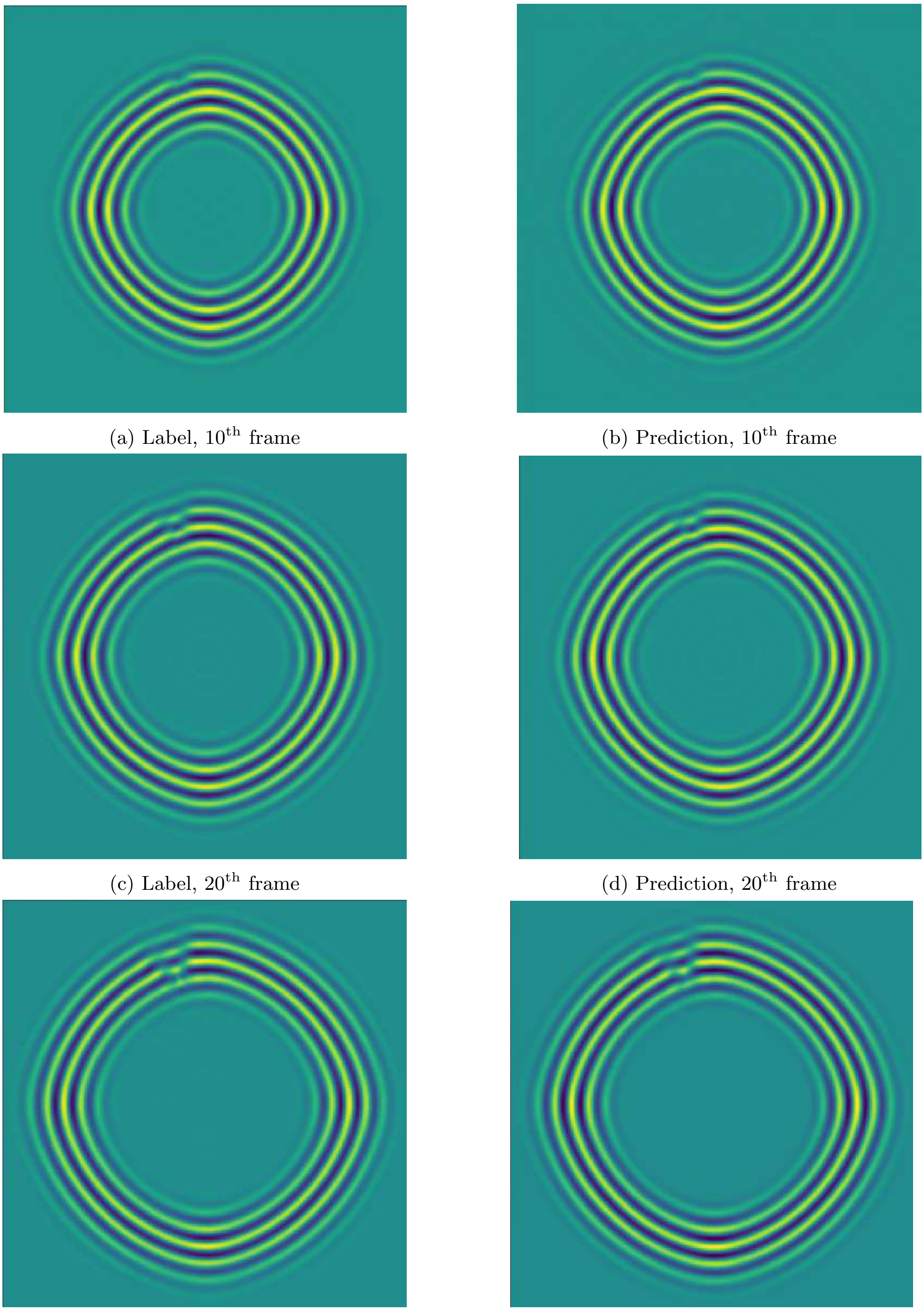
7.6.1 Evaluation of the surrogate DL model

In this section, I present the evaluation of the proposed DL model based on numerical test data of 95 delamination cases representing the frames of full wavefield propagation, which were not shown in the proposed DL model during training. The proposed DL model was evaluated using numerical test data to demonstrate its capability to predict the interaction of Lamb waves with delaminations of various locations, shapes and sizes.

Three different representative cases were selected from the numerical dataset to demonstrate the performance of the developed DL model. Figures 7.5, 7.6, and 7.7 shows three different frames from three selected numerical test cases. Frames in the left column represent the labels against which the prediction of the proposed DL model are compared. Frames on the right column represent the predictions by the DL model. Particular frames were selected to illustrate the interaction of propagating Lamb waves with the delamination, namely 10th, 20th and 30th frames after the interaction with the delamination. These frame numbers were determined based on the A_0 mode velocity and modelled delamination location.

As can be seen in the first scenario (Fig 7.5), the delamination occurred at the upper-left side of the plate. In the second scenario (Fig 7.6), the delamination occurred at the top-left of the plate, while in the third scenario (Fig 7.7) the delamination occurred at the top-centre of the plate.

In all presented cases (Figs. 7.5–7.7), the changes in wave velocity due to delamination are well reproduced by the DL model. The wave reflections from the delamination are very well predicted in Fig. 7.6, whereas in some cases and frames, the reflection patterns differ between the label and the DL prediction. For comparison, see Fig. 7.7e to Fig. 7.7f. It should be noted that these reflections have much smaller amplitudes than the main wavefront, and the proposed DL model is cannot accurately reproduce all the detailed intricacies of Lamb wave reflections. Achieving further improvement might require a more complex DL model. Nevertheless, testing results are satisfactory and very promising.



(a) Label, 10th frame

(b) Prediction, 10th frame

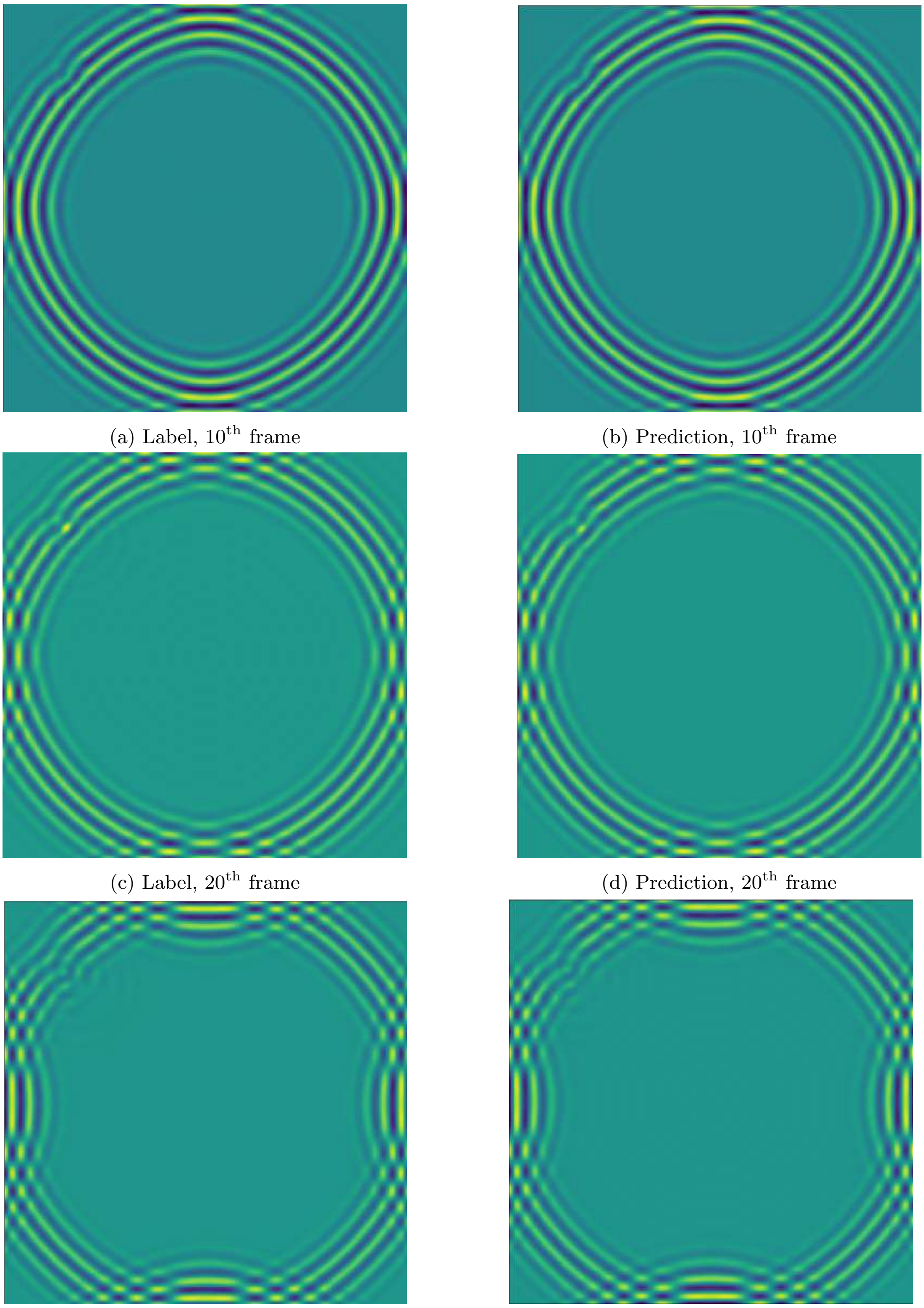
(c) Label, 20th frame

(d) Prediction, 20th frame

(e) Label, 30th frame

(f) Prediction, 30th frame

Figure 7.5: First scenario: comparison of predicted frames with the label frames at 10th, 20th, and 30th frame after the interaction with delamination.



(a) Label, 10th frame

(b) Prediction, 10th frame

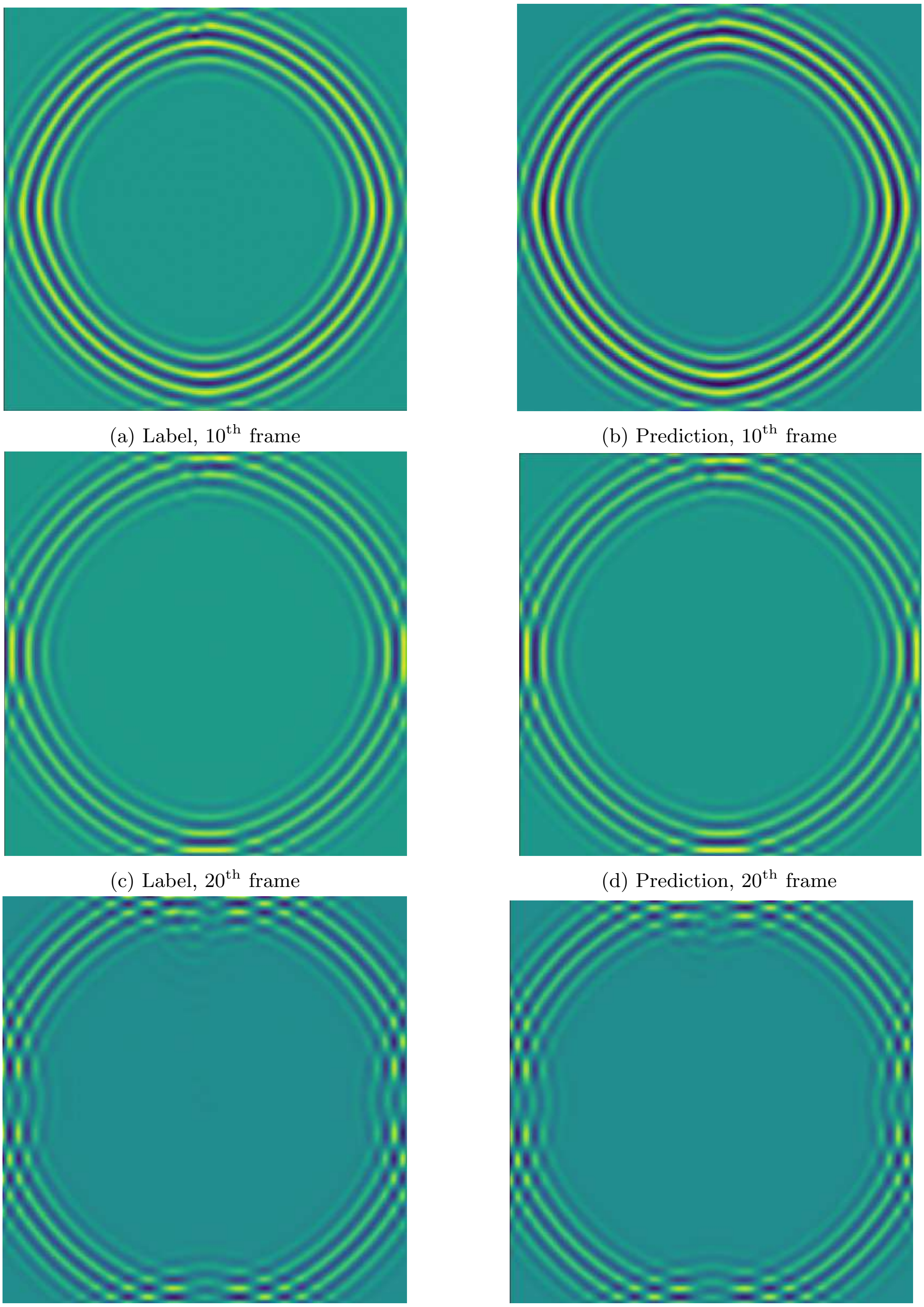
(c) Label, 20th frame

(d) Prediction, 20th frame

(e) Label, 30th frame

(f) Prediction, 30th frame

Figure 7.6: Second scenario: comparison of predicted frames with the label frames at 10th, 20th, and 30th frame after the interaction with delamination.



(a) Label, 10th frame

(b) Prediction, 10th frame

(c) Label, 20th frame

(d) Prediction, 20th frame

(e) Label, 30th frame

(f) Prediction, 30th frame

Figure 7.7: Third scenario: comparison of predicted frames with the label frames at 10th, 20th, and 30th frame after the interaction with delamination.

From all three scenarios, it can be confirmed that the proposed DL-based surrogate model has accurately reconstructed the full wavefield containing delamination with minimal error. Furthermore, the PSNR and Pearson CC values of all these three scenarios are shown in Table 7.2. The mean PSNR value was 21.8 dB, and the mean Pearson CC value was 0.98 on all of the test data. It confirms that the predictions by the proposed DL model are accurate.

Table 7.2: DL surrogate model evaluation metrics for three numerical cases

scenario		PSNR	Pearson CC
first	10th frame	23.3 dB	0.96
	20th frame	23.4 dB	0.98
	30th frame	23.7 dB	0.98
second	10th frame	21.4 dB	0.96
	20th frame	22.1 dB	0.98
	30th frame	22.6 dB	0.98
third	10th frame	21.8 dB	0.97
	20th frame	22.1 dB	0.98
	30th frame	22.3 dB	0.99

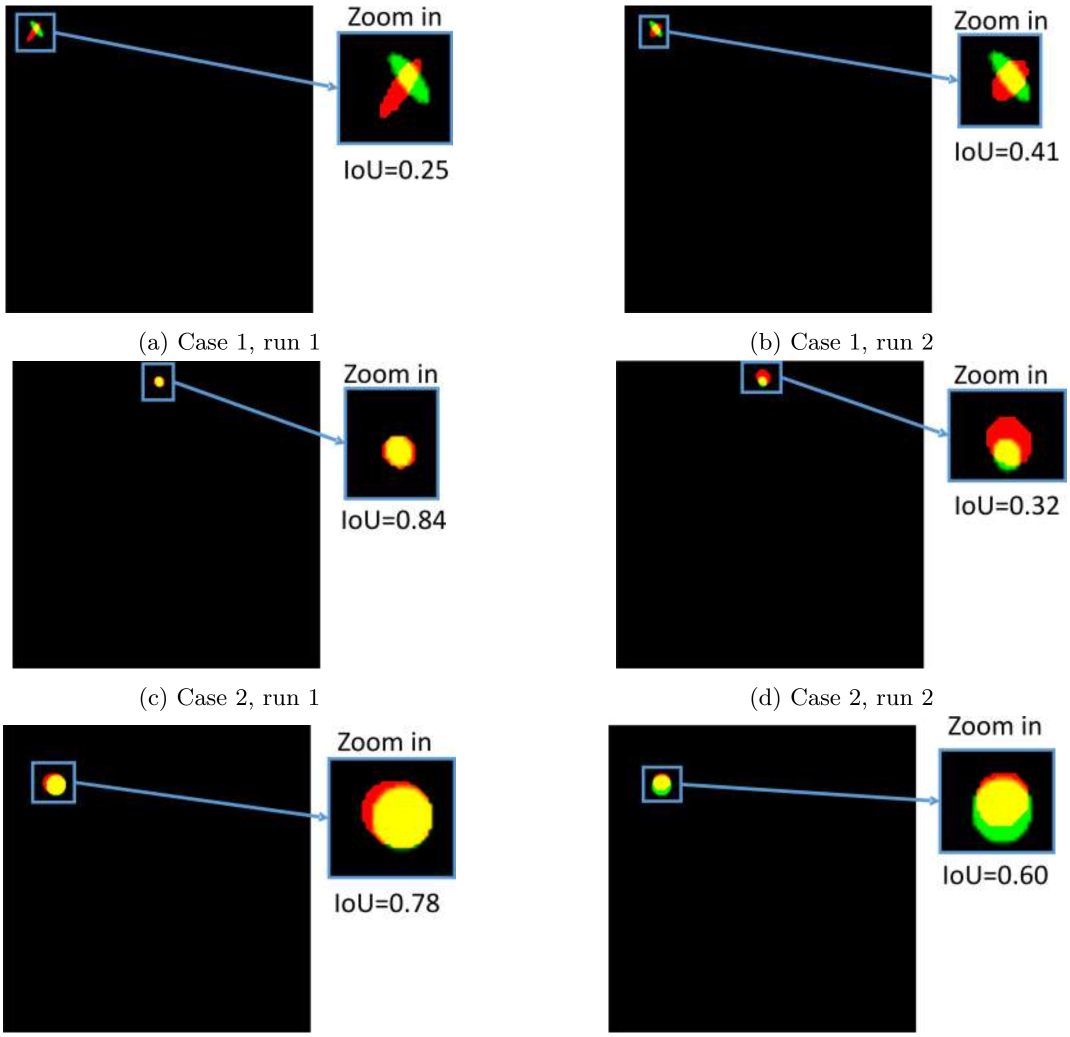
7.6.2 Delamination identification results

The delamination identification results obtained using PSO aided by the DL-based surrogate model are presented in Fig. 7.8. Several runs were performed due to the meta-heuristic nature of the PSO algorithm, and the results from two runs are selected for illustration purposes. The following cases were selected, namely case 1 (Fig. 7.8a, Fig. 7.8b), case 2 (Fig. 7.8c, Fig. 7.8d), and case 3 (Fig. 7.8e, Fig. 7.8f), where the damage identification difficulty can be ranked from highest to lowest. The most challenging case is case 1, in which the delamination is in the corner of the plate. For case 2, the delamination is located very close to the top edge of the plate, where edge reflections can overshadow reflections from delamination. The delamination in case 3 is quite large and far away from the plate's edges, making it relatively easy to detect.

Actually, the damage identification difficulty level is reflected in the obtained IoU values, which are shown in the zoomed-in regions around delamination in Fig. 7.8. On average, the lowest IoU values were obtained for case 1.

The visualisation of damage identification results is presented in such a way that the delamination ground truth is shown in green colour, the DL model's prediction in red, and the intersection of the two is represented by colour mixing, resulting in a yellow colour. Therefore, the more yellow pixels, the greater the overlap of delaminations and, consequently, the higher the accuracy.

It should be noted that despite low *IoU* values in certain cases, the identification algorithm performed remarkably well because delamination was localised accurately for each scenario.



(a) Case 1, run 1

(b) Case 1, run 2

(c) Case 2, run 1

(d) Case 2, run 2

(e) Case 3, run 1

(f) Case 3, run 2

Figure 7.8: Delamination identification results; green - ground truth, red - prediction, yellow - intersection.

The delamination identification results in terms of IoU values are presented in Table 7.3, with results from chapter 5 included for comparison. The method introduced in chapter 5 is completely different from the one presented here but relies on the same dataset. However, the frame size was larger, namely (512×512) pixels versus (256×256) pixels here, providing better resolution of damage identification.

Although, according to Table 7.3, the current results are not as good as compared to the previous approach (as of chapter 5), the advantage of the proposed method is that it can be easily extended to the cases in which only a limited number of signals are available in comparison to full wavefield data. This is of great significance in practical applications for structural health monitoring where only signals from sensor locations are available.

It should also be stressed that the complexity of the proposed DL model and dataset size used for training are constrained by the memory of a single Nvidia Tesla V100 GPU (32 GB memory), which was available. Undoubtedly, the surrogate DL model can be improved by using a larger number of frames in the sequence of ConvLSTM layers. This is expected to result in improved damage identification with a more accurate surrogate DL model.

Table 7.3: Damage identification evaluation metrics for three numerical cases

case number		IoU	
		current	chapter 5
1	run 1	0.25	0.74
	run 2	0.41	
	run 3	0.34	
2	run 1	0.84	0.76
	run 2	0.32	
	run 3	0.34	
3	run 1	0.78	0.88
	run 2	0.60	
	run 3	0.85	

7.7 Conclusions

In this chapter, a novel DL-based surrogate model is presented. The developed model adopts the architecture of an autoencoder-decoder in conjunction with ConvLSTM for predicting a full wavefield containing interacting Lamb waves with delamination. In the proposed model, the encoder and decoder parts are jointly trained on a synthetic dataset comprising frames that contain wave patterns of delamination reflections and changes in the wavefront due to delamination. Subsequently, the encoder is trained separately on reference data without delamination. The delamination information in the form of binary images is also provided alongside the reference images to the encoder part for training the encoder, resulting in the final prediction of frames propagating in the plate with delamination. In simple words, this DL-based surrogate model takes full wavefield frames of propagating waves in a healthy plate as input and predicts full wavefields in a plate containing a single delamination.

The proposed DL model has demonstrated excellent performance, as evidenced by the results. The proposed DL model is helpful for predicting full wavefield data from the time of excitation initiation to the desired simulation time. The predicted wavefield from the proposed architecture can be used for inverse problems in NDT (shown here) and SHM (possibly in future). The wavefield prediction by the proposed DL model is ultrafast, making it suitable for objective functions that require multiple evaluations, which would be unfeasible with conventional forward solvers like those based on p-FEM or SEM.

Chapter 8

Conclusions and Future work

This chapter encompasses the conclusions drawn from the research undertaken in this dissertation. Moreover, it delves into potential avenues for future research building upon the current work.

8.1 Conclusions

GW-based NDT/SHM systems have gained prominence due to their capacity for reliable inspection of structures. These systems often rely on data-driven methods, which necessitate substantial datasets derived from ongoing structural monitoring efforts. In recent times, non-contact systems have garnered attention for damage detection and localization in NDT/SHM applications. They utilize SLDV to capture full wavefields of elastic wave propagation initiated by a fixed piezoelectric transducer. The primary objective of this dissertation was to explore the viability of employing artificial neural networks-based approaches for identifying delaminations in composite laminated structures. Consequently, this dissertation introduces a novel approach, featuring the utilization of full wavefields from Lamb wave propagation in composite laminates in conjunction with deep learning techniques for delamination identification. The full wavefields generated by elastic wave propagation in composite laminates encapsulate extensive, intricate information about plate discontinuities like delaminations or edges. This rich information can serve as the foundation for training DL models to conduct damage identification via an end-to-end methodology. DL approaches enable the use of raw registered data without the necessity of conventional feature engineering, extraction, and classification. Thus, they offer an end-to-end structure that autonomously uncovers latent features within high-dimensional input data. Although DL typically demands more time for training, it compensates with rapid and efficient testing, outpacing conventional signal processing and machine learning methods.

The primary findings of this dissertation can be summarized as follows:

- Introduction of an innovative ANNs-based method that employs an end-to-end approach for many-to-one sequence prediction, allows the identification of delamination location, size, and shape. This model can simultaneously process a sequence of a predefined number of full wavefield frames to detect and estimate delaminations. Consequently, utilizing a specific number of full wavefield frames proves sufficient for accurate delamination identification. The model exhibited promising results and displayed adaptability by effectively detecting multiple delaminations in real-world scenarios, even though it had been initially trained solely on single delamination scenarios.
- Full wavefield acquisition via SLDV is time consuming. The necessity to enhance the signal-to-noise ratio (SNR) through averaging procedure, often requiring several hours of automatic measurements. As a solution, a DL model for super-resolution image reconstruction was developed, capable of accurately recovering high-resolution full wavefield frames from low-resolution SLDV data. Comparative analysis against the traditional Compressive Sensing (CS) technique revealed the DL model's superiority in reconstructing full wavefield frames. Furthermore, when applied to experimentally obtained full wavefield frames via SLDV, the DLSR model demonstrated its ability to generalize. Consequently, it is noteworthy that DL-based super-resolution frame reconstruction outperforms conventional CS techniques. The outputs of the DLSR approach can be feed into the proposed ConvLSTM approach for the purpose of delamination identification.
- This study adopts a novel DL-based methodology to delve into the intricacies of GW propagation within composite structures composed of various delamination scenarios. Central to this strategy lies in the utilization of an advanced deep ConvLSTM autoencoder-based surrogate model, meticulously crafted to produce extensive full wavefield data simulating the characteristics of propagating GWs within these composite structures. Furthermore, this state-of-the-art DL model, utilized to predict full wavefield patterns, establishes its groundbreaking application within the realm of the inverse problem associated with damage identification.

8.2 Future work

The research conducted in this study primarily involved training DL models designed for the detection and localization of delamination in CFRP structures. This foundational work can serve as a stepping stone for identifying various other types of defects within CFRP structures. It is important to note that the dataset created for training the models in this research required approximately three months of computational effort. However, given the advancements in computational power, generating such a dataset can be accomplished within a reasonable timeframe. Consequently, this dataset can be employed to train DL models capable of simultaneously detecting and localizing different types of defects in an end-to-end manner, potentially enhancing the performance of the developed models.

To further improve the performance of the proposed models, training on experimental data could be considered, allowing them to learn more intricate patterns. It is worth mentioning that the studies presented here are focused on a single type of signal at a carrier frequency of 50 kHz. As a progression, a new dataset could be generated with a higher excitation frequency or a broadband frequency signal. Additionally, the proposed approaches have demonstrated feasibility for delamination identification. Therefore, the next logical step would be to collect a comprehensive experimental dataset comprising full wavefields from structures with stiffeners, rivets, as well as basic plate-like structures. Such experimental data, as opposed to naive numerical datasets, would significantly enhance the performance of the proposed models.

Furthermore, the model developed for super-resolution image reconstruction in this research was trained using a low-resolution dataset with a compression ratio equivalent to 19.2% of the Nyquist sampling interval. This approach can be extended to various compression ratios, offering opportunities for further exploration.

In this research work, only a specific number of frames (f_w) were selected to feed into the proposed DL models (due to limited GPU memory availability). In the future, the more GPU memory could be acquired so that all of the 512 frames of the full wavefield could be inputted into the DL models. Further, due to higher GPU memory, more dense DL models can also be implemented. The surrogate model would also be capable of training with (512×512) pixels instead of (256×256) pixels, resulting in improved performance of the surrogate modelling, and will be more efficient for the inverse problems.

In this research work, most of the convolutional neural networks are employed. In the future, various other DL approaches, specifically physics-informed neural networks, generative adversarial networks, graph neural networks, and transformer-based networks can be applied, particularity for the reconstruction of full wavefield and simulation of full wavefield tasks.

References/Bibliography

[1] C. R. Farrar and K. Worden. “An introduction to structural health monitoring”. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 365.1851 (2007), pp. 303–315. ISSN: 1364503X. DOI: 10.1098/rsta.2006.1928.

[2] S. Alampalli and M. Ettouney. *Structural identification, damage identification and structural health monitoring.* Tech. rep. 2007, p. 65310L. DOI: 10.1117/12.715000.

[3] T. Stepinski, T. Uhl, and W. Staszewski. *Advanced structural damage detection: from theory to engineering applications.* John Wiley & Sons, 2013.

[4] L. Deng and D. Yu. “Deep learning: methods and applications”. In: *Foundations and trends in signal processing* 7.3–4 (2014), pp. 197–387.

[5] Q. K. Al-Shayea. “Artificial neural networks in medical diagnosis”. In: *International Journal of Computer Science Issues* 8.2 (2011), pp. 150–154.

[6] M. Pashaei, H. Kamangir, M. J. Starek, and P. Tissot. “Review and evaluation of deep learning architectures for efficient land cover mapping with UAS hyperspatial imagery: A case study over a wetland”. In: *Remote Sensing* 12.6 (2020), p. 959.

[7] C. R. Farrar, H. Sohn, F. M. Hemez, M. C. Anderson, M. T. Bement, P. J. Cornwell, S. W. Doebling, J. F. Schultze, N Lieven, and A. N. Robertson. “Damage prognosis: current status and future needs”. In: *Report. Los Alamos, NM, USA: Los Alamos National Laboratory* (2003).

[8] C. R. Farrar and K. Worden. *Structural Health Monitoring: A Machine Learning Perspective.* 2012. ISBN: 9781119994336. DOI: 10.1002/9781118443118.

[9] A. Rytter. “Vibrational Based Inspection of Civil Engineering Structures”. 1993.

[10] P. J. Shull. *Nondestructive evaluation: theory, techniques, and applications.* CRC press, 2002.

[11] M. Jawaid, M. Thariq, and N. Saba. *Structural Health Monitoring of Biocomposites, Fibre-Reinforced Composites and Hybrid Composites.* Woodhead Publishing, 2018.

[12] K. Worden and J. M. Dulieu-Barton. “An Overview of Intelligent Fault Detection in Systems and Structures”. In: *Structural Health Monitoring* 3.1 (2004), pp. 85–98. ISSN: 14759217. DOI: 10.1177/1475921704041866.

[13] P. D. Samuel and D. J. Pines. *A review of vibration-based techniques for helicopter transmission diagnostics.* Vol. 282. 1-2. 2005, pp. 475–508. ISBN: 1301314900. DOI: 10.1016/j.jsv.2004.02.058.

[14] D. C. Montgomery. "The DMAIC process". In: *Introduction to statistical quality control. 6th ed. Hoboken, NJ: John Wiley & Sons Inc* (2009), pp. 45–60.

[15] S. Gopalakrishnan, M. Ruzzene, and S. Hanagud. *Computational techniques for structural health monitoring.* Springer Science & Business Media, 2011.

[16] D. Balageas, C. P. Fritzen, and A. Güemes. *Structural Health Monitoring.* Wiley-ISTE, 2010, pp. 1–495. ISBN: 9781905209019. DOI: 10.1002/9780470612071.

[17] F. K. Chang. "Introduction to health monitoring: context, problems, solutions". In: *Presentation at the 1st European pre-workshop on structural health monitoring, Paris, France.* 2002.

[18] F Lamonaca, P. F. Sciammarella, C Scuro, D. L. Carni, and R. S. Olivito. "Internet of Things for Structural Health Monitoring". In: *2018 Workshop on Metrology for Industry 4.0 and IoT, MetroInd 4.0 and IoT 2018 - Proceedings* September (2018), pp. 95–100. DOI: 10.1109/METROI4.2018.8439038.

[19] J.-J. Lee, Y.-S. Park, C.-B. Yun, K.-Y. Koo, and J.-H. Yi. "An Overview of Information Processing Techniques for Structural Health Monitoring of Bridges". In: *Journal of the Computational Structural Engineering Institute of Korea* 21.6 (2008), pp. 615–632.

[20] D. A. Tibaduiza Burgos, R. C. Gomez Vargas, C. Pedraza, D. Agis, and F. Pozo. *Damage identification in structural health monitoring: A brief review from its implementation to the use of data-driven applications.* 2020. DOI: 10.3390/s20030733.

[21] A. Güemes, A. Fernandez-Lopez, A. R. Pozo, and J. Sierra-Pérez. "Structural Health Monitoring for Advanced Composite Structures: A Review". In: *Journal of Composites Science* 4.1 (2020), p. 13. ISSN: 2504-477X. DOI: 10.3390/jcs4010013.

[22] R. L. Crane. "Introduction to structural health monitoring". In: *Comprehensive Composite Materials II* March (2017), pp. 355–357. DOI: 10.1016/B978-0-12-803581-8.10350-9.

[23] C. Liang, F. P. Sun, and C. A. Rogers. "Coupled electro-mechanical analysis of adaptive material systems-determination of the actuator power consumption and system energy transfer". In: *Journal of intelligent material systems and structures* 8.4 (1997), pp. 335–343.

[24] P. Fiborek, P. H. Malinowski, P. Kudela, T. Wandowski, and W. M. Ostachowicz. "Time-domain spectral element method for modelling of the electromechanical impedance of disbonded composites". In: *Journal of Intelligent Material Systems and Structures* 29.16 (2018), pp. 3214–3221. ISSN: 15308138. DOI: 10.1177/1045389X18758193.

[25] K. Worden and G. Manson. "The application of machine learning to structural health monitoring". In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 365.1851 (2007), pp. 515–537. ISSN: 1364503X. DOI: 10.1098/rsta.2006.1938.

[26] A. Deraemaeker and K. Worden. *New trends in vibration based structural health monitoring.* Vol. 520. Springer Science & Business Media, 2012.

[27] A. Raghavan. "Guided-wave structural health monitoring". In: *Dissertation Abstracts International, Volume: 68-11, Section: B, page: 7465.;Adviser: Carlos E. Cesn* 975.2 (2007), pp. 91–114.

[28] H. Mei, M. F. Haider, R. Joseph, A. Migot, and V. Giurgiutiu. "Recent advances in piezoelectric wafer active sensors for structural health monitoring applications". In: *Sensors (Switzerland)* 19.2 (2019). ISSN: 14248220. DOI: 10.3390/s19020383.

[29] Z. Tian, L. Yu, and C. Leckey. "Delamination detection and quantification on laminated composite structures with Lamb waves and wavenumber analysis". In: *Journal of Intelligent Material Systems and Structures* 26.13 (2015), pp. 1723–1738. ISSN: 15308138. DOI: 10.1177/1045389X14557506.

[30] B. Park, Y. K. An, and H. Sohn. "Visualization of hidden delamination and debonding in composites through noncontact laser ultrasonic scanning". In: *Composites Science and Technology* 100 (2014), pp. 10–18. ISSN: 02663538. DOI: 10.1016/j.compscitech.2014.05.029. URL: http://dx.doi.org/10.1016/j.compscitech.2014.05.029.

[31] S. Sikdar, P. Fiborek, P. Kudela, S. Banerjee, and W. Ostachowicz. "Effects of debonding on Lamb wave propagation in a bonded composite structure under variable temperature conditions". In: *Smart Materials and Structures* 28.1 (2019). ISSN: 1361665X. DOI: 10.1088/1361-665X/aaefaa.

[32] D. Girolamo, H. Y. Chang, and F. G. Yuan. "Impact damage visualization in a honeycomb composite panel through laser inspection using zero-lag cross-correlation imaging condition". In: *Ultrasonics* 87 (2018), pp. 152–165. ISSN: 0041624X. DOI: 10.1016/j.ultras.2018.02.014. URL: https://doi.org/10.1016/j.ultras.2018.02.014.

[33] M. D. Rogge and C. A. Leckey. "Characterization of impact damage in composite laminates using guided wavefield imaging and local wavenumber domain analysis". In: *Ultrasonics* 53.7 (2013), pp. 1217–1226. ISSN: 0041624X. DOI: 10.1016/j.ultras.2012.12.015. URL: http://dx.doi.org/10.1016/j.ultras.2012.12.015.

[34] P. Kudela, M. Radzienski, and W. Ostachowicz. "Impact induced damage assessment by means of Lamb wave image processing". In: *Mechanical Systems and Signal Processing* 102 (2018), pp. 23–36. ISSN: 10961216. DOI: 10.1016/j.ymssp.2017.09.020. URL: https://doi.org/10.1016/j.ymssp.2017.09.020.

[35] J. L. Rose, W. Zhu, and M. Zaidi. "Ultrasonic NDT of titanium diffusion bonding with guided waves". In: (1998).

[36] P. Fromme, P. D. Wilcox, M. J. S. Lowe, and P. Cawley. "On the development and testing of a guided ultrasonic wave array for structural integrity monitoring". In: *ieee transactions on ultrasonics, ferroelectrics, and frequency control* 53.4 (2006), pp. 777–785.

[37] P Wilcox, M Evans, B Pavlakovic, D Alleyne, K Vine, P Cawley, and M Lowe. "Guided wave testing of rail". In: *Insight-Non-Destructive Testing and Condition Monitoring* 45.6 (2003), pp. 413–420.

[38] H. Gao and J. L. Rose. "Ice detection and classification on an aircraft wing with ultrasonic shear horizontal guided waves". In: *IEEE transactions on ultrasonics, ferroelectrics, and frequency control* 56.2 (2009), pp. 334–344.

[39] Y. Lugovtsova, J. Bulling, C. Boller, and J. Prager. "Analysis of guided wave propagation in a multi-layered structure in view of structural health monitoring". In: *Applied Sciences (Switzerland)* 9.21 (2019). ISSN: 20763417. DOI: 10.3390/app9214600.

[40] W. Wang, L. Li, Y. Fan, and Z. Jiang. "Piezoelectric transducers for structural health monitoring of joint structures in cylinders: A wave-based design approach". In: *Sensors (Switzerland)* 20.3 (2020), pp. 0–26. ISSN: 14248220. DOI: 10.3390/s20030601.

[41] M. Mitra and S. Gopalakrishnan. "Guided wave based structural health monitoring: A review". In: *Smart Materials and Structures* 25.5 (2016), p. 0. ISSN: 1361665X. DOI: 10.1088/0964-1726/25/5/053001. URL: http://dx.doi.org/10.1088/0964-1726/25/5/053001.

[42] P. Cawley, M. J. Lowe, D. N. Alleyne, B. Pavlakovic, and P. Wilcox. "Practical long range guided wave testing: Applications to pipes and rail". In: *Materials Evaluation* 61.1 (2003), pp. 66–74. ISSN: 00255327.

[43] R. K. Munian, D. R. Mahapatra, and S. Gopalakrishnan. "Lamb wave interaction with composite delamination". In: *Composite Structures* 206.April (2018), pp. 484–498. ISSN: 02638223. DOI: 10.1016/j.compstruct.2018.08.072. URL: https://doi.org/10.1016/j.compstruct.2018.08.072.

[44] T Wandowski, P Kudela, P Malinowski, and W Ostachowicz. "Lamb waves for damage localisation in panels". In: *Strain* 47.5 (2011), pp. 449–457.

[45] M. Radzieński, P. Kudela, A. Marzani, L. De Marchi, and W. Ostachowicz. "Damage identification in various types of composite plates using guided waves excited by a piezoelectric transducer and measured by a laser vibrometer". In: *Sensors (Switzerland)* 19.9 (2019). ISSN: 14248220. DOI: 10.3390/s19091958.

[46] H. Lamb. "On waves in an elastic plate". In: *Proceedings of the Royal Society of London. Series A, Containing papers of a mathematical and physical character* 93.648 (1917), pp. 114–128.

[47] Z. Chang and A. Mal. "Scattering of Lamb waves from a rivet hole with edge cracks". In: *Mechanics of Materials* 31.3 (1999), pp. 197–204. ISSN: 01676636. DOI: 10.1016/S0167-6636(98)00060-X.

[48] M. S. Hameed and Z. Li. "Transverse Damage Localization and Quantitative Size Estimation for Composite Laminates Based on Lamb Waves". In: *IEEE Access* 7.December (2019), pp. 174859–174872. ISSN: 21693536. DOI: 10.1109/ACCESS.2019.2957515.

[49] J. L. Rose. "A baseline and vision of ultrasonic guided wave inspection potential". In: *J. Pressure Vessel Technol.* 124.3 (2002), pp. 273–282.

[50] M. J. S. Lowe and P Cawley. “Long range guided wave inspection usage–current commercial capabilities and research directions”. In: *Department of Mechanical Engineering, Imperial College, London* (2006).

[51] W. Ostachowicz, P. Kudela, P. Malinowski, and T. Wandowski. “Damage localisation in plate-like structures based on PZT sensors”. In: *Mechanical Systems and Signal Processing* 23.6 (2009), pp. 1805–1829. ISSN: 08883270. DOI: 10.1016/j.ymssp.2008.10.011.

[52] S. S. Kessler, S. M. Spearing, and C. Soutis. “Damage detection in composite materials using Lamb wave methods”. In: *Smart Materials and Structures* 11.2 (2002), pp. 269–278. ISSN: 09641726. DOI: 10.1088/0964-1726/11/2/310.

[53] P. Kudela, M. Radzienski, W. Ostachowicz, and Z. Yang. “Structural Health Monitoring system based on a concept of Lamb wave focusing by the piezoelectric array”. In: *Mechanical Systems and Signal Processing* 108 (2018), pp. 21–32. ISSN: 10961216. DOI: 10.1016/j.ymssp.2018.02.008. URL: https://doi.org/10.1016/j.ymssp.2018.02.008.

[54] E. K. Dimitriadis, C. R. Fuller, and C. A. Rogers. “Piezoelectric actuators for distributed vibration excitation of thin plates”. In: (1991).

[55] M. J. Schulz, P. F. Pai, and D. J. Inman. “Health monitoring and active control of composite structures using piezoceramic patches”. In: *Composites Part B: Engineering* 30.7 (1999), pp. 713–725. ISSN: 13598368. DOI: 10.1016/S1359-8368(99)00034-7.

[56] R. A. Badcock and E. A. Birt. “Use of 0-3 piezocomposite embedded Lamb wave sensors for detection of damage in advanced fibre composites”. In: *Smart Materials and Structures* 9.3 (2000), pp. 291–297. ISSN: 09641726. DOI: 10.1088/0964-1726/9/3/307.

[57] A. M. Thomas, K. R. Pradeep, and P. Mathew. “Structural health monitoring using piezoelectric sensors and actuators”. In: *Applied Mechanics and Materials*. Vol. 857. Trans Tech Publ. 2017, pp. 255–260.

[58] Z. Su and L. Ye. “Lamb wave-based quantitative identification of delamination in CF/EP composite structures using artificial neural algorithm”. In: *Composite Structures* 66.1-4 (2004), pp. 627–637. ISSN: 02638223. DOI: 10.1016/j.compstruct.2004.05.011.

[59] F. Ricci, E. Monaco, L. Maio, N. D. Boffa, and A. K. Mal. “Guided waves in a stiffened composite laminate with a delamination”. In: *Structural Health Monitoring* 15.3 (2016), pp. 351–358. ISSN: 17413168. DOI: 10.1177/1475921716636335.

[60] S. Das and P. Saha. “A review of some advanced sensors used for health diagnosis of civil engineering structures”. In: *Measurement: Journal of the International Measurement Confederation* 129.January (2018), pp. 68–90. ISSN: 02632241. DOI: 10.1016/j.measurement.2018.07.008. URL: https://doi.org/10.1016/j.measurement.2018.07.008.

[61] S. R. Anton and H. A. Sodano. “A review of power harvesting using piezoelectric materials (2003-2006)”. In: *Smart Materials and Structures* 16.3 (2007). ISSN: 09641726. DOI: 10.1088/0964-1726/16/3/R01.

[62] Z. Su and L. Ye. *Identification of damage using Lamb waves: from fundamentals to applications*. Vol. 48. Springer Science & Business Media, 2009.

[63] G. E. Stavroulakis. *Lecture Notes in Applied and Computational Mechanics: Introduction*. Vol. 56 LNACM. 2013. ISBN: 9783642339677.

[64] P. Kedziora, M. Barski, and M. Chwał. "Piezoelectric transducers". In: *Key Engineering Materials*. Vol. 542. Trans Tech Publ. 2013, pp. 75–80.

[65] M. S. Hameed, Z. Li, J. Chen, and J. Qi. "Lamb-wave-based multistage damage detection method using an active PZT sensor network for large structures". In: *Sensors (Switzerland)* 19.9 (2019). ISSN: 14248220. DOI: 10.3390/s19092010.

[66] S. Sikarwar, Satyendra, S. Singh, and B. C. Yadav. "Review on pressure sensors for structural health monitoring". In: *Photonic Sensors* 7.4 (2017), pp. 294–304. ISSN: 21907439. DOI: 10.1007/s13320-017-0419-z.

[67] O. S. Salawu. "Detection of structural damage through changes in frequency: A review". In: *Engineering Structures* 19.9 (1997), pp. 718–723. ISSN: 01410296. DOI: 10.1016/S0141-0296(96)00149-6.

[68] Y. Nemirovsky, A. Nemirovsky, P. Muralt, and N. Setter. "Design of a novel thin-film piezoelectric accelerometer". In: *Sensors and Actuators, A: Physical* 56.3 (1996), pp. 239–249. ISSN: 09244247. DOI: 10.1016/S0924-4247(96)01324-6.

[69] D. L. DeVoe and A. P. Pisano. "Surface micromachined piezoelectric accelerometers (PiXLs)". In: *Journal of Microelectromechanical Systems* 10.2 (2001), pp. 180–186. ISSN: 10577157. DOI: 10.1109/84.925733.

[70] A. Sabato, C. Niezrecki, and G. Fortino. "Wireless MEMS-Based Accelerometer Sensor Boards for Structural Vibration Monitoring: A Review". In: *IEEE Sensors Journal* 17.2 (2017), pp. 226–235. ISSN: 1530437X. DOI: 10.1109/JSEN.2016.2630008.

[71] W. Li, Y. Yuan, J. Yang, H. Deng, and L. Yuan. "In-Fiber Integrated Sensor Array With Embedded Weakly Reflective Joint Surface". In: *Journal of Lightwave Technology* 36.23 (2018), pp. 5663–5668.

[72] Z. H. Warsi, S. M. Irshad, F. Khan, M. A. Shahbaz, M. Junaid, and S. U. Amin. "Sensors for Structural Health Monitoring: A Review". In: *2019 Second International Conference on Latest trends in Electrical Engineering and Computing Technologies (INTELLECT)*. IEEE. 2019, pp. 1–6.

[73] B. Torres, I. Payá-Zaforteza Ignacio, P. A. Calderón, and J. M. Adam. "Analysis of the strain transfer in a new FBG sensor for Structural Health Monitoring". In: *Engineering Structures* 33.2 (2011), pp. 539–548. ISSN: 01410296. DOI: 10.1016/j.engstruct.2010.11.012. URL: http://dx.doi.org/10.1016/j.engstruct.2010.11.012.

[74] J. R. Casas and P. J. S. Cruz. "Fiber optic sensors for bridge monitoring". In: *Journal of bridge engineering* 8.6 (2003), pp. 362–373.

[75] M. Majumder, T. K. Gangopadhyay, A. K. Chakraborty, K. Dasgupta, and D. K. Bhattacharya. "Fibre Bragg gratings in structural health monitoring-Present status and applications". In: *Sensors and Actuators, A: Physical* 147.1 (2008), pp. 150–164. ISSN: 09244247. DOI: 10.1016/j.sna.2008.04.008.

[76] H. N. Li, D. S. Li, and G. B. Song. “Recent applications of fiber optic sensors to health monitoring in civil engineering”. In: *Engineering Structures* 26.11 (2004), pp. 1647–1657. ISSN: 01410296. DOI: `10.1016/j.engstruct.2004.05.018`.

[77] S. Delepine-Lesoille, E. Merliot, C. Boulay, L. Quétel, M. Delaveau, and A. Courteville. “Quasi-distributed optical fibre extensometers for continuous embedding into concrete: Design and realization”. In: *Smart Materials and Structures* 15.4 (2006), pp. 931–938. ISSN: 09641726. DOI: `10.1088/0964-1726/15/4/005`.

[78] P. J. Bourne-Webb, B Amatya, K Soga, T Amis, C Davidson, and P Payne. “Energy pile test at Lambeth College, London: geotechnical and thermodynamic aspects of pile response to heat cycles”. In: *Géotechnique* 59.3 (2009), pp. 237–248.

[79] H. Guo, G. Xiao, N. Mrad, and J. Yao. “Fiber optic sensors for structural health monitoring of air platforms”. In: *Sensors* 11.4 (2011), pp. 3687–3705.

[80] N. Takeda, Y. Okabe, J. Kuwahara, S. Kojima, and T. Ogisu. “Development of smart composite structures with small-diameter fiber Bragg grating sensors for damage detection: Quantitative evaluation of delamination length in CFRP laminates using Lamb wave sensing”. In: *Composites science and technology* 65.15-16 (2005), pp. 2575–2587.

[81] H. Tsuda. “Ultrasound and damage detection in CFRP using fiber Bragg grating sensors”. In: *Composites science and technology* 66.5 (2006), pp. 676–683.

[82] D. C. Betz, G. Thursby, B. Culshaw, and W. J. Staszewski. “Lamb wave detection and source location using fiber Bragg grating rosettes”. In: *Smart Structures and Materials 2003: Smart Sensor Technology and Measurement Systems*. Vol. 5050. International Society for Optics and Photonics. 2003, pp. 117–127.

[83] D. C. Betz, G. Thursby, B. Culshaw, and W. J. Staszewski. “Structural damage location with fiber Bragg grating rosettes and Lamb waves”. In: *Structural health monitoring* 6.4 (2007), pp. 299–308.

[84] H.-N. Li, D.-S. Li, and G.-B. Song. “Recent applications of fiber optic sensors to health monitoring in civil engineering”. In: *Engineering structures* 26.11 (2004), pp. 1647–1657.

[85] Q. Zhen and W. Piyawattanamatha. “New endoscopic imaging technology based on MEMS sensors and actuators”. In: *Micromachines* 8.7 (2017). ISSN: 2072666X. DOI: `10.3390/mi8070210`.

[86] E. S. Cochran, J. F. Lawrence, A. Kaiser, B. Fry, A. Chung, and C. Christensen. “Comparison between low-cost and traditional MEMS accelerometers: a case study from the M7. 1 Darfield, New Zealand, aftershock deployment”. In: *Annals of Geophysics* 54.6 (2012).

[87] F.-K. Chang. *Structural health monitoring 2013: a roadmap to intelligent structures: proceedings of the ninth international workshop on structural health monitoring, september 10–12, 2013*. DEStech Publications, Inc, 2013.

[88] H. Saboonchi, D. Ozevin, and M. Kabir. “MEMS sensor fusion: Acoustic emission and strain”. In: *Sensors and Actuators, A: Physical* 247 (2016), pp. 566–

578. ISSN: 09244247. DOI: 10.1016/j.sna.2016.05.014. URL: http://dx.doi.org/10.1016/j.sna.2016.05.014.

[89] A. Dixit and S. Bhalla. "Prognosis of fatigue and impact induced damage in concrete using embedded piezo-transducers". In: *Sensors and Actuators, A: Physical* 274 (2018), pp. 116–131. ISSN: 09244247. DOI: 10.1016/j.sna.2018.03.005. URL: https://doi.org/10.1016/j.sna.2018.03.005.

[90] H. Debéda, R. Lakhmi, V. Pommier-Budinger, and C. Lucat. "Study of free-standing electroded PZT thick-films: from materials to microsystems". In: *Key Engineering Materials*. Vol. 605. Trans Tech Publ. 2014, pp. 55–58.

[91] J.-S. Kim, K. Vinoy, and V. K. Varadan. "Wireless health monitoring of cracks in structures with MEMS-IDT sensors". In: *Smart Structures and Materials 2002: Smart Electronics, MEMS, and Nanotechnology*. Vol. 4700. International Society for Optics and Photonics. 2002, pp. 342–353.

[92] V. K. Varadan. "Nanotechnology: MEMS and NEMS and their applications to smart systems and devices". In: *Smart materials, structures, and systems*. Vol. 5062. International Society for Optics and Photonics. 2003, pp. 20–43.

[93] R. O. Guldiken, O. Onen, M. Gul, and F. N. Catbas. "A structural health monitoring system with ultrasonic MEMS transducers". In: *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2011*. Vol. 7981. International Society for Optics and Photonics. 2011, 79810F.

[94] W. J. Staszewski, B. C. Lee, and R. Traynor. "Fatigue crack detection in metallic structures with Lamb waves and 3D laser vibrometry". In: *Measurement Science and Technology* 18.3 (2007), pp. 727–739. ISSN: 13616501. DOI: 10.1088/0957-0233/18/3/024.

[95] Ł. Doliński, M. Krawczuk, M. Palacz, W. Waszkowiak, and A. Żak. "The influence of the grid density of measurement points on damage detection in an isotropic plate by the use of elastic waves and laser scanning Doppler vibrometry". In: *Sensors* 21.21 (2021), p. 7394.

[96] P. Kudela, M. Radzieński, and W. Ostachowicz. "Identification of cracks in thin-walled structures by means of wavenumber filtering". In: *Mechanical Systems and Signal Processing* 50-51 (2015), pp. 456–466. ISSN: 10961216. DOI: 10.1016/j.ymssp.2014.05.041.

[97] W. Ostachowicz, M. Radzieński, and P. Kudela. "50th anniversary article: Comparison studies of full wavefield signal processing for crack detection". In: *Strain* 50.4 (2014), pp. 275–291. ISSN: 14751305. DOI: 10.1111/str.12098.

[98] V. Rahul, S. Alokita, K. Jayakrishna, V. R. Kar, M. Rajesh, S. Thirumalini, and M. Manikandan. *Structural health monitoring of aerospace composites*. August. Elsevier, 2018, pp. 33–52. ISBN: 9780081022917. DOI: 10.1016/B978-0-08-102291-7.00003-4. URL: http://dx.doi.org/10.1016/B978-0-08-102291-7.00003-4.

[99] L. Yu, Z. Tian, X. Li, R. Zhu, and G. Huang. "Core–skin debonding detection in honeycomb sandwich structures through guided wave wavefield analysis". In: *Journal of Intelligent Material Systems and Structures* 30.9 (2019), pp. 1306–1317.

[100] M. A. Fakih, S. Mustapha, and A. Abdul-Aziz. “Robust Localization and Classification of Barely Visible Indentations in Composite Structures by Fusion of Ultrasonic Damage Indices”. In: *Journal of Nondestructive Evaluation, Diagnostics and Prognostics of Engineering Systems* 2.3 (2019). ISSN: 25723898. DOI: `10.1115/1.4044177`.

[101] G. F. Gomes, Y. A. D. Mendéz, P. d. S. L. Alexandrino, S. S. da Cunha Jr, and A. C. Ancelotti Jr. “The use of intelligent computational tools for damage detection and identification with an emphasis on composites–A review”. In: *Composite Structures* 196 (2018), pp. 44–54.

[102] B. Yang, F. Z. Xuan, P. Jin, C. Hu, B. Xiao, D. Li, Y. Xiang, and H. Lei. “Damage Localization in Composite Laminates by Building in PZT Wafer Transducers: A Comparative Study with Surface-Bonded PZT Strategy”. In: *Advanced Engineering Materials* 21.3 (2019), pp. 1–12. ISSN: 15272648. DOI: `10.1002/adem.201801040`.

[103] W. Li, Y. Cho, and J. D. Achenbach. “Detection of thermal fatigue in composites by second harmonic Lamb waves”. In: *Smart Materials and Structures* 21.8 (2012). ISSN: 09641726. DOI: `10.1088/0964-1726/21/8/085019`.

[104] A. Zak, M. Radzieński, M. Krawczuk, and W. Ostachowicz. “Damage detection strategies based on propagation of guided elastic waves”. In: *Smart Materials and Structures* 21.3 (2012). ISSN: 09641726. DOI: `10.1088/0964-1726/21/3/035024`.

[105] N. Toyama and J. Takatsubo. “Lamb wave method for quick inspection of impact-induced delamination in composite laminates”. In: *Composites Science and Technology* 64.9 (2004), pp. 1293–1300. ISSN: 02663538. DOI: `10.1016/j.compscitech.2003.10.011`.

[106] N. Rauter and R. Lammering. “Impact damage detection in composite structures considering nonlinear lamb wave propagation”. In: *Mechanics of Advanced Materials and Structures* 22.1-2 (2015), pp. 44–51. ISSN: 15376532. DOI: `10.1080/15376494.2014.907950`.

[107] W. J. Staszewski, S. Mahzan, and R. Traynor. “Health monitoring of aerospace composite structures - Active and passive approach”. In: *Composites Science and Technology* 69.11-12 (2009), pp. 1678–1685. ISSN: 02663538. DOI: `10.1016/j.compscitech.2008.09.034`. URL: `http://dx.doi.org/10.1016/j.compscitech.2008.09.034`.

[108] R. Kruse, C. Borgelt, C. Braune, S. Mostaghim, and M. Steinbrecher. *Computational intelligence: a methodological introduction*. Springer, 2016.

[109] I. Nunes and H. S. Da Silva. *Artificial neural networks: a practical course*. Springer, 2018.

[110] C. C. Aggarwal et al. *Neural networks and deep learning*. Springer, 2018.

[111] B. Blachowski and N. Pnevmatikos. “Neural network based vibration control of seismically excited civil structures”. In: *Periodica Polytechnica Civil Engineering* 62.3 (2018), pp. 620–628.

[112] H. Simon et al. “Neural networks and learning machines”. In: *Upper Saddle River: Pearson Education* 3 (2009).

[113] S. Khan, H. Rahmani, S. A. A. Shah, and M. Bennamoun. “A guide to convolutional neural networks for computer vision”. In: *Synthesis Lectures on Computer Vision* 8.1 (2018), pp. 1–207.

[114] W. S. McCulloch and W. Pitts. “A logical calculus of the ideas immanent in nervous activity”. In: *The bulletin of mathematical biophysics* 5.4 (1943), pp. 115–133.

[115] D. Hebb. “Organization of behavior. New York: Wiley”. In: *J. Clin. Psychol* 6.3 (1949), pp. 335–307.

[116] A. L. Hodgkin and A. F. Huxley. “A quantitative description of membrane current and its application to conduction and excitation in nerve”. In: *The Journal of physiology* 117.4 (1952), p. 500.

[117] E Rosenblatt. *Principles qfneurod)’namics.* 1962.

[118] B. Widrow and M. E. Hoff. *Adaptive switching circuits.* Tech. rep. Stanford Univ Ca Stanford Electronics Labs, 1960.

[119] M. Minsky and S. A. Papert. *Perceptrons: An introduction to computational geometry.* MIT press, 2017.

[120] K. Fukushima and S. Miyake. “Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition”. In: *Competition and cooperation in neural nets.* Springer, 1982, pp. 267–285.

[121] D. H. Hubel and T. N. Wiesel. “Receptive fields and functional architecture of monkey striate cortex”. In: *The Journal of physiology* 195.1 (1968), pp. 215–243.

[122] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. “Backpropagation applied to handwritten zip code recognition”. In: *Neural computation* 1.4 (1989), pp. 541–551.

[123] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel. “Handwritten digit recognition with a back-propagation network”. In: *Advances in neural information processing systems.* 1990, pp. 396–404.

[124] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[125] P. Werbos. “Beyond regression:” new tools for prediction and analysis in the behavioral sciences”. In: *Ph. D. dissertation, Harvard University* (1974).

[126] T. W.-s. Chow and D. S.-y. Cho. *Neural networks and computing: Learning algorithms and applications.* Vol. 7. World Scientific, 2007.

[127] S. Russell and P. Norvig. “Artificial intelligence: a modern approach”. In: (2002).

[128] N. Abbas, Y. Nasser, and K. El Ahmad. “Recent advances on artificial intelligence and learning techniques in cognitive radio networks”. In: *EURASIP Journal on Wireless Communications and Networking* 2015.1 (2015), pp. 1–20.

[129] K. P. Murphy. *Machine learning: a probabilistic perspective.* MIT press, 2012.

[130] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference, and prediction.* Springer Science & Business Media, 2009.

[131] J. R. Quinlan. “Induction of decision trees”. In: *Machine learning* 1.1 (1986), pp. 81–106.

[132] L. Breiman. “Random forests”. In: *Machine learning* 45.1 (2001), pp. 5–32.

[133] A. Liaw, M. Wiener, et al. “Classification and regression by randomForest”. In: *R news* 2.3 (2002), pp. 18–22.

[134] E. E. Osuna. “Support vector machines: Training and applications”. PhD thesis. Massachusetts Institute of Technology, 1998.

[135] R. E. Schapire. “The boosting approach to machine learning: An overview”. In: *Nonlinear estimation and classification* (2003), pp. 149–171.

[136] R. E. Schapire and Y. Freund. “Boosting: Foundations and algorithms”. In: *Kybernetes* (2013).

[137] Y. LeCun, Y. Bengio, and G. Hinton. “Deep learning”. In: *nature* 521.7553 (2015), pp. 436–444.

[138] Y. Bengio, I. Goodfellow, and A. Courville. *Deep learning.* Vol. 1. MIT press Massachusetts, USA: 2017.

[139] J. Schmidhuber. “Deep learning in neural networks: An overview”. In: *Neural networks* 61 (2015), pp. 85–117.

[140] K. He, X. Zhang, S. Ren, and J. Sun. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2016, pp. 770–778.

[141] M. Słoński. “Bayesian neural networks and Gaussian processes in identification of concrete properties”. In: *Computer Assisted Methods in Engineering and Science* 18.4 (2011), pp. 291–302.

[142] K.-L. Du and M. N. Swamy. *Neural networks and statistical learning.* Springer Science & Business Media, 2013.

[143] N. Buduma and N. Locascio. *Fundamentals of deep learning: Designing next-generation machine intelligence algorithms.* ” O’Reilly Media, Inc.”, 2017.

[144] A. G. Barto, R. S. Sutton, and C. W. Anderson. “Neuronlike adaptive elements that can solve difficult learning control problems”. In: *IEEE transactions on systems, man, and cybernetics* 5 (1983), pp. 834–846.

[145] U. Michelucci. *Applied Deep Learning - A Case-Based Approach to Understanding Deep Neural Networks; Apress Media, LLC: New York, NY, USA, 2018.* Tech. rep. ISBN 978-1-4842-3789-2.

[146] J. Moolayil, J. Moolayil, and S. John. *Learn Keras for Deep Neural Networks.* Springer, 2019.

[147] D. Osinga. *Deep Learning Cookbook: Practical Recipes to Get Started Quickly.* ” O’Reilly Media, Inc.”, 2018.

[148] A. L. Maas, A. Y. Hannun, and A. Y. Ng. “Rectifier nonlinearities improve neural network acoustic models”. In: *Proc. icml.* Vol. 30. 1. Citeseer. 2013, p. 3.

[149] K. He, X. Zhang, S. Ren, and J. Sun. "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification". In: *Proceedings of the IEEE international conference on computer vision.* 2015, pp. 1026–1034.

[150] J. Kukačka, V. Golkov, and D. Cremers. "Regularization for deep learning: A taxonomy". In: *arXiv preprint arXiv:1710.10686* (2017).

[151] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. "Dropout: a simple way to prevent neural networks from overfitting". In: *The journal of machine learning research* 15.1 (2014), pp. 1929–1958.

[152] J. Sietsma and R. J. Dow. "Creating artificial neural networks that generalize". In: *Neural networks* 4.1 (1991), pp. 67–79.

[153] S. Ruder. "An overview of gradient descent optimization algorithms". In: *arXiv preprint arXiv:1609.04747* (2016).

[154] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. "Learning representations by back-propagating errors". In: *nature* 323.6088 (1986), pp. 533–536.

[155] D. C. Ciresan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber. "Flexible, high performance convolutional neural networks for image classification". In: *Twenty-second international joint conference on artificial intelligence.* 2011.

[156] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun. "What is the best multi-stage architecture for object recognition?" In: *2009 IEEE 12th international conference on computer vision.* IEEE. 2009, pp. 2146–2153.

[157] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems* 25 (2012), pp. 1097–1105.

[158] B. Miller and L. Ziemiański. "Identification of Mode Shapes of a Composite Cylinder Using Convolutional Neural Networks". In: *Materials* 14.11 (2021), p. 2801.

[159] M. Słoński, K. Schabowicz, and E. Krawczyk. "Detection of flaws in concrete using ultrasonic tomography and convolutional neural networks". In: *Materials* 13.7 (2020), p. 1557.

[160] K. O'Shea and R. Nash. "An introduction to convolutional neural networks". In: *arXiv preprint arXiv:1511.08458* (2015).

[161] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, et al. "Recent advances in convolutional neural networks". In: *Pattern Recognition* 77 (2018), pp. 354–377.

[162] D. Scherer, A. Müller, and S. Behnke. "Evaluation of pooling operations in convolutional architectures for object recognition". In: *International conference on artificial neural networks.* Springer. 2010, pp. 92–101.

[163] K. Simonyan and A. Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).

[164] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. "Going deeper with convolutions". In: *Proceed-*

ings of the IEEE conference on computer vision and pattern recognition. 2015, pp. 1–9.

[165] H. Salehinejad, S. Sankar, J. Barfett, E. Colak, and S. Valaee. “Recent advances in recurrent neural networks”. In: *arXiv preprint arXiv:1801.01078* (2017).

[166] T. Mikolov, A. Joulin, S. Chopra, M. Mathieu, and M. Ranzato. “Learning longer memory in recurrent neural networks”. In: *arXiv preprint arXiv:1412.7753* (2014).

[167] M. Schuster and K. K. Paliwal. “Bidirectional recurrent neural networks”. In: *IEEE transactions on Signal Processing* 45.11 (1997), pp. 2673–2681.

[168] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. “Learning phrase representations using RNN encoder-decoder for statistical machine translation”. In: *arXiv preprint arXiv:1406.1078* (2014).

[169] S. Hochreiter and J. Schmidhuber. “Long short-term memory”. In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[170] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber. “Learning precise timing with LSTM recurrent networks”. In: *Journal of machine learning research* 3.Aug (2002), pp. 115–143.

[171] A. Graves and J. Schmidhuber. “Framewise phoneme classification with bidirectional LSTM and other neural network architectures”. In: *Neural networks* 18.5-6 (2005), pp. 602–610.

[172] C.-B. Yun and J. Min. “Smart sensing, monitoring, and damage detection for civil infrastructures”. In: *KSCE Journal of Civil Engineering* 15.1 (2011), pp. 1–14.

[173] Z. Su, L. Ye, and Y. Lu. “Guided Lamb waves for identification of damage in composite structures: A review”. In: *Journal of sound and vibration* 295.3-5 (2006), pp. 753–780.

[174] J. S. Hall, P. McKeon, L Satyanarayan, J. E. Michaels, N. F. Declercq, and Y. H. Berthelot. “Minimum variance guided wave imaging in a quasi-isotropic composite plate”. In: *Smart Materials and Structures* 20.2 (2011), p. 025013.

[175] P. Kijanka, A. Manohar, F. Lanza di Scalea, and W. J. Staszewski. “Damage location by ultrasonic Lamb waves and piezoelectric rosettes”. In: *Journal of Intelligent Material Systems and Structures* 26.12 (2015), pp. 1477–1490.

[176] B. Park, Y.-K. An, and H. Sohn. “Visualization of hidden delamination and debonding in composites through noncontact laser ultrasonic scanning”. In: *Composites science and technology* 100 (2014), pp. 10–18.

[177] W. Leong, W. Staszewski, B. Lee, and F. Scarpa. “Structural health monitoring using scanning laser vibrometry: III. Lamb waves for fatigue crack detection”. In: *Smart Materials and Structures* 14.6 (2005), p. 1387.

[178] H. Sohn, D. Dutta, J. Yang, M. DeSimio, S. Olson, and E. Swenson. “Automated detection of delamination and disbond from wavefield images obtained using a scanning laser vibrometer”. In: *Smart Materials and Structures* 20.4 (2011), p. 045017.

[179] M. D. Rogge and C. Leckey. “Local guided wavefield analysis for characterization of delaminations in composites”. In: *AIP Conference Proceedings*. Vol. 1511. 1. American Institute of Physics. 2013, pp. 963–970.

[180] X. Zhang, L. Han, L. Han, and L. Zhu. “How well do deep learning-based methods for land cover classification and object detection perform on high resolution remote sensing imagery?” In: *Remote Sensing* 12.3 (2020), p. 417.

[181] A. Mardanshahi, V. Nasir, S. Kazemirad, and M. Shokrieh. “Detection and classification of matrix cracking in laminated composites using guided wave propagation and artificial neural networks”. In: *Composite Structures* 246 (2020), p. 112403.

[182] A. De Fenza, A. Sorrentino, and P. Vitiello. “Application of Artificial Neural Networks and Probability Ellipse methods for damage detection using Lamb waves”. In: *Composite Structures* 133 (2015), pp. 390–403.

[183] Z. Su and L. Ye. “Lamb wave-based quantitative identification of delamination in CF/EP composite structures using artificial neural algorithm”. In: *Composite Structures* 66.1-4 (2004), pp. 627–637.

[184] D. Perfetto, A. De Luca, M. Perfetto, G. Lamanna, and F. Caputo. “Damage detection in flat panels by guided waves based artificial neural network trained through finite element method”. In: *Materials* 14.24 (2021), p. 7602.

[185] B. Feng, D. J. Pasadas, A. L. Ribeiro, and H. G. Ramos. “Locating defects in anisotropic CFRP plates using ToF-based probability matrix and neural networks”. In: *IEEE Transactions on Instrumentation and Measurement* 68.5 (2019), pp. 1252–1260.

[186] D. Chetwynd, F. Mustapha, K. Worden, J. Rongong, S. Pierce, and J. Dulieu-Barton. “Damage localisation in a stiffened composite panel”. In: *Strain* 44.4 (2008), pp. 298–307.

[187] J. Melville, K. S. Alguri, C. Deemer, and J. B. Harley. “Structural damage detection using deep learning of ultrasonic guided waves”. In: *AIP Conference Proceedings*. Vol. 1949. 1. AIP Publishing LLC. 2018, p. 230004.

[188] I. Tabian, H. Fu, and Z. Sharif Khodaei. “A convolutional neural network for impact detection and characterization of complex composite structures”. In: *Sensors* 19.22 (2019), p. 4933.

[189] V. Ewald, R. M. Groves, and R. Benedictus. “DeepSHM: A deep learning approach for structural health monitoring based on guided Lamb wave technique”. In: *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2019*. Vol. 10970. International Society for Optics and Photonics. 2019, 109700H.

[190] H. Song and Y. Yang. “Noncontact super-resolution guided wave array imaging of subwavelength defects using a multiscale deep learning approach”. In: *Structural Health Monitoring* (2020).

[191] M. Rautela and S Gopalakrishnan. “Ultrasonic guided wave based structural damage detection and localization using model assisted convolutional and recurrent neural networks”. In: *Expert Systems with Applications* (2020), p. 114189.

[192] P. Kudela, J. Moll, and P. Fiborek. “Parallel spectral element method for guided wave based structural health monitoring”. In: *Smart Materials and Structures* 29.9 (2020), p. 095010. ISSN: 0964-1726. DOI: `10.1088/1361-665X/ab9e10`. URL: `https://iopscience.iop.org/article/10.1088/1361-665X/ab9e10`.

[193] P. Kudela, J. Moll, and P. Fiborek. “Parallel spectral element method for guided wave based structural health monitoring”. In: *Smart Materials and Structures* 29.9 (2020), p. 095010.

[194] P. Kudela and A. Ijjeh. *Synthetic dataset of a full wavefield representing the propagation of Lamb waves and their interactions with delaminations.* 2021.

[195] S. H. I. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo. “Convolutional LSTM network: A machine learning approach for precipitation nowcasting”. In: *Advances in neural information processing systems.* 2015, pp. 802–810.

[196] S. A. Taghanaki, K. Abhishek, J. P. Cohen, J. Cohen-Adad, and G. Hamarneh. “Deep semantic segmentation of natural and medical images: a review”. In: *Artificial Intelligence Review* 54.1 (2021), pp. 137–178.

[197] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez. “The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition.* Vol. 2016-Decem. Packt Publishing, 2016, pp. 3234–3243. ISBN: 9781467388504. DOI: `10.1109/CVPR.2016.352`.

[198] O. Miksik, V. Vineet, M. Lidegaard, R. Prasaath, M. Nießner, S. Golodetz, S. L. Hicks, P. Pérez, S. Izadi, and P. H. Torr. “The semantic paintbrush: Interactive 3D mapping and recognition in large outdoor spaces”. In: *Conference on Human Factors in Computing Systems - Proceedings.* Vol. 2015-April. Packt Publishing, 2015, pp. 3317–3326. ISBN: 9781450331456. DOI: `10.1145/2702123.2702222`.

[199] A. Milioto, P. Lottes, and C. Stachniss. “Real-Time Semantic Segmentation of Crop and Weed for Precision Agriculture Robots Leveraging Background Knowledge in CNNs”. In: *Proceedings - IEEE International Conference on Robotics and Automation.* IEEE. Packt Publishing, 2018, pp. 2229–2235. ISBN: 9781538630815. DOI: `10.1109/ICRA.2018.8460962`. arXiv: `1709.06764`.

[200] R. Szeliski. *Computer vision: algorithms and applications.* Packt Publishing, 2010.

[201] S. Minaee, Y. Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz, and D. Terzopoulos. “Image segmentation using deep learning: A survey”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).

[202] S. Ghosh, N. Das, I. Das, and U. Maulik. “Understanding deep learning techniques for image segmentation”. In: *ACM Computing Surveys* 52.4 (2019), pp. 1–35. ISSN: 15577341. DOI: `10.1145/3329784`. arXiv: `1907.06119`.

[203] S. Santurkar, D. Tsipras, A. Ilyas, and A. Madry. “How does batch normalization help optimization?” In: *Advances in Neural Information Processing Systems.* Vol. 2018-Decem. 2018, pp. 2483–2493. arXiv: `1805.11604`.

[204] F. Chollet. *Keras.* `https://github.com/fchollet/keras/`. 2015.

[205] D. P. Kingma and J. Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).

[206] E. J. Candes, J. K. Romberg, and T. Tao. “Stable signal recovery from incomplete and inaccurate measurements”. In: *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences* 59.8 (2006), pp. 1207–1223.

[207] D. L. Donoho. “Compressed sensing”. In: *IEEE Transactions on information theory* 52.4 (2006), pp. 1289–1306.

[208] S. S. Chen, D. L. Donoho, and M. A. Saunders. “Atomic decomposition by basis pursuit”. In: *SIAM review* 43.1 (2001), pp. 129–159.

[209] E. Van Den Berg and M. P. Friedlander. “Probing the Pareto frontier for basis pursuit solutions”. In: *Siam journal on scientific computing* 31.2 (2009), pp. 890–912.

[210] E Berg. *van den & Friedlander, MP SPGL1: A solver for large-scale sparse reconstruction.* 2019.

[211] M. Lustig, D. Donoho, and J. M. Pauly. “Sparse MRI: The application of compressed sensing for rapid MR imaging”. In: *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine* 58.6 (2007), pp. 1182–1195.

[212] Z. Gao, L. Dai, S. Han, I Chih-Lin, Z. Wang, and L. Hanzo. “Compressive sensing techniques for next-generation wireless communications”. In: *IEEE Wireless Communications* 25.3 (2018), pp. 144–153.

[213] F. J. Herrmann, M. P. Friedlander, and O. Yilmaz. “Fighting the curse of dimensionality: Compressive sensing in exploration seismology”. In: *IEEE Signal Processing Magazine* 29.3 (2012), pp. 88–100.

[214] O. Mesnil and M. Ruzzene. “Sparse wavefield reconstruction and source detection using compressed sensing”. In: *Ultrasonics* 67 (2016), pp. 94–104.

[215] A. Perelli, T. Di Ianni, L. De Marchi, N. Testoni, and N. Speciale. “Compressive sensing with warped frequency models in lamb waves damage detection procedures”. In: *2012 IEEE International Ultrasonics Symposium.* IEEE. 2012, pp. 154–157.

[216] T. Di Ianni, L. De Marchi, A. Perelli, and A. Marzani. “Compressive sensing of full wave field data for structural health monitoring applications”. In: *IEEE transactions on ultrasonics, ferroelectrics, and frequency control* 62.7 (2015), pp. 1373–1383.

[217] Y. K. Esfandabadi, L. De Marchi, N. Testoni, A. Marzani, and G. Masetti. “Full wavefield analysis and damage imaging through compressive sensing in lamb wave inspections”. In: *IEEE transactions on ultrasonics, ferroelectrics, and frequency control* 65.2 (2017), pp. 269–280.

[218] R. Dahl, M. Norouzi, and J. Shlens. “Pixel recursive super resolution”. In: *Proceedings of the IEEE international conference on computer vision.* 2017, pp. 5439–5448.

[219] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu. "Residual dense network for image super-resolution". In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2018, pp. 2472–2481.

[220] Z. Zhang and V. Sze. "FAST: A framework to accelerate super-resolution processing on compressed videos". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops.* 2017, pp. 19–28.

[221] B. Park and H. Sohn. "Reconstruction of laser ultrasonic wavefield images from reduced sparse measurements using compressed sensing aided super-resolution". In: *AIP Conference Proceedings.* Vol. 1806. 1. AIP Publishing. 2017.

[222] Y. Keshmiri Esfandabadi, M. Bilodeau, P. Masson, and L. De Marchi. "Deep learning for enhancing wavefield image quality in fast non-contact inspections". In: *Structural Health Monitoring* 19.4 (2020), pp. 1003–1016.

[223] A. Ijjeh, S. Ullah, M. Radzienski, and P. Kudela. "Deep learning super-resolution for the reconstruction of full wavefield of Lamb waves". In: *Mechanical Systems and Signal Processing* 186 (2023), p. 109878.

[224] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang. "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network". In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2016, pp. 1874–1883.

[225] S. Duczek and U. Gabbert. "Anisotropic hierarchic finite elements for the simulation of piezoelectric smart structures". In: *Engineering Computations* 30.5 (2013), pp. 682–706. ISSN: 0264-4401. DOI: 10.1108/EC-08-2013-0005.

[226] C. Anitescu, C. Nguyen, T. Rabczuk, and X. Zhuang. "Isogeometric analysis for explicit elastodynamics using a dual-basis diagonal mass formulation". In: *Computer Methods in Applied Mechanics and Engineering* 346 (2019), pp. 574–591. ISSN: 00457825. DOI: 10.1016/j.cma.2018.12.002.

[227] F. Mossaiby, M. Joulaian, and A. Düster. "The spectral cell method for wave propagation in heterogeneous materials simulated on multiple GPUs and CPUs". In: *Computational Mechanics* 63.5 (2019), pp. 805–819. ISSN: 0178-7675. DOI: 10.1007/s00466-018-1623-4.

[228] W. Ostachowicz, P. Kudela, M. Krawczuk, and A. Zak. *Guided Waves in Structures for SHM: The Time-Domain Spectral Element Method.* Wiley, 2012, pp. 1–337. ISBN: 9780470979839. DOI: 10.1002/9781119965855.

[229] W. H. L. Pinaya, S. Vieira, R. Garcia-Dias, and A. Mechelli. "Autoencoders". In: *Machine learning.* Elsevier, 2020, pp. 193–208.

[230] E.-R. Ardelean, A. Coporîie, A.-M. Ichim, M. Dînșoreanu, and R. C. Mureșan. "A study of autoencoders as a feature extraction technique for spike sorting". In: *Plos one* 18.3 (2023), e0282810.

[231] S. Jo, C. Park, D.-W. Ryu, and S. Ahn. "Adaptive surrogate estimation with spatial features using a deep convolutional autoencoder for CO2 geological sequestration". In: *Energies* 14.2 (2021), p. 413.

[232] S. Nikolopoulos, I. Kalogeris, and V. Papadopoulos. "Non-intrusive surrogate modeling for parametrized time-dependent partial differential equations using

convolutional autoencoders". In: *Engineering Applications of Artificial Intelligence* 109 (2022), p. 104652.

[233] J. K. Sharma, R. Soman, P. Kudela, E. Chatzi, and W. Ostachowicz. "Wave Propagation Modeling via Neural Networks for Emulating a Wave Response Signal". In: *European Workshop on Structural Health Monitoring.* Springer. 2022, pp. 512–520.

[234] S. A. Zargar and F.-G. Yuan. "Impact diagnosis in stiffened structural panels using a deep learning approach". In: *Structural Health Monitoring* 20.2 (2021), pp. 681–691.

[235] H. Peng, J. Yan, Y. Yu, and Y. Luo. "Structural surrogate model and dynamic response prediction with consideration of temporal and spatial evolution: An encoder–decoder ConvLSTM network". In: *International Journal of Structural Stability and Dynamics* 21.10 (2021), p. 2150140.

[236] J. Keneddy and R. C. Eberhart. "Particle Swarm Optimization". In: *Proceedings of the IEEE International Joint Conference on Neural Networks.* 1995, pp. 1942–1948. ISBN: 0780327683.

[237] V. Miranda, Lester James, A. Moser, and S. K. Cronin. *PYSWARMS a research toolkit for Particle Swarm Optimization in Python.* URL: `https://pyswarms.readthedocs.io/`.