DEVELOPMENT AND APPLICATION OF BIG DATA ANALYTICS AND ARTIFICIAL INTELLIGENCE FOR STRUCTURAL HEALTH MONITORING AND METAMATERIAL DESIGN

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Rih-Teng Wu

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

December 2020

Purdue University

West Lafayette, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF DISSERTATION APPROVAL

ACKNOWLEDGMENTS

The author appreciates the committee members, Dr. Jahanshahi, Dr. Dyke, Dr. Delp, Dr. Bertino, Dr. Irfanoglu and Dr. Semperlotti for providing valuable insights and comments to this dissertation. The author especially thanks Dr. Jahanshahi, Dr. Bertino and Dr. Semperlotti for collaborating on part of the works in this dissertation. The author also appreciates the Electric Power Research Institute (EPRI) in United States and the National Center for Research on Earthquake Engineering (NCREE) in Taiwan for providing datasets used in this dissertation. The views and opinions of the author expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

ABSTRACT

Wu, Rih-Teng Ph.D., Purdue University, December 2020. Development and Application of Big Data Analytics and Artificial Intelligence for Structural Health Monitoring and Metamaterial Design . Major Professor: Mohammad R. Jahanshahi.

Recent advances in sensor technologies and data acquisition platforms have led to the era of Big Data. The rapid growth of artificial intelligence (AI), computing power and machine learning (ML) algorithms allow Big Data to be processed within affordable time constraints. This opens abundant opportunities to develop novel and efficient approaches to enhance the sustainability and resilience of Smart Cities. This work, by starting with a review of the state-of-the-art data fusion and ML techniques, focuses on the development of advanced solutions to structural health monitoring (SHM) and metamaterial design and discovery strategies. A deep convolutional neural network (CNN) based approach that is more robust against noisy data is proposed to perform structural response estimation and system identification. To efficiently detect surface defects using mobile devices with limited training data, an approach that incorporates network pruning into transfer learning is introduced for crack and corrosion detection. For metamaterial design, a reinforcement learning (RL) and a neural network based approach are proposed to reduce the computation efforts for the design of periodic and non-periodic metamaterials, respectively. Lastly, a physics-constrained deep auto-encoder (DAE) based approach is proposed to design the geometry of wave scatterers that satisfy user-defined downstream acoustic 2D wave fields. The robustness of the proposed approaches as well as their limitations are demonstrated and discussed through experimental data or/and numerical simulations. A roadmap for future works that may benefit the SHM and material design research communities is presented at the end of this dissertation.

1. INTRODUCTION

1.1 Motivations

During the past decades, significant efforts have been dedicated to develop reliable methods in structural health monitoring (SHM) [1–11]. The health assessment for the target structure of interest is achieved through the interpretation of collected data. At the beginning of the 21st century, the rapid advances in sensor technologies and data acquisition platforms have led to the new era of Big Data, where a huge amount of heterogeneous data is collected by a variety of sensors. The increasing accessibility and diversity of the data resources provide new opportunities for SHM while the aggregation of information obtained from multiple sensors to make robust decisions remains a challenging problem. Recently, the rapid developments of artificial intelligence (AI) techniques have brought a huge impact across numerous disciplines, including but not limited to engineering applications, human-computer interactions, biomedical sciences, intelligent gaming, and material discovery. Advances in machine learning (ML) algorithms and computing powers not only enable the processing of Big Data within affordable time constraints, but also discover the embedded patterns from the data automatically without imposing human subjective judgements. By learning the information from various forms of data, e.g., 1-D time series signals and 2-D images, the decision making is potentially more robust and can be achieved more efficiently. In this work, data analysis tools are developed based on ML techniques including multi-layer perceptron (MLP) [12], deep convolutional neural network (CNN) [13] and reinforcement learning (RL) [14], for applications in SHM and material design.

The current practices in building and bridge inspections are labor-intensive and time-consuming. One viable solution is to infer the health state of the structure based on the vibration measurements collected from the sensors instrumented on the structure. Given the input ground excitation and structural output responses, a numerical model of the structure is established based on a proposed CNN, as it is often difficult to acquire a precise finite element model (FEM) for the structure in real world. After training with the input and output response measurements, the proposed CNN model is used to estimate the dynamic response and the fundamental frequencies of the structure. The performance of the CNN model is compared with a baseline model proposed in [15]. Comprehensive experiments including numerical simulations and experimental data are used to validate the robustness of the proposed CNN-based approach.

Another critical issue is how to develop efficient data analysis tools for SHM in the notion of Internet of Things (IoT) [16]. The IoT consists of a set of edge sensors and central server units. In IoT, data analysis can take place at the edge or at a server, depending on the application's requirements [17, 18]. For civil infrastructure, the inspection target is usually enormous in size or length. In this context, data analysis achieved by swarms of autonomous inspection robots, i.e., edge sensors, can replace the current manual inspections and result in efficient decision making [19–22]. However, these small robots are typically limited in computing and memory resources. The incorporation of deep learning-based approaches will be infeasible without appropriate considerations in the algorithm. To this end, this study introduces a solution based on network pruning [23] to utilize pre-trained deep CNNs for efficient edge computing. Results from comprehensive experiments on two pre-trained networks (i.e., VGG16 [24] and ResNet18 [25]) and two types of prevalent surface defects (i.e., crack and corrosion) are presented and discussed in details with respect to performance, memory demands, and the inference time for damage detection. It is demonstrated that the proposed approach significantly enhances resource efficiency without decreasing damage detection performance [26, 27].

Besides applications in SHM, this work further proposes AI-based approaches for material discovery and design. Conventional material design processes require precise physical modeling of the material along with the extensive use of optimization methods to achieve target performance. This approach is computationally intensive and severely limits the possibility to explore the vast design space offered by engineered materials. In addition, these optimization techniques typically target properties defined in the physical space (e.g., displacements and stresses) and do not allow direct access to properties in a transformed space. A typical example is the inability to prescribe a target dynamic behavior in the reciprocal-space, where quantities like frequency-wavenumber dispersion and band structure are defined. This study presents two AI-based design frameworks for the design of periodic and non-periodic material systems. For periodic materials, a RL-based approach is proposed in order to design the unit-cell properties according to a user-defined dispersion behavior. For non-periodic materials, a neural network based approach capable of learning the behavior of individual material units is presented. In this case, the design of the engineered material is achieved by assembling the neural network representation of individual units within a general optimization framework that targets a user-defined material response. Interestingly, this latter framework is capable of synthesizing different material assemblies (based on the available cells) while requiring only one-time network training. Numerical examples are provided for both the periodic and non-periodic material designs to demonstrate the performance of the proposed framework.

Furthermore, a physics-constrained deep auto-encoder (DAE) based approach is proposed to design the geometry of wave scatterers that satisfy the target downstream pressure fields. The control of acoustic and elastic waves via material design has been an interesting subject with various applications such as non-destructive evaluation of structural components, biomedical devices, high-resolution imaging, radar, and remote sensing. To date, there is still a lack of powerful and efficient design methodologies since the conventional optimization-based approaches suffer from the computation burden in parameter search whenever a design query is made. The proposed network consists of a geometry estimator and a DAE that provides the geometry estimator with physics constraints during the learning process. By joint optimization, the estimation of scatterer geometry is strengthened with the latent representations of the target pressure fields learned by the DAE. Once the training is finished, the design inference is quasi-instantaneous given a target 2D pressure fields. The generalization capability of the proposed network is further validated through a dataset generated with new shapes of wave scatterers. Numerical simulations and design examples are presented to demonstrate the robustness of the proposed approach.

1.2 Scope

Chapter 2 provides a comprehensive review for the state-of-the-art data fusion and ML techniques for SHM applications. Challenges of each technique are addressed, and a roadmap is provided for future research. In Chapter 3, a CNN-based approach is introduced for the first time for structural dynamic response estimation and system identification. The performance of the proposed approach as well as the physical interpretation of the CNN network are discussed in detail. Chapter 4 proposes an efficient damage detection approach for edge computing based on network pruning combined with transfer learning. Case studies including crack and corrosion detection are conducted to investigate the performance of the proposed approach in terms of the required inference time, memory storage demands, and damage detection accuracy. In Chapter 5, two design frameworks for periodic and non-periodic materials are proposed. For periodic materials, a RL-based approach is developed to determine the properties of each material unit, according to a user-defined behavior in a transformed space, e.g., a dispersion with a band gap in the frequency domain. For non-periodic materials, a fully-connected NN is used to model the material unit, and the design of the metamaterial is achieved by the combination of the duplicated networks. Chapter 6 presents a DAE-based approach to design the geometry of acoustic wave scatterers that achieve a user-defined downstream 2D pressure fields. The proposed approach is the first demonstration of multi-objective inverse design for wave scattering applications. A summary of conclusions and future works is provided in Chapter 7.

2. REVIEW OF DATA FUSION AND MACHINE LEARNING APPROACHES IN STRUCTURAL HEALTH MONITORING AND SYSTEM IDENTIFICATION

2.1 Introduction

2.1.1 Background

All civil structures are inevitably faced with aging problems, possible excessive loading conditions, inappropriate usages, or natural hazards such as earthquake and hurricane. It is crucial to identify the current health condition of a target structure periodically and after the strike of a natural hazard. In the recent decades, several researchers have put their efforts in monitoring the health state of structural systems to identify existing damage in structures. A damage in a structural system can be the result of changes in material properties, geometric configuration, or boundary conditions, which typically influences the performance of the structural system. In 2007, Farrar and Worden [28] defined SHM as "the process of implementing a damage identification strategy for aerospace, civil and mechanical engineering." Recent advances in sensor and information technologies have led to the era of Big Data, that have enhanced data acquisition and that inevitably lead to more opportunities in SHM [29, 30]. The increasing availability of heterogeneous data (e.g., displacement, acceleration, strain, global positioning system (GPS), LiDAR, depth [31, 32], ultrasonic, ground penetration radar (GPR), infrared, microwave, etc.) provides more information to the scientists to deal with conventional problems. As a result, how to integrate information from multiple sources and make a more robust decision for applications in SHM is an important and yet challenging task. This chapter reviews state-of-the-art ML and data fusion algorithms that can be used in SHM problems.

2.1.2 Motivation

Data fusion is the integration of various types of data to obtain more information from the combined data instead of considering each dataset separately [33, 34]. It can be used to generate new raw data or more informative new data based on the original raw data. Once the new data is generated, it is often expected to be more helpful in decision making process than using the original datasets. The concept of data fusion has been widely applied in various disciplines including but not limited to automated target recognition, environmental monitoring, robotics, medical applications, quality control of manufacturing process, and condition monitoring of complex machinery [35]. For instance, in the case of traffic control, the traffic state for a certain area is monitored through the combination of acoustic, image, and other sensor data measured near the road side. Discussions regarding the applications such as advanced traveler information systems, automatic incident detection, advanced driver assistance, network control, crash analysis and prevention, traffic demand estimation, traffic forecasting and traffic monitoring, and accurate position estimation are reviewed in [36]. In the context of SHM, data fusion is adopted to enhance the decision making for the health evaluation of the structures. However, due to the complex nature of the real world applications, the uncertainties and imperfections must be considered during the data processing. In this regard, data fusion helps reduce the uncertainty by increasing the information completeness [37, 38]. The outcome of a decision making process is usually essential for risk analysis.

In general, data integration can be achieved in various levels of fusion depending on the task being performed. There are three levels of data fusion commonly used: data-level, feature-level, and decision-level [39–41]. In data-level fusion, the raw data from multiple sources are directly combined before further process. These raw data must have the same physical meaning (i.e., measuring the same physical quantities). For instance, sound signal and image data cannot be integrated at data-level. In feature-level fusion, the input data can be heterogeneous (i.e., measuring different physical parameters). Statistical features or signatures are often extracted from the original raw data, and these features are selected or concatenated prior to further analysis. In decision-level fusion, the final assessment result is obtained by integrating the decisions, that are obtained from different data sources, through particular combination rules.

In this study, the author presents a comprehensive review of popular data fusion concepts and their applications in SHM. Among all these techniques, data registration, Bayesian probabilistic approaches, Dempster-Shafer evidential approach, fuzzy reasoning, state estimation, machine learning algorithms, and weighted combinations are the most common data fusion methods applied in SHM. Furthermore, challenges and opportunities for using data fusion techniques are discussed.

2.1.3 Scope

The organization of the rest of this chapter is as following. In Section 2.2, the theoretical basis and applications of common data fusion techniques in SHM are provided. Section 2.3 discusses the challenges for data fusion in SHM as well as the road map for future research.

2.2 Data Fusion Techniques in SHM

This section starts with the introduction of the benefits stemmed from data fusion, the essential data preprocessing step before applying data fusion, followed by the review of the popular techniques that have been used to integrate data for SHM applications. For each data fusion technique, the basic concept of the technique as well as its recent applications in SHM are presented. An illustrative example for data fusion in structural response estimation and damage detection is provided at the end of this section.

2.2.1 Prior to Data Fusion

Why Data Fusion? A Perspective from the Observability and Identifiability of System

When dealing with a problem of interest, one would inevitably need to identify whether a solution to the problem exists given the available information. There could be too many unknowns in the system and, therefore, the augmentation of data to help with problem solving might be required. This leads to the concepts of observability and identifiability. The observability of a system is defined as whether the states of a system can be identified by a set of measurements. On the other hand, the identifiability of a system is defined as whether the measurements lead to unique or finite solutions for the parameters of the system [42]. Obviously, the question of observability is indivisible from identifiability since the state of the system is determined by the parameters of the system.

In general, data fusion improves the observability and the identifiability of a system by providing additional information for the decision makers. For instance, the observability of a 2-DOF system is discussed under the availabilities of acceleration and displacement measurements in [42]. It is shown that the system is observable only when the displacement measurement of the first mode and the acceleration measurement of the second mode are available. Moreover, it is demonstrated that using multiple sensors with different characteristics is potentially more beneficial than using one specific type of sensor alone. A particular sensor with its own inherent noise and Nyquist frequency may prevent the sensor from measuring the signal outside its bandwidth. For instance, GPS sensors for displacement measurements typically perform better in low frequencies, while the accelerometers perform better in high frequencies. By exploiting the measurements from both GPS and accelerometers, the accuracy of displacement estimation is enhanced since the GPS measurements implicitly impose constraints on the displacement computed from acceleration measurement, hence the drifting in displacement induced by double integration is eliminated. This addresses the underlying benefit of heterogeneous data fusion [43, 44].

Preprocessing - Data Registration

When dealing with the aggregation of information from multiple sensors, the registration problem occurs if the sensors are not located at the same position [45]. For instance, the same target point may appear at different pixel coordinates of the images captured from two cameras at different locations. Before further exploitation of the data, the first step is to register the sensor readings into a common frame of reference [46–48]. In other words, the data registration is an essential step to transform the data obtained from multiple sensors onto a common coordinate system.

Although the registration techniques often work with 2D or 3D data, they can be applied to 1D data as well. In general, data registration is achieved by first searching the best similarity between the recordings, and then the recordings are mapped to the common frame of reference based on the similarity. The similarity can be determined by finding the maximum correlation between the sensor readings, template matching, regression, control points, or features invariant to the sensor location. Depending on the data type, mapping functions like bilinear, warping, or least square optimization could be employed if necessary. A survey of the sensor registration techniques is presented in [49].

It is noted that data registration is not limited to the transformation of the original sensor recording. The registration of higher level information onto a common coordinate system can also be helpful for decision making as well. In [50], the aggregation of detection results from different video frames improves the accuracy for crack detection since a crack that is missed in a frame would be detected in other frames. Instead of considering only one video frame, the crack detection results from different video frames are registered into a global spatiotemporal coordinate system, and the final detection result is enhanced by exploiting the underlying spatiocoherence of the data.

2.2.2 Bayesian Theory

The basis of Bayesian theory is originated from conditional probability that expresses the probability of occurrence of event x given event y, which is denoted as $P(x \mid y)$. For data fusion, Bayes' rule gives an expression for the probability of a hypothesis being true given some prior knowledge or observations [41, 51]. When it comes to SHM, a hypothesis could be some specified damage states of a structure, or whether a certain damage exists or not. An observation may come from a series of experiments or a set of sensor measurements. By computing the probability of each damage state given the available information, decision makers can make better decisions about the retrofit plan for the structures. The fundamental equation of Bayes' rule is given in Equation 2.1.

$$P(H_i \mid D) = \frac{P(D \mid H_i)P(H_i)}{P(D)} = \frac{P(D \mid H_i)P(H_i)}{\sum\limits_{i} P(D \mid H_i)P(H_i)}, \tag{2.1}$$

where $P(H_i \mid D)$ is the posterior probability that hypothesis H_i is true given available data D. $P(D \mid H_i)$ is the likelihood function, and $P(H_i)$ is the prior probability. Figure 2.1 illustrates the schematic relationship between the prior, likelihood, and the posterior distributions. In this example, the prior distribution is normally distributed with mean 50 and standard deviation 15, and the likelihood function is normally distributed with mean 70 and standard deviation 5. Since the variance of prior distribution is high (i.e., not informative), the resulting posterior distribution tends to be closer to the likelihood function.

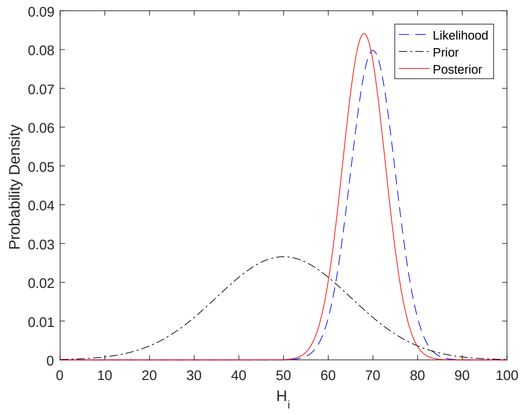


Fig. 2.1.: Schematic illustration of prior, likelihood, and the posterior distribution.

Assuming the data are collected independently from n different sensors, the posterior probability of H_i given data $D^1 \cap D^2 \cap ... \cap D^n$ can be expressed as Equation 2.2, known as the Naïve Bayes rule:

$$\begin{aligned} P(H_i \mid D^1 \cap D^2 \cap ... \cap D^n) &= \frac{P(D^1 \cap D^2 \cap ... \cap D^n \mid H_i)P(H_i)}{\sum\limits_i P(D^1 \cap D^2 \cap ... \cap D^n \mid H_i)P(H_i)} \\ &= \frac{P(D^1 \mid H_i)P(D^2 \mid H_i)...P(D^n \mid H_i)P(H_i)}{\sum\limits_i P(D^1 \mid H_i)P(D^2 \mid H_i)...P(D^n \mid H_i)P(H_i)}, \end{aligned} \tag{2.2}$$

where D^n corresponds to the n^{th} observation (i.e., declaration) made by the n^{th} sensor, and H_i is the i^{th} hypothesis.

Prior Probability

There are two types of prior probabilities $P(H_i)$. One is subjective (i.e., informative) prior while the other one is objective (i.e., non-informative) prior. The subjective prior is established based on one's subjective belief or prior knowledge about the hypothesis while the objective prior is established based on little or indirect information about the hypothesis. General rules for selecting prior probability are [52]: (a) include as many reasonable priors as possible; (b) eliminate unreasonable priors; (c) confirm that the prior does not require information that is difficult to elicit; and (d) verify the capability to compute measures of robustness without much difficulty.

To address the effects of priors/model selection, sensitivity analysis is essential in Bayesian inference. The performances of the inference under different reasonable choices of prior distribution or probability models should be investigated through sensitivity analysis [52].

Posterior Probability

In general, there are two procedures for estimating the posterior probability $P(H_i \mid D)$ [52, 53]:

(a) Parametric Identification: Formulate a class of mathematical models for a specific system or physical phenomenon, and identify the unknown parameters in the model. In this step, the parameters of interest as well as their relation to the problem are determined. For instance, whether a building is damaged or not can be associated with the amount of reduction in the column stiffness, or changes in the mode frequencies of the building.

(b) Model Selection: Select an appropriate class of mathematical models for parametric identification. The selection is usually determined by user's judgement, which involves the choice of the likelihood function and the prior distribution.

Considerations in Model Selection

The performance of Bayesian-based applications highly depends on the selection of appropriate models. This requires a thorough uncertainty formulation and assumptions made with solid engineering knowledge. Without a fundamental understanding of the underlying system or physical phenomenon, a suitable model class will be hard to be constructed. For instance, the prior distribution of the yielding strength of steel may follow a log-normal distribution with median and variance determined from experimental data [54]. In general, the considerations for choosing an appropriate model are:

(a) Prefer simpler model. A complicated model may work well on the training data but fail to predict the outcome of the new data in the future.

(b) The prior should not be absorbed into the normalizing constant (i.e., the denominator in Equation 2.2) when determining the model class.

(c) A more informative prior based on previous experience is preferred.

(d) Model selection favors a model with physical meaning. It is difficult to estimate the uncertain parameters of an empirical model prior to data acquisition. Consequently, the number of choices of prior distribution are relatively higher for an empirical model.

After the formulation of the likelihood function and the prior distribution, the posterior distribution can be obtained from Equation (2.1). It is shown in [55] that for globally identifiable cases, the posterior distribution has the asymptotic behavior given a large amount of data [56] and therefore can be approximated accurately by a Gaussian distribution. For the general cases where the posterior may not be approximate by Gaussian distribution, it is often computationally intensive to obtain a close form expression of the posterior distribution when the dimensionality of the parameter space is high. To deal with this issue, sampling algorithm such as Markov Chain Monte Carlo (MCMC) and Gibbs sampler [57–60] are often adopted to generate the samples from the posterior distribution. The MCMC algorithm avoids the computation of the denominator in (2.1) by exploiting the relative probability density between two parameter vectors. For more discussion regarding model selection, the reader is referred to [53].

Application

In [61, 62], a Bayesian framework is proposed for SHM. In their paper, the probability that the concerned model stiffness parameters would be less than a fraction of the undamaged stiffness parameters given the available modal data is computed based on Bayes theorem. The uncertainty of model parameters are introduced as well. The damage state of the structure is evaluated through the probability of reduction in structural parameters (e.g., stiffness of the structure). In [63], the safety of a bridge structure is updated with measured dynamic responses. In this study, measurements of ground excitation and acceleration at the midspan of the bridge deck are aggregated to infer the change in structural reliability. In [64], a Bayesian approach is proposed for structural system identification and damage detection. By using the simulated floor acceleration responses, the proposed approach is demonstrated to be robust against measurement noise, and the uncertainty of the structural parameters can be quantified. However, damage with little severity may not be detected. Moreover, the employed optimization algorithm is only applicable for globally identifiable cases, and the identification performance depends on the selection of appropriate structural model. In [65], model updating of building structures is achieved through Bayesian methodology with noisy incomplete modal data. The proposed approach uses a computationally efficient iterative scheme to avoid the convergence difficulties faced in non-linear optimization problem. System natural frequencies, mode shapes and stiffness are estimated through the proposed method using numerical examples. In [66], the simulated acceleration measurements are employed to perform the system identification of buildings. The issue of high-dimension optimization is relieved by the transition MCMC algorithm, and the simulation results indicate decent performances achieved by th proposed approach. However, the identification results could be altered significantly due to poor knowledge about the damping in the structure. In [67, 68], modal data computed from vibration measurements is used to quantify and localize the damage in bridges. It is demonstrated though a laboratory test that the proposed approach depends on the model classes, damage magnitudes and locations, as well as the configuration of sensor instrumentation. In [69], a Naïve Bayes based approach is proposed to monitor the health condition of building structures under various ground excitations. Acceleration measurements collected from the strong ground motion and the velocity response recorded from the micro-vibrations are used to detect damage in structures under the earthquake mode and the micro-vibration mode, respectively. Although the performance in the earthquake mode is not as good as the micro-vibration mode, experimental results have shown that the proposed approach is able to rapidly determine the condition and location of the damage. In [70], the crack length of a steel specimen is estimated through Bayes theorem. Simulated samples generated through Markov Chain Monte Carlo algorithm and the experimental data are fused to derive the probability of a certain crack length at a specific number of loading cycles. The failure probability is defined as the probability that the crack length is larger than a user defined critical crack length. In [71], a theoretical probability framework is proposed to estimate the reliability index of the composite wing structure at future times. In this framework, Bayesian inference is employed to evaluate the current damage state of the system and update the joint probability density function (PDF) of the damage size at various locations. After that, probabilistic models for future aerodynamic loads and damage propagation are used to predict the joint PDF of damage extents at future times. Finally, the local and global failure criteria are combined to compute the lower bound and the upper bound of the probability of system failure at future times. In [72], an experimental validation is conducted to test the proposed probability framework in [71]. Multiple crack propagation trajectories recorded from a series of fatigue experiments are employed to estimate the remaining fatigue life of the specimen. In [73], the health condition of a rotating machine is monitored by means of a two-stage Bayesian inference data fusion technique. Prior to data fusion, time domain features and frequency domain features are extracted from the signals of interest including acceleration, current, voltage, and temperature. The overall health assessment is achieved by first performing local data fusion that integrates the information from specific components, then perform global data fusion that combines the diagnostics results of the specific components.

In [74], recursive Bayesian framework is used to update the parameters for the crack growth model, as well as the probability distribution of the crack size and crack growth rate. Data from acoustic emission, periodic inspection, and the empirical crack growth model are fused to update the crack information for better health assessment. In [75], the accuracy of the load estimation for a cable-stayed bridge is enhanced by Bayesian inference. The elongation recorded by the fiber optical sensors, the load measured by the elasto-magnetic sensors as well as the thermal expansion information are fused to give a better estimation of the cable load. In [76], a Bayesian based model updating approach is proposed to refine the finite element model of a coupled floor slab. In this study, the ambient acceleration measurements collected through field test are used to update the structural model. It is demonstrated that the model with partition walls achieves better performances in identifying the natural frequencies and mode shapes. In [77], Bayesian fusion is employed in the scope of masonry structures survey. The probability distribution for the elastic modulus of the granite blocks is updated by integrating the information from prior knowledge, the data from sonic test, the data from ultrasonic test, as well as the data from compressive strength test. In [78], the acoustic emission source in plate-like structures is localized based on a Bayesian approach. Multiple two-dimensional Gaussian distributions obtained from time difference data at different frequencies are merged to provide the final probability distribution of the acoustic emission source location. In [79], a Bayesian computational sensor network is proposed to perform SHM in a small scale structure. The damage in an aluminum panel is mapped by a small mobile robot equipped with vision and ultrasonic sensors. It is indicated that the fusion of vision and ultrasonic information reduces the uncertainty in damage localization. In [80], Bayesian inference is employed to perform damage detection and system identification for a 4-DOF shear type building structure. In this study, only partial acceleration measurements are used to update the posterior PDF of the structural parameters (e.g., mass, damping and stiffness). Experimental results have demonstrated that the proposed approach accurately estimates the parameter distributions in the presence of measurement noise and incomplete measurement data. In [81], a Bayesian inference-based regularization approach is proposed to identify the unknown traffic loads through the vibration measurements of the bridge. To this end, the accelerations, strains and displacement measurements are fused through a state space model. Results from a 27 bar truss bridge indicate the effectiveness and robustness of the proposed approach against noisy data. In [82], an efficient approach for modal identification is proposed to relieve the problem of computation time and convergence stability encountered by the method proposed in [83] when the number of measured DOFs increases. Although fast computation is achieved by the proposed approach, modal identification may fail for structures with no separated modes, e.g., a structure with close stiffness in both the two principal directions. In [84–86], modal parameters and structural parameters of buildings are identified through the aggregation of the ambient acceleration measurements. Various setups of sensor instrumentation are considered to test the proposed approach. Comprehensive experiments show that the proposed approach fails to achieve good performance in some cases of sensor placement. Also, identification performance depends on other factors such as modeling error and the noise level. In [87], a frequency-domain fast Bayesian FFT approach is proposed for modal identification through force vibration tests. By utilizing the input excitation and the acceleration measurements, data from simulation and field tests have demonstrated the robustness of the proposed approach. However, the input excitation must be large enough to eliminate the bias in the identification results due to the ambient responses of the structure. In [88], the stiffness parameters of shear buildings are estimated through the incorporation of transmissibility matrix, random matrix and Bayes' theorem. Based on the acceleration measurements, the proposed approach requires no evaluation of the inverse of the covariance matrix and hence is computationally efficient. Although the proposed method only needs the information from the structural responses, the number of independent measurements must be larger than the number of independent inputs (excitations). Chen *et al.* [89] use local binary patterns [90] and support vector machine [91] to autonomously detect cracks on metallic surfaces. The inspection videos of a nuclear power plant are recorded with an underwater camera system, and the frames of the videos are extracted to perform crack detection. Next, Bayes theorem is adopted to aggregate the information from different frames. It is shown that the performance of the detection algorithm is significantly enhanced with data fusion technique. In [92], Bayesian system identification is employed to identify the reduction in structural stiffness through the use of modal data (e.g., mode shapes and mode frequencies). The Gibbs sampling algorithm [59, 93] is adopted to generate samples from the posterior distribution of the stiffness scaling parameters. It is claimed that the proposed approach is capable of incorporating all sources of uncertainties based on given available data. In [94], a Bayesian based approach is proposed to provide one unified ground motion prediction by combining the predictions from multiple earthquake early warning (EEW) algorithms. The aggregation of information can be achieved even if these algorithms do not have compatible source models. Results have demonstrated that the proposed approach can operate in real-time and evaluate whether the prediction from these algorithms is a false alarm. In [50], the detection of cracks for nuclear power plant reactors is enhanced through the Naïve Bayes approach. By incorporating the spatiotemporal coherence of cracks in the adjacent video frames, the proposed approach outperforms other algorithms based on the fusion of multiple video frames.

Although the Bayesian inference-based approaches seem to be more popular in nondestructive testing [95], it requires reasonable assumptions to work properly. In [96], the classical inference-based approach, which requires fewer assumptions, is adopted to aggregate the data from ultrasonic and radiographic sensors to conduct nondestructive inspection. The detection accuracy is improved by the fusion of all the individual data. Table 2.1 and 2.2 summarize the above applications using Bayesian approaches in damage identification/quantification and system identification/response estimation, respectively. The information of input data source, validation approach, and limitations/concerns are presented.

2.2.3 State Estimation Methods

In SHM, the displacement of a structural system can be the indicative of damage extent within the structure [97]. Under controlled situations, the displacement of a structure can be easily measured through linear variable displacement transducer (LVDT) or optical sensors. In practice, the acquisition of displacement information is much more difficult than the acceleration information due to the difficulty in finding a reference plane for the displacement sensor. Double integrating the acceleration to obtain displacement is not reliable because of the amplification of low-frequency noise. Therefore, state estimation techniques are often dedicated to give a better estimation of the desirable information (e.g., displacement) given only limited available information (e.g., acceleration).

Kalman Filter Concepts

Kalman filter [98] is a powerful mathematical tool that can be used to predict the state of a dynamic system in the presence of uncertainty. It takes advantage of the information regarding the previous state of the system, and it is suitable for real-time implementation and embedded systems.

Consider a state vector x_k is governed by linear stochastic differential equation:

$$x_k = A_k x_{k-1} + B_k u_k + w_k, \tag{2.3}$$

with a measurement:

$$z_k = H_k x_k + v_k, \tag{2.4}$$

where variables w_k and v_k denote the process noise and measurement noise, respectively. Matrix A_k defines the relationship between the state at time step k and the state at time step $k-1$ in the absence of driving function u_k and process noise w_k. Matrix B_k defines the optional driving function u_k. In Equation (2.4), z_k is the actual measurement acquired from sensors while $H_k x_k$ is the measurement one expects to get based on the estimation x_k (i.e., the predicted measurement).

The core concept of Kalman filter is that it aims to find the best weight between the predicted and the observed measurements. Assume that the predicted and the observed measurements belong to two Gaussian distributions [99]. The best estimation between these two measurements would be the overlapping region of these two Gaussian distributions. Figure 2.2 illustrates this concept with 1-D Gaussian distributions. In this example, the best estimated state from Kalman filter would lie in the overlapping region between the distributions of the predicted and the observed measurements. During each time step, Kalman filter recursively updates the state estimation and its uncertainty. Figure 2.3 illustrates the recursive update process.

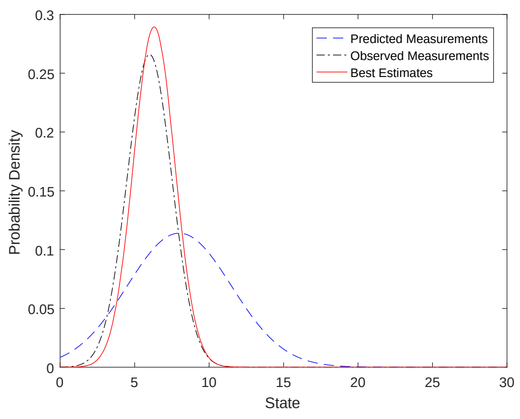


Fig. 2.2.: 1-D example for the best estimation between the predicted and the observed measurements.

Kalman Filter in Structural Dynamics

In the context of structural dynamics, consider the second-order linear equation of motion [53]:

$$M\ddot{x}(t) + C\dot{x}(t) + Kx(t) = T_0F(t), \tag{2.5}$$

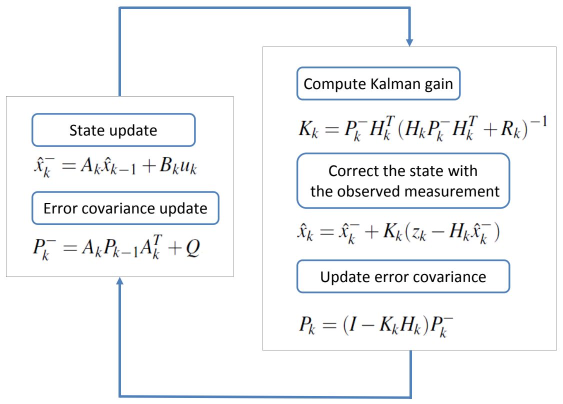

State update
$\hat{x}_k^- = A_k\hat{x}_{k-1} + B_k u_k$
Error covariance update
$P_k^- = A_k P_{k-1} A_k^T + Q$
Compute Kalman gain
$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1}$
Correct the state with the observed measurement
$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H_k\hat{x}_k^-)$
Update error covariance
$P_k = (I - K_k H_k)P_k^-$

Fig. 2.3.: Flowchart for Kalman filter. See [99] for derivation in detail.

where M, C, K denote the mass, damping, and stiffness matrices, respectively. T_0 denotes the force distribution matrix, and F denotes the excitation. In this case, state vector can be defined as:

$$X(t) = \begin{bmatrix} x(t) \\ \dot{x}(t) \end{bmatrix}. \tag{2.6}$$

Consider discrete time series, equation (2.5) can be formulated as a state-space representation:

$$X_{k+1} = AX_k + BF_k, \tag{2.7}$$

where A and B are the matrices define the state-space transition function. (See [53] for derivation in detail.)

It is noted that Kalman filter is an estimator of Bayes' theorem. The Kalman filter estimates the posterior mean and covariance of the state vector (estimation) conditioned on all the previous measurements. For nonlinear systems, the extended Kalman filter, the unscented Kalman filter, or the particle filter should come into practice instead of using the Kalman filter.

The Extended Kalman Filter (EKF)

The EKF has been the standard state-estimation method for dealing with nonlinear systems for the last several decades [100, 101]. Consider the state vector x_k governed by nonlinear stochastic differential equation:

$$x_k = \mathbf{F}(x_{k-1}, u_k, w_k), \tag{2.8}$$

with a measurement:

$$z_k = \mathbf{H}(x_k, v_k), \tag{2.9}$$

where **F** and **H** stand for the functions that define the nonlinearity. For a nonlinear system, the EKF applies the nonlinear transformations to the mean and covariance estimates through the Taylor series expansion. In EKF, only the first two moments of the conditional probability density function (PDF) are updated.

For the EKF to be reliable, it is only suitable for the systems that are almost linear in the updating time intervals since the high order terms in the Taylor series expansion are neglected. Several disadvantage are addressed in [102, 103]:

(a) The linearized transformation (i.e., Taylor series expansion) is only applicable if the error propagation can be well approximated by a linear fundtion during the updating process.

(b) The Taylor series expansion requires the computation of the Jacobian matrices. This could be difficult to compute and may induce additional computation efforts.

Therefore, the EKF may be inappropriate to be applied to higher order nonlinearities as it is often encountered in civil engineering applications.

The Unscented Kalman Filter (UKF)

Unlike the EKF method, the UKF algorithm do not require the computation of Jacobian matrices [103, 104]. A set of sigma points is first selected deterministically from the statistics of the transformation, and the mean and covariance estimates of the state vector can be described by the weighted combination of these sigma points. After the nonlinear transformation, the sigma points are able to estimate the posterior mean and covariance accurately to the second order (third order for Gaussian inputs) for any nonlinearity [102, 103]. Although the UKF is capable of dealing with higher order nonlinearity than the EKF, it may not work well if the error distribution is not Gaussian [102].

The Particle Filter (PF)

The PF (i.e, sequential Monte Carlo) methods is capable of dealing with the nonlinear system with the state vector described as a non-Gaussian posterior PDF [105–107]. In general, the PF method generates samples randomly to approximate the posterior PDF based on the Monte Carlo simulation. Once the samples are obtained, the posterior mean and the variance of the state vector can be computed through optimization techniques (e.g., minimization of the mean squared error (MMSE)) [103]. However, PF method could suffer from the computation issue since it may require to generate lots of samples depending on the problem of interest.

The Discontinous EKF and The Discontinuous UKF

Liu *et al.* [108] have shown that the EKF and UKF algorithms may fail to converge because of the unobservable states or parameters. This problem lies in many non-smooth systems such as the transition between the elastic and plastic responses, the initiation of a crack, or the initiation of a sliding. Switching behavior upon these discrete events makes the state-space equations non-differentiable. Recently, Chatzis *et al.* [109, 110] propose a D-modification version of the EKF and UKF, named as the discontinuous EKF and the discontinuous UKF, to address the unidentifibility in non-smooth systems. The D-modification ensures the unidentifiable parameters remain invariant over such time intervals, and therefore the estimates will not diverge. Table 2.3 summarizes the comparison between various Kalman filter algorithms.

Application

Smyth *et al.* propose a multi-rate Kalman fiter approach to estimate the displacement by the aggregation of acceleration and displacement measurements [111]. The multi-rate kalman filter is used to account for the lower sampling rate of the displacement sensors (e.g., LVDT, GPS) compared with the acceleration sensors. According to the simulation results in a single degree of freedom (SDOF) system, it is demonstrated that the displacement estimates of the proposed method is superior to the single-rate kalman filter. Also, a smoothing technique is introduced to improve the accuracy of the displacement estimates. However, this smoothing technique can only by applied to offline estimation since it requires the filtering over the entire sequence of the available measurements. In [112], Kalman filter-based approaches [111,113] are used to determine the intrastride variation in speed for human running. Data from GPS and the inertial measurement units (IMU) are fused to estimate the speed with fluctuation induced by cyclic pedal strokes. It is shown that the data fusion scheme achieves better estimates compared with the GPS and IMU used alone. In [114], the displacement measurements from a camera and the acceleration signals from a accelerometer are integrated through the multi-rate Kalman filter with smoothing technique proposed in [111]. It is demonstrated that data fusion using kalman filter not only estimates the velocity response, but also improves the accuracy of the estimated displacements and enlarges the frequency bandwidth of the displacement estimates. In [115], an adaptive EKF is proposed to detect variations in the structural parameters. Compared to the classical least-squares estimation (LSE) method, the EKF approach requires only acceleration measurements and therefore avoids the error induced from numerical integration. However, LSE is numerically more stable and easy to converge than the EKF approach. In [116], the EKF formulation proposed in [115] is modified to deal with systems with unknown excitations. Although the EKF-based approach is able to perform system identification, it may suffer computation burden if the test structure has a large number of DOFs.

In [117], the ground excitation and the acceleration measurements are integrated to estimate the vibration responses and structural parameters of a SDOF and a 2-DOF system. To this end, the nonlinear hysteretic Bouc–Wen system [118] is introduced to model the system nonlinearity. Based on the simulation results, it is demonstrated that the UKF yields more accurate estimation than the EKF when the system is highly nonlinear. Moreover, the UKF is more robust against the measurement noise than the EKF. In [44], the non-collocated acceleration and displacement measurements are fused to determine the displacement response and the hysteretic parameters of a 3-DOF system. The performance of the UKF, generic PF, and the Gaussian mixture sigma point particle filter (GMSPPF) [119] are compared. According to the simulation results, the UKF and GMSPPF are the most efficient techniques in terms of validation with the final identified hysteretic parameters. Also, the GMSPPF method achieves the best performance when estimating the time invariant model parameters.

In [120], the Kalman filter approach developed in [111] is employed to estimate the displacement time history of a full-scale seven-story reinforced concrete wall building based on collocated GPS and accelerometers. It is shown that the adopted data fusion scheme is able to estimate the displacement time series with millimeter precision. In [121], the fuzzy Kalman filter is used to reduce the failure risk of an integrated vehicle health maintenance system. To this end, data from accelerometer, pressure sensor, torque sensor, and liquid quality sensor are integrated through fuzzy logic and state estimation. In [122], a new approach based on Kalman filter is developed to estimate the displacement responses from the fusion of acceleration and displacement measurements. This method considers the acceleration measurement bias explicitly in the system dynamics of the Kalman filtering. The acceleration error is modeled as a combination of an offset bias (e.g., mechanical hysteresis), installation error (e.g., misalignment), and a zero-mean stochastic noise process. Based on the data collected from a small cantilever beam, it is claimed that the proposed approach is superior to the methods developed in [111].

In [123], a dual implementation of the Kalman filter is proposed to estimate the displacements and the velocities of a structure with only a limited number of noise-contaminated acceleration measurements. Floor acceleration responses are fused to obtain the displacements, velocities, and the input forces of the structure. It is claimed that the proposed method eliminates the numerical problems caused by the un-observability and rank deficiency of the augmented Kalman filter [124]. Additionally, the drift effect in the estimated displacements and input force observed in the Gillijn and DeMoor filter [125] is reduced by the proposed method with a good priori knowledge of the covariance of the unknown input force. In [126], a augmented Kalman filter is proposed to circumvent the divergence problem encountered in the estimation of input force when only the acceleration measurements are available. It is claimed that by adding the dummy displacement measurements in the Kalman filter updating process, the drift in the estimated response can be eliminated. However, the proposed approach is only applicable when the steady-state position of the system is known. In [127], the displacement response of a bridge is estimated through Kalman filter. The acceleration and strain measurements are incorporated into the state-space model. The proposed approach requires no reference points, which is necessary for the conventional displacement transducers. In [128], a Kalman filter based approach is proposed to identify the stiffness, mass and damping of a structure simultaneously. The input force and output acceleration measurements of a structure are aggregated to estimate the structural parameters in real-time operation. Although this approach requires complete measurements from each model DOF, numerical simulations have demonstrated that the proposed method is robust against noisy data and can be operate online. In [129], a smoothing based Kalman filter approach is proposed to estimate the displacement time history by aggregating the velocity from a laser Doppler vibrometer (LDV) and the displacement from a LiDAR sensor. Results from a series of lab-scale tests show that the proposed approach outperform the method in [117] with a lower estimation error and can be operated in real-time as well.

In [130], the seismic-induced damage in buildings is evaluated on the basis of a probabilistic framework. Four Bayesian filters, including the EKF, UKF, ensemble Kalman filter [131], and the PF are used to estimate the structural response as well as the uncertainties of estimations and their performances are compared. The response estimations are achieved by integrating the limited acceleration measurements and the information of a nonlinear model. It is pointed out that the EKF is the most suitable filter among all the filters being compared. In [132], a multi-rate UKF is implemented to monitor the response of a high rise building with the fusion of GPS and acceleration data. An artificial zero mean white noise measurement is added to avoid the drifting of the displacement estimates induced by low frequency integration errors. Through the data obtained from a densely instrumented building, it is shown that the proposed approach yields accurate estimates of the structural response. In [133], the dynamic response and physical representation of a structural system is estimated through the transformed subspace state-space system identification (T-SSID) [134] and the UKF. The acceleration and displacement signals are aggregated through T-SSID and UKF separately, and the performances of the two algorithms are discussed. The results indicate that T-SSID method is vulnerable to noise with low frequency content. Also, the T-SSID method is unable to operate under a reduced observation scheme (i.e, it requires the information from every floor in the form of either acceleration or displacement), and T-SSID needs offline implementation. Although the UKF method overcomes the limitations of the T-SSID method, the variation in the identified structural parameters for the UKF method is higher. This disadvantage is resolved by the fusion of acceleration and displacement measurements.

In [103], it is pointed out that the UKF appears to be more robust than the EKF when dealing with higher order nonlinearities. The EKF fails to give accurate estimations when encountering non-Gaussian conditional probability density functions. Finally, it is commonly seen that the state estimation approaches are incorporated within the probability framework. A typical example is the application in [74] as aforementioned. In [135], the UKF is used to perform the model calibration and response estimation for a 7-DOF shear building model. Two modeling schemes are considered, namely the cantilever and shear building models. The cantilever model outperforms the shear building model in the identification process. It is claimed that the fusion of acceleration and displacement measurements fails to enhance the response estimation in the presence of large modeling error. In [109], the discontinuous EKF (DEKF) is proposed to deal with the divergence problem in non-smooth dynamic systems. The impact problem, the nonlinear hysteretic Bouc-Wen model, and the elasto-plastic system are selected as case studies to validate the proposed algorithm. It is shown that the DEKF outperforms the EKF in all the selected non-smooth systems. When compared with the UKF algorithm, the proposed DEKF performs on a par with or better than the UKF in some cases. However, the good performance of UKF may depend on the initial estimates and the noise in the measurement signals. More recently, Chatzis and Chatzi [110] have proposed the discontinuous UKF (DUKF) to enhance the parameter estimates in non-smooth dynamic systems. Similar case studies in [109] are used to test the proposed algorithm. Results indicate that the DUKF improves the estimates and behaves more consistently than the standard UKF for non-smooth systems. Also, the DUKF does not require the detection of the transition event, which is an advantage over the DEKF algorithm. In [136], the DUKF algorithm is applied to the identification of a rocking body subjected to ground motions. The DUKF outperforms the UKF since the latter suffers from the divergence of the friction coefficient during the time periods of unidentifiability. Table 2.4 summarizes the above applications using KF approaches in system identification/response estimation. The information of input data source, validation approach, and limitations/concerns are presented.

2.2.4 Dempster-Shafer Theory

Dempster-Shafer (DS) theory is an approach for combining degrees of belief based on evidence. It is a generalized form of the Bayesian theory for considering multiple unions of hypothesis which are mutually exclusive [137, 138]. Different from probability theory, DS allows the quantification of ignorance. For example, in probability theory, the probabilities of the occurrences of all possible events sums to one. In the DS theory, a specified degree of belief can be assigned to the hypothesis where it is not knowing which event would occur. The elements for mathematical formulation of the DS theory are introduced in [137].

In terms of data fusion using multiple sensors, the DS theory allows us to integrate the beliefs derived from different sensors [138]. Suppose that the observations from two sensors are available, and the two sensors could have different observations. Denote m_i is the mass function assigned to the observations based on sensor 1, m_j is the mass function assigned to the observations based on sensor 2, A_k is an observed hypothesis from sensor 1, and $A_{k'}$ is an observed hypothesis from sensor 2. Then, the belief in outcome A being true can be expressed as:

$$Belief(A) = m_i \oplus m_j(A) = \frac{\sum\limits_{A_k \cap A_{k'} = A} m_i(A_k) m_j(A_{k'})}{1 - K}, \tag{2.10}$$

where

$$K = \sum_{A_l \cap A_{l'} = \phi} m_i(A_l) m_j(A_{l'}), \tag{2.11}$$

and $m_i \oplus m_j(A)$ stands for the combined proposition A, that is the intersection of the observed hypothesis A_k from sensor 1 and the observed hypothesis $A_{k'}$ from sensor 2.

Conflicts

Conflicts may happen when the application's frame of discernment is not well defined. If the values of mass function is strongly skewed, the belief of the hypothesis computed from Equation (2.10) would be unreasonable [138]. There are several methods to solve the conflict problem proposed by researchers. Yager *et. al.* [139] proposed a term as the ground probability mass assignment, that is the DS's basic probability mass assignment without the normalization by $(1 - K)$ (i.e., the denominator of Equation (2.10)). The probability mass tends to be extremely small when new conflicting observations are being introduced. Inagaki [140] introduced an approach to make a balance between DS's method and Yager's method by means of a factor k. This factor helps to reduce conflicts and make DS theory more reliable. However, how to choose the optimal value of k becomes an issue, and it is hard to be generalized for different applications. In [138], a dynamic weight calibration scheme is introduced to fuse the information from multiple sensors. The calibration scheme tends to assign larger weights to the information that are acquired more recently.

Application

Bao *et. al.* [141] employed DS theory to identify the damage locations on a 3D truss structure. The global stiffness matrix can be expressed as the sum of stiffness matrices of its substructures. [142] developed a multi-sensor data fusion scheme to perform the early detection of fire events. In this study, the DS theory was used to estimate the probability of fire event by means of information from temperature, humidity and vision sensors. In [143], the information from multiple piezoceramic sensors located at different locations of a two-story concrete frame was fused based on DS theory. A weighted fusion damage index was proposed to indicate the damage states of different areas of the frame. In [144], the damage detection of a building structure was performed by the combination of posterior probability support vector machine and DS theory. Support vector machine output the posterior probabilities from the recording of multiple sensors. These probabilities were fused to determine whether the structure was damaged or not based on DS theory. Also, the performance of data fusion at different levels were discussed. It was shown that fusion at decision level achieved the best accuracy when data was subjected to disturbance. In [145], the fault diagnosis of spark plug in an internal combustion engine was performed using the combination of artificial neural network, least square support vector machine, and the DS theory. Acoustic signals and vibration signals were fed into the two classifiers separately to classify different types of damages. The result indicated that using the

DS theory to fuse the results of the two classifiers achieved better classification accuracy than the result of each individual classifier alone. In [146], the fault diagnosis of an induction motor was performed based on neural network and the DS theory. The motor stator current signal and two acceleration signals were fed into three neural network separately. The outcomes of the three classifier were integrated by the DS theory. The analysis results indicated that decision-level fusion enhanced the robustness of fault diagnosis. In [147], a multi-stage data fusion approach was proposed to detect damage in structural systems. Local damage indicators were built based on the flexibility approach, which uses the change of the flexibility of the structural element as a way to quantify damage. The DS theory was employed to integrate all local decisions and make the final decision. In [148], the accuracy of pavement crack detection was enhanced by the combination of 2D gray-scale image analysis and 3D laser scanning method. The 2D method may fail to detect the cracks in the presence of shadows and tire marks, while the 3D method is unable to quantify the cracks when there is no obvious depth changes in the pavement. As a result, the 2D and 3D crack characteristics were fused based on the DS combination rules to perform more accurate crack detection. However, the proposed approach is highly dependent on the selection of the thresholds (e.g., the mass value for the 2D detection results). In [149], the DS theory and the Hadamard product [150] approach are employed to detect the honeycomb defects in concrete specimens. Measurements from the impact-echo (IE), ultrasonic pulse echo (US) and ground penetrating radar (GPR) sensors are used to perform feature-level fusion. Results demonstrate that both data fusion approaches perform better than the best single sensor, and the Hadamard product method outperform the DS theory when there is no nearly zero entries in the features. Table 2.5 summarizes the above applications using DS approaches in damage identification/quantification. The information of input data source, validation approach, and limitations/concerns are presented.

2.2.5 Fuzzy Logic

Different from classical set theory, fuzzy logic is often used to address vagueness and imprecision that exists in real world events. It is particularly useful when the boundaries between sets of categories are not well-defined, or the mathematical formulation of the system of interest is not fully understood [41,151]. For instance, whether a person is tall or short depends on subjective judgement of the observer, and there is no valid boundary value between the definition of tall and short. Therefore, fuzzy logic is applicable when the noise content of the data is high, or the measurement of the sensor is not precise enough. The key elements in fuzzy set theory are the membership functions that define the boundaries between the fuzzy set variables. The basic concept and procedure for fuzzy logic is introduced in the following section.

Analysis Procedure

Typically, each member in a fuzzy set consists of the variable value as well as the associated membership functions of the variable in one or more sets. A membership function provides a graphical representation of the boundaries between each set. The values for the membership function of the variable lie in the interval between 0 and 1, while zero means the variable is not a member of the set and one means the variable is a member of the set. Values between 0 and 1 indicate the variable may be partially a member of one or more sets. For instance, the temperature values could be a variable in environmental monitoring, while the sets consist of cool, medium, warm, and hot. The membership function is then used to point out the current temperature belongs to which set. Originally, bell-shaped curves have been employed to define membership function. However, they are replaced with triangle or trapezoid curves in many applications for simplicity [41]. A general analysis procedure is described as follows:

(a) Define the variable of interest and its characteristics.

(b) Define the membership functions associated with the variable.

(c) Define the production rules for aggregating the output from the membership functions, and compute the fuzzy values of samples based on the production rules.

(d) Convert the fuzzy values to a fixed and discrete output that can be employed to make inference about the characteristic of the sample.

Application

Jiang *et. al.* [152] combined fuzzy logic and neural network to detect damage in a 7-story shear-beam type building model. The damage detection in the first stage was performed separately by the use of fuzzy neural networks (FNN) with three types of inputs: normalized damage signature index (NDSI), normalized frequency change ratio (NFCR), and normalized mode change ratio (NMCR). In the second stage, the results of FNN models were fed into a fusion center to create a new output for damage pattern recognition. This was achieved by computing the weighting of the three FNN models based on the results in the first stage, and the new output was determined based on the weighted combination of different FNN models. It was concluded that combining data fusion and FNN techniques would perform better than using single FNN models alone. In [153], the false alarm rate for the detection of near-surface crack was reduced based on the fusion of images from eddy current test, magnetic flux leakage, and thermography test. The image fusion was achieved through the fuzzy AND operator [154, 155], fuzzy OR operator, and the shift-invariant wavelet transform [156]. In [157], fuzzy sets and the DS theory were employed to perform stress identification of two real world structures. It was shown that the identification results obtained from data fusion were more robust against the effect of noise than the results without data fusion. In [158], the damage detection of a wind turbine gearbox was conducted through fuzzy inference. The vibration data and the oil-debris data served as inputs for fuzzy inference, and the output damage level was determined using the defined membership functions. Table 2.6 and 2.7 summarize the above applications using fuzzy logic approaches in damage identification/quantification and system identification/response estimation, respectively. The information of input data source, validation approach, and limitations/concerns are presented.

2.2.6 Machine Learning Approaches

Among all the machine learning techniques, artificial neural network (ANN) and support vector machine (SVM) are the most popular techniques employed to perform data fusion in SHM in the past decades. Usually, this type of fusion is achieved through feature-level fusion. Features extracted from raw data or pre-processed data are fed into machine learning algorithms to perform regression or classification tasks. Recently, deep learning-based approaches have gain lots of attention due to their capability in automatically learning meaningful feature representations from the raw data. The input to the learning model does not need to be the handcrafted features as commonly seen in the conventional machine learning algorithms, and the generalization of the model is achieved through the use of large amount of data. The basic concepts of ANN, SVM, and deep learning are introduced in the following sections.

Artificial Neural Network (ANN)

ANN is originally inspired by the human nervous system that consists of billions of connected neurons [12]. It is a supervised learning method that is widely used in various disciplines. A supervised learning means the algorithm is first provided with a training dataset and the corresponding answers, then the parameters in the network are updated (i.e., learned by means of the training data). Typically, an ANN consists of multiple layers including an input layer, several hidden layers, and an output layer. In each layer, there are a number of neurons (i.e., nodes, connected with every nodes in the previous layer). Each node in the hidden layer and the output layer can be formulated as a weighted combination of all the nodes (including a bias term) in the previous layer, and the combination is further passed into a activation function to model the nonlinearity. In ANN, all the layers are fully connected. Figure

2.4 shows a typical configuration for ANN with one hidden layer. The inputs x_i are weighted summed, and a bias term is added to obtain y_j, and y_j is passed through the activation function f to get the output of node j (i.e., z_j). The network is established by providing a set of training samples, and its performance is verified by new test data. A general analysis procedure for ANN is described as follows:

(a) Determine the input of the network. Usually, the inputs to the network are the feature representations of the raw data.

(b) Design the network structure (i.e., the number of hidden layers and the number of nodes in each layer).

(c) Train the network with sufficient number of training samples. Usually, the parameters of the network is updated by back-propagation algorithms. This process involves the optimization of a cost function.

(d) Evaluate the network using the test dataset that is not used during the training stage.

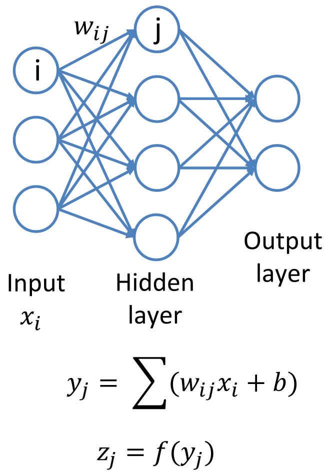


Fig. 2.4.: An illustration of ANN with one hidden layer.

Support Vector Machine (SVM)

SVM is a supervised machine learning algorithm as well. The fundamental concept of SVM is that it aims to find the best hyperplane to separate the data [159, 160]. The original data points are projected by a kernel function into a higher dimension feature space. An illustrative example is provided in Figure 2.5. In this space, SVM tends to find the hyperplane that separates the data with the largest margin. The hyperplane is defined by a subset of data points called support vectors that lie on the margin. A general analysis procedure for SVM is described as follows:

(a) Determine the input of SVM. Usually, the inputs are the feature representations of the raw data.

(b) Train the SVM model with sufficient number of training samples. The separation plane is determined by the optimization of a cost function.

(d) Evaluate the model using the test dataset that is not used during the training stage.

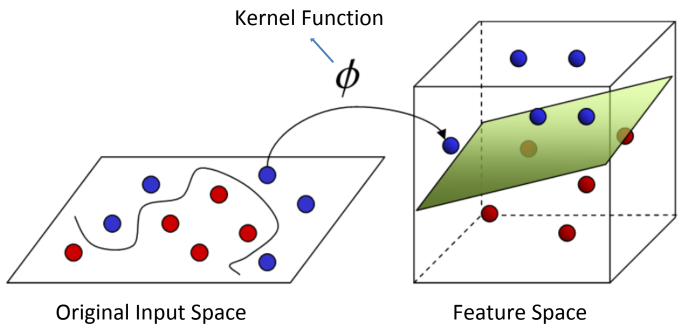


Fig. 2.5.: SVM illustration: the original data are projected into a higher dimensional feature space by a kernel function where a hyperplane separates the data.

Deep Learning

Different from the conventional machine learning algorithms, a deep learning algorithm uses a deeper (larger) network to conduct the regression or classification tasks. Although it requires intensive computation to establish the network from a large amount of training data, the network is able to learn the high level representations from the raw data autonomously. This is particularly useful since the selection of handcrafted features is subjective and may not contain enough information for the network to perform well. For instance, the deep convolution neural network (CNN) leads to a great success in object classification using large amount of image data [161, 162]. The spatial invariant features such as edges/contrast of an object is automatically learned during the training stage. Figure 2.6 illustrates a CNN configuration with one convolution layer followed by activation and one pooling layer. In the convolution layer, the kernel w_{ij}^1 performs convolution operation on the input data to generate the first feature map, the kernel w_{ij}^2 convolves with the input data and generates second feature map. These feature map will be transformed using activation such as RELU function [163], and the pooling layer extracts the dominant features from the feature map by applying operations such as retaining only the maximum value in the window that slides over the feature map. These features can be adopted to other classifiers to perform learning tasks. Figure 2.7 shows a schematic configuration of CNN that consists of multiple convolution and pooling layers. The convolution and pooling layers extract features from the input data, and the features are employed to determine whether there is a damage present in the input data.

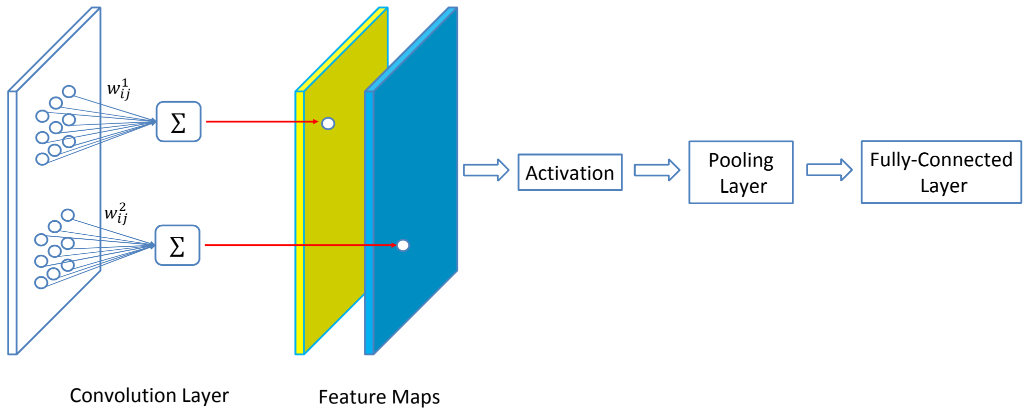


Fig. 2.6.: A sample for CNN Configuration.

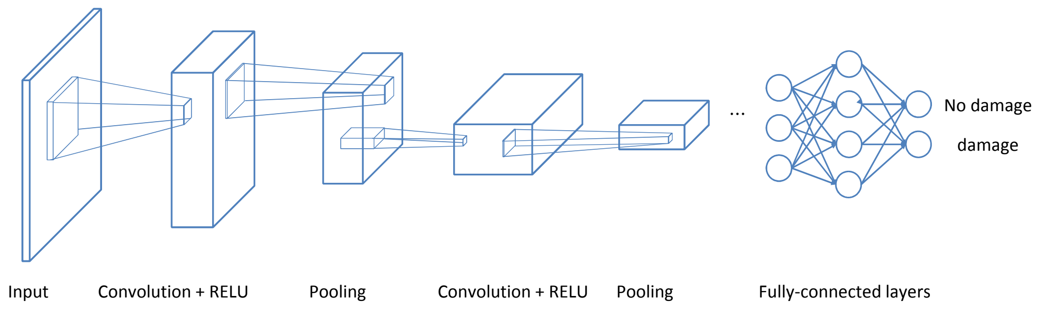


Fig. 2.7.: A schematic configuration for CNN.

Another deep learning algorithm, named as the deep Auto-encoder (DAE) [164, 165], is categorized as an unsupervised learning approach. The DAE algorithm is particularly useful when there is no enough training samples to conduct supervised learning. During the training of a DAE, the output of the network is identical with the input, and the network tries to minimize the difference between the input and the output. Figure 2.8 shows a sample of DAE configuration. Usually, the DAE consists of fully connected layers, and the output of the DAE is set to be the same as the input. Therefore, the DAE aims to reconstruct the input data, and the hidden nodes in the network become a high level representation of the input data. These representations can be further applied to other classifiers or clustering methods for learning tasks.

Application

Liu *et al.* [166] employ the wavelet packet transformation (WPT) algorithm [167, 168] and ANN to perform the detection of damage occurrence, location, and extent. Data fusion is achieved by integrating the first 16 components of wavelet packet relative energy (WPRE) computed from vibration signals. It is indicated that using several ANN models performs better than using only one single ANN model. In [169–174], the use of ANN to approximate nonlinear functions in engineering mechanics is discussed. The nonlinear restoring force is estimated through the displacement

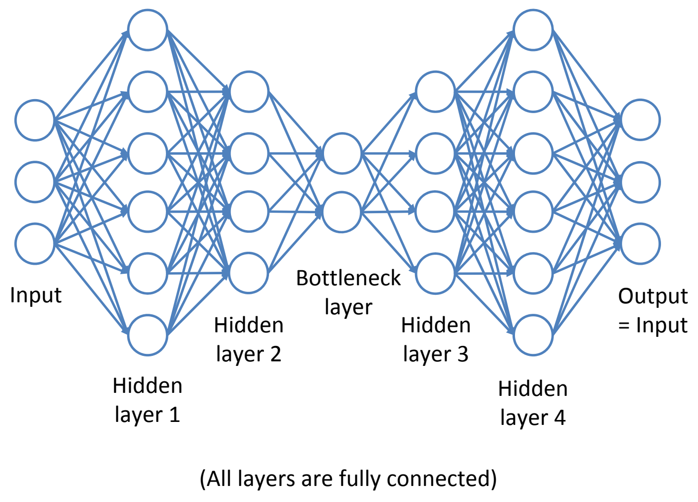


Fig. 2.8.: A sample for DAE configuration.

and velocity inputs. In [175, 176], ANN is used to estimate the displacement and acceleration transmissibility functions as well as the restoring force of a viscous fluid damper. In [177], the ANN and the least square SVM (LS-SVM) [178] are employed to detect the damage location and severity. The performance of these two algorithms are evaluated separately, and the selection of input features are optimized by using particle swarm harmony search (PSHS) algorithm [179, 180]. Similar to [166], data fusion is conducted by fusing the first 6 components of WPRE of vibration signals as well as the time-domain statistical features. It is shown that the learning model with input features optimized by PSHS algorithm achieves a higher accuracy, and LS-SVM performs better than ANN. Stramondo *et al.* [181] use maximum likelihood criterion to detect damage area after earthquake. Two types of remote sensing sensors are employed to rapidly construct the damage map of urban areas after a seismic event happened. One is synthetic aperture radar (SAR) and the other one is optical satellite data. Features such as the complex coherence of two images, the intensity correlation of two images, pixel to pixel difference, and the normalized difference vegetation index (NDVI) play an important role in the classification process. It is claimed that the combination of SAR and optical data leads to better classification results than using SAR or optical data alone. [182] compares the performances of different fusion algorithms, such as supervised and unsupervised methods, when dealing with the detection of flooded regions. Data sources in this application are multi-temporal and multi-modal images. It is shown that the supervised algorithms (e.g., ANN or SVM) do not always outperform the unsupervised method (fuzzy C-mean clustering algorithm). The performance of the learning algorithm is highly dependent on the input features, and the unsupervised method may be more favorable when there is no enough training samples available in disaster scenarios.

In [183], a framework of data-driven model for real-time SHM is proposed. Various machine learning algorithms including Naïve Bayes and AdaBoost are employed to perform damage detection and localization. Damage in structure is simulated by changes in stiffness and mass. Statistical features such as mean, standard deviation, and skewness are extracted from the vibration time history signals. These features are then selected and used to monitor the damage state of a multi-story shear building structure. In [184], a new damage indicator is proposed for bridge health monitoring based on image data. The vehicle types and the associated moving loads are first determined by the AdaBoost [185] technique and the Cascade classifier [186], then the unit influence surface of displacements is constructed based on images. The experimental results from a four-span bridge demonstrate the effectiveness of the proposed method in the damage detection and localization of the bridge. In [187], neural network and fuzzy inference are combined to evaluate the structural condition of a cable bridge. Heterogeneous data fusion is performed by integrating the GPS displacement data and wind velocity data. The GPS and wind velocity data are first fed into the neural network to generate features, and then these features serve as inputs to fuzzy inference system to output the health index of the structure. However, this method highly relies on the user defined fuzzy rules. For instance, the output would be risky if the wind is identified as very strong and the GPS signal is identified as normal. In [188], the corrosion cracking of a U-bend pipe is monitored by means of Bayesian Gaussian process and multiple sensor fusion. The Bayesian Gaussian process aims to find the posterior distribution of the test output given a random test input, available training samples, and the predefined likelihood and noise functions. The degree of degradation of the structure element is determined through Bayesian Gaussian process, and the results of multiple ultrasonic sensors are fused based on principal component analysis to compute the final damage index of the element. In [189], the health state of a truss-type structure is investigated by using multiple signal classification (MUSIC) ANN. The MUSIC algorithm produces a high-resolution spectral estimation even when the noise content of the signal is high. The vibration signals acquired from multiple sensors installed at different locations of the truss structure are first processed by the MUSIC algorithm. The output of the MUSIC algorithm is then inserted into a neural network to evaluate the damage location and damage severity of the structure. In [190], ambient vibration signals from undamaged and damaged structures are fused to perform damage detection and quantification. Wavelet features are extracted from the vibration signals, and the dominant features are determined through principal component analysis. The k-means clustering algorithm adopted in this study is shown to be able to characterize the damage patterns correctly. In [191], the state vector of a linear dynamic system with multiple sensors is estimated through the combination of multiple Kalman filters and one ANN. Each sensor measurement is fed into one individual Kalman filter, and the estimates from each Kalman filter serve as inputs to the ANN to determine the final state vector of the system. Based on the simulation results, the proposed network yields better performance than the fusion scheme proposed in [192]. In [193], the ANN and an ordinary differential equation (ODE) solver are combined to predict the relative displacement and velocity time history of a 12-story concrete frame. The ANN is adopted to model the soil-structure interaction by the aggregation of the displacement, velocity and ground acceleration information. Damage detection is achieved based on the difference between the prediction of ANN and the measured response. Although the proposed approach detects the damage accurately, it remains challenging to correlate the quantified detection results with the visual observation in the specimen. In [194], the performances of various supervised learning algorithms are compared for vision-based pavament crack detection. Among the decision tree, the k-nearest neighbor (KNN), the ANN and the adaptive neuro-fuzzy inference system (ANFIS) [195] algorithms, the ANN and ANFIS not only provide superior performance but also are more flexible in terms of the output format than the other two algorithms. Although the ANFIS method requires longer training time, it achieves comparable performances with the ANN and has higher interpretability over ANN. In [196], the KNN classifier is used to perform fault diagnosis of a fixed-axis gearbox. Features extracted from the vibration and sound signals are fed into the KNN classifier to discriminate the normal and faulty conditions. Results indicate that the aggregation of vibration and sound features achieves better performance than using the individual signal alone. In [197], the detection of honeycomb in concrete specimens is achieved through the machine learning-based approach. Features computed from the measurements of impact-echo (IE), ultrasonic pulse echo (US) and ground penetrating radar (GPR) sensors are aggregated through the density based clustering algorithm (DBSCAN) [198]. Compared to the individual evaluation methods, it is shown that data fusion enhances the detectability of the honeycomb.

Recently, Cha *et al.* [199] has employed CNN to detect cracks on concrete surfaces. The deep CNN achieves good performance without the use of handcrafted features. Results demonstrate that CNN-based approach is capable of dealing with real world challenges such as the varying illumination conditions. Abdeljaber *et al.* [200] have developed a CNN-based damage detection approach for a planar steel frame. In this study, the 1-D acceleration signals collected from 30 joints of the planar frame are used to train 30 CNN networks. Each CNN model determines whether the corresponding joint is damaged or not. It is shown that the CNN-based approach accurately detect and localize the damage, although the proposed method fail to localize the damage when the joints along the structure's line of symmetry were damaged. In 2017, Chen and Jahanshahi [50, 201, 202] have proposed a deep learning-based data fusion framework to detect and localize cracks on the metallic surfaces of nuclear power plant reactors. Deep CNN is used to detect cracks in the video frames, and the Naïve Bayes method is incorporated to account for the inherent spatiotemporal coherence of cracks in the adjacent video frames. It is shown that the proposed method outperforms other approaches with a higher hit rate by aggregating the information from multiple video frames.

In terms of the applications using DAE, a deep convolutional selective Autoencoder is proposed to conduct the early detection of combustion instability [203]. The hi-speed video frames of the combustor are fed into the network to determine whether the combustion is in the stable state or not. In [204], the DAE is employed to characterize cracks in composite coupon specimens that frequently used in aircraft applications. To this end, the DAE is trained by using only the images of intact surface to reconstruct the input image. When an test image with crack is fed into the DAE, the regions with higher reconstruction error indicate the crack locations. In [205], DAE is used to perform the fault diagnosis for the flight data. Signals from 13 sensors (e.g., time, measured load, ambient temperature) are concatenated to form the input to the DAE. The network is trained using the signals from the nominal (healthy) state of the flight, and the fault diagnosis is achieved through the clustering of the reconstruction errors of the 13 attributes. In [206], the detection of anomaly in gas turbine is achieved through the use of DAE. The DAE serves as a feature extractor for the input signals of exhaust gas temperature, and the features are fed into a ANN for anomaly detection. Results demonstrate that using the DAE learned features achieves better performance than the handcrafted features. Table 2.8 and 2.9 summarize the above applications using machine learning approaches in damage identification/quantification and system identification/response estimation, respectively. The information of input data source, validation approach, and limitations/concerns are presented.

2.2.7 Weighted Combination and Voting

A straightforward and simple approach for data fusion is the weighted combination of data. This is often not applicable to the original data, but is achieved in feature-level or decision-level data fusion. The simplest method is averaging all the data with the same physical quantity from all the sources, which means each data source has equal weight contribution. However, sometimes there are several health states inferred from multiple sensors with different resolutions and precisions. The decision made from the sensor with less precision and confidence might be assigned smaller weight contribution prior to the fusion. For classification tasks, the selection of appropriate thresholds is needed to assign the predicted damage pattern. Another method is voting scheme that could be applied for data integration when there are multiple decision makers. The rules of voting being applied is task-oriented and depends on the nature of the output from individual decision makers. For instance, majority voting is commonly seen when there are multiple classifiers used to perform classification task. If the user prefers a model to make conservative decisions, then the voting rule may be designed to be dominated by the most conservative decision made by all the decision-makers [41].

Although voting based data fusion approach is straightforward in concept and easy to implement, it is noted that the algorithm designer should be careful to avoid the presence of dictatorship during the fusion process. In 1950, K. J. Arrow proposed a theorem to address the difficulty in aggregating individual's preference for social welfare system [207]. This theorem is known as the Arrow's impossibility theorem. Without going into the mathematical details, an explanation quoted from [208] is provided as follows: "Any constitution (people get together to come to a conclusion of any sort) that respect the four conditions of unanimity, transitivity, independence of irrelevant alternatives, and no dictator compromise a contradictory set for which there is no mathematical resolution." This implies that when the first three conditions (i.e., unanimity, transitivity and independence of irrelevant alternatives) are satisfied, there must be a dictator existing among all the individuals. Consider the following example where there are three individuals given their preference over three events A, B and C:

1. $A > B > C$
2. $B > C > A$
3. $C > A > B$

The objective is to find the best social choice of the event that satisfies all the above three preferences. According to pairwise comparison, the social choice would prefer A over B (from preferences of voters 1 and 3), prefer B over C (from preferences of voters 1 and 2), and prefer C over A (from preferences of voters 2 and 3). In this situation, the majority vote fails to give the best choice to satisfy the preference of all individuals. In other words, if the majority vote choose event B as the best choice, it is basically ignoring voter 3 and hence result in dictatorship. Back to the discussion of data fusion, suppose there are three models that try to infer the potential damage state of a structure. In this case, the fusion result could be unreliable if the algorithm designer attempts to aggregate the decisions from the three models through majority voting scheme.

Application

Lu and Michaels have applied feature and sensor fusion to detect damage in aluminum specimens [209]. To this end, different features are computed from the diffuse ultrasonic signals, and the overall detection performance is improved by adopting voting fusion schemes on multi-sensors. In [210], images generated from guided wave signals are used to localize the damage in plates. Multiple images are obtained by passing the wave signals through filters with various frequency bands, and the images are fused at pixel-level. Although the results indicate the fused image achieves better localization performance than the individual images, the fused image may fail to detect the damage under the employed fusion scheme if there is no damage shown in the individual images. In [211], the maximum and the average data fusion strategy are used to postprocess the phased-array ultrasonic inspection data. Results indicate that the average fusion scheme produces a clearer visualization for the ultrasonic data. Bai *et. al.* [212] proposes an approach for detecting crack locations in beams by obtaining the overall Katz's fractal dimension (KFD) curve [213]. It is demonstrated that the overall KFD curve, that is computed from averaging all the single KFD curves, leads to a more reliable results for locating cracks compared to a single KFD curve. In [214], the cracks on a metallic specimen is detected through the aggregation of images. The images taken from the Eddy Current sensor, the Giant Magnetoresistance (GMR) sensor, and the Thermography sensor are fused at pixel-level to determine the presence of a crack. The performance of various fusion schemes including the wavelet coefficient fusion, singular value decomposition fusion, and the simple average method are compared and discussed. It is shown that the simple average method achieves the best performance among all the fusion schemes being considered. In [215], the damage diagnosis for a long-span suspension bridge is conducted through the fusion of damage indices from changes in modal frequency, eigenvalues of principal component, wavelet-packet energy, and changes in the covariance of covariance matrix of accelerations. The fusion techniques including the weighted average method, Bayesian theory, and DS theory are used and their performances are investigated. It is claimed that the weighted average method and the DS theory are not sensitive to the conflicting evidences and hence achieve more robust damage indicators. In [216], the damage localization for complex structures is enhanced through sensor fusion and statistical modeling. It is demonstrated that using maximum-likelihood estimate to localize the damage may fail due to the sparsity of sensor instrumentation in large structures. The localization is improved by means of the Neyman-Pearson criterion, and the contribution of each sensor pair is weighted according to the uncertainty (described by a Rayleigh-distributed random process) in the estimate of localization.

In [217], the performance of fusion techniques at different levels (i.e., data, feature, and decision level) are investigated to find the best separability between the undamaged and damaged regions of a fuselage rib. The Fisher's discriminant ratio [218] as well as simple voting schemes are utilized in this study. It is found that the fusion technique with the best performance varies with different specimens and features selected. In [219], the fault detection and the diagnosis of an industrial steam turbine is achieved based on SVM and an adaptive neuro-fuzzy inference system (i.e., the combination of ANN and fuzzy logic) [220]. The results of these two systems are integrated by ordered weighted averaging operator. It is demonstrated that the performance of the fusion scheme is better than each individual system. In [221], the information from a RGB sensor and a depth sensor are aggregated to measure the 3D displacement time history of a moving target. The coordinates of the desirable target points are first extracted from the color image based on the corner detection algorithm developed by Geiger *et al.* [222], and then they are mapped to the depth image to acquire the depth measurement. Once the depth and the pixel coordinates are obtained, the 3D world coordinates of the taget points relative to the sensor can be computed based on the pinhole camera model [223]. In this study, the weights for the information from the color image and the depth image are identical. By comparing with the measurement of an LVDT sensor from the shaking table test, it is shown that the proposed method is able to measure the displacements accurately. Also, the effects of various test conditions (i.e., amplitude, frequency, sampling rate, spatial distortion, and the relative motion between the target and the sensor) are comprehensively studied. In [224], an cost-effective autonomous data acquisition system is developed to detect, localize, and quantify the defects (e.g., potholes) on the road surfaces. This system consists of four RGB-D cameras, a GPS sensor, and two triaxial accelerometers. The RGB-D cameras aim to capture features (e.g., edges, corners, or texture) for recognizing the defects as well as compute the 3D world coordinates of the target points. The GPS sensor is used to record the location of defects and the speed of the vehicle, while the accelerometers are employed for the alignment of the sensor platform. Furthermore, the defect detection and quantification method is adopted from the approach proposed in [29]. Based on several field tests, it is pointed out that the developed system is able to monitor the pavement surfaces more frequently. However, factors like sunlight interference, motion blur, and rolling shutter distortion may affect the performance of the proposed system. In [225], the acceleration-based method developed by Lee *et al.* [226] and the strain-based method proposed by Shin *et al.* [227] are integrated to estimate the displacement responses of beam-like structures. According to the results of the numerical simulation and a full-scale field test, the proposed method is able to estimate the nonzero mean displacement and reduce the high-frequency noise. Moreover, there is no calibration technique required for the proposed method since the scaling factor for the strain-displacement relationship can be derived directly from the power spectral densities of the estimated displacements from the acceleration and strain at the first mode frequency. In [227], the information from the acceleration and strain sensor are equally weighted. In [228], the data fusion method developed in [225] is adopted in a wireless sensing network to monitor the displacement of a bridge. According to the laboratory and field tests, the proposed sensing system enables the time synchronization of the strain and acceleration measurements and further enhances the sensing precision of strain. In [229], the flexibility matrix of a simply-supported beam is estimated through the aggregation of the acceleration and angular velocity measurements. The proposed formulation is an extension of the single measurement-based method developed by Bernal [230]. Results from the numerical simulations indicate that the proposed approach estimates the flexibility matrix within a reasonable error tolerance. In [231], model updating of a simply-supported beam is conducted through the incorporation of acceleration and angular velocity measurements. The identifications of elastic modulus and boundary conditions of the beam demonstrate the superior performance of the proposed data fusion approach than using acceleration alone. In [232], the safety evaluation of railway bridges is achieved through the aggregation of acceleration and strain measurements. The displacements at every sensor location are first estimated using acceleration and strain information. Then, the dense displacements of the bridge is computed using mode shape approximation and modal expansion. Numerical and experimental results indicate a good agreement between the estimated dense displacements and the measured displacements. In [233], a kernel density estimation [234] based approach is proposed to perform crack localization for ferromagnetic metals. The joint density of crack location is obtained by aggregating the partial density that includes only the detected crack location from three individual test (i.e., eddy current test, magnetic flux leakage, and thermography test). Results have shown that the proposed fusion approach substantially reduces false alarms under various test conditions. In [235], the estimation of displacements is enhanced by aggregating the displacements obtained from time-synchronized image data and acceleration measurements. Complimentary filters are used to achieve a better frequency response for the combined displacements. It is demonstrated that the proposed approach is able to deal with various noise level and signals with different frequency characteristics. Table 2.10 and 2.11 summarize the above applications using weighting approaches in damage identification/quantification and system identification/response estimation, respectively. The information of input data source, validation approach, and limitations/concerns are presented.

2.2.8 An Illustrative Example for Data Fusion in Structural Response Estimation and Damage Detection

Consider a linear SDOF system:

$$m\ddot{x}(t) + c\dot{x}(t) + kx(t) = f(t), \tag{2.12}$$

where the mass $m = 1$, the dampling $c = 1.5$, and the stiffness $k = 1$. $f(t)$ is the ground excitation. $x(t)$, $\dot{x}(t)$ and $\ddot{x}(t)$ are the displacement, velocity and accleration response, respectively.

In this example, Kalman filter is employed to estimate the displacement response of the system through the noise-contaminated displacement measurements and known ground excitation. Following the procedure described in Section 2.2.3 with a discrete time step $\Delta t = 0.005$ (sec) , a state-space representation can be formulated as:

$$X_{k+1} = AX_k + Bf_k, \tag{2.13}$$

where $A = \begin{bmatrix} 1 & 0.005 \\ -0.005 & 0.9925 \end{bmatrix}$, $B = \begin{bmatrix} 1.25 \times 10^{-5} \\ 0.005 \end{bmatrix}$. The process noise and the measurement noise are assumed to be zero mean Gaussian white noise with covariance 0.01. The input excitation is zero mean Gaussian white noise with variance 1.

Figure 2.9(a) shows the Kalman filter estimation versus the true displacement response. The Root-Mean-Square-Error (RMSE) between the true response and the estimated response is 0.0031, which indicates a decent performance achieved by Kalman filter. Figure 2.9(b) depicts the estimated displacement response using $k = 1$ and $k = 0.8$ to simulate the undamaged and damaged structure, respectively. The maximum absolute displacement of the damaged structure is 0.1360, which is 8% larger than the undamaged structure (0.1259). This demonstrates that the damage detection can be achieved through the statistical measures of the discrepancy between the structural behavior of the damaged and undamaged structure. Moreover, features extracted from time-domain and frequency-domain responses can be further fed into clustering or machine learning algorithms to detect the existence of damage, providing that sufficient training data is generated either numerically or experimentally. As an example for damage detection, the Kalman filter simulation is repeated 1000 times for both the $k = 1$ and $k = 0.8$ cases. Damage detection is achieved through the fusion of frequency-domain features computed from the 2000 estimated displacement responses of the undamaged and damaged structure. The estimated response from Kalma filter is decomposed by the wavelet analysis with the Haar wavelet at level 4, and the 16 wavelet energy components are fed into a SVM classifier with linear kernel for damage detection. The training data is selected based on random permutation with 70% out of the 2000 samples, and the rest of the samples (not being seen by the classifier during the training process) are used for validation. Figure 2.10 shows the receiver operating characteristic (ROC) curve of the SVM classifier tested on the validation data. The area under curve (AUC) is 0.9704, which indicates a satisfactory performance in detecting the damage. According to the ROC curve, a detection accuracy of 92.43% is achieved if the user-defined tolerance of false positive rate is 10%.

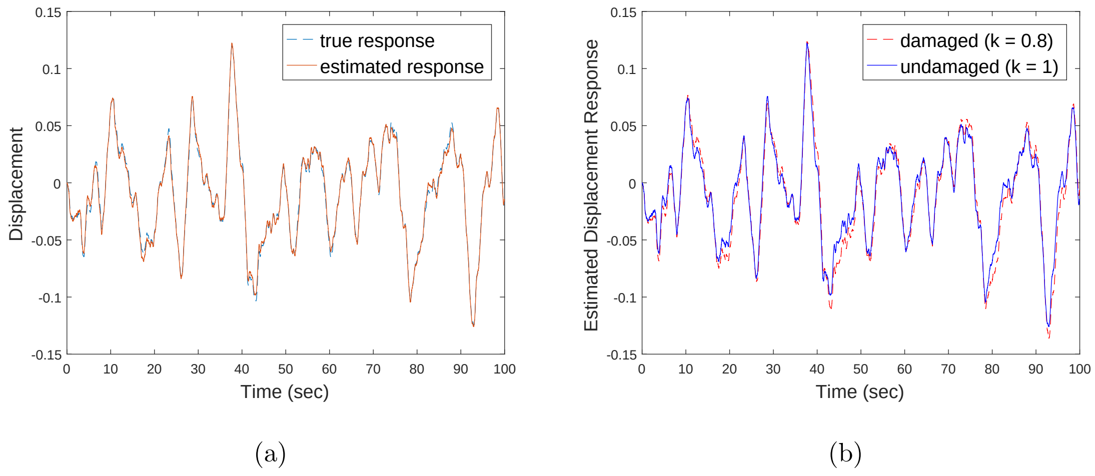


Fig. 2.9.: An illustrative example for data fusion: (a) displacement estimation using Kalman filter and (b) damage detection using the estimated response.

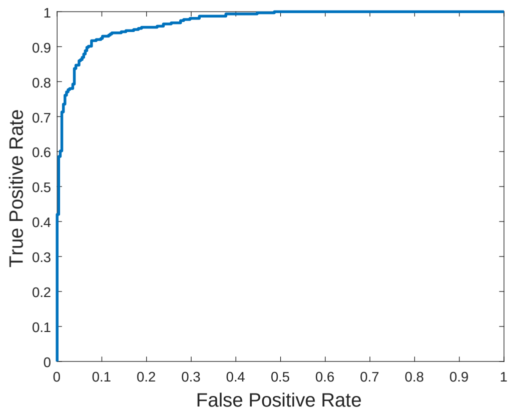


Fig. 2.10.: Damage dection using SVM with frequency-domain features: the ROC curve of the validation data. (AUC: 0.9704)

Note that in this particular example, state estimation approach is applicable since the state-space relationship is well-understood. For cases in which the state-space representation is unknown, data-driven based approaches could be more favorable than the model-based approaches. In addition, when comparing the performance of different approaches/algorithms, one should consider the following aspects but not limited to:

- Robustness against noise.
- The underlying assumptions in each approach.
- Algorithm's sensitivity against any modification in the assumption.
- Computation efficiency, which points to the applicability in real-time practice.
- Whether the hyper-parameters of the algorithm are tuned properly.
- Any fundamental limitation of the selected algorithm.
- Algorithm's scalability towards real world problems.

2.3 Challenges in Data Fusion and Machine Learning-based Approaches

As more inexpensive sensors and data acquisition systems come into practice, the availability of data and various categories of data source bring many possibilities to SHM. Although there have been several research efforts in data science, how to aggregate useful information from data and make robust decisions still remain challenging problems. There is no universal data fusion technique that works fine in all the applications across disciplines [236]. The performance of data fusion approaches varies with different applications and depends on the format and quality of the data as well as the objective of fusion. During the data acquisition stage, the noise content in the signal may be different even if all the sensors come from the same manufacturer and are measuring the same physical quantities. Possible conflicts between types of data, the incompleteness of data, and the lack of conciseness are the common problems when the data source is heterogeneous. Misleading information may also present in the recorded data if the sensors suffer from inappropriate usage or need to be maintained. These are all related issues that exist in the nature of data.

In terms of the challenges in data fusion techniques, the Bayesian probability framework and the state estimation method are capable of accounting for the uncertainties of the parameters during the analysis process. However, these approaches often require subjective decisions about the prior probabilities and model selections. The performance may vary with the selection of models and hence a sensitivity analysis is crucial for examination of robustness. Similar to Bayesian approach, the DS theory needs a subjective choice for the frame of discernment. Poor selection of frame of discernment and inappropriate combination rules may lead to unreasonable fusion results. Although Fuzzy logic is suitable for addressing vagueness and imprecision between each judgement, it requires well-defined membership functions and production rules to achieve acceptable inference. The lack of mathematical foundation could lead to pros and cons when the domain of application changes. In regard to machine learning approaches, the performance of the learning model is highly related to the training data being used. The number of training samples needs to be sufficient, and the training dataset should be able to capture the general characteristics of the system. Additional test dataset is also necessary to avoid overfitting in the training dataset. Lastly, eventhough the weighted combination and voting approaches are easy to be implemented and fast for real-time computation, how to assign appropriate weights to each sensor or decision maker is the key issue of these approaches. So far, most of the applications use constant weights to aggregate information from sources. But a dynamic weight calibration scheme is necessary when dealing with the long term health monitoring. Also, the selection of appropriate thresholds for making inference from the fused result is essential. A fixed threshold may become unreliable when the input data varies a lot.

With the issues and challenges in data fusion and machine learning-based techniques, the road map for future research is discussed as follows. Within the recent few years, deep learning-based approaches have been quite popular across various research fields due to the enhancement of computing capability. However, for the deep learning-based approaches to work well, it is essential to have sufficient training data. This is sometimes intractable for real world problems. For instance, the damage scenarios for a 30-story building are hard to define, not to mention the difficulties in collecting the data for all the damage scenarios from the real structure. Therefore, the learning task should be conducted in an unsupervised manner to avoid the need of large amount of labeled data. This means that the learning model should be established using only the data from the nominal (healthy) state of the system, and later on the model can be used to determine the damage scenario based on the reconstruction performance. A typical example is the use of DAE to diagnose the fault in flight data [205]. In terms of the network configuration for deep learning, the generative models like Restricted Boltzmann Machines (RBM) [237] and deep belief network [238] may outperform the discriminative models if the learning process is performed in unsupervised manner [239]. The Long Short Term Memory (LSTM) unit [240, 241] could also be adopted in the network and may have the potential to enhance the performance due to its ability in learning long-term dependencies. More recently, a new configuration named Capsule network is proposed to enhance the performance in object recognition [242]. Unlike the pooling layer in CNN, which only extracts the dominant feature based on the maximum operation, the routing between capsules extracts the features more effectively. The dynamic routing mechanism between the lower and the higher level of capsules allows the learning model to recognize multiple objects even if the objects overlap with each other. This could potentially enhance the performance of vision-based data fusion approaches.

Although machine learning-based approaches often benefit from data augmentation, one should be aware of the “curse of dimensionality” [243] that may occur when the dimension of the data, i.e., the number of variables, increases rapidly. As the number of variables increases, the complexity in algorithm grows exponentially, which not only leads to the difficulty in finding the optimal solution but also requires more training data to achieve convergence. Moreover, as the dimension increases, some distance metrics such as Euclidean measure become uniform among the data samples, resulting the poor performance of the similarity-based algorithms such as clustering or nearest neighbor classification. One possible solution to this issue is to search the optimal solution in a small subset of the full space. To this end, dimensionality reduction algorithms such as principal component analysis [244] or singular value decomposition [245] may be helpful, but one should be careful to select the subspace appropriately to consider the situation where different clusters may lie in different subspaces [246].

Another related issue that needs further investigation is the quality of data. Whether the data quality is good or not often requires subjective judgement and is task-oriented [247]. The quality metrics could be in terms of sensor resolution, signal-to-noise ratio, correctness, completeness, accessibility, trustness, or timeliness. For instance, an image that is perfect for one specific task may be inappropriate for another. Most data fusion applications in the past do not consider the effect of data quality, which could be problematic if the input data contains wrong information. An example for considering the data quality is [110], where the unidentifiability of the parameters in non-smooth systems are addressed to solve the divergence issue in parameter estimates. Also, the early detection of sensor malfunction needs to be incorporated into the data fusion framework and should be prior to any data processing. Although the completeness in data is usually beneficial for decision making, this is not always true in certain situations. For instance, in disaster response system, how to use partial information to make decisions is crucial since the time constraint is critical for saving lives. In this case, the data fusion algorithms that are capable of operating in real-time are more favored, and the data transmission between the sensor and the base station needs to be efficient. To this end, a reliable data reduction scheme is necessary. This may be achieved by the DAE established through off-line operations, and then the DAE automatically extracts the dominant feature representations in lower dimension for data transmission.

In summary, there is room for improvement of data fusion and machine learning-based techniques as well as the unresolved problems. Note that the data fusion approaches are not limited to the techniques presented in this chapter. For instance, Game theory could be potentially applicable for data fusion in SHM. In [248], Game theory is adopted to aggregate the hyperspectral images for groundcover classification in precision agriculture. It is expected to see more extensive research in adopting data fusion and machine learning techniques in SHM.

Table 2.1.: Applications using Bayesian approaches in structure damage identification/quantification. (The acceleration, displacement and velocity measurements are denoted as acc., disp. and vel., respectively.)

literature	data source	numerically validated	experimentally validated	limitations/concerns
[61, 62]	modal parameters	✓		no validation from real structures.
[63]	acc. measurements and ground excitations	✓		the employed structural model is assumed to be identical to the actual structure.
[64]	acc. measurements	✓		identification performance depends on the selection of appropriate strucutral model, and small damage may not be detected.
[67]	modal data computed from acc. measurements	✓	✓	performance depends on the sensor placements, damage magnitudes and locations, as well as the model classes and the modeling errors.
[68]	modal data computed from acc. measurements		✓	damage with small magnitudes may not be detected.

Table 2.1.: Continued

literature	data source	numerically validated	experimentally validated	limitations/concerns
[69]	acc. from ambient vibration and strong motion		✓	less accurate for cases in strong motions.
[70]	crack length measurements and fatigue cycle	✓	✓	simple polynomial assumptions are made for stress intensity range.
[71]	ultrasonic guided waves	✓		damage propagation is assumed to be driven by fatigue only.
[72]	crack propagation trajectories	✓	✓	uncertainties in loadings are not considered.
[73]	acc., current, voltage, and temperature signals		✓	performance depends on the selection of optimal sensors and engineered features.

Table 2.1.: Continued

literature	data source	numerically validated	experimentally validated	limitations/concerns
[74]	crack information from acoustic emission, empirical model and digital imaging		✓	performance depends on the correlation between the features and the crack growth rate.
[78]	PDFs of time difference data at multiple frequencies		✓	modeling error and measurement noise are assumed to be Gaussian.
[79]	image and ultrasonic data		✓	damage distant from the robots is poorly localized.
[89]	video frames		✓	crack detection performance depends on the environmental conditions during data collection.
[50]	video frames		✓	the proposed crack detection approach requires lots of training data and GPU computation.

Table 2.2.: Applications using Bayesian approaches in structure system identification/response estimation.

literature	data source	numerically validated	experimentally validated	limitations/concerns
[64]	acc. measurements	✓		identification performance depends on the selection of appropriate strucutral model, and small damage may not be detected.
[65]	modal data	✓		structural behavior is assumed to be well approximated by linear dynamics.
[66]	acc. measurements	✓		poor understanding about the damping in the structure may lead to unsatisfactory identification results.
[75]	elongation, load and thermal expansion coefficient measurements		✓	load estimate performance depends on appropriate data calibration and prior knowledge.
[76]	ambient acc. measurements		✓	performance depends on the selection of appropriate finite element model.

Table 2.2.: Continued

literature	data source	numerically validated	experimentally validated	limitations/concerns
[77]	sonic, ultrasonic and compressive strength test measurements		✓	subjective judgements are made about the confidence interval of different test methods.
[80]	acc. measurements	✓	✓	Gaussian and inverse Gamma distributions are assumed to be the prior PDFs of structural parameters and prediction error covariance, respectively.
[81]	acc., disp. and strain measurements	✓		noise levels are assumed to be identical for different sets of noisy sequences.
[82]	ambient acc. measurements	✓	✓	the proposed approach only applies to the structures with separated modes.
[84–86]	ambient acc. measurements	✓	✓	performance depends on the sensor setups, modeling error and signal-to-noise ratio.

Table 2.2.: Continued

literature	data source	numerically validated	experimentally validated	limitations/concerns
[87]	acc. measurements and input excitations	✓	✓	ambient response is not modeled in the proposed approach and hence may lead to bias in identification results.
[88]	acc. measurements	✓	✓	number of independent response measurements must be larger than the number of independent excitations.
[92]	mode shapes and mode frequencies	✓	✓	parameter identification is assumed to be performed with low-amplitude vibration data.
[94]	predictions from multiple EEW systems		✓	ground motion intensity is assumed to be log-normal distributed with fixed variance.

Table 2.4.: Applications using KF approaches in structure system identification/response estimation.

literature	data source	numerically validated	experimentally validated	limitations/concerns
[111]	acc. and disp. measurements	✓		the proposed smoothing technique is only suitable for offline estimation.

Table 2.4.: Continued

literature	data source	numerically validated	experimentally validated	limitations/concerns
[114]	disp. from imaging and acc. measurements		✓	reduction in the sampling rate of the disp. from imaging may induce slight drift in the fused disp. estimates.
[115]	acc. measurements	✓		the proposed EKF approach requires considerations in numerical stability.
[116]	acc. measurements	✓		the proposed approach is computation intensive if the structure has many DOFs.
[117]	ground acc. and acc. responses	✓		the ground acc. and structural acc. response are assumed to be measurable.
[44]	acc. and disp. measurements	✓		measurement noise is not considered.
[120]	GPS and acc. measurements		✓	networks of densely collocated GPS and seismic instruments are necessary.
[122]	acc. and disp. measurements		✓	performance depends on the displacement sampling rates.

Table 2.4.: Continued

literature	data source	numerically validated	experimentally validated	limitations/concerns
[123]	floor acc. measurements	✓		performance depends on the fine-tuning of the covariance of the unknown input.
[126]	acc. measurements		✓	only applicable when the steady-state position of the system is available.
[127]	acc. and strain measurements	✓	✓	performance may depend on the amplitude of displacement response.
[128]	acc. and force measurements	✓		requires complete measurements from each model DOF.
[129]	vel. and disp. measurements		✓	time-varying bias is not considered in acc. measurements.
[130]	limited acc. measurements and output from a nonlinear model	✓		optimized sensor placement is not discussed for the proposed approach.
[132]	acc. and GPS measurements		✓	an artificial white noise observation is required for on-line monitoring.

Table 2.4.: Continued

literature	data source	numerically validated	experimentally validated	limitations/concerns
[133]	acc. and disp. measurements		✓	performance depends on the selection of noise matrices and the assumption of known structured model.
[135]	acc. and disp. measurements	✓		performance depends on reasonable modeling assumptions.
[109]	ground acc. input and output of model	✓		performance depends on the initial estimates of the parameters. the initial condition requires a pre-estimate.
[110]	disp. and output of model	✓		parameters are assumed to be invariant during unobservable time intervals.
[136]	ground acc. and disp. measurements	✓		parameters are assumed to be invariant during unobservable time intervals.

Table 2.3.: Comparison between various Kalman filter (KF) algorithms. (Algo.: algorithm, DEKF: discontinuous EKF, DUKF: discontinuous UKF)

Algo.	Properties	Applications
KF	1. Real-time operation. 2. Limited to linear system.	[111], [112], [114], [120], [121], [122], [123], [126], [128], and [129]
EKF	1. Can be used in nonlinear system. 2. Not suitable for system with higher order nonlinearity.	[117], [130], and [103]
UKF	1. Applicable for system with high nonlinearity. 2. Not appropriate for system with non-Gaussian error distribution. 3. Performance depends on initial estimates of parameters.	[117], [44], [130], [132], [133], [103], and [135]
PF	1. Applicable when state vector is non-Gaussian. 2. Computationally intensive and hence only for offline operation.	[44], and [130]
DEKF	1. Applicable for non-smooth systems. 2. Capable of dealing with unobservable states or parameters. 3. Requires detection of transition events in non-smooth system.	[109]
DUKF	1. Applicable for non-smooth systems. 2. Capable of dealing with unobservable states or parameters.	[110], and [136]

Table 2.5.: Applications using DS approaches in structure damage identification/quantification.

literature	data source	numerically validated	experimentally validated	limitations/concerns
[141]	damage probability assignment functions from different datasets		✓	damage localization are assumed to be completely characterized by modal strain energy.
[143]	recordings from multiple piezo-ceramic sensors		✓	requires active sensing.
[144]	acc. measurements	✓		damage are assumed to be characterized by wavelet features.
[147]	mode shapes	✓		requires a finite element model.
[148]	2D images and 3D laser scanning		✓	performance depends on the selection of thresholds.

Table 2.5.: Continued

literature	data source	numerically validated	experimentally validated	limitations/concerns
[149]	IE, US and GPR measurements		✓	damage is assumed to be fully-characterized by engineered wave features.

Table 2.6.: Applications using fuzzy logic approaches in structure damage identification/quantification.

literature	data source	numerically validated	experimentally validated	limitations/concerns
[152]	outputs of three fuzzy neural networks	✓		comprehensive training dataset is required to find the optimal weights for fusion.
[153]	images from eddy current test, magnetic flux leakage and thermography		✓	combination rules are simple and straightforward, could be inadequate for complex data.

Table 2.7.: Applications using fuzzy logic approaches in structure system identification/response estimation.

literature	data source	numerically validated	experimentally validated	limitations/concerns
[157]	strain measurements	✓		requires transient time history analysis on the finite element model.

Table 2.8.: Applications using machine learning approaches in structure damage identification/quantification. (GPS: global positioning system; vel.: velocity; acc.: acceleration; ANFIS: adaptive neuro-fuzzy inference system; ANN: artificial neural network; IE: impact-echo; US: ultrasonic pulse echo; GPR: ground penetration radar.)

literature	data source	numerically validated	experimentally validated	limitations/concerns
[166]	acc. measurements	✓		damage assessment is assumed to be characterized by wavelet energies.
[177]	acc. measurements	✓		damage assessment is assumed to be characterized by wavelet energies and time-domain statistical features.
[183]	acc. and disp. measurements	✓		damage is assumed to be characterized by time-domain statistical features.
[184]	images		✓	performance may depend on the camera resolution.

Table 2.8.: Continued

literature	data source	numerically validated	experimentally validated	limitations/concerns
[187]	GPS and wind vel.		✓	performance depends on user-defined fuzzy rules.
[188]	ultrasonic measurements		✓	damage index is determined by the first principal component only.
[189]	acc. measurements		✓	damage index is assumed to be characterized by the amplitudes of natural frequencies.
[190]	acc. measurements	✓		the proposed approach requires a database of baseline measurements for optimal signal selection.
[194]	images		✓	ANFIS algorithm is more interpretable but requires longer training time than ANN.
[197]	IE, US and GPR measurements		✓	requires contact sensing.
[199]	images		✓	requires sufficient training data and computing capability.
[200]	acc. measurements		✓	damage localization may fail when damage are presented in symmetric joints.

Table 2.8.: Continued

literature	data source	numerically validated	experimentally validated	limitations/concerns
[50]	video frames		✓	requires sufficient training data and computing capability.
[204]	images		✓	detection performance depends on the selection of threshold for reconstruction error.
[205]	13 time series data (e.g., time, temperature, loads)		✓	requires sufficient dataset for undamaged structure.

Table 2.9.: Applications using machine learning approaches in structure system identification/response estimation. (Disp.: displacement; vel.: velocity; ANN: artificial neural network; acc.: acceleration.)

literature	data source	numerically validated	experimentally validated	limitations/concerns
[169–173]	disp. and vel. measurements	✓		only applicable for simple ANN configuration.

Table 2.9.: Continued

literature	data source	numerically validated	experimentally validated	limitations/concerns
[174]	disp. and vel. measurements		✓	only applicable for simple ANN configuration.
[175]	disp. and vel. measurements	✓		the proposed approach is based on heuristic methods rather than rigorous mathematical formulation.
[176]	disp. and vel. measurements	✓	✓	the proposed prototype is only adequate for specific nonlinear function mapping.
[193]	disp., vel. and ground acc.		✓	hard to correlate between the estimated response and the observed damage.

Table 2.10.: Applications using weighting approaches in structure damage identification/quantification.

literature	data source	numerically validated	experimentally validated	limitations/concerns
[209]	ultrasonic measurements		✓	additional considerations may be required if the wetting area of aluminum specimens is large.

Table 2.10.: Continued

literature	data source	numerically validated	experimentally validated	limitations/concerns
[210]	images		✓	fusion scheme may fail if individual images fail to detect damage.
[212]	operating deflection shapes	✓	✓	equal weighting is assigned to each data due to simple averaging.
[214]	images		✓	equal weighting is assigned to each data due to simple averaging.
[215]	mode frequency and acc. measurements	✓		no cases of strong motion, only ambient vibration is considered.
[216]	ultrasonic measurements		✓	damage pattern is assumed to be point-like scatters.
[217]	ultrasonic measurements		✓	the most appropriate statistical feature may depend on problem of interest.
[224]	images, GPS and acc. measurements		✓	performance may depend on environmental factors, e.g., illumination condition.

Table 2.10.: Continued

literature	data source	numerically validated	experimentally validated	limitations/concerns
[233]	images from eddy current test, magnetic flux leakage and thermography test		✓	performance depends on fusion rules.

Table 2.11.: Applications using weighting approaches in structure system identification/response estimation.

literature	data source	numerically validated	experimentally validated	limitations/concerns
[225]	acc. and strain measurements	✓	✓	heuristic approach is employed to investigate the influence of sensor quantity.
[228]	acc. and strain measurements		✓	displacement estimation is achieved through contact sensing.
[229]	acc. and angular vel. measurements	✓		measurement noise is not considered.

Table 2.11.: Continued

literature	data source	numerically validated	experimentally validated	limitations/concerns
[231]	acc. and angular vel. measurements	✓	✓	only concentrated point excitation is considered.
[232]	acc. and strain measurements	✓	✓	large estimation errors may occur for beams with fixed-end boundary condition.
[235]	image and acc. measurements	✓	✓	performance depends on the design of complementary filters.

3. DEEP CONVOLUTIONAL NEURAL NETWORK FOR RESPONSE ESTIMATION AND SYSTEM IDENTIFICATION IN STRUCTURAL HEALTH MONITORING

3.1 Introduction

3.1.1 Motivation and Related Work

During the past decades, many scientists have made efforts to propose effective structural health monitoring (SHM) techniques for civil buildings and structures. The estimation of the structural vibration responses has been intensively investigated since it provides useful information to infer the health state of the structure as well as the inherent structural characteristics. Through the nonlinear time-history analysis, the performance level of the structure under various earthquake intensities can be determined through the maximum drift estimation. For instrumented structures, data recorded from the acceleration sensors can be employed to compute the fundamental frequencies and the mode shapes. Changes in the identified characteristics serve as an indicator for health assessment. Recently, the advances in machine learning (ML) techniques have led to more opportunity for research in SHM. These ML approaches attempt to learn the underlying mechanisms from the available measurements and, subsequently, predict the possible outcomes given the new input. More recently, the emergence of deep learning approaches provide new opportunities for ML based researches. Contrary to the conventional ML techniques, deep learning is usually referred to the establishment of a larger and deeper network with the aid of large amount of data. Due to the recent developments in computation and sensor technology, the accessibility in data acquisition is much easier compared to the past, and therefore increases the applicability of deep learning approaches. This study proposes a deep CNN to estimate the dynamic response of a linear SDOF system, a nonlinear SDOF system, and a MDOF three-story steel frame. The conventional MLP approach is adopted to serve as a reference for the performance of the proposed deep CNN-based method.

The MLP has been widely used to estimate the dynamic response or the characteristics of a system. Masri *et al.* [15] used MLP to model the dynamic response of a linear and a nonlinear system. The network was trained using the simulated vibration data to represent the behavior of the healthy structure, and the damage detection was achieved later through the discrepancy between the network's prediction and the vibration response from a damaged structure. In Masri *et al.* [249], similar concept in Masri *et al.* [15] was adopted to model a four degree of freedom dynamic system using MLP. The damage identification was conducted through the least-square error between the measured output from a damaged structure and the predicted response from the trained MLP with the data from the healthy structure. Pei *et al.* [250] addressed how to fit a nonlinear function with MLP, and employ MLP to model a nonlinear dynamic system. In [169–172, 174], the ability of MLP to approximate nonlinear functions in engineering mechanics was discussed. Also, these works were dedicated to develop constructive methods for the initialization of the MLP configuration where the MLPs were trained to predict the nonlinear restoring force by using the displacement and velocity inputs. In [173, 175, 176], the authors demonstrated the use of MLP to fit functions like first-order polynomial, exponential, and Gaussian function. Data from laboratory test was used to validate the function fitting concept. The approximations of the displacement and acceleration transmissibility functions as well as the restoring force of a viscous fluid damper were presented. In [251], the MLP was used to estimate the acceleration response of a four degree of freedom structure given the acceleration and excitation measurement at previous time steps. The input-output relationship of MLP is then adopted to estimate the structural parameters such as mass and stiffness of the structure based on an equivalent auto-regressive moving average model. In [252], the damage in building structures was localized and quantified through the use of a MLP to identify the stiffness reduction in the damaged story given the changes in the natural frequency.

Cury and Crémona [253] presented a damage classification scheme for beam-type structure using MLP. Data histograms from the vibration signals, natural frequencies, and the mode shapes served as inputs to the network. The effect of noise contaminated signals was also discussed. Arangio and Beck [254] incorporated Bayesian inference with MLP to achieve the damage detection, localization, and quantification for bridge structures. Similar to the scheme in [15], the damage was detected through the error between the measured data and the prediction of a MLP trained by the undamaged structure. In [255], the active control of a base-isolated building was conducted through the use of a MLP. The authors used MLP to approximate the nonlinear control law of the active controller and reduced the vibration response of the building by applying appropriate force achieved from the controller. Kao and Loh [256] presented a health monitoring method for dam inspection using MLP. The long-term deformation of the dam was estimated through MLP given the information extracted from the water level and the temperature distribution of the dam body. In [257], MLP was used to fit the load-deflection curve of a carbon fiber reinforced polymer (CFRP) reinforced concrete (RC) slab. Derkevorkian *et. al.* [193] combined the MLP and an ordinary differential equation (ODE) solver to predict the relative displacement and velocity time history, where the MLP was adopted to model the soil-structure interaction. The difference between the prediction of MLP and the measured data served as a tool to detect damage.

CNN has been quite popular in speech recognition, image classification, and time-series modeling [13]. For instance, there is significant achievement in CNN-based approaches for speaker/gender identification and spoken words/music recognition using 1-D acoustic signals [258]. Due to the ability to learn the spatial invariant features automatically, CNN leads to a great success in object classification using large amount of image data [161, 162]. Recently, Cha *et. al.* [199] used CNN to detect cracks in concrete surfaces. The results demonstrated that CNN-based approach is capable of dealing with real world challenges such as the varying illumination conditions. Abdeljaber *et. al.* [200] proposed a CNN-based damage detection approach for a planar steel frame. The CNN was employed to extract the dominant features from 1-D acceleration signals to perform the damage detection and localization. Tompson *et. al.* [259] adopted CNN to accelerate the 2D and 3D motion simulations of fluids and smoke where the CNN was used to infer fluid pressure, which is the solution of the Poisson equation that requires a large number of computation iterations. Chen and Jahanshahi [50] used CNN to detect and localize cracks on the metallic surfaces of nuclear power plant components. The crack detection results from different video frames are aggregated to enhance the final detection result. Atha and Jahanshahi [260] proposed a CNN-based approach for corrosion detection on metallic surfaces. It is shown that the CNN is capable of learning appropriate classification features automatically without the need of human hand-crafted features.

Although CNN has been explored in many research fields, most of the studies use CNN as a feature extractor to conduct classification/recognition tasks. Only a few works use CNN to perform regression or function approximation problems. This study presents a deep CNN-based approach for the response estimation in structural dynamic systems. Results of CNN are compared with the results obtained from a MLP-based approach. The robustness of the proposed approach against noise is discussed through the use of various noise-contaminated vibration signals. Data employed in this study consists of both numerical simulation and laboratory test data. Finally, the structural characteristics of the test steel frame is identified through the response predicted by the CNN model.

3.1.2 Contribution and Scope

To the best understanding of the author, this chapter presents the first attempt to adopt deep CNN for response estimation (regression) in structural dynamic problems. The deep CNN is employed to estimate the vibration responses of a linear SDOF system, a nonlinear SDOF system, and a three-story steel frame. The results of MLP-based approach serves as a reference for the proposed CNN method, and the system identification of the steel frame is achieved through the estimations obtained from the CNN model. The rest of the chapter is organized as follows: Section "Methodology" presents the formulation and configuration of MLP and CNN. Section "Single Degree of Freedom (SDOF) System" discusses the analysis results from a linear and a nonlinear SDOF system. Section "Multi-Degree of Freedom (MDOF) System" presents the results of vibration data collected from a shaking table test of a three-story steel frame where the CNN model is used to conduct the system identification of the frame. Section "Physical Interpretation of Convolution Layer" discusses the physical interpretations for the convolution layer. The concluding remarks and further discussions are addressed in the last Section.

3.2 Methodology

3.2.1 Multilayer Perceptron (MLP)

MLP, first inspired by the biological neural system, typically consists of one input layer, multiple hidden layers, and one output layer [12]. Fig. 3.1 depicts a MLP configuration with only one hidden layer. Each node in the hidden or output layer is fully-connected with all the nodes in the previous layer, and these connections stand for the weighted combination of the nodes in the previous layer. For instance, consider the output value for node j indicated in Fig. 3.1 that is computed through the following two equations:

$$z_j = \sum (w_{ij} x_i + b) \ , \tag{3.1}$$

$$y_j = f(z_j) \ . \tag{3.2}$$

where x_i stands for all the inputs, w_{ij} represents the weight between node i and j, b is a bias term for regularization, z_j is a temporary value, and y_j is the output value for node j, which is computed by passing z_j through an activation function $f(\cdot)$.

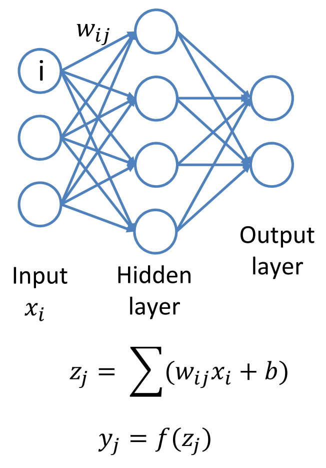


Fig. 3.1.: A sample MLP configuration.

The activation function allows MLP to deal with many problems across various research fields. It has been demonstrated that MLP is capable of approximating any linear or nonlinear function by providing appropriate constraints. There are two activation functions considered in this study: hyperbolic tangent ($tanh(x)$, Eq. 3.3) and the rectified linear unit (RELU, Eq. 3.4).

$$f(x) = tanh(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} \ , \tag{3.3}$$

$$f(x) = max(0, x) \ . \tag{3.4}$$

To establish a MLP model, a set of training data is required for the model to learn the explicit or implicit relationship between the input and the output data. The training of MLP is conducted through the minimization of the difference between the prediction of the model and the desirable target value. The weights are usually optimized through the standard Levenberg–Marquardt backpropagation algorithm [261, 262]. The computation of gradients of the activation functions is involved in the optimization process, and therefore the backpropagation may favor the activation function whose gradient is of simpler form due to computation efficiency. The gradients of $tanh(x)$ and RELU are given in Eq. 3.5 and 3.6, respectively. Although the $tanh(x)$ function is widely adopted in regression problems, the easy computation of the gradient of RELU makes RELU popular to use in recent years. After the training stage, the MLP model is evaluated through the test dataset that is not included in the training process.

$$f'(x) = 1 - (\frac{1 - e^{-2x}}{1 + e^{-2x}})^2 = 1 - tanh^2(x) \ , \tag{3.5}$$

$$f'(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{otherwise} \end{cases} \tag{3.6}$$

In this study, the vibration signals (i.e., the displacement, velocity, excitation, and the acceleration time history) of the structures obtained from either numerical simulations or experimental data are employed to be the input and output of the MLP model. The choice of inputs, outputs, as well as the MLP configuration are discussed in the following sections.

3.2.2 Convolution Neural Network

CNN, best known for its ability to extract features from raw data automatically [161], usually consists of layers of convolution, activation, pooling, and fully-connected layers. The major differences between MLP and CNN are the convolution and the pooling layers. To illustrate how the convolution layer works, Fig. 3.2 shows an example of 4×3 input convolved with two kernels with size 3×3. The kernels slide over the whole input signal with a specified stride size (e.g., the stride size is 1 in Fig. 3.2), perform point-wise multiplications, and sum up all the values to obtain the output. For instance, the output of the first kernel is $\alpha = 1\times a+2\times b+3\times c+4\times d+5\times e+6\times f+7\times g+8\times h+9\times i$, and $\beta = 4\times a+5\times b+6\times c+7\times d+8\times e+9\times f+10\times g+11\times h+12\times i$, resulting a 2×1 output vector. The second kernel does the same operation but with different weight values, and hence the total output is of size 2×2. Fig. 3.3 demonstrates an example of convolution operations. The 6×3 input is convolved with two kernels with size of 3×3, resulting a 4×2 output. The blue dashed window and the red dotted window indicate the operations performed by kernels 1 and 2, respectively. Typically, the output of each kernel is referred to as the feature map or feature channel.

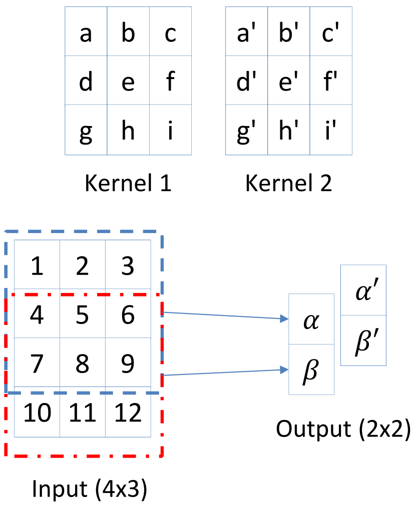


Fig. 3.2.: Illustration of a convolution layer.

Depending on the size of the input and the kernel, there could be a large number of feature maps obtained from the convolution layer. The pooling layer subsamples the feature map and extracts the dominant information in the map. Fig. 3.4 shows an example of max pooling layer. The max pooling operation extracts the maximum value in the specified filter window, and hence reduces the size of the feature map. For instance, in Fig. 3.4, the maximum value 6 in the 2×2 window is extracted, and the pooling filter slides over the whole feature map with a stride of size 2 to obtain the 2×2 output.

Fig. 3.5 shows the configuration of a typical CNN network. In Fig. 3.5, the input data is convolved with three 3×3 filters. During the convolution, the filter scans through the input data with a user-specified stride size [199]. The inputs

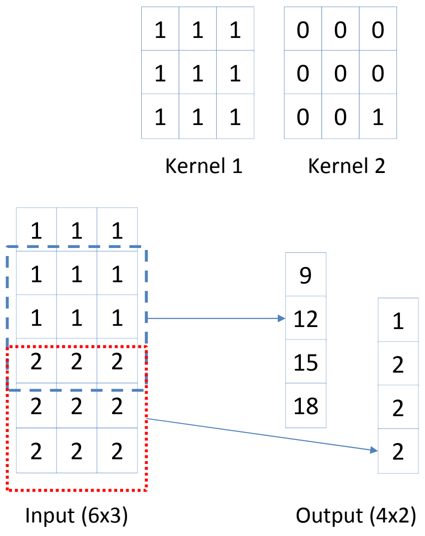


Fig. 3.3.: An example of convolution operations.

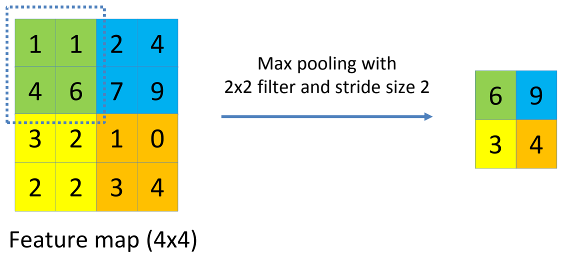


Fig. 3.4.: Illustration of a pooling layer.

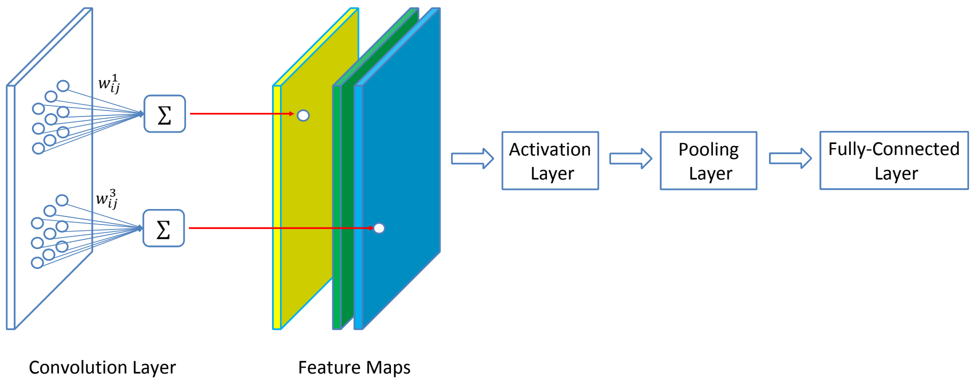


Fig. 3.5.: Typical CNN configuration.

within the filter window are multiplied with the weights (e.g., w_{ij}^1) and summed to obtain the feature value in the next layer. Therefore, all the inputs share the same weights to form one feature map. As shown in Fig. 3.5, three feature maps are obtained after the input layer is convolved with three filters. The pooling layer serves as a subsampling mechanism, which is used to reduce the output dimensionality but keep the most salient information at the same time. This information is then passed into the fully-connected layer to conduct the classification or regression tasks. Note that the configuration in Fig. 3.5 is just for illustration purposes. For deep CNNs, the network usually consists of multiple convolution and pooling layers stacked together to form a large network. Although pooling layer is usually required for image classification tasks, it may not be necessary in the context of regression problem since it only extracts the dominant information and eliminates the details that might be crucial for the regression problem. Similar to MLPs, in order to introduce nonlinearity in the system, activation functions are incorporated into CNNs. In this study, there are two activation functions considered ($tanh(x)$ Eq. 3.3 and RELU Eq. 3.4), and the pooling layer is removed from the network.

Similar to the training of MLP, the establishment of CNN model is achieved through the training and the test datasets. The weights in the filters are optimized through the minimization of the difference between the prediction of the model and the desirable target value. The optimization algorithm is the Backpropagation approach [263]. After the training, the CNN model is evaluated through the test dataset.

The differences between the configurations of MLP and CNN provide insights for why CNN is potentially more suitable for response estimation in structure dynamics. The convolution layers in CNN may serve as filters to remove noise more efficiently. Moreover, contrary to MLP, CNN is capable of dealing with various sizes of inputs while avoiding the drastically increasing in the number of weights in the network. For instance, in a MLP with one hidden layer of 15 nodes, the number of weights will increase from 45 to 1500 if the input size changes from 3 to 100. On the contrary, the number of CNN weights will only depend on the size of convolution kernels, and hence it is not sensitive to changes in the input size.

This study uses the vibration signals (i.e., the displacement, velocity, excitation, and the acceleration time history) of the structures obtained from either numerical simulation or experimental data as the input and output of the CNN model. The choice of inputs, outputs, as well as the CNN configuration are discussed in detail in the following sections.

3.3 Single Degree of Freedom (SDOF) System

In this section, MLP and CNN are used to estimate the response of a linear and a nonlinear SDOF system adopted from Masri *et. al.* [15]. The performances of MLP and CNN with respect to noise contaminated input and output signals are compared and discussed. For linear SDOF system, there are two cases of input-output relationship considered: (1) use acceleration, velocity and excitation to estimate displacement; and (2) use only excitation to estimate acceleration. For nonlinear SDOF system, the following input-output relationship is considered: use displacement and velocity to predict restoring force. In this study, five noise levels (i.e., 1%, 2%, 5%, 10% and 30%) are added to the signal to investigate the robustness of MLP and CNN against noisy data.

3.3.1 Linear SDOF System

Consider the following linear SDOF system:

$$m\ddot{x} + c\dot{x} + kx = f \ , \tag{3.7}$$

where $m = 1.0$ stands for mass, $k = 200$ stands for stiffness, $c = 1.5$ stands for damping, and f is the excitation. The input excitation is white noise with zero mean and variance 1. Fig. 3.6 depicts an illustration of this SDOF system. The sampling frequency and the data length is set to 200 (Hz) and 8300 (sec), respectively. Therefore, a total of $1,660,000$ data points are generated using the standard Newmark-Beta [264] numerical integration method. Fig. 3.7 depicts an example of the generated original vibration signals as well as the 10% noise-contaminated signals.

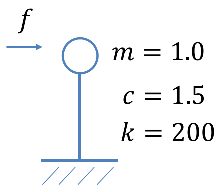
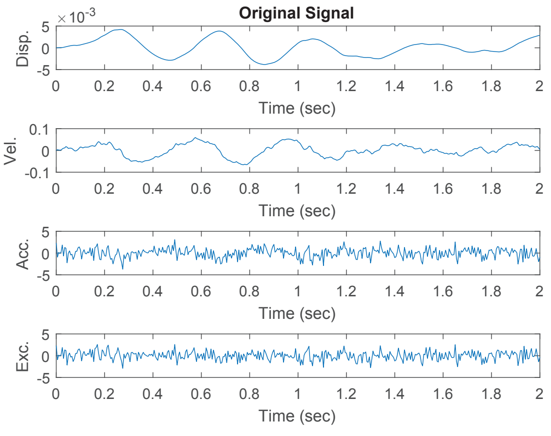
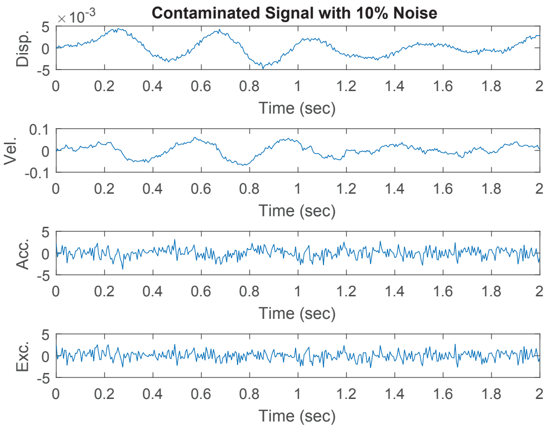


Fig. 3.6.: Illustration of the SDOF system.



(a)



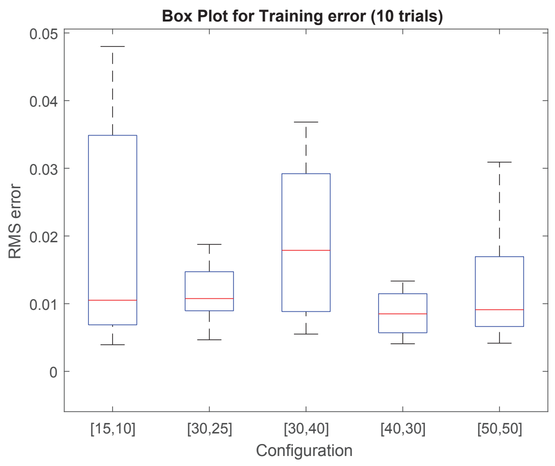
(b)

Fig. 3.7.: Vibration signals (i.e., displacement (Disp.), velocity (Vel.), acceleration (Acc.) and excitation (Exc.)) of the linear SDOF system: (a) original signals, and (b) signals contaminated with 10% noise.

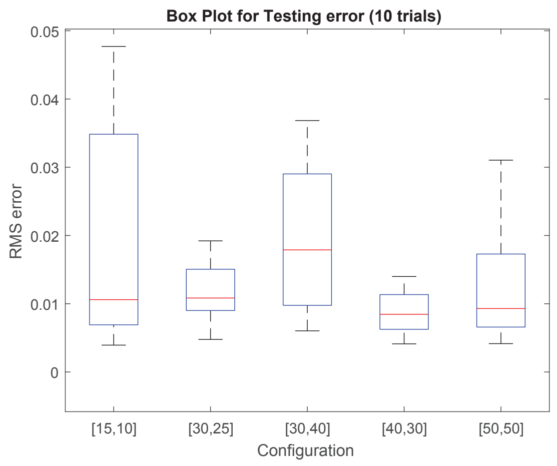
Preliminary test for MLP configuration

In this section, a preliminary test is conducted to determine the appropriate configuration for MLP. In Masri *et. al.* [15], two hidden layers with 15 and 10 hidden nodes, denoted as $[15, 10]$, were selected as the configuration for MLP. To investigate the sensitivity of different configurations, five configurations (i.e., $[15, 10]$, $[30, 25]$, $[30, 40]$, $[40, 30]$, $[50, 50]$) are selected to perform ten repeated trials for each configuration. The preliminary test uses acceleration, velocity and excitation at time k to estimate displacement at time k (i.e., $\dot{x}(k), \ddot{x}(k), f(k) \Rightarrow x(k)$, point-wise estimation). The datasets are divided randomly into 70% training set and 30% test set. Before training, the output signals are scaled to make it comparable to the inputs, and the inputs are normalized to $[-1, 1]$. No noise is added for the preliminary test. Root-mean-square (RMS) error indicator is selected for performance evaluation in this study.

Fig. 3.8 shows the variations and box plots of training and testing errors of the five MLP configurations using the ten repeated trials. The training and testing error of each configuration are almost identical, which demonstrates the capability of generalization achieved by MLP. According to Fig. 3.8, it clearly indicates that $[40, 30]$ outperforms other configurations with a smaller variation and lower median of RMS error for both training and test datasets. As a result, this study selects $[40, 30]$ as the MLP configuration to conduct point-wise prediction for SDOF systems.



(a)



(b)

Fig. 3.8.: Error variation for different MLP configurations: (a) training error, and (b) test error.

CNN configuration for SDOF system

Fig. 3.9 depicts the proposed CNN configuration for SDOF system. As mentioned above, the pooling layer is deployed in this study for conducting response estimation. The input to the network is three 100×1 vectors, which stand for velocity, acceleration, and excitation, respectively. The output of network is a 100×1 displacement vector. Note that the first convolution layer is designed to be of size $9 \times 3 \times 3$. The intuition behind this is that the convolution layer can act as a filter to remove the noise from the input signals. There is no overlap between each sample, and hence a total of $1,660,000/100 = 16600$ samples are used for training and testing. The datasets are divided randomly into 70% training set and 30% test set. Before training, the outputs are scaled to make it comparable to the inputs, and the inputs are normalized to $[-1, 1]$. For the case of using excitation to predict acceleration ($f \Rightarrow \ddot{x}$), the input to the network is a 100×1 vector, and the first and second convolution layers in Fig. 3.9 are replaced with two filters size of $9 \times 1 \times 1$ and $9 \times 1 \times 8$, respectively.

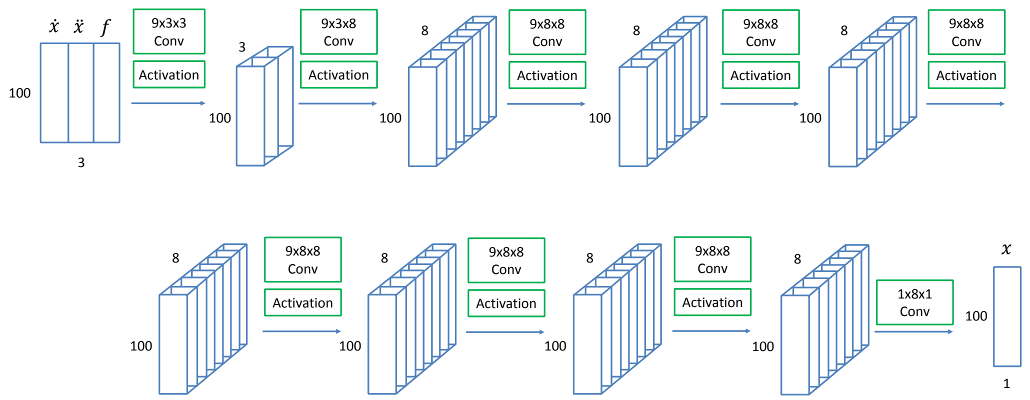


Fig. 3.9.: CNN configuration for SDOF system.

Results and discussions

For real world problems, the presence of noise is inevitable during the data collection process. The source of noise may come from sensor itself, ambient vibration, or even temperature variations. It is impossible to have a clean (i.e., ideal) signal for real world applications. In this study, there are five noise level (i.e., 1%, 2%, 5%, 10% and 30%) considered to simulate the practical situations. The noise is added independently to each vibration signal, and each noise signal is obtained by multiplying the noise level with the standard deviation of each vibration signal [253]. Each network is trained with the noisy input and noisy output data.

Table 3.1 to 3.4 shows the RMS error of using velocity, acceleration, and excitation to predict the displacement, for MLP-*tanh*, CNN-*tanh*, MLP-RELU, and CNN-RELU, respectively. The second and third columns show the error with respect to the noisy target, and the fourth and fifth columns show the error with respect to the ideal target. In general, both algorithms achieve good estimations with a low RMS error even if the signal is contaminated with noise. The performances of the two activation functions, *tanh* and RELU, are similar. Also, the training and testing RMS errors for both MLP and CNN are very similar, which means that there is no overfitting for the two models (overfitting occurs when the learning model performs well on the training data but poor on the test data). Since the RMS errors with respect to the ideal target are always smaller than the RMS errors with respect to the noisy target, this demonstrates that both MLP and CNN are capable of learning the "true response" given noisy input and output data. For ideal case (i.e., 0% noise), MLP outperform CNN. However, for noisy cases, CNN achieves lower RMS errors for all cases. Compared with the performance of MLP, Tables 3.1 and 3.2 show that CNN reduces the test RMS error for 1%, 2%, 5%, 10% and 30% noisy cases by 22%, 56%, 75%, 80% and 82%, respectively. This indicates that CNN is more robust against noise.

As a representative case, Fig. 3.10 and 3.11 show the error distribution of 10% noise data, *tanh*, for MLP and CNN, respectively. The red solid line is the fitted normal distribution curve superimposed on the histogram for comparison reasons. The test error standard deviation for CNN is 0.0669, which is five times lower than the test error standard deviation for MLP (i.e., 0.3319). In other words, the error distribution of CNN is more centered to zero than that of MLP. Fig. 3.12 and Fig. 3.13 depict the prediction of 10% noisy data with *tanh* activation function for MLP and CNN with respect to the ideal target, respectively. The blue dashed line indicates the ideal target while the red line is the prediction from the learning model. The MLP prediction captures the wave shape of the ideal target but there exists some small unwanted high frequency oscillations. On the contrary, the CNN prediction almost matches with the ideal target. Fig. 3.14 shows the test RMS error variation versus different noise levels for both MLP and CNN. As the noise level increases, the RMS error of CNN increases much slower than the MLP.

Table 3.1.: RMS Error - MLP-*tanh* Results ($\dot{x}, \ddot{x}, f \Rightarrow x$)

Noise (%)	Train	Test	Train (w.r.t Ideal)	Test (w.r.t Ideal)
0	0.012	0.012	-	-
1	0.037	0.037	0.035	0.035
2	0.073	0.073	0.069	0.069
5	0.181	0.181	0.170	0.170
10	0.356	0.356	0.332	0.332
30	0.812	0.812	0.730	0.730

Another case considered in the linear SDOF system is to use excitation f to estimate the acceleration $\ddot{x}$. This is of particular interest since it is difficult to acquire the information of displacement responses in real world. Tables 3.5 to 3.8 present the RMS error of using excitation to predict the acceleration for MLP-*tanh*, CNN-*tanh*, MLP-RELU and CNN-RELU, respectively. In this case, there is still no overfitting for both MLP and CNN model since the RMS errors for training and testing are similar. Also, there is no obvious discrepancy between the performances of *tanh* and RELU activation functions. The RMS error with respect to the ideal target is slightly smaller

Table 3.2.: RMS Error - CNN-*tanh* Results ($\dot{x}, \ddot{x}, f \Rightarrow x$)

Noise (%)	Train	Test	Train (w.r.t Ideal)	Test (w.r.t Ideal)
0	0.027	0.026	-	-
1	0.031	0.030	0.029	0.027
2	0.040	0.039	0.031	0.030
5	0.076	0.075	0.044	0.043
10	0.146	0.146	0.068	0.067
30	0.380	0.379	0.135	0.134

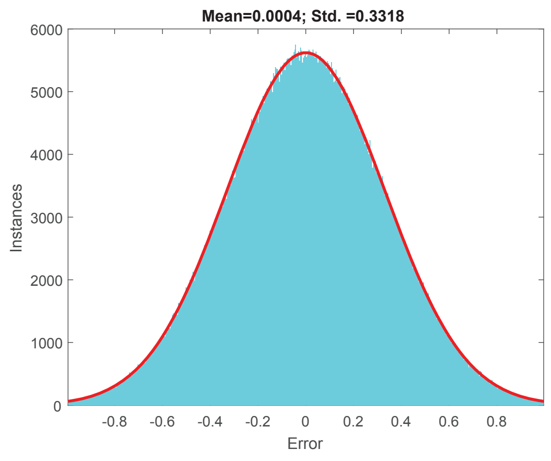
Table 3.3.: RMS Error - MLP-RELU Results ($\dot{x}, \ddot{x}, f \Rightarrow x$)

Noise (%)	Train	Test	Train (w.r.t Ideal)	Test (w.r.t Ideal)
0	0.007	0.007	-	-
1	0.049	0.049	0.047	0.047
2	0.072	0.072	0.068	0.068
5	0.176	0.176	0.165	0.165
10	0.356	0.356	0.332	0.332
30	0.817	0.817	0.736	0.736

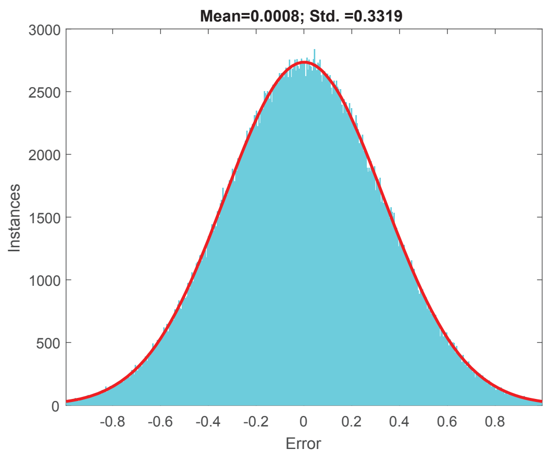
than the error with respect to the noisy target. Compared with the performance of MLP, Tables 3.5 and 3.6 show that CNN reduces the test RMS error for 1%, 2%, 5%, 10% and 30% noisy cases by approximately 6% to 8%. Although CNN outperforms MLP, the reduction in the RMS error is not as significant as the case of using $\dot{x}, \ddot{x}, f$ to predict x. As a representative case, Fig. 3.15 and 3.16 depict the error distribution of 10% noise data, using *tanh* activation function, for MLP and CNN, respectively. The

Table 3.4.: RMS Error - CNN-RELU Results ($\dot{x}, \ddot{x}, f \Rightarrow x$)

Noise (%)	Train	Test	Train (w.r.t Ideal)	Test (w.r.t Ideal)
0	0.025	0.024	-	-
1	0.029	0.029	0.026	0.026
2	0.044	0.043	0.036	0.036
5	0.081	0.081	0.052	0.052
10	0.146	0.146	0.069	0.069
30	0.381	0.381	0.139	0.138



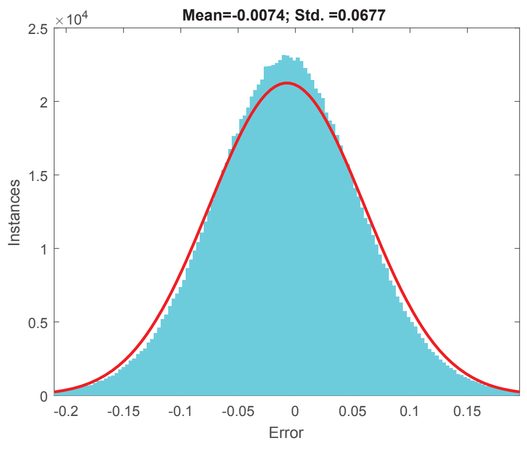
(a)



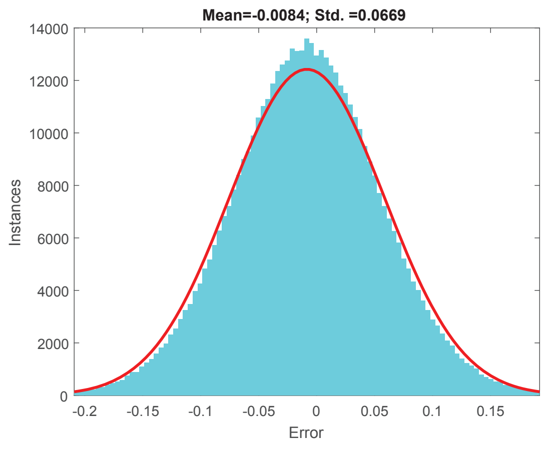
(b)

Fig. 3.10.: MLP error histogram - *tanh*, 10% noise. (a) training error histogram, and (b) test error histogram.

red solid line is the fitted normal distribution curve superimposed on the histogram for comparison reasons. The test error standard deviation of CNN is 0.4756, which is lower than the test error standard deviation of MLP (i.e., 0.5133). Fig. 3.17 and 3.18 present the prediction of 10% noise data with *tanh* activation function for MLP and CNN with respect to the ideal target, respectively. The blue dashed line indicates the



(a)



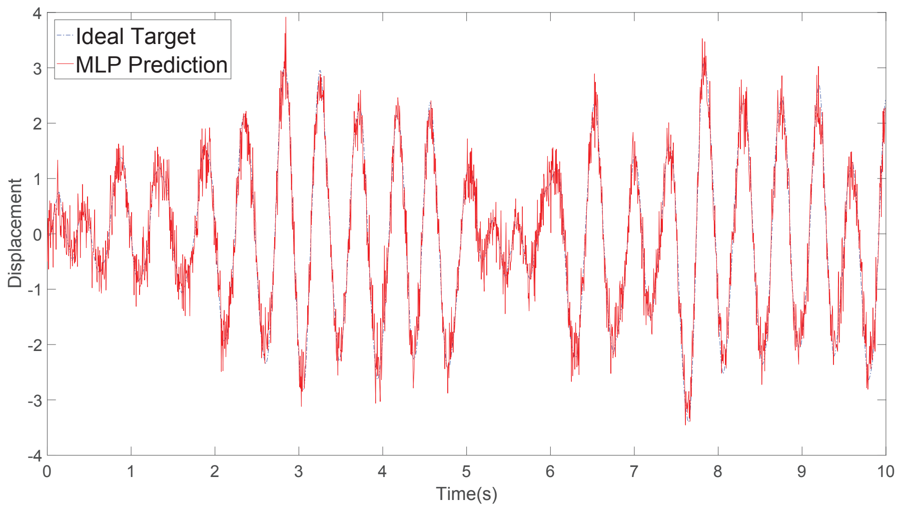
(b)

Fig. 3.11.: CNN error histogram - *tanh*, 10% noise. (a) training error histogram, and (b) test error histogram.

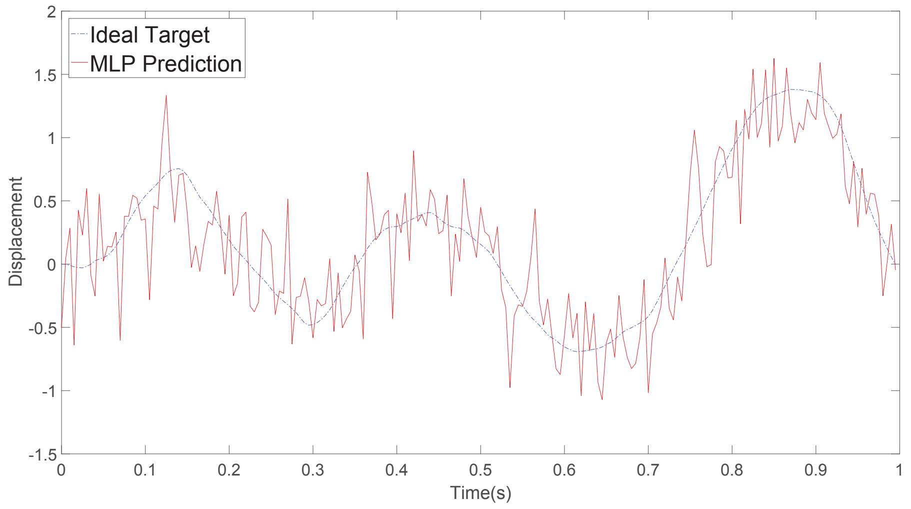
ideal target while the red line is the prediction from the learning model. In this case, the CNN matches slightly better with the ideal target compared with the prediction of MLP by a 7% reduction of test RMS error. Fig. 3.19 shows the test RMS error variation versus different noise levels for both MLP and CNN. The RMS error for CNN is always lower than MLP for all noise levels.

Table 3.5.: RMS Error - MLP-*tanh* Results ($f \Rightarrow \ddot{x}$)

Noise (%)	Train	Test	Train (w.r.t Ideal)	Test (w.r.t Ideal)
0	0.531	0.531	-	-
1	0.532	0.532	0.532	0.532
2	0.529	0.529	0.529	0.528
5	0.531	0.531	0.528	0.528
10	0.523	0.523	0.513	0.513
30	0.660	0.660	0.582	0.582



(a)



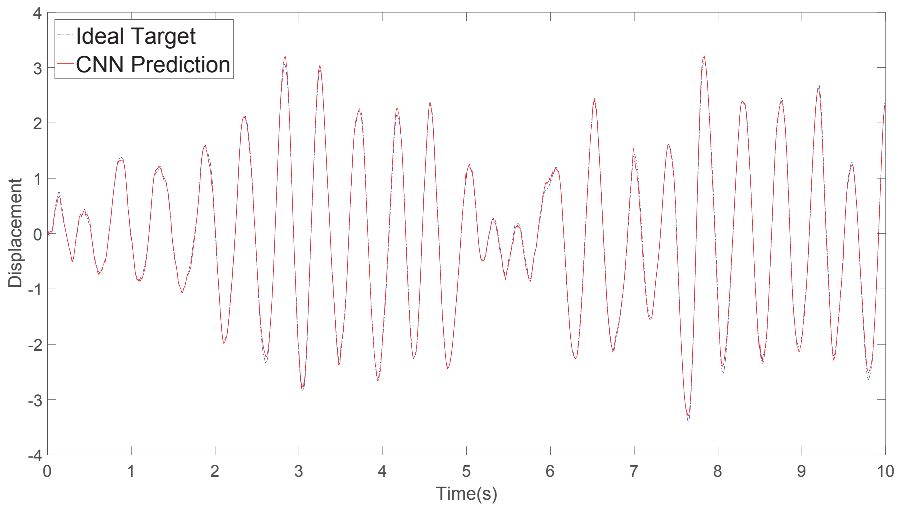
(b)

Fig. 3.12.: MLP prediction versus ideal target (*tanh*, 10% noise): (a) 10 (sec) of the estimated displacement and (b) the first one second of the estimated response.

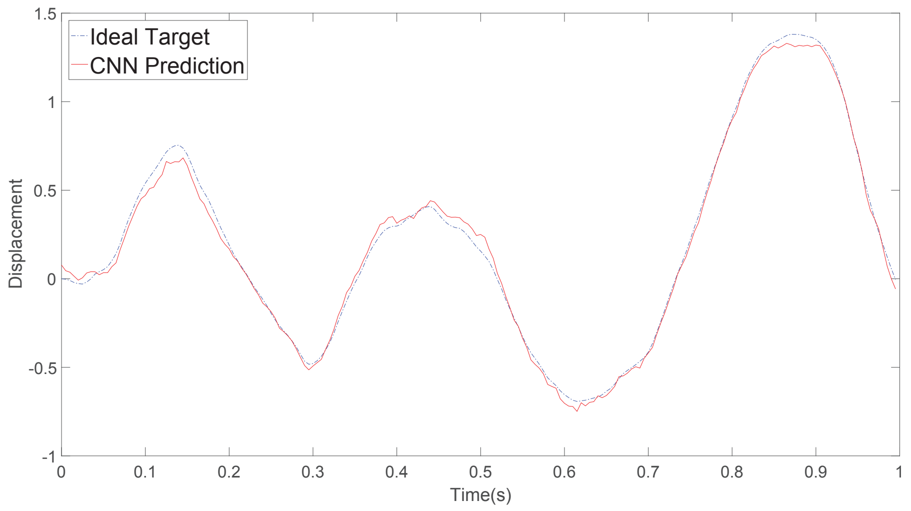
3.3.2 Nonlinear SDOF System

Consider the following nonlinear SDOF system:

$$g(x, \dot{x}) = ax + bx^3 + c\dot{x} \, , \tag{3.8}$$



(a)



(b)

Fig. 3.13.: CNN prediction versus ideal target ($tanh$, 10% noise): (a) 10 (sec) of the estimated displacement and (b) the first one second of the estimated response.

where $a = (2\pi)^2$ is the linear stiffness coefficient, $b = 10$ is the nonlinear stiffness coefficient, $c = 1.25$ stands for the damping coefficient, and $g(x, \dot{x})$ is the restoring force. The input excitation is the white noise signal with zero mean and variance large enough to drive the system into the nonlinear phase. Similar to the linear

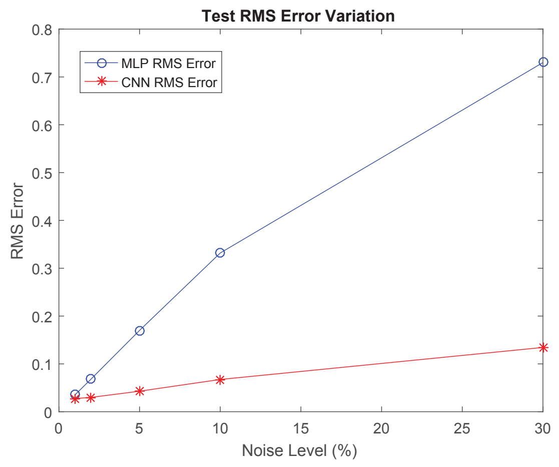


Fig. 3.14.: Test RMS error variations for MLP and CNN ($\dot{x}, \ddot{x}, f \Rightarrow x$; $tanh$).

Table 3.6.: RMS Error - CNN-$tanh$ Results ($f \Rightarrow \ddot{x}$)

Noise (%)	Train	Test	Train (w.r.t Ideal)	Test (w.r.t Ideal)
0	0.492	0.490	-	-
1	0.493	0.492	0.493	0.491
2	0.490	0.489	0.490	0.489
5	0.492	0.490	0.489	0.488
10	0.488	0.486	0.477	0.476
30	0.632	0.632	0.551	0.549

SDOF system, the sampling rate and the data length are 200 (Hz) and 8300 (sec), respectively. Runge-Kutta method is employed to generate the numerical vibration responses. Fig. 3.20 plots the restoring force versus the displacement and the velocity time history. Nonlinear response behavior is observed from the response plot.

In this case, the displacement and the velocity are used to predict the restoring force (i.e., $x, \dot{x} \Rightarrow g(x, \dot{x})$). Similar to the linear SDOF system, configuration $[40, 30]$ is selected for MLP model. The input to MLP is the displacement and velocity at

Table 3.7.: RMS Error - MLP-RELU Results ($f \Rightarrow \ddot{x}$)

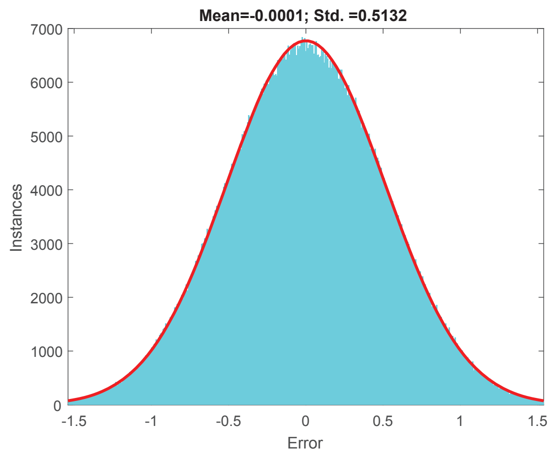
Noise (%)	Train	Test	Train (w.r.t Ideal)	Test (w.r.t Ideal)
0	0.531	0.531	-	-
1	0.532	0.532	0.532	0.532
2	0.529	0.529	0.529	0.529
5	0.531	0.530	0.528	0.528
10	0.523	0.523	0.513	0.513
30	0.660	0.660	0.582	0.582

Table 3.8.: RMS Error - CNN-RELU Results ($f \Rightarrow \ddot{x}$)

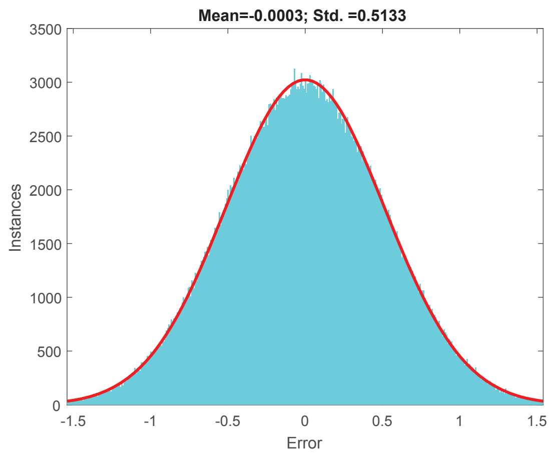
Noise (%)	Train	Test	Train (w.r.t Ideal)	Test (w.r.t Ideal)
0	0.497	0.495	-	-
1	0.498	0.496	0.498	0.496
2	0.496	0.494	0.495	0.494
5	0.497	0.495	0.494	0.493
10	0.493	0.491	0.482	0.481
30	0.637	0.637	0.557	0.555

time k, and the output of it is the restoring force at time k. The dataset is randomly divided into 70% training and 30% test dataset. Before training, the output is scaled to make it comparable to the inputs, and the inputs are normalized to $[-1, 1]$.

In terms of the CNN model, the input to the network is two 100×1 vectors, and the first and second convolution layer in Fig. 3.9 are replaced with two filters with size $9 \times 2 \times 2$ and $9 \times 2 \times 8$, respectively. There is no overlapping between each

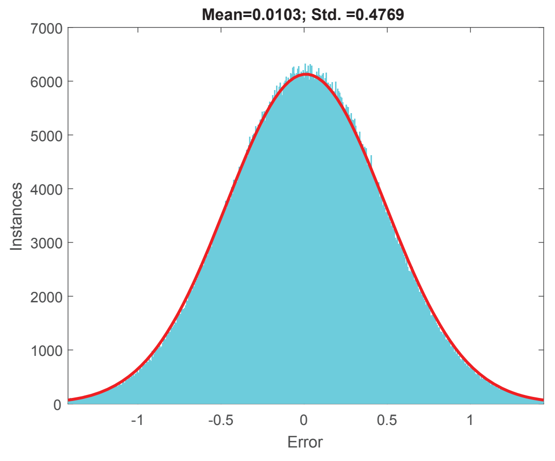


(a)

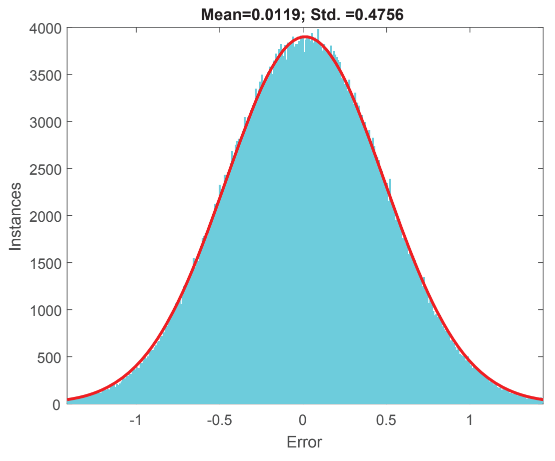


(b)

Fig. 3.15.: MLP error histogram - 10% noise. (a) training error histogram, and (b) test error histogram.



(a)



(b)

Fig. 3.16.: CNN error histogram - 10% noise. (a) training error histogram, and (b) test error histogram.

sample, and hence a total of $1,660,000/100 = 16600$ samples are used for training and testing. The datasets are divided randomly into 70% training set and 30% test

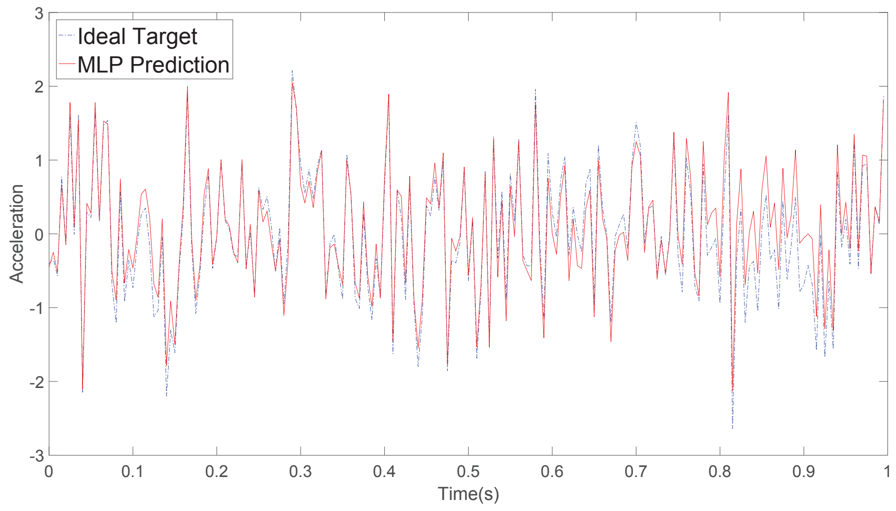


Fig. 3.17.: MLP prediction versus ideal target (10% noise).

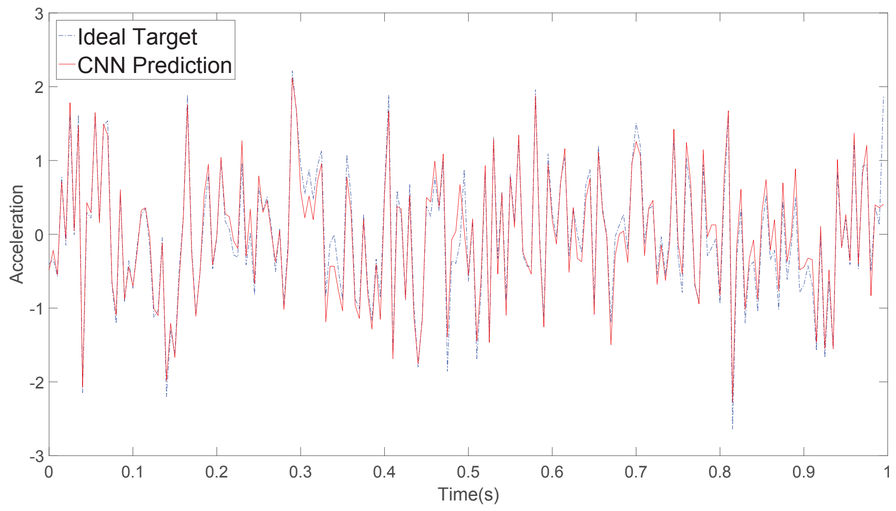


Fig. 3.18.: CNN prediction versus ideal target (10% noise).

set. Before the training of CNN, the output is scaled to make it comparable to the inputs, and the inputs are normalized to $[-1, 1]$.

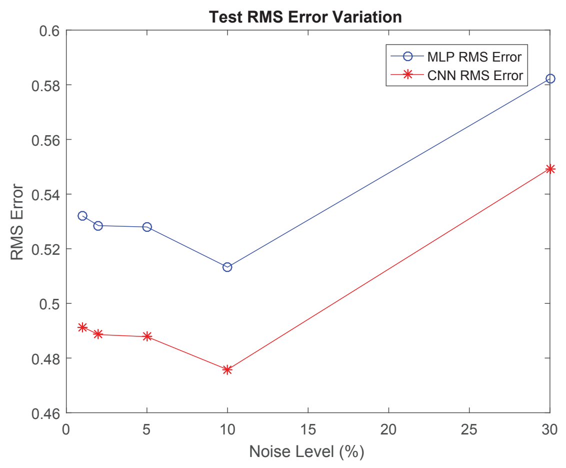
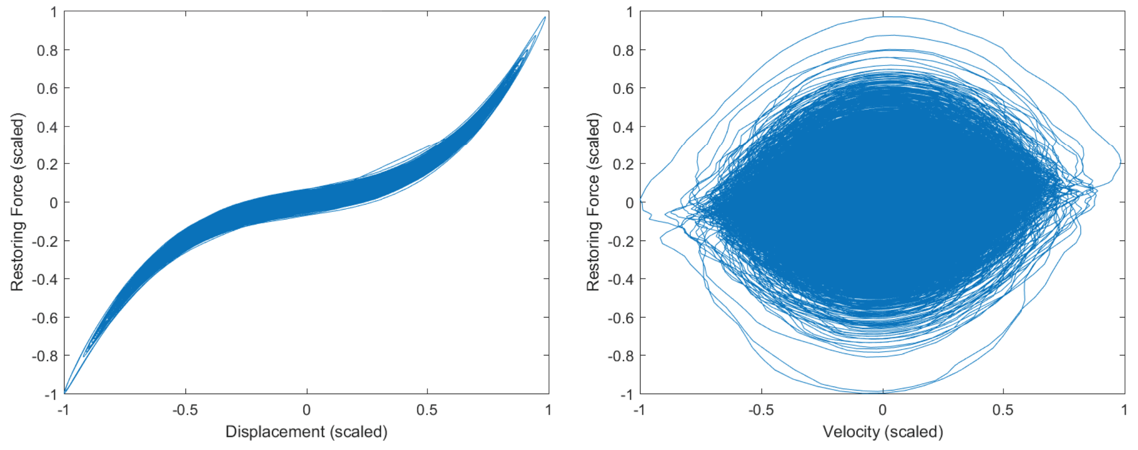


Fig. 3.19.: Test RMS error variation ($f \Rightarrow \ddot{x}$).



(a) (b)

Fig. 3.20.: (a) Nonlinear restoring force versus displacement and (b) restoring force versus velocity.

Results and discussions

Five noise levels (i.e., 1%, 2%, 5%, 10% and 30%) are considered for performance evaluation purposes. Tables 3.9 to 3.12 show the RMS error of using displacement and velocity to predict the restoring force, for MLP-*tanh*, CNN-*tanh*, MLP-RELU, and CNN-RELU, respectively. In general, both MLP and CNN algorithms achieve small

RMS errors for estimating the restoring force under the presence of noise. There is no overfitting phenomenon observed for both MLP and CNN models since the RMS errors of training and test dataset are close. For the ideal case (i.e., noise level 0%), MLP outperform CNN. For noise level greater than 1%, CNN is superior to MLP with smaller RMS errors with respect to the ideal target. Compared with the performance of MLP, Tables 3.9 and 3.10 show that CNN reduces the test RMS error for 2%, 5%, 10% and 30% noisy cases by 21%, 45%, 68% and 73%, respectively. According to the comparison between the RMS errors with respect to the noisy target and the ideal target, both MLP and CNN are capable of eliminating the effect of noise and learning to estimate the ideal target. The performances of *tanh* and RELU function are also close to each other.

As a representative case, Fig. 3.21 and 3.22 present the error distribution of 10% noise data with *tanh* as activation function for MLP and CNN, respectively. The red solid line is the fitted normal distribution curve superimposed on the histogram for comparison reasons. The test error standard deviation for CNN is 0.1869, which is three times lower than the test error standard deviation of MLP (i.e., 0.5810). This means that the error distribution of CNN is more centered to zero than that of MLP. Fig. 3.23 and 3.24 show the response prediction of 10% noise data, *tanh* activation function, for MLP and CNN with respect to the ideal target, respectively. The red line is the predicted response, and the blue dashed line is the ideal target. The CNN prediction almost matches with the ideal target response while the MLP prediction captures the target wave shape but exhibits some undesirable high frequency oscillations. Fig. 3.25 depicts the test RMS error variation versus different noise levels for both MLP and CNN. This figure indicates that the RMS errors of CNN increases much slower than the RMS errors of MLP as the noise level increased.

To further test the ability of the proposed CNN approach in generalization, the nonlinear SDOF responses are re-simulated using a random excitation with length of 50 (sec). Similarly, the displacement, velocity and restoring force responses are contaminated with 10% noise. The trained CNN model with 10% noisy measurements are employed to predict the noise-free restoring force given the noisy displacement and velocity measurements. Fig. 3.26 shows the prediction from CNN and the target restoring force. The CNN model accurately captures the dynamic behavior of the system even when the responses are generated using a new excitation. The overall RMS error between CNN prediction and target response is 0.1706, which is close to the test RMS error 0.188 indicated in Table 3.10. This further demonstrates the trained CNN model generalizes very well on new data. Note that when applying new data, the scaling factors for input and output measurements must be identical to the scaling factors used during training stage.

Table 3.9.: RMS Error - Nonlinear SDOF MLP-*tanh* Results

Noise (%)	Train	Test	Train (w.r.t Ideal)	Test (w.r.t Ideal)
0	0.060	0.059	-	-
1	0.105	0.100	0.094	0.089
2	0.158	0.158	0.127	0.127
5	0.391	0.392	0.313	0.313
10	0.752	0.753	0.581	0.581
30	2.155	2.152	1.609	1.606

3.4 Multi-Degree of Freedom (MDOF) System

In this section, vibration responses from a three-story steel frame tested on a shake table is used to establish the MLP and CNN models to predict the dynamic behavior of the structural system. The objective is to predict the roof acceleration $\ddot{x}_3$ given only the information of the ground excitation f. The prediction from CNN model is further employed to perform the system identification of the steel frame.

Table 3.10.: RMS Error - Nonlinear SDOF CNN-*tanh* Results

Noise (%)	Train	Test	Train (w.r.t Ideal)	Test (w.r.t Ideal)
0	0.213	0.179	-	-
1	0.186	0.166	0.180	0.159
2	0.163	0.137	0.133	0.100
5	0.308	0.292	0.200	0.173
10	0.521	0.514	0.208	0.188
30	1.500	1.501	0.451	0.439

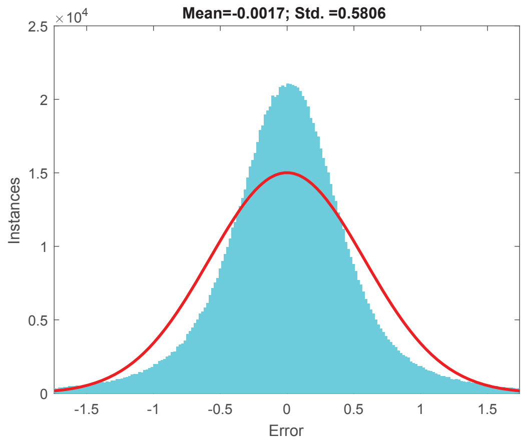
Table 3.11.: RMS Error - Nonlinear SDOF MLP-RELU Results

Noise (%)	Train	Test	Train (w.r.t Ideal)	Test (w.r.t Ideal)
0	0.251	0.250	-	-
1	0.104	0.104	0.093	0.092
2	0.249	0.249	0.231	0.230
5	0.399	0.398	0.323	0.322
10	0.788	0.788	0.626	0.626
30	2.181	2.175	1.644	1.640

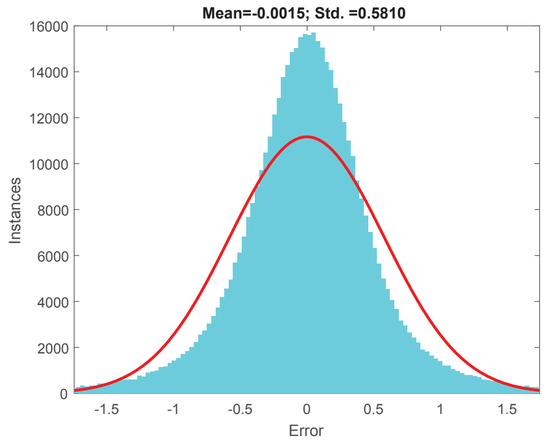
To account for the complexity in the real MDOF system, the input to both MLP and CNN models is a three seconds excitation signal, which is a 600×1 vector for sampling frequency $200(Hz)$. The output of the network is a 1×1 roof acceleration signal at the end of the corresponding three-second time interval. Each input sample has a overlapping of 599 points with each other. For instance, the input and output for the first sample are $f(600), f(599), ..., f(1)$ and $\ddot{x}_3(600)$, respectively. The input and

Table 3.12.: RMS Error - Nonlinear SDOF CNN-RELU Results

Noise (%)	Train	Test	Train (w.r.t Ideal)	Test (w.r.t Ideal)
0	0.252	0.217	-	-
1	0.140	0.140	0.132	0.132
2	0.150	0.150	0.117	0.117
5	0.279	0.279	0.152	0.151
10	0.517	0.519	0.198	0.200
30	1.500	1.504	0.451	0.450



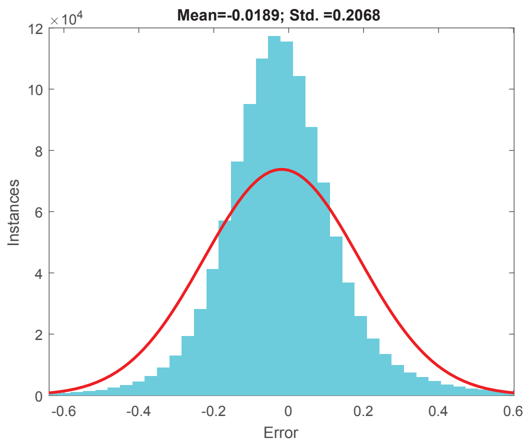
(a)



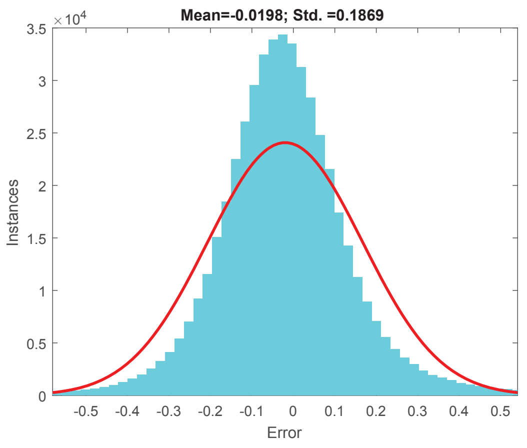
(b)

Fig. 3.21.: MLP error histogram - *tanh*, 10% noise. (a) training error histogram, and (b) test error histogram.

output for the second sample are $f(601), f(600), ..., f(2)$ and $\ddot{x}_3(601)$, respectively. Therefore, the first 599 points of roof acceleration is discarded during the training and testing. The MLP configuration is selected as $[40, 30]$, and the proposed CNN configuration is shown in Fig. 3.27. The dataset consists of $197,612$ samples, and it is



(a)



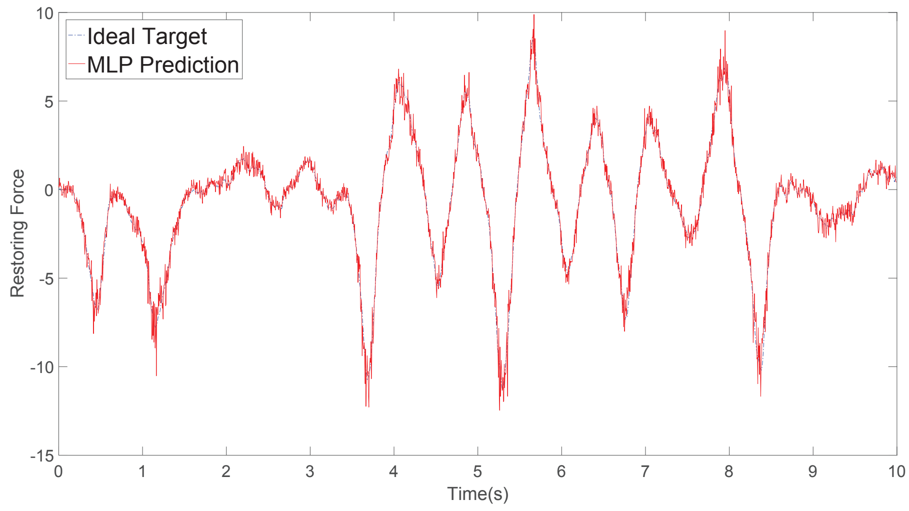
(b)

Fig. 3.22.: CNN error histogram - *tanh*, 10% noise. (a) training error histogram, and (b) test error histogram.

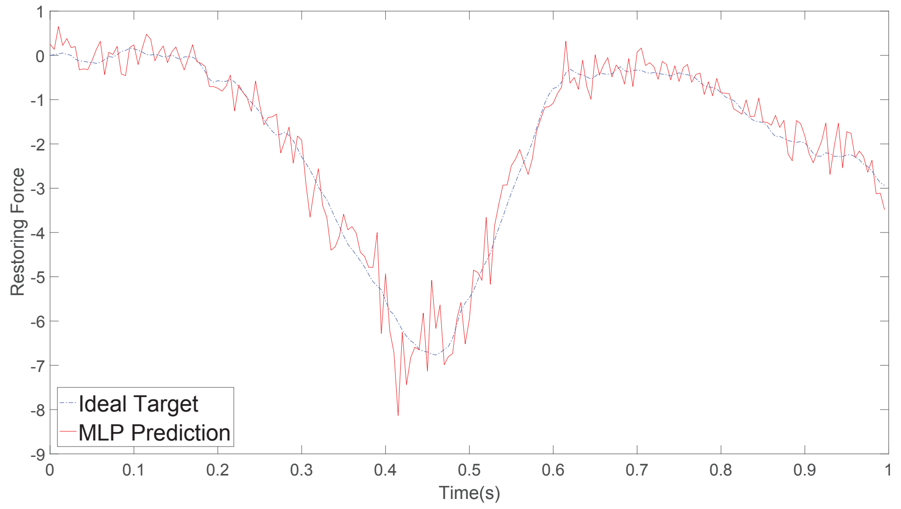
randomly divided into 85% training and 15% test dataset. Before training, the input and output are scaled to $[-1, 1]$. The *tanh* activation function is used in this case.

3.4.1 Shake Table Test Setup

Fig. 3.28 shows the full-scale three-story steel frame built at the National Center for Research on Earthquake Engineering (NCREE) in Taiwan. The frame is designed as a moment resisting frame with each floor height equal to $3(m)$. The length, width and thickness of the steel floor slab are $3(m)$, $2(m)$ and $3(cm)$, respectively. The dimensions of each beam and column is H150 $\times$ 150 $\times$ 7 $\times$ 10. Accelerometers and linear variable differential transformer (LVDT) sensors are installed at each floor slab to measure the floor acceleration and the floor displacements, respectively. The steel frame is excited with the ElCentro earthquake, random excitation, and the TCU076 ground motion recorded during the 1999 Chi-Chi earthquake in Taiwan. In this study, only the responses recorded in the transverse direction of the frame are used for the analysis.



(a)

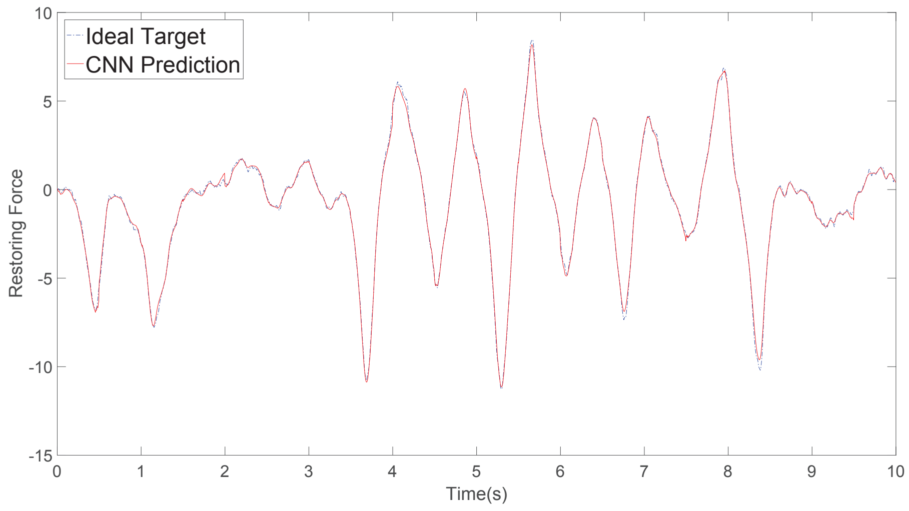


(b)

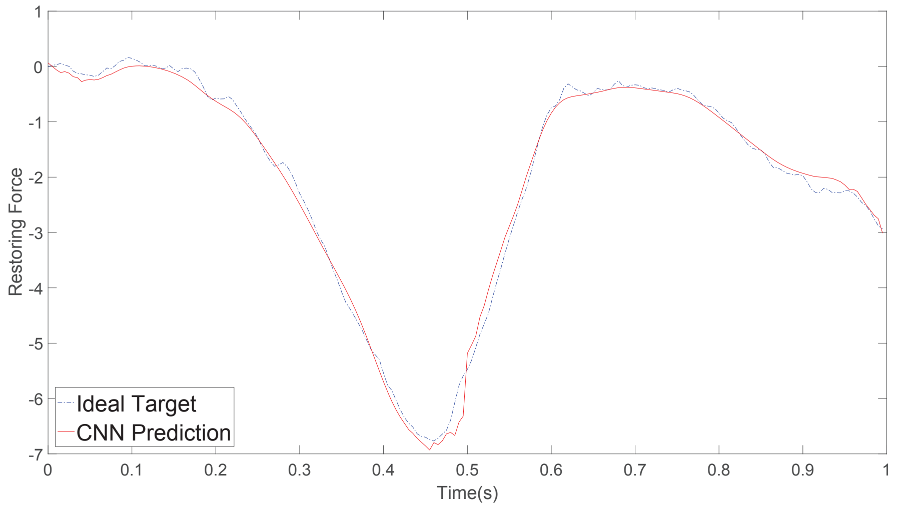
Fig. 3.23.: MLP prediction versus ideal target ($tanh$, 10% noise): (a) 10 (sec) of the estimated restoring force and (b) the first one second of the estimated response.

3.4.2 Shake Table Test Results

Table 3.13 presents the training and test RMS error for both MLP and CNN model. The RMS errors for CNN is 19.38% lower than the errors of MLP. Since the errors for training and testing are the same, there is no overfitting for both the



(a)



(b)

Fig. 3.24.: CNN prediction versus ideal target (*tanh*, 10% noise): (a) 10 (sec) of the estimated restoring force and (b) the first one second of the estimated response.

MLP and CNN models. Fig. 3.29 and 3.30 show the error distribution for MLP and CNN predictions, respectively. The red solid line is the fitted normal distribution curve superimposed on the histogram for comparison reasons. The test error standard deviation for CNN is 0.1042, which is 19% lower than the test error standard deviation

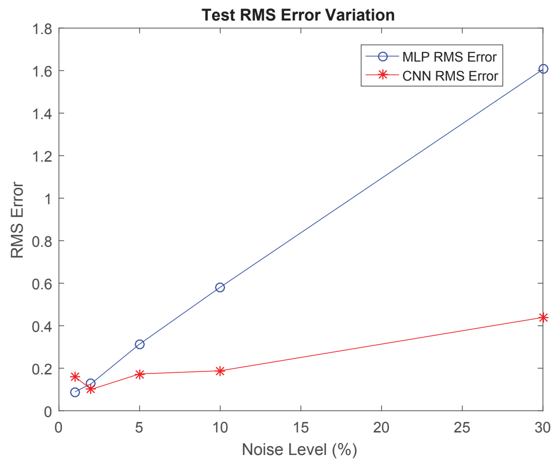


Fig. 3.25.: Test RMS error variation

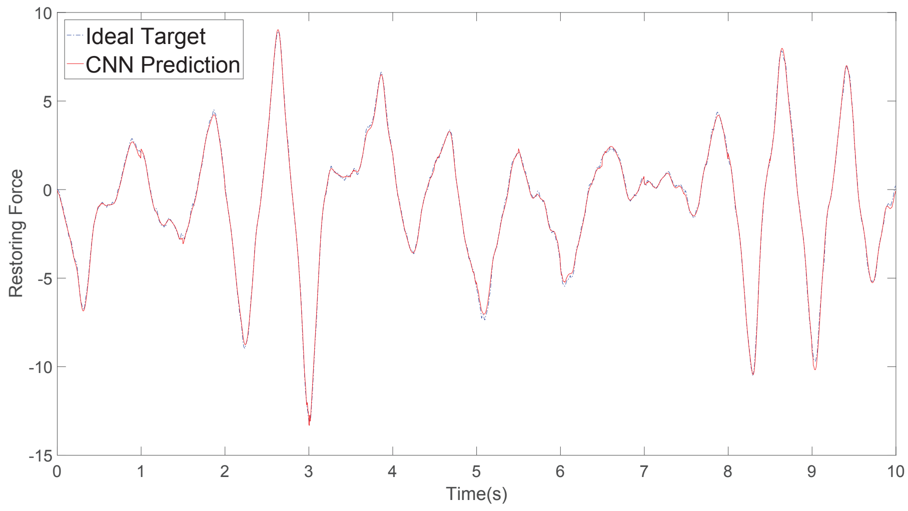
Table 3.13.: RMS Error - MLP and CNN

Method	Train	Test
MLP	0.129	0.129
CNN	0.104	0.104

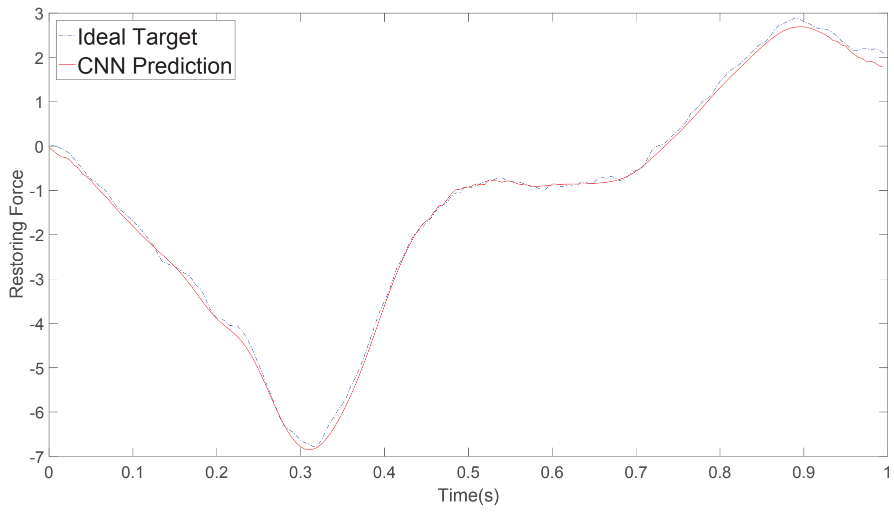
of MLP (i.e., 0.1287). It is seen that the CNN prediction errors is more centered to zero than that of MLP. Fig. 3.31 and 3.32 show a ten seconds predicted response versus target response for MLP and CNN, respectively. The blue dashed line is the target response while the red line is the estimation from the prediction models. The CNN estimation matches slightly better with the target response.

3.4.3 System Identification

This section demonstrates that the prediction from the established CNN model can be further used to perform system identification. Fig. 3.33 shows the frequency response of the recorded NCREE roof acceleration data and the predicted roof acceleration from CNN. Table 3.14 presents the identified first three natural frequencies in



(a)



(b)

Fig. 3.26.: CNN prediction versus ideal target using measurements from new excitation: (a) 10 (sec) of the estimated restoring force and (b) the first one second of the estimated response.

the transverse direction from the NCREE measurements and the CNN estimations. The identified mode frequencies from CNN is very close to the ones obtained from

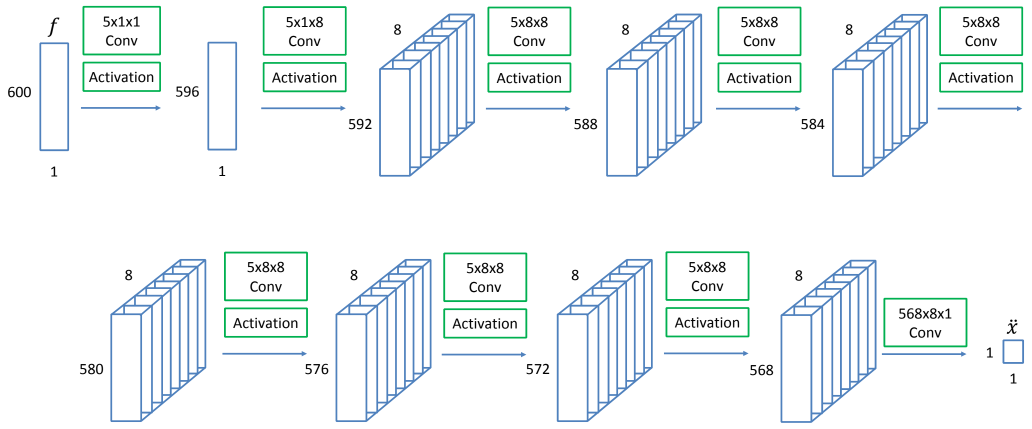


Fig. 3.27.: CNN configuration for MDOF system.

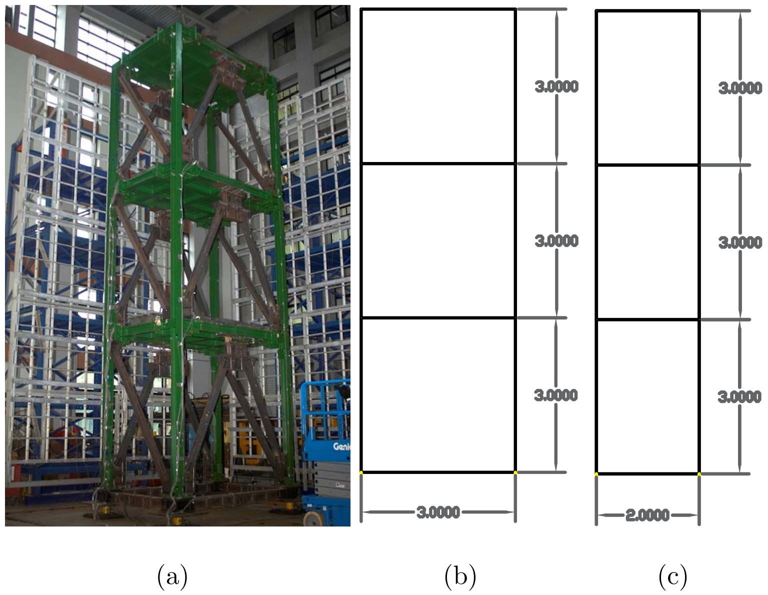
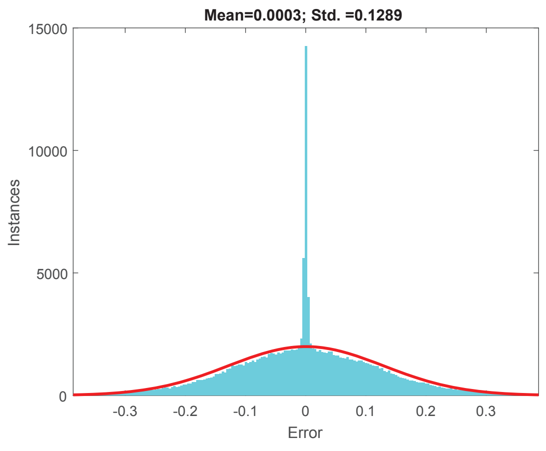
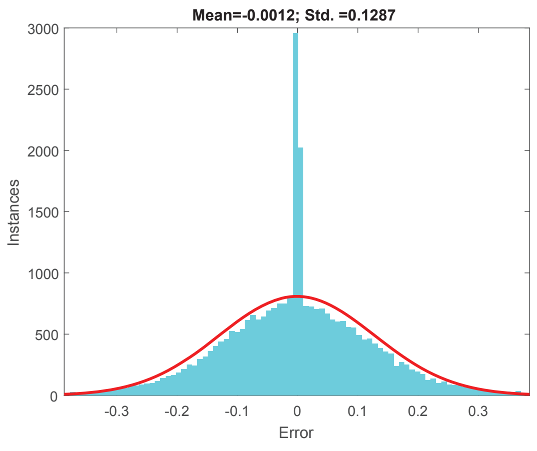


Fig. 3.28.: Three-story steel frame at NCREE: (a) frame photo, (b) frame front view and (c) frame side view, unit: (m). (Image (a) courtesy of NCREE)

NCREE data. The identification error is below 8% for setting the NCREE data as the ground truth.

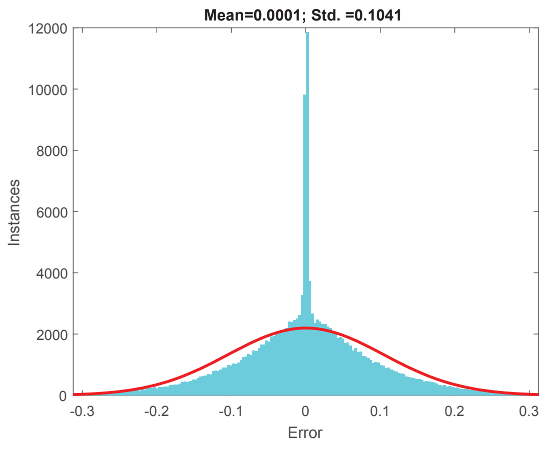


(a)

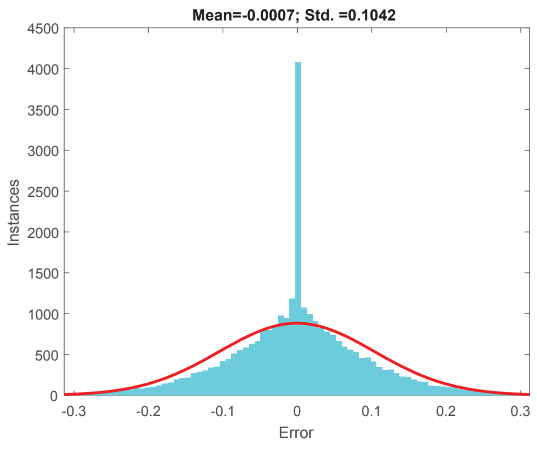


(b)

Fig. 3.29.: MLP error histogram. (a) training error histogram, and (b) test error histogram.



(a)



(b)

Fig. 3.30.: CNN error histogram. (a) training error histogram, and (b) test error histogram.

3.5 Physical Interpretation of Convolution Layer

Although the learning capability of conventional MLP has been demonstrated in several studies, how to interpret the physical meaning of the MLP model remains a

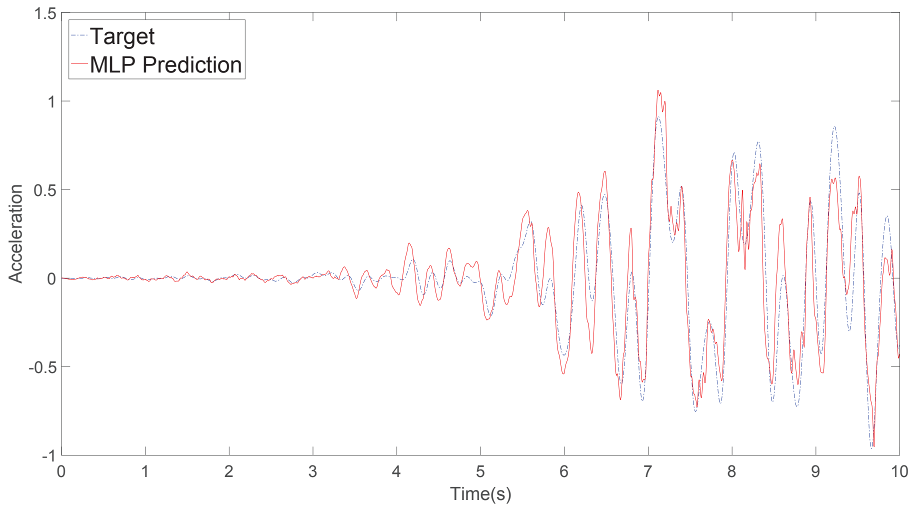


Fig. 3.31.: MLP prediction versus target.

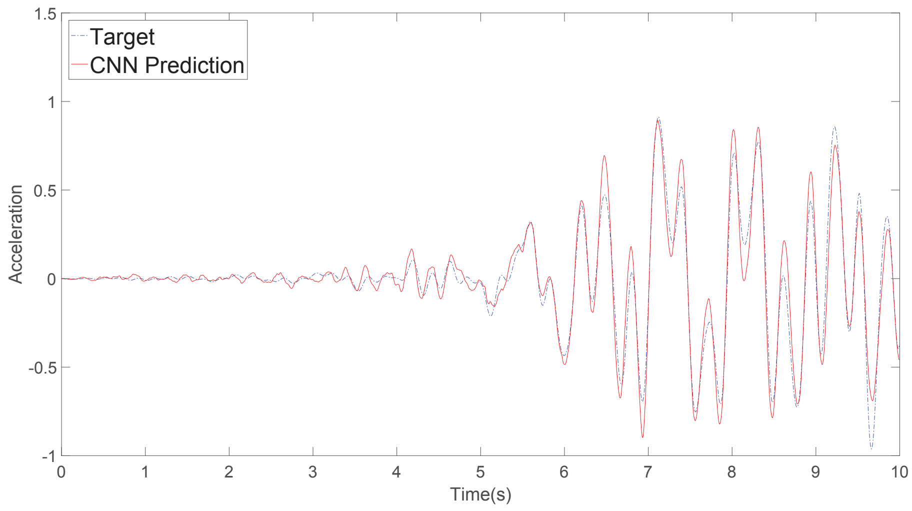
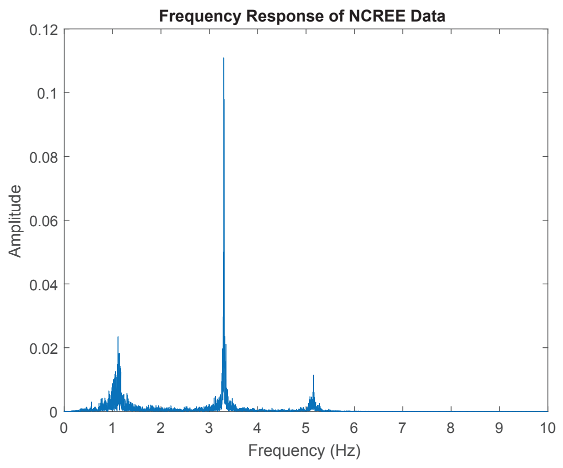
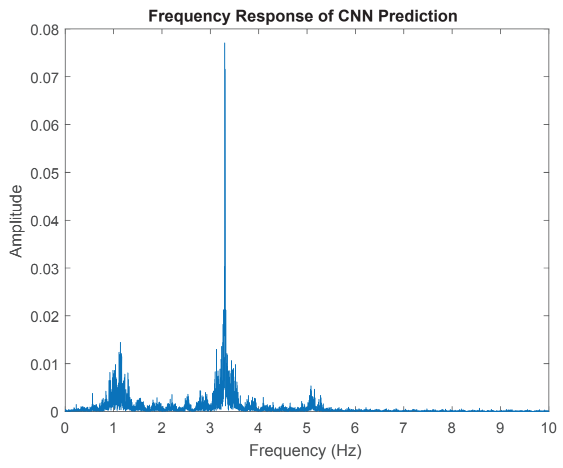


Fig. 3.32.: CNN prediction versus target.

challenging problem, and hence the MLP is often considered as a "black box" methodology. The CNN, however, has a higher interpretability over the MLP algorithm. For instance, studies in image classifications have shown that the convolution layer in CNN acts as a feature detector to extract spatial invariant features like edges and color contrast [162]. The convolution kernels become useful filters after the network is trained using large amount of data. In this section, the physical interpretation



(a)



(b)

Fig. 3.33.: Frequency response of (a) NCREE data and (b) the CNN prediction.

Table 3.14.: Identified Natural Frequencies in Transverse Direction

Mode	NCREE (Hz)	CNN (Hz)	Error (%)
1	1.113	1.140	2.43
2	3.297	3.297	0.00
3	5.513	5.081	-7.84

of the convolution layer for regression problem is provided. The authors show that the integration operator can be approximated through the convolution kernels. Also, the outputs of the convolution layer preserve the dominant signature in frequency domain. The CNN models trained using 1%, 5%, 10% and 30% noisy data with *tanh* activation function in Section "Single Degree of Freedom (SDOF) System" are selected as examples to illustrate the interpretation.

3.5.1 Approximate the Integration Operator with the Convolution Layer

In this section, it will be shown that for cases where integration is involved in the response estimation (e.g., $\dot{x}, \ddot{x}, f \Rightarrow x$), the convolution layer acts as an integration operator. Consider the example in Section "Single Degree of Freedom (SDOF) System", where the velocity, acceleration and excitation are used to estimate the displacement (i.e., $\dot{x}, \ddot{x}, f \Rightarrow x$). It is reasonable to assume that the CNN model attempts to conduct the numerical integration given the available data. The integration of a time series signal $s(t)$ over a time interval T is given as follows:

$$g(t) = \int_{t-T}^{t} s(\tau)d\tau \ , \tag{3.9}$$

where $g(t)$ stands for the integration of $s(t)$. Note that Eq. 3.9 can be written in terms of convolution operation:

$$g(t) = \int_{t-T}^{t} s(\tau)d\tau = \int_{-\infty}^{\infty} s(\tau)\mathbb{1}_{[t-T,t]}d\tau = s(t) * h(t), \tag{3.10}$$

where $\mathbb{1}_{[t-T,t]}$ is the indicator function, $h(t) = \mathbb{1}_{[t-T,t]}$, and $*$ denotes the convolution operation. Applying the Fourier transform to $g(t)$:

$$F\{g(t)\} = F\{s(t)\}F\{h(t)\} = S(\omega)H(\omega) \ , \tag{3.11}$$

where F denotes the Fourier transform, $S(\omega)$ and $H(\omega)$ are the Fourier transforms of $s(t)$ and $h(t)$, respectively. Notice that convolution in time domain equals to the multiplication in frequency domain. The theoretical displacement time series $x(t)$ can be derived from setting the signal $s(t)$ as the acceleration $\ddot{x}(t)$ and applying the convolution operation twice:

$$x(t) = \ddot{x}(t) * h(t) * h(t) \ . \tag{3.12}$$

Next, consider the first output channel in the first convolution layer shown in Fig. 3.9. The authors assume that the convolution layers in CNN attempt to estimate the ground truth displacement time history. The estimated displacement time series $\hat{x}(t)$ from CNN is expressed as:

$$\hat{x}(t) = \dot{x}(t) * w_1 + \ddot{x}(t) * w_2 + f(t) * w_3 \ , \tag{3.13}$$

where $w_1(9 \times 1)$, $w_2(9 \times 1)$ and $w_3(9 \times 1)$ are the first, second and the third column of the first kernel in the convolution layer, respectively. By using the equation of motion (i.e., Eq. 3.7) and Eq. 3.10, Eq. 3.13 can be arranged as:

$$\begin{aligned}\hat{x}(t) &= \ddot{x}(t) * (h(t) * w_1 + w_2) + f(t) * w_3 \\ &= \ddot{x}(t) * (h(t) * w_1 + w_2) + [\ddot{x}(t) * (m + ch(t)) + k\hat{x}(t)] * w_3 \\ &= \ddot{x}(t) * [h(t) * w_1 + w_2 + mw_3 + ch(t) * w_3] + k\hat{x}(t) * w_3.\end{aligned} \tag{3.14}$$

Introducing the delta function $\delta(t)$ for the left of Eq. 3.14:

$$\hat{x}(t) * [\delta(t) - kw_3] = \ddot{x}(t) * [h(t) * w_1 + w_2 + mw_3 + ch(t) * w_3] \ . \tag{3.15}$$

To see whether or not the convolution operation in our CNN model is able to approximate the integration operator, Fourier transform is applied to both Eqs. 3.12 and 3.15. The Fourier transform of Eq. 3.12 is:

$$X(\omega) = \ddot{X}(\omega)H^2(\omega) \ , \tag{3.16}$$

and the Fourier transform of Eq. 3.15 is:

$$\hat{X}(\omega) = \ddot{X}(\omega)\frac{[H(\omega)W_1(\omega) + W_2(\omega) + mW_3(\omega) + cH(\omega)W_3(\omega)]}{1 - kW_3(\omega)} \equiv \ddot{X}(\omega)H_{CNN}(\omega) \ , \tag{3.17}$$

where all the capital letters denote the corresponding functions after the transformation, ω in *rads* is the variable in the frequency domain, and $H_{CNN}(\omega)$ denotes the filter corresponding to the CNN kernel.

As a result, the authors compare $H^2(\omega)$ with $H_{CNN}(\omega)$ in the frequency domain, and see if they have similar frequency responses. The trained values of w_1, w_2 and w_3 of the CNN kernels for 1%, 5%, 10% and 30% noise levels are listed in Table 3.15.

Since the kernels are not flipped during the operation in the convolution layer, set $h(t) = \mathbb{1}_{[t-9,t]}$ and flip $h(t)$ to compute $H^2(\omega)$. Fig. 3.34 shows the frequency responses $H^2(\omega)$ and $H_{CNN}(\omega)$ computed using 256-point Fourier transform. The black solid line is the $H^2(\omega)$, and the other lines are the $H_{CNN}(\omega)$ obtained using 1%, 5%, 10% and 30% noise levels in data. The correlation coefficient R between each $H_{CNN}(\omega)$

and $H^2(\omega)$ is listed in the figure. It is demonstrated that $H_{CNN}(\omega)$ is similar to $H^2(\omega)$, which means that CNN kernel is performing operations to approximate the integration operator. The $H_{CNN}(\omega)$ for 1% noise data has the highest correlation coefficient (i.e., $R = 0.98$) with the theoretical $H^2(\omega)$. As the noise level increases, $H_{CNN}(\omega)$ gradually deviates from the analytical frequency response that leads to the decreasing of R values (i.e., $R = 0.93$, $R = 0.90$ and $R = 0.85$ for 5%, 10% and 30% noise data, respectively). This is expected since the convolution layer needs to perform filtering in addition to the integration operation to eliminate noise. The subsequent layers will further fine-tune the processed data. As shown in Fig. 3.34, the CNN kernel $H_{CNN}(\omega)$ filters out more high frequency signal than the $H^2(\omega)$. Note that the authors are discussing about the first output channel of the first convolution layer in the CNN model. The displacement $\hat{x}(t)$ is not the final estimation of the CNN model, and hence $H_{CNN}(\omega)$ is not identical to $H^2(\omega)$. The $\hat{x}(t)$ is passed through all the other layers in CNN to obtain the final estimation.

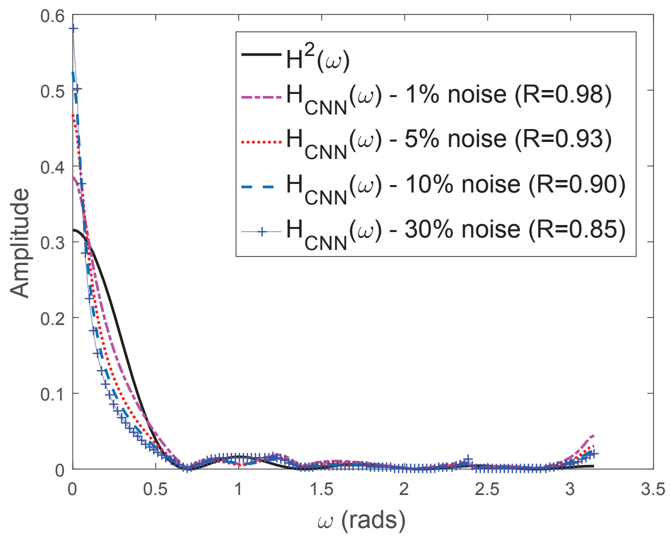


Fig. 3.34.: Frequency responses of $H^2(\omega)$ and $H_{CNN}(\omega)$ for 1%, 5%, 10% and 30% noise level in data.

Table 3.15.: Values of CNN Kernels for 1%, 5%, 10% and 30% Noisy Data

Noise Level											
1%			5%			10%			30%		
w_1	w_2	w_3	w_1	w_2	w_3	w_1	w_2	w_3	w_1	w_2	w_3
0.23	0.14	-0.36	0.21	0.15	-0.34	0.17	0.17	-0.31	0.11	0.17	-0.27
0.07	0.06	-0.01	0.06	0.07	0.01	0.03	0.08	0.02	-0.04	0.10	0.05
0.52	0.12	-0.05	0.51	0.13	-0.03	0.50	0.14	-0.02	0.43	0.10	-0.01
0.15	0.35	-0.27	0.15	0.35	-0.26	0.15	0.36	-0.25	0.15	0.33	-0.24
0.26	-0.33	0.12	0.26	-0.33	0.12	0.28	-0.34	0.10	0.35	-0.37	0.06
0.12	-0.07	0.25	0.13	-0.07	0.26	0.15	-0.08	0.25	0.21	-0.11	0.23
0.39	-0.05	0.06	0.40	-0.05	0.07	0.41	-0.06	0.07	0.42	-0.05	0.07
-0.22	0.14	-0.22	-0.21	0.14	-0.21	-0.21	0.14	-0.20	-0.17	0.15	-0.18
-0.11	-0.41	0.23	-0.10	-0.40	0.24	-0.10	-0.41	0.24	-0.04	-0.39	0.25

3.5.2 Dominant Feature Extraction

In this section, the authors show that the CNN extracts the useful information from the input and preserves the dominant characteristic of the signal throughout the whole network. Fig. 3.35 depicts the time series of the CNN estimation, the ideal target, and their corresponding Fast Fourier transform (FFT). Five seconds of the time series with a $200(Hz)$ sampling rate is shown, and hence there is a total of 1000 data points. The ideal target signal has a dominant frequency located at $2.15(Hz)$, and so does the estimation from CNN. The authors observe that this frequency signature is preserved in every output of the convolution layers. Fig. 3.36 shows the time series of input signals, the FFT of the input signals, and the FFT of the first convolution layer (i.e., layer 2) output. Fig. 3.37 shows the FFT of the second last convolution layer (i.e., layer 17) output. According to Fig. 3.36, the input signals have the same dominant frequency feature observed in the ideal target except for the third input signal (i.e., excitation). During the network training, CNN tries to extract the useful feature and eliminate the irrelevant information. As shown in Fig. 3.36 and 3.37, every output channel in layers 2 and 17 has exhibited the dominant frequency signature. From the inputs to the output, the network maintains the useful signature and eliminate the irrelevant high frequency signals to achieve the estimation.

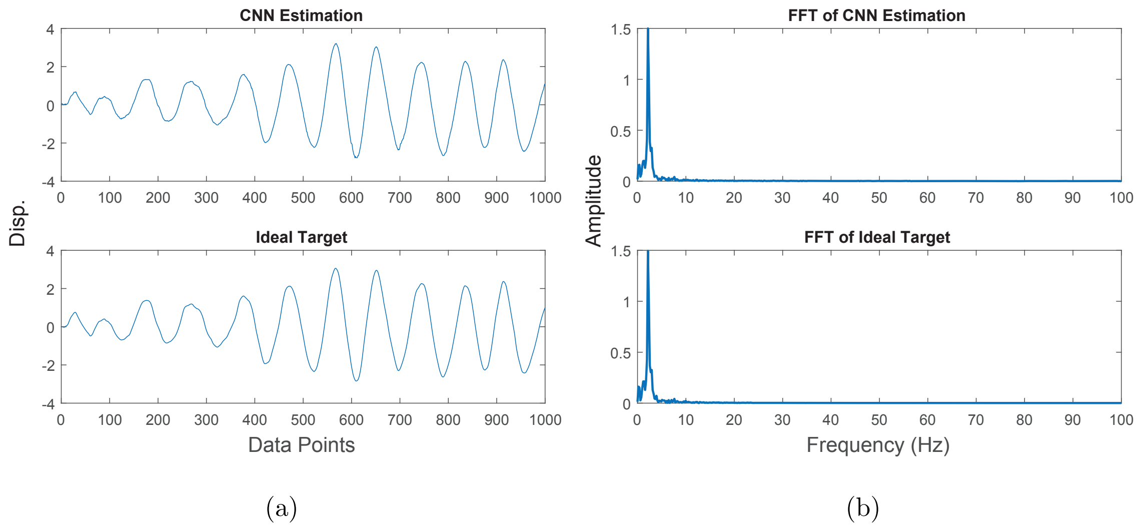
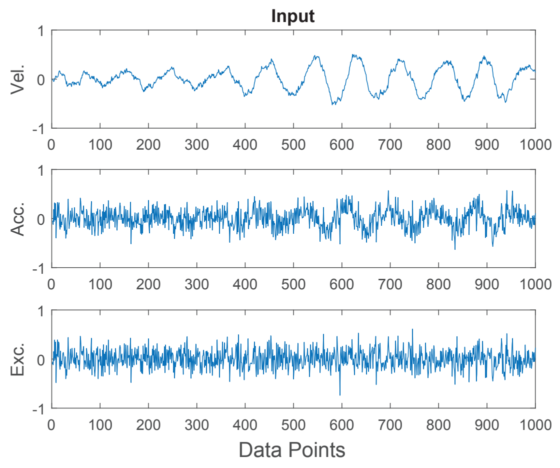


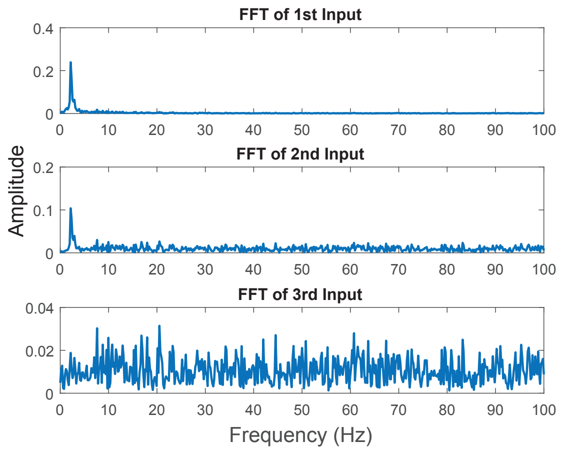
Fig. 3.35.: Time series and frequency responses: (a) time series of CNN estimation and ideal target, and (b) FFT of CNN estimation and ideal target.

3.6 Concluding Remarks

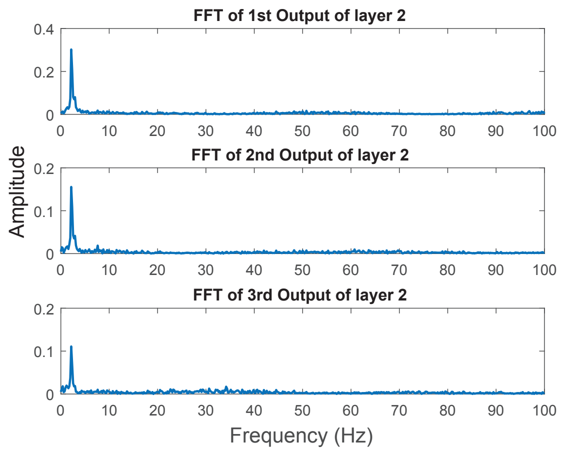
This chapter presents a deep CNN-based approach for the vibration response estimation of a linear SDOF system, a nonlinear SDOF system, and a three-story steel frame tested by NCREE in Taiwan. For the SDOF system, five noise levels (i.e., 1%, 2%, 5%, 10% and 30%) are considered to account for the measurement in real world conditions. According to the linear SDOF results of using velocity, acceleration, and excitation to predict displacement, the proposed CNN approach achieves lower RMS errors for all noisy levels compared with the MLP model that is proposed in previous studies. There is no overfitting phenomenon for both MLP and



(a)



(b)



(c)

Fig. 3.36.: Time series and frequency responses: (a) time series of input signals, (b) FFT of input, (c) FFT of layer 2 output.

CNN methods, and both methods are capable of predicting the ideal target given the noisy input and output training data. The error distribution of CNN is more centered to zero compared to MLP. The CNN predicted time history matches with the ideal target pretty well while the MLP predicted time history captures the general trend but exhibits some undesirable high frequency oscillations. As the noise level is increased, the prediction errors increase faster for MLP with respect to the CNN,

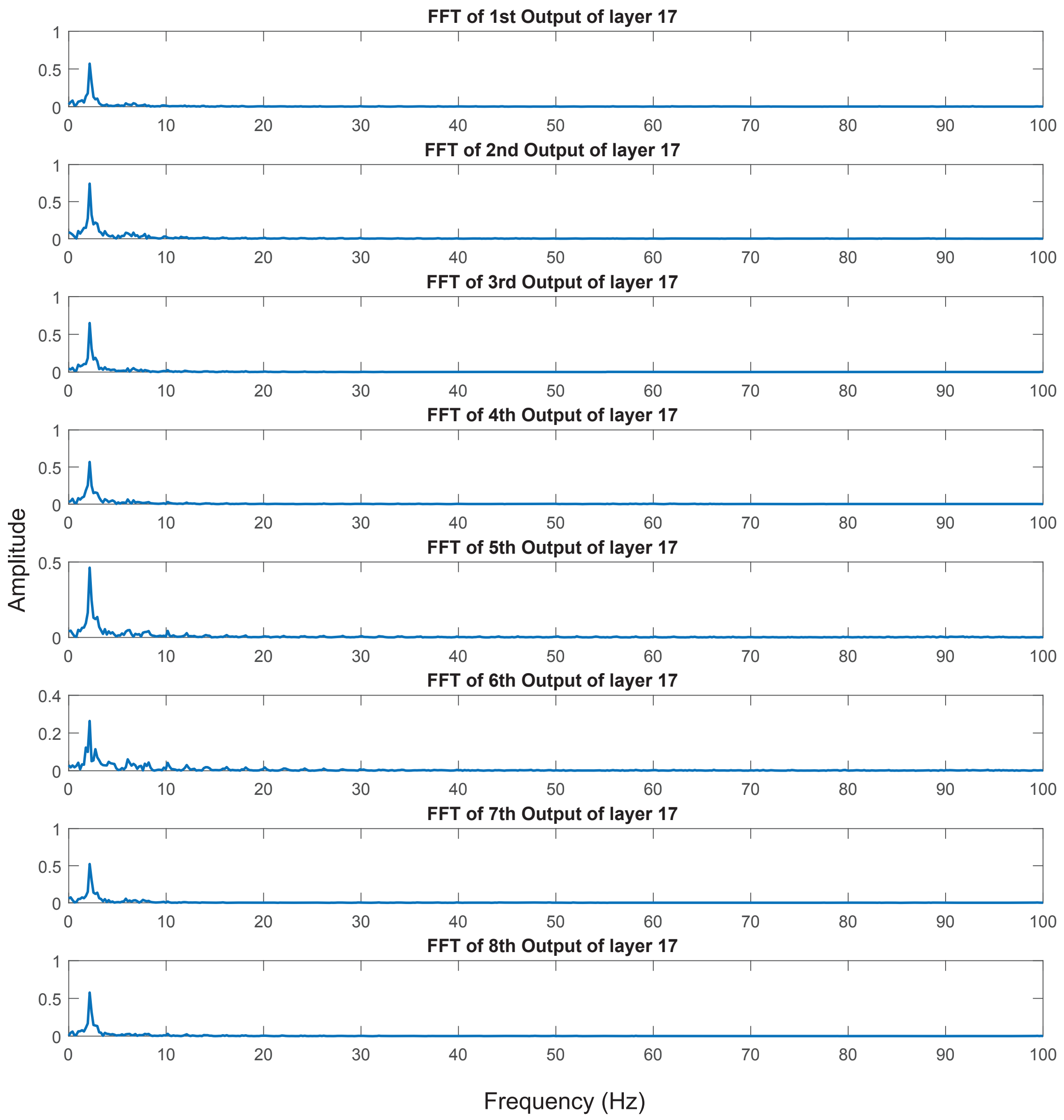


Fig. 3.37.: FFT for the output of the 17^{th} layer.

which means CNN is more robust than the conventional MLP. In the case of using excitation to predict acceleration, the CNN RMS error is slightly lower than the MLP RMS error.

For the results of nonlinear SDOF system, the CNN-based approach works better than the MLP for noise levels greater than 1%. Similarly, no overfitting is observed for both MLP and CNN methods, and the two approaches are able to learn the true behavior from the noisy training data. The error distribution of the CNN prediction is more centered to that of MLP prediction, and the predicted response time history from CNN matches better with the ideal response. The RMS error for CNN increases much slower than the error for MLP as the noise level increases. In the case of MDOF system, the proposed CNN approach is validated through the shake table test on a full-scale three-story steel frame. The ground excitation is used to estimate the roof acceleration response. From the RMS error of CNN and MLP, it is shown that the CNN prediction error is lower than the MLP by 19.38%. The error distribution of CNN prediction is more centered to zero, and the predicted time history from CNN matches slightly better with the target response. Further, the established CNN model is used to identify the natural frequencies of the steel frame. The difference between the identified frequencies from the CNN prediction and the NCREE measurements is below 8%.

In terms of the training time for model establishment, the conventional MLP is more computational efficient than the deep CNN. Table 3.16 tabulates the training times of MLP and CNN models for one noise level. The conventional MLP runs roughly ten times faster than the CNN. Although there is no significant effect observed for the selection of activation functions, RELU runs faster than *tanh* by approximately 25%. The MLP algorithm is implemented in MATLAB R2016a, and the CNN algorithm is implemented using the MatConvNet [265] library. NVIDIA's GeForce GTX 1070 GPU is employed to enhance the computation efficiency for both the MLP and the CNN. Note that as discussed previously, if the network configuration is fixed, the number of weights in MLP will increase drastically as the number of input increases, while the number of weights in CNN will remain the same. This illustrates a potential advantage of CNN over MLP when implementing in a sensor network for field application. As the number of sensor increases, the increasing in MLP training time should be more prominent and eventually exceeds the training time required by CNN. This will need further justification since there are other factors that may affect the training time, e.g., the number of multiplication operations, the network configuration, and the chosen optimization algorithm.

Table 3.16.: Model Training Time Comparison between MLP and CNN

Case	CNN-*tanh* Training time (hr)	CNN-RELU Training time (hr)	MLP-*tanh* Training time (hr)	MLP-RELU Training time (hr)
SDOF $(\dot{x}, \ddot{x}, f \Rightarrow x)$	2.0	1.6	0.2	0.1
SDOF $(f \Rightarrow \ddot{x})$	1.9	1.5	0.1	0.05
Nonlinear SDOF $(x, \dot{x} \Rightarrow g(x, \dot{x}))$	2.0	1.5	0.2	0.14
MDOF $(f \Rightarrow \ddot{x}_3)$	13.0	-	1.5	-

In Section "Physical Interpretation of Convolution Layer", the physical interpretation of the convolution layer is discussed. The authors have shown that the integration operator can be approximated by the convolution layer. This is achieved through the comparison between the frequency responses of the theoretical integration operator and the convolution kernel. The frequency response of the convolution kernel appears to be similar to the integration operator, and the convolution kernel attempts to eliminate more high frequency signals since the CNN model is trained using noise-contaminated data. Moreover, the convolution layers eliminate the irrelevant information and preserve the dominant frequency signature throughout the whole network. All the output signals from the convolution layers exhibit the dominant frequency component that appears in the ideal target signal.

In general, while both MLP and the proposed CNN approach estimate the dynamic response for the systems of interest, the proposed CNN approach is more robust against noise-contaminated data. The convolution layers act as filters, which is an advantage of using CNN. Therefore, the proposed CNN is more desirable for the response estimation of an real structure. Although the conventional MLP is more computational efficient, it remains challenging to interpret the physical meaning in the MLP model. The CNN, however, has a higher interpretability to understand the underlying mechanism during the training process, and therefore it is more favorable than the conventional MLP algorithm. As part of the future work, the authors plan to interpret the performance of convolution layers for MDOF system.

4. PRUNING DEEP CONVOLUTIONAL NEURAL NETWORK FOR EFFICIENT DECISION MAKING IN STRUCTURAL HEALTH MONITORING

4.1 Introduction

4.1.1 Background and Motivation

According to the 2017 ASCE infrastructure report [2], the estimated cost for infrastructure rehabilitation such as bridges, dams and levees, requires billions of dollars. Early detection of deficiencies in structural components and surface defects can help in reducing the retrofit costs [3–10,266–273], but current practice in structure health monitoring (SHM) still requires manual inspection which is labor-intensive and time-consuming. The development of cost-effective and autonomous SHM approaches is an urgent need in order to enhance the efficiency of inspection processes [274].

Recent advances in sensor technology and artificial intelligence provide opportunities for developing novel SHM approaches. The employment of sensor networks along with the communication capabilities among sensors have established the notion of Internet of Things (IoT) [16]. The IoT consists of a set of edge sensors and central server units. In an IoT system, decision making can take place at the edge or at a server, depending on the application's requirements [17, 18]. For civil infrastructure, the inspection target is usually enormous in size or length. A decision making scheme where all the data collected at the edge sensor are transmitted to a server and the decisions made on the server are sent back to the edge device would be extremely inefficient, particularly that the transmission bandwidth is often limited. A preferable solution, which is referred to as edge computing [17], is to deploy edge devices that have the capability to analyze data and make decisions about data acquisition without the support of a remote server. For instance, a self-driving vehicle is equipped with sensors such as GPS, the ultrasonic sensor, the camera and the Light Detection and Ranging (LiDAR) sensor. The information acquired from the sensors enables the vehicle to make real-time decision without central control unit. For the foreseeable future, with the development of swarm robotics [275] being mature, an autonomous inspection system will leverage the coordination between robots. Each individual robot, as an edge device, can make its own decision and communicate with other robots to find any potential presence of damage efficiently. In this way, the bandwidth and time spent on the data transmission are saved and more rapid inspections are achieved. However, such an approach requires the optimization in memory and computing costs of SHM algorithms deployed at the edge devices as these devices may have limited memory and computational resources.

Deep convolutional neural networks (DCNN), due to their ability to automatically extract features, have been successfully used in computer vision since the breakthrough of the 2012 ImageNet challenge [162]. The convolution kernels in DCNN capture the spatial invariant characteristics such as edges and contrast from the input image where these features are then used to make inference about the image. Since 2017, the rapid growth of DCNN-based approaches for damage detection in civil engineering has shown huge potential [50,199,260,276–285]. However, the high computation and memory demands required for DCNN make it inappropriate for deployment on mobile inspection devices, such as UAVs and robots. Furthermore, a DCNN needs a large volume of training data, which is sometimes not feasible for a newly discovered damage pattern, to avoid network overfitting. To address these issues, the concepts of transfer learning and network pruning have been introduced [23,260,284,286].

In general, there are four schemes to perform image-based damage detection using machine learning techniques. **Scheme 1**: Extract handcrafted features, e.g., local binary patterns [287] and histogram of gradients [288], from the input images. Use these features to train a classifier, e.g., support vector machine (SVM) [89]. **Scheme 2**: Design a DCNN from scratch and train it using sufficient labeled training data, e.g., the NB-CNN network by [50]. **Scheme 3**: Start from a pre-trained network, e.g., VGG16 [24], replace the fully-connected (FC) layers with new FC layers or other classifiers like SVM and k-nearest neighbor (KNN). The pre-trained network serves as an autonomous feature extractor. **Scheme 4**: Start from a pre-trained network, fine-tune the network and reduce its size by pruning technique [23]. The first scheme is able to classify the images with low memory demands and fast inference time, but it may fail to achieve good performance when the images are collected under various environmental conditions, e.g., illumination, due to the use of engineered features. Depending on the size of the designed DCNN, scheme 2 could be deployed onto computing platforms with small memory and low computing power. However, it is necessary to have enough labeled data for training. To address the problem of insufficient training data, scheme 3, referred to as *transfer learning*, can be adapted to a new classification problem with a small amount of training data by using a pre-trained network. However, pre-trained networks are usually quite large in size, and hence suffer from high memory and computation costs. The proposed scheme 4 exploits the advantages of transfer learning, and enhances the efficiency in memory cost and inference time for field applications by the Taylor expansion-based network pruning technique [23].

4.1.2 Related Work in Network Pruning

Recently, there have been some efforts in the area of network pruning. The work in [286] removed the filters if their values were below a pre-defined threshold after fine-tuning with a strong regularization term. This approach requires a long time for network fine-tuning. In [289], an approach called dynamic surgery network is proposed to evaluate the connections between neurons where the network structure was continuously maintained during the process of determining the least important connections. In [290], a three-stage pruning pipeline is proposed: pruning, K-means quantization, and Huffman coding. The first stage removed the unimportant connections between neurons, while the last two stages reduced the memory demands of the network by the quantization of the weight values and Huffman encoding for lossless data compression. The work in [291] implemented network pruning by solving a convex optimization problem, in which the sparsest set of weights for each layer were identified and then set to zero. In [292], an incremental network quantization approach was proposed to reduce the memory storage. Instead of using the full-precision (i.e., 32-bit floating-point), the proposed approach converted the weight values to be either powers of two or zero to enhance the storage efficiency. In [293], a network pruning approach is proposed based on the capability of parallelism of the computing units. The work in [294] evaluated whether a filter can be removed or not based on the outputs of its next layer rather than its own layer.

In [295], an iterative pruning approach is proposed to perform multi-tasks classification from a single network. Starting with a network trained for one task, the neurons with weights of smaller absolute magnitude were set to zeros, and these neurons were re-trained for another classification task. Following this iterative pruning and re-training process, the resulting network was able to deal with multi-tasks at the same time. In [296], the existing pruning algorithms are evaluated in terms of classification accuracy. Extensive experiments showed that for pruning algorithms that use a predefined target network architecture, similar performance can be achieved by training the target architecture from scratch. For pruning algorithms that automatically determine the target architecture by the evaluation of the global importance of the filters, the value of these algorithms lie in the searching of efficient architectures or sparsity patterns. In [297], the filters with high percentage of zero activations were removed from the network by evaluating the network on the validation dataset. Most recently, [298] proposed a reinforcement learning-based pruning algorithm to reduce the computation costs of deep neural networks. The proposed algorithm removed the filters through the rewards obtained from the predictions of the network. It is noted that some of the aforementioned approaches were relatively naive and none of the them have been focusing on efficient edge computing for filed applications related to civil infrastructure condition assessment.

In this chapter, the VGG16 [24] network is first analyzed to demonstrate the efficiency of inference before and after network pruning. Two types of surface defects, i.e., crack and corrosion, are considered to show the effectiveness of the proposed approach for damage detection. NVIDIA TITAN X GPU and Nvidia Jetson TX2 GPU development kit are selected as the computing platforms representing the server and the edge device, respectively. The ResNet18 [25] network, which has a size smaller than VGG16, is then used for comparison with VGG16 in terms of damage detection performance, memory demands and inference time. Moreover, an optimization method is proposed to further reduce the inference time of VGG16 by enhancing the efficiency in feature computation.

4.1.3 Contribution and Scope

The contribution of this work is as follows. (1) Efficient DCNNs are designed and developed by using transfer learning and network pruning. (2) Experimental results on crack and corrosion detection with different computing platforms are presented to show the efficiency of the proposed DCNNs with respect to memory cost and inference time. (3) Two pre-trained networks, i.e, VGG16 and ResNet18, are experimentally tested for damage detection to show the effect of different pre-trained networks on detection performance, memory demands and inference time. (4) An optimized feature extraction approach is proposed to further reduce the inference time of VGG16.

The rest of the chapter is organized as follows. Section 4.2 describes the image datasets and computing platforms used in this work. Section 4.3 elaborates on the concept and formulation of the network pruning technique as well as the proposed optimization method for the VGG16 feature extraction. The results and discussions about the conventional transfer learning approaches and the proposed efficient DCNN are provided in Section 4.4. Section 4.5 outlines conclusions and final remarks.

4.2 Datasets and Computing Platforms

There are two types of common surface defects investigated in this work, i.e., the crack and corrosion damage. The crack images are collected from the underwater inspection videos of internal nuclear power plant components [299]. The image pixel resolution is 720×540, and a total of $147,344$ and $149,460$ crack and non-crack image patches with size 120×120 pixels are cropped from the original images [50]. Figure 4.1 shows the crack and non-crack samples of the dataset. Due to the nature of metallic surfaces, the non-crack samples contain welds and grind marks, which makes it hard to differentiate them from the crack samples even for human inspectors. Implementing the proposed approach (i.e., Scheme 4) requires significantly less training data than a scheme that would train a DCNN from scratch (i.e., Scheme 2). To this end, only $29,468$ crack (training: $25,048$, testing: $4,420$) and $29,780$ non-crack (training: $25,313$, testing: $4,467$) image patches are used in this study. All the image patches are scaled to size 224×224 pixels for the pre-trained networks.

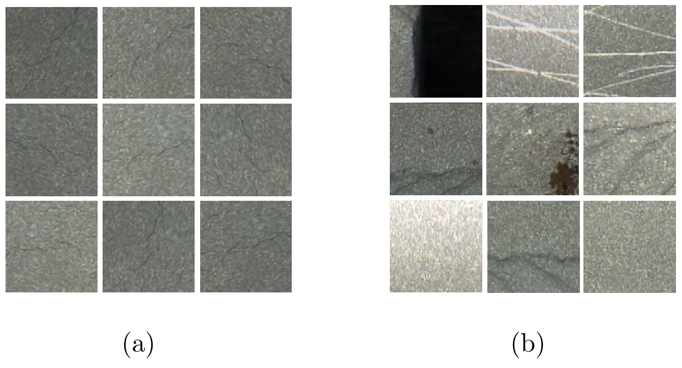
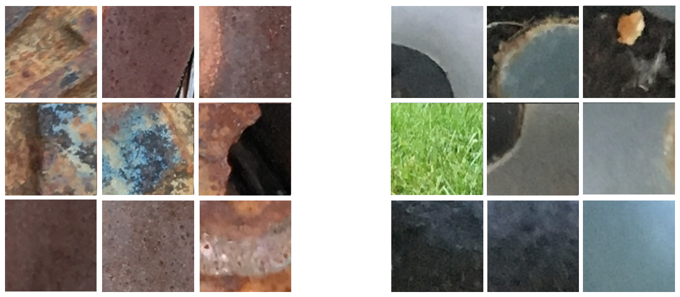


Fig. 4.1.: Crack dataset samples: (a) crack and (b) non-crack samples.

Figure 4.2 shows corrosion and non-corrosion samples from the corrosion dataset collected on metallic surfaces. The image patches with size of 128×128 pixels are cropped from a total of 926 images captured using different digital cameras [260]. This dataset contains a total of $33,039$ corrosion (training: $28,083$, testing: $4,956$) and $34,148$ non-corrosion (training: $29,026$, testing: $5,122$) image patches. Note that the underlying characteristics of the two datasets used in this study are completely different. A crack damage is usually thin in width, long in length, and darker than the background image. The crack data set belongs to metallic submerged under water. Crack detection on metallic surfaces is quite challenging since the cracks are tiny and there is less contrast between the crack and its background compared to concrete surfaces. Furthermore, the existence of scratches, welds, grind marks, and varying light condition as well as light reflection make the crack detection task more challenging. The corrosion data set consists of online images as well as images that are captured by using different cameras by the research team. A corrosion damage often covers a large area with contrast in color and texture. The use of datasets varying in nature demonstrates that the proposed approach is quite flexible.



(a) (b)

Fig. 4.2.: Corrosion dataset samples: (a) corrosion and (b) non-corrosion samples.

Two different computing setups are used for experiments. The first is a server setup with high computing power and the second is an embedded chipset to represent the reduced computation capacity of an edge device. The experimental server setup includes an Intel Xeon processor E52620, 2.1 GHz with 16 GB RAM and an NVIDIA Titan X GPU with 3584 CUDA cores at a base clock rate of 1417 MHz and 12 GB GDDR5X memory. For the embedded chipset, the NVIDIA Jetson TX2 GPU is chosen as the platform since NVIDIA has demonstrated its capability of performing integrated operation for GPU computation and navigation [300, 301]. The Jetson TX2 is a developer kit equipped with a GPU with 256 CUDA cores and a combined

8GB LPDDR4 RAM along with a CPU with dual-core NVIDIA Denver2 + quad-core ARM Cortex-A57. Note that the main concerns are the inference time performance and the accuracy, not the training time, of the DCNN models.

4.3 Methodology

Although the proposed approach is validated through the detection of crack and corrosion damage, it can be applied to any vision-based decision making problem. As usually numerous types of damage exist (e.g., pavement pothole, concrete spalling, exposed rebars), it is often very difficult to acquire sufficient data for network training at the beginning. Whenever a DCNN needs to be used to deal with a new damage detection problem from images, transfer learning is usually the best choice due to the limited available training data. However, as discussed in Section 4.1, such transfer learning-based methods are inappropriate when the decision making processes are required to be made on edge devices. The use of network pruning is introduced to enhance the resource efficiency for on-board computations, and thus allows one to deploy DCNNs that are very accurate, have low storage and computing costs, and make decisions very quickly at the edge. Network pruning can be executed on the server machine, and the pruning terminates when either the detection performance on the test dataset starts to decrease or drops below a user-defined performance tolerance.

Figure 4.3 illustrates the pruning flowchart for constructing an efficient DCNN through transfer learning and network pruning. The methodology starts with a pre-trained network (e.g., VGG16), then modifies its fully-connected (FC) layers and re-trains the network with a training dataset. As the network is originally designed for the ImageNet dataset consisting of 1000 image categories, it is very large in size and may contain redundant convolution kernels that do not contribute to the new detection problems considered herein, i.e., the detection of crack and corrosion. To reduce the network size, the importance of convolution kernels are evaluated through training data, and the kernels are removed based on their ranking in importance. Next, the pruned network is fine-tuned again to ensure its performance for damage detection. Based on the detection performance, the user can determine whether or not to further prune the network following the same procedure. Section 4.3.1 elaborates on the pruning technique that is used, and Section 4.3.2 describes the proposed optimization method for feature extraction in VGG16.

4.3.1 Network pruning

Network pruning is closely related to the biological brain behavior. At the beginning, the brain learns to handle multi-tasks, and it becomes more efficient by disconnecting some of the neurons if it is asked to focus on a specific task. The concept of network pruning can be traced back to the work done by LeCun *et al.* in 1990 [302], in which the generalization of a neural network is enhanced by the removal of the parameters selected using a second-order Taylor expansion.

To evaluate the parameters' contribution in the network, the existing techniques include, but not limited to, the investigation of the statistical values of activations, the absolute value of kernel weights, the mutual information between the activations and the network predictions [286,303,304], regularization-based techniques [305–307], and the Taylor expansion-based approaches [23]. In this work, the Taylor expansion-based algorithm is adopted for the following two reasons [23]: (1) Compared to the regularization-based approaches, the Taylor expansion-based approach uses the global rescaling of criteria and therefore no computation on per layer sensitivity analysis is needed, and (2) The Taylor expansion-based algorithm is demonstrated to be more robust against other pruning methods in the sense that the algorithm keeps maintaining higher classification accuracy as the pruning proceeds.

Intuitively, the pruning algorithm aims to find the set of kernel weights W' that leads to the minimum changes in the cost function $C(\cdot)$:

$$\min|\Delta C(W)| = \min|C(D|W') - C(D|W)| \text{ , s.t. } ||W'|| \leq B. \tag{4.1}$$

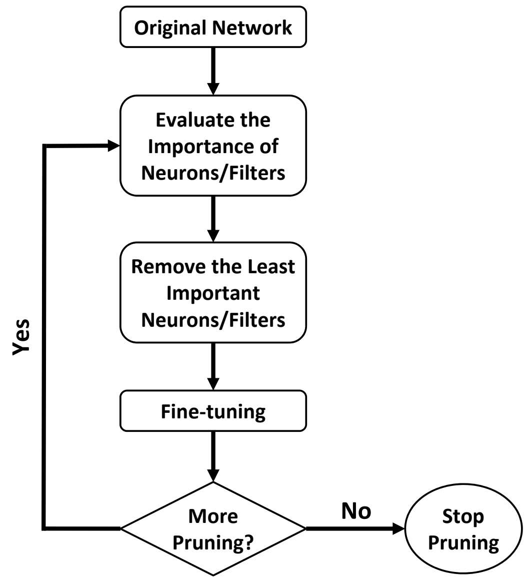

Original Network
Evaluate the Importance of Neurons/Filters
Remove the Least Important Neurons/Filters
Fine-tuning
More Pruning?
Yes
No
Stop Pruning

Fig. 4.3.: Network pruning flowchart.

where D denotes the training data, and W denotes the set of the kernel weights obtained after network training is finished. B is the number of non-zero elements in W'. Notice that it requires $2^{|W|}$ evaluations on the kernel weights to solve Eq. 4.1 if considering all the combinations among the elements in W. To reduce the computational costs for solving W', a greedy method is employed that iteratively searches among all the subsets of W' to be removed. Thus, such method removes subsets one at a time, as shown in Figure 4.3.

For DCNNs, the convolution feature maps can be considered as the elements of W'. Let h_i denote the i^{th} feature map in a DCNN, $i = 1, ..., L$, where L is the total number of feature maps in the DCNN. By assuming independence of parameters, the cost function $C(D|W)$ can be written as:

$$C(D|W) \approx C(D|h_i) = C(D, h_i) \ . \tag{4.2}$$

Let $C(D, h_i = 0)$ denotes the cost function $C(\cdot)$ where the feature map h_i is being removed. Then, the difference in cost function is:

$$|\Delta C(h_i)| = |C(D, h_i = 0) - C(D, h_i)| \ . \tag{4.3}$$

Using Taylor expansion, the term $C(D, h_i = 0)$ is approximated as:

$$C(D, h_i = 0) = C(D, h_i) - \frac{\partial C}{\partial h_i} h_i + R_1(h_i = 0) \; , \tag{4.4}$$

where $R_1(h_i = 0)$ is the first-order remainder, i.e., the higher-order terms. By ignoring $R_1(h_i = 0)$ and substituting Eq. 4.4 into Eq. 4.3, the change in the cost function is estimated as:

$$|\Delta C(h_i)| = \left| \frac{\partial C}{\partial h_i} h_i \right| \; . \tag{4.5}$$

Conceptually, the pruning algorithm removes the feature map h_i that leads to a near-zero gradient of the cost function. The difference in cost function can be computed through the standard back-propagation approach [308]. The pruning algorithm is implemented in Python 2 using PyTorch [309] version 0.2 with CUDA 8.0, cuDNN 6.0.21, and Ubuntu 16.04.

Pruning VGG16 Network

VGG16 has been widely adopted for SHM applications in recent two years [260, 310–312]. As the winner of the 2014 ImageNet challenge, VGG16 is the first network that introduces the concept of deep neural networks. It has a total of $120,000,000$ parameters, and the network architecture consists of simple concatenation of convolution and pooling layers [24].

To use VGG16 for crack detection, the last FC layer in VGG16 is replaced with a binary output layer to indicate the presence of cracks in the input image patch. The stochastic gradient decent (SGD) algorithm [308] with learning rate 0.001 and momentum 0.9 is used for network fine-tuning. The parameters of the whole original network are first fine-tuned with the crack training dataset, and the convolutional feature maps are removed based on the pruning steps illustrated in Figure 4.3. At each pruning iteration, the contributions of the convolutional kernels are ranked using training data, and 512 kernels are removed from a total of $4,224$ convolution kernels in VGG16. Next, the whole pruned network is again fine-tuned using the same learning rate with the training data to enhance its detection performance, and the user decides whether to further prune the network or not. In this work, the number of fine-tuning epochs for the original network is 20, and the number of fine-tuning epochs for the pruned network is 10. Seven pruning iterations are conducted for VGG16 to remove 84% of the convolution kernels. Following the same procedure, another DCNN is constructed for corrosion detection using the corrosion dataset.

Pruning ResNet18 Network

As the winner of the 2015 ImageNet challenge, ResNet18 introduces the concept of residual learning to make easier inference on the mapping between the input and output [25]. The network consists of the concatenations of residual learning blocks, as shown in Figure 4.4, to enhance network training through identity mapping. The total number of parameters in ResNet18 is $9,000,000$, which is approximately 13 times less than VGG16.

To adopt ResNet18 for crack detection, the FC layer is replaced with a binary output layer. Using the same network fine-tuning and pruning procedure as VGG16, six pruning iterations are conducted for ResNet18 to remove 79% of the convolution kernels from a total of $3,904$ kernels.

It is noted that the residual learning block requires the input and output dimensions to be identical, as shown in Figure 4.4. During the training phase, the output $F(x)$ of a residual learning block consists of weight layers and nonlinear activations is added to the input x. The resulting $F(x) + x$ is then passed to the subsequent layers for further processing. Since the convolution kernels are removed based on ranking of importance, the dimension compatibility of the input and output of the residual learning blocks is not guaranteed. For instance, given an input x of dimension $224 \times 224 \times 32$, the output $F(x)$ after two convolution layers with kernel sizes of $3 \times 3 \times 32 \times 32$ is still $224 \times 224 \times 32$ by zero-padding. After pruning, the two convolution layers could have kernels with different sizes, e.g., $3 \times 3 \times 32 \times 32$ and

$3 \times 3 \times 32 \times 16$, resulting in an output dimension of $224 \times 224 \times 16$ for $F(x)$. Therefore, the residual learning blocks are disabled during the fine-tuning of the pruned networks.

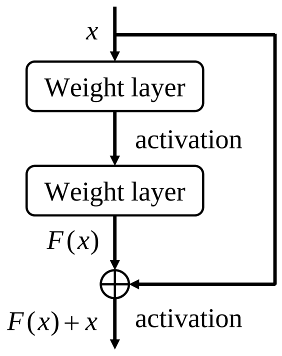


Fig. 4.4.: An illustration of a sample ResNet18 building block.

4.3.2 An optimization for feature extraction in VGG16

To detect and locate the presence of damage given an input image frame, one common approach is to first scan the whole frame by sliding windows of the same size of the patch size used in network training. Next, each window is passed to the network for damage identification, and the damage in the input image is localized by grouping the overlapping/neighboring sliding windows identified as damage patches [50]. The step size between adjacent windows is user-defined. For instance, a step size of 8 pixels between each sliding window is used by [50]. A smaller step size will enhance the spatial resolution of the damage detection result while the time required to process the whole image will increase due to higher number of image patches to be processed. Notice that the overlapping regions between adjacent sliding windows will be large if the chosen step size is small, which means that the two adjacent windows share a large number of features, as shown in Figure 4.5(a). The time required to compute the features can be reduced by eliminating the redundant computations in the overlapping regions.

To reduce the computation in processing the whole image, an optimization approach inspired by SPPNet [313] is proposed to compute the VGG16 features. Figure 4.5(b) illustrates the proposed approach. Given an input image, the image is passed to the network for processing up to the layer before the FC layers. The resulting convolutional feature map is then used to extract the features of the sliding windows in the original image according to their corresponding locations in the map. For instance, a step size of 8 pixels between the sliding windows in the original image corresponds to a step size of 1 pixel in the convolutional feature map after three executions of pooling operations. Therefore, repetitive convolution operations are saved and the inference time for the whole frame greatly decreases. This can further improve the efficiency of DCNNs for deployment onto edge devices. Note that the proposed optimization approach can be applied to any network that consists of concatenations of convolution and pooling layers, e.g., AlexNet [162]. For networks with more complicated configuration, other optimization schemes may be required to eliminate the repetitive computation.

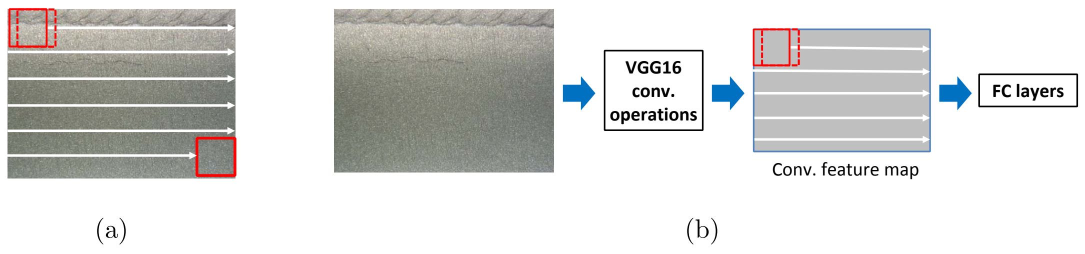


Fig. 4.5.: An optimization of feature extraction for VGG16: (a) pass each sliding window into VGG16 separately, and (b) pass the whole image frame into VGG16 to generate a convolutional feature map, then extract the feature of each window from the corresponding location on the feature map.

4.4 Results and Discussions

This section evaluates the damage detection performance, the inference time, and the memory requirement of the conventional transfer learning-based approaches (i.e., Scheme 3 in Section 4.1) compared to the proposed network pruning (i.e., Scheme 4 in Section 4.1) approach. The results on the crack and corrosion datasets are presented. In the experimental evaluation, the server setup and the edge device are used as computing setups with different computational capacities. During the training phase, only the server setup is used since all the heavy computations can be executed at a cloud server in practice. In the network testing phase, both the server setup and the edge device are used to compare the inference time and memory demands. The recorded inference time is the time required to classify $3,720$ image patches with size 224×224, assuming that one 720×540 image has $3,720$ sliding windows. Instead of using only $3,720$ test image patches, the detection performance is based on the detection accuracy for all the image patches in the test datasets (i.e., $8,887$ patches for crack and $10,078$ patches for corrosion) to avoid any biased judgement.

4.4.1 Conventional transfer learning without network pruning

Conventional transfer learning approaches use a pre-trained network as a feature extractor, and train a new classifier based on these input features and the corresponding image labels. In this section, VGG16 [24] is adopted to extract $1 \times 4,096$ features given an input image patch of 224×224. The last two FC layers of VGG16 are replaced with a SVM or a KNN classifier. Three types of classifiers are considered, i.e., KNN, the C-Support vector classifier (SVC), and the linear support vector classifier (SVMH). The Scikit-Learn library [314] is used, where the SVC classifier is implemented using libsvm [315] and SVMH classifier using liblinear [316]. The feature generation is implemented using Tensorflow library [317] with Keras wrapper [318].

Table 4.1 presents the experimental results for image-based damage assessment based on Scheme 3. The detection accuracy of the KNN classifier is slightly better than the SVC and the SVMH classifiers for the crack dataset while the SVC classifier achieves better accuracy for the corrosion dataset. Among the three classifiers, the SVMH classifier has the least memory demand (i.e., 0.03 (MB)) and the smallest inference time (i.e., 234.8 (sec) and 245.7 (sec) for crack and corrosion data, respectively) when deployed on the edge device. This means that it takes approximately 4 minutes to process one 720×540 image on the edge device which is inefficient for practical applications. For both crack and corrosion datasets, even with the server setup, it takes roughly 29 seconds to process one image for the SVMH classifier. Note that 99% of the inference time is spent on the feature extraction due to the heavy architecture of the network, which demonstrates the necessity of network pruning for efficient inference.

The feature extraction part is implemented using the Tensorflow python library [317], which utilizes the parallel processing capabilities of a GPU when available. The classifiers on the other hand use the Scikit Learn library [319] which is a CPU-only implementation. Due to the powerful TITAN X GPU in the server setup, the feature extraction part is eight times faster on the server when compared to the edge device. However, the classifier part does not get similar speed-up on the server since it utilizes only the CPU.

4.4.2 Transfer learning with network pruning

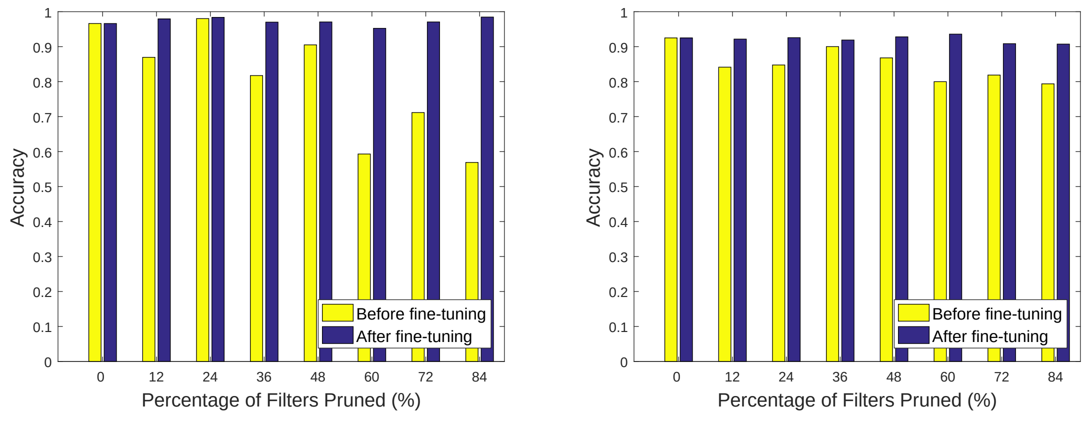
Pruning VGG16

To achieve efficient inference on the edge devices, the Taylor expansion-based network pruning technique described in Section 4.3.1 is used to reduce the size of the pre-trained VGG16 network. Pruning is conducted on the server as it requires computations for ranking the convolution kernels. The pruned networks are then deployed to both the server and the edge device to test the inference time and the memory demands. Whenever the pruning algorithm removes 512 convolution kernels in each pruning iteration, the network is fine-tuned with 10 epochs. Figure 4.6 shows the

Table 4.1.: Scheme 3 damage detection results: VGG16 used as feature extractor and KNN, SVC and SVMH as classifier.

Classifier	Data	Model size (MB)	Server setup inference time (Sec)	Edge device inference time (Sec)	Accuracy (%)
KNN	crack	3355	96.1	587.6	94.6
SVC	crack	169	124.6	417.7	89.3
SVMH	crack	0.03	29.5	234.8	85.5
KNN	corrosion	3787	89.0	548.4	82.8
SVC	corrosion	191	131.7	492.5	89.8
SVMH	corrosion	0.03	29.3	245.7	84.0

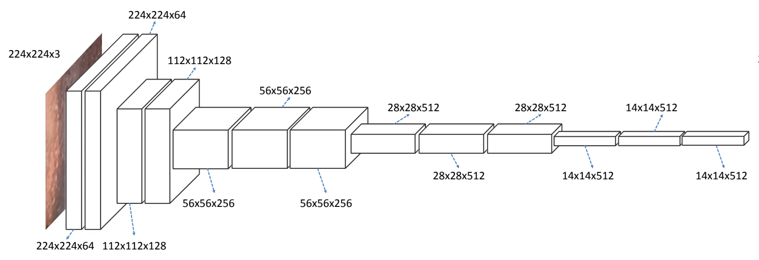
detection performance before and after fine-tuning for both the crack and corrosion datasets. In general, the detection performance drops immediately when the number of convolution kernels is reduced, and then the accuracy is improved by fine-tuning with the training data. For the crack dataset, the accuracy before fine-tuning ranges from 56.9% to 98.0% while the accuracy after fine-tuning lies in the range between 95.3% and 98.5%, for the pruned networks. It is noted that the detection accuracy after fine-tuning does not exhibit a monotonic decreasing behavior. This demonstrates that after pruning, the network still has the capacity to detect damage adequately since the original network is designed for ImageNet challenge to classify 1,000 categories. Also, a small oscillation is observed in the accuracy after fine-tuning because the number of fine-tuning epochs is fixed, and the experiment is conducted under the assumption that there is no sufficient training data. With enough fine-tuning and training data, the oscillation in detection accuracy should be reduced. For corrosion dataset, the accuracy before fine-tuning ranges from 79.4% to 90.0% while the accuracy after fine-tuning lies in the range between 90.7% to 93.6%, for the pruned networks. This indicates that network fine-tuning is crucial to ensure good detection performance after pruning convolution kernels.

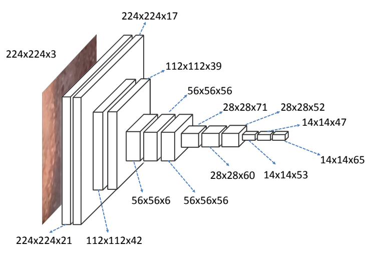


(a) (b)

Fig. 4.6.: Detection performance of pruned VGG16 versus percentage of pruned filters: (a) accuracy of the $8,887$ test image patches from the crack dataset, and (b) accuracy of the $10,078$ test image patches from the corrosion dataset. (Refer to Section 4.4.2 for more discussion about the stopping criterion for pruning and the effects of different fine-tuning datasets.)

Figure 4.7 depicts the original VGG16 convolution feature maps versus the pruned network where 84% of convolution kernels are removed for crack detection. The numbers of the removed kernels in each convolution layer are: {1: 43; 2: 47; 3: 86; 4: 89; 5: 195; 6: 200; 7: 200; 8: 441; 9: 452; 10: 460; 11: 459; 12: 465; 13: 447} , where i: j denotes that j convolution kernels are removed in the i^{th} convolution layer. As shown in Figure 4.7, after pruning the network size is significantly reduced, and the memory demands of VGG16 drops from 525 (MB) to 125 (MB), which is approximately an 80% reduction in memory. Figure 4.8 presents the distribution of the percentage of the pruned kernels in each convolution layer for both the crack and corrosion data using VGG16 network. According to the figure, the following observations are made: (1) The distribution for the corrosion data is close to the crack data, (2) A similar number of pruned kernels are deleted in the convolution layers between pooling layers, and (3) Network pruning removes more percentage of kernels in the deeper layer. This demonstrates that when using transfer learning for a new task very different from the original task, lower-level feature representations extracted from the first few layers could be more useful than the higher-level features in the deeper layer.





(a) (b)

Fig. 4.7.: VGG16 convolution feature map dimensions: (a) original dimension of feature maps, and (b) reduced feature map dimension after pruning 84% of the convolution kernels.

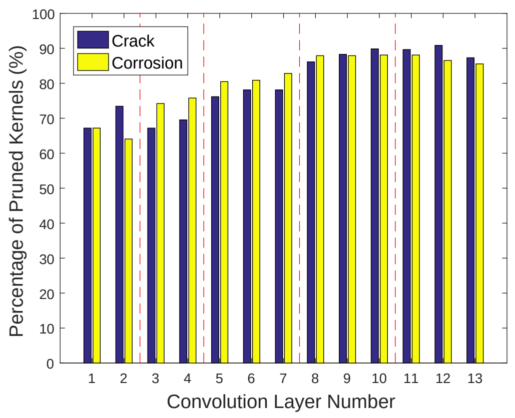
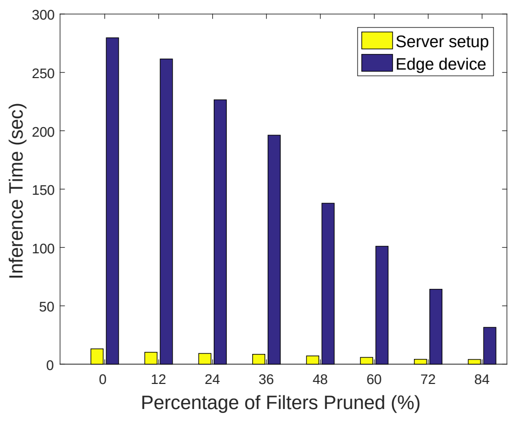


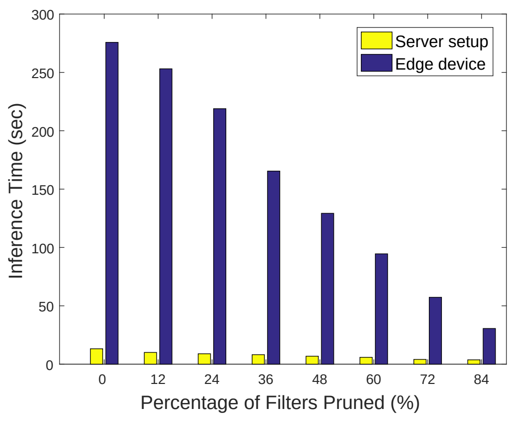
Fig. 4.8.: The distribution of the percentage of the pruned kernels in each convolution layer for the crack and corrosion datasets after 84% kernels being eliminated in the VGG16 network. The dashed lines indicate the locations of the pooling layers.

Figure 4.9 shows the variation of the inference time when using pruned networks for damage detection. For crack dataset, the inference time on edge device decreases from 279.7 (sec) to 31.6 (sec) that corresponds to a reduction factor of 8.9. For the corrosion dataset, the inference time on edge device drops from 275.7 (sec) to 30.6 (sec) that corresponds to a reduction factor of 9.0. This demonstrates how network pruning enhances the time efficiency on the edge. For the server setup, the inference time of crack and corrosion datasets decreases from 13.1 (sec) to 4.0 (sec) and 13.2 (sec) to 3.7 (sec), respectively. The corresponding reduction factors are 3.3 and 3.5, which is approximately 0.38 times of the reduction factors on the edge device. The main reason for this difference is the $cudaMalloc$ function which allocates the memory on the GPU. The time taken by this function does not vary significantly when the size of the model changes. For our implementation, $cudaMalloc$ takes 2.3 (sec) on the server setup, i.e., approximately 60% of the total inference time in case of the pruned model, which makes the speedup less considerable. In case of the edge device, however, $cudaMalloc$ takes 9.9 (sec) which is only 33% of the inference time for a pruned network.

In addition, a CNN network (i.e., Scheme 2 in Section 4.1), which is a specialized network developed for crack detection by [50], is deployed on the server setup for comparison with the pruned VGG16 network. The specialized CNN takes 32.0 (sec) to complete the inference, which is quite close to the time 31.6 (sec) achieved by the VGG16 network with 84% kernels pruned. It is noted that the NB-CNN network uses an smaller input patch size of 120×120 pixels, meaning that the pruned VGG16 would be more efficient in inference time than the specialized CNN if using the same input size. This shows that network pruning provides a resource-efficient solution when using a large network with several layers and parameters at the beginning. However, the specialized CNN network is superior to the pruned VGG16 network in terms of memory demands and damage detection performance. The specialized CNN requires only 11.1 (MB) memory, and its detection accuracy is 99.5% due to the use of larger training data (i.e., $240,000$ image patches).



(a)

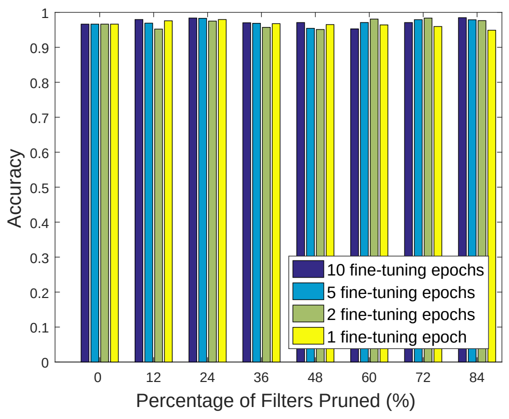


(b)

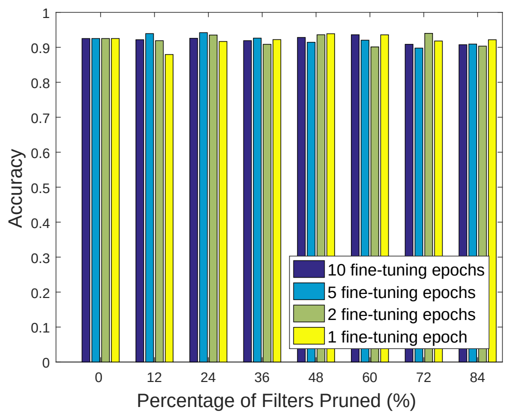
Fig. 4.9.: Inference time of VGG16 versus percentage of pruned filters: (a) crack dataset, and (b) corrosion dataset. Inference time: the total time (sec) required for the forward-pass of 3720 image patches of 224×224 pixels.

It is shown that the pruned networks need fine-tuning with the training data to improve the detection performance. The more the fine-tuning is performed on a network, the longer it takes to complete the pruning on the server machine. Therefore, a sensitivity analysis is performed on the required number of fine-tuning epochs for both the crack and the corrosion datasets. Figure 4.10 shows the detection performance when fine-tuning the network with $1, 2, 5$, and 10 epochs. As shown in the figure, there is no significant difference in the detection accuracy when using different numbers of fine-tuning epochs. For the crack dataset, the detection accuracy of the network with 84% kernels pruned is 94.9% and 98.5% when using 1 and 10 fine-tuning epochs, respectively. Similarly, for the corrosion dataset, the detection performance of the pruned network with 84% kernel removal is 92.2% and 90.7% when using 1 and 10 fine-tuning epochs, respectively. To eliminate the effects of random initialization, five repeat trials are conducted for each case of fine-tuning epochs when pruning 84% of the convolution kernels. The boxplots shown in Figure 4.11 are used to show the variation of the F-Score in damage detection.

According to Figure 4.11, the variation in F-score is the smallest for both the crack and corrosion datasets when using 5 fine-tuning epochs. The median F-Score for crack dataset is 0.963 and 0.981 for 1 and 10 fine-tuning epochs, respectively. The median F-Score for corrosion dataset is 0.905 and 0.931 for 1 and 10 fine-tuning epochs, respectively. These results show that using 10 epochs to fine-tune the network may achieve slightly better performance than using 1 epoch. Figure 4.12 illustrates the variation of time spent on network pruning. For the crack dataset, it takes a total of 1.55 (hr) and 7.16 (hr) to prune 84% of kernels using 1 and 10 fine-tuning epochs, respectively. For the corrosion dataset, it takes 1.65 (hr) and 7.75 (hr) to prune 84% of kernels with 1 and 10 fine-tune epochs, respectively. This demonstrates that using a smaller number of fine-tuning epochs reduces the time required for pruning on the server machine, and doing this does not lead to a significant decrease in the detection performance.

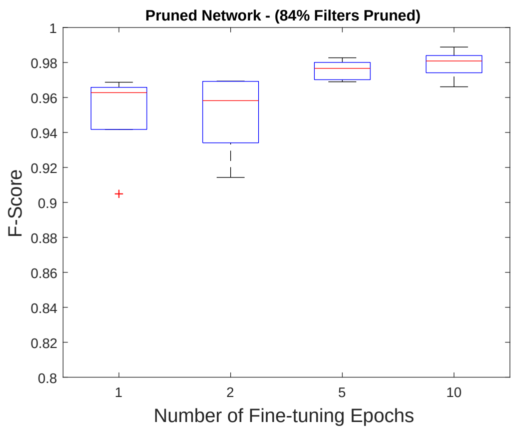


(a)

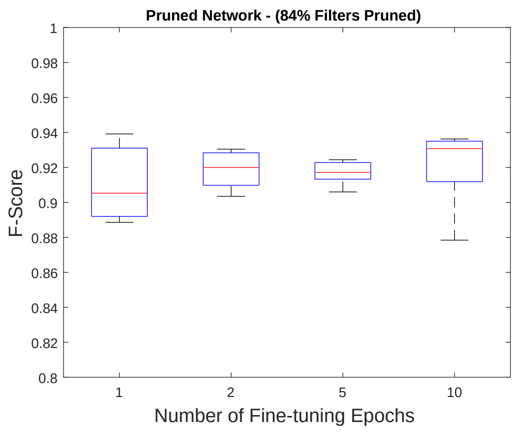


(b)

Fig. 4.10.: Effect of fine-tuning epochs on detection performance for VGG16 network: (a) accuracy of the crack test dataset, and (b) accuracy of the corrosion test dataset.

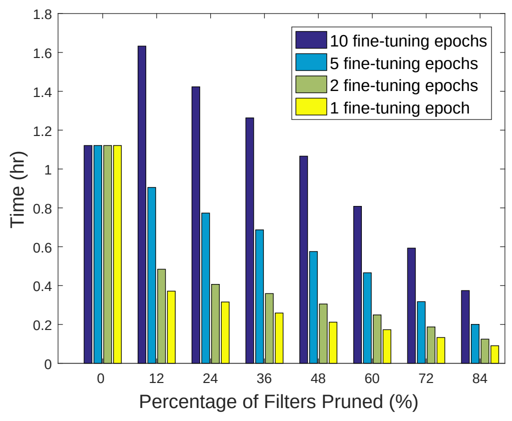


(a)

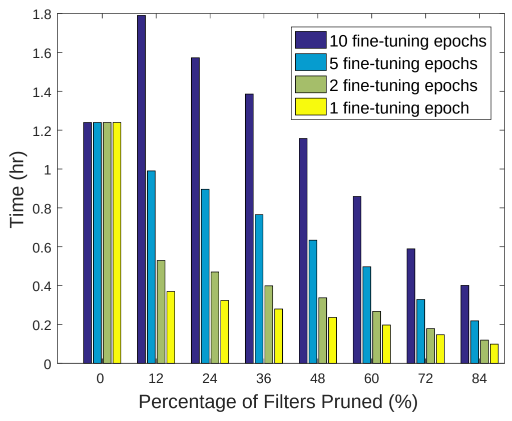


(b)

Fig. 4.11.: Repeated trials for damage detection with 1, 2, 5 and 10 fine-tuning epochs for pruned VGG16 network: (a) F-score of the detection results for the crack test data, and (b) F-score of the detection results for the corrosion test data.



(a)



(b)

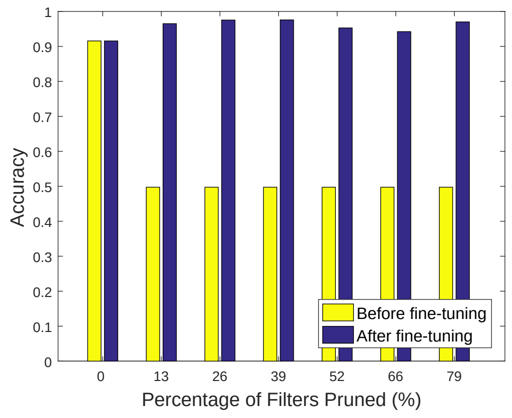
Fig. 4.12.: Effect of fine-tuning epochs on pruning time of VGG16: (a) crack dataset, and (b) corrosion dataset.

Pruning ResNet18

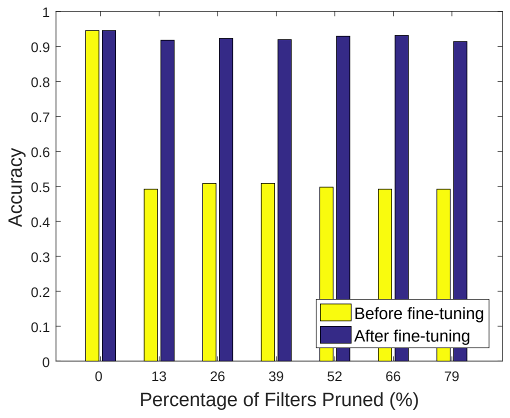
Besides VGG16, the pre-trained ResNet18 network is also adopted for detecting surface defects. The original ResNet18 network is 44 (MB), which is 8.4% of the original VGG16 network size (525 (MB)) and therefore should be more efficient with respect to inference time and memory costs. Figure 4.13 shows the detection performance of ResNet18 for the crack and corrosion datasets. Similar to VGG16, the detection accuracy decreases every time the algorithm removes 512 convolution kernels. For the crack dataset, the accuracy before fine-tuning is 49.7% while for the corrosion dataset, the accuracy before fine-tuning ranges from 49.2% to 50.8%, for the pruned networks. After fine-tuning on the pruned networks, the accuracy improves and lies between 94.2% and 97.6% for the crack dataset. For the corrosion dataset, the detection accuracy increases to values between 91.8% and 93.1% after network fine-tuning. This again demonstrates that it is essential to fine-tune the network after network pruning. Figure 4.14 compares the inference time of VGG16 and ResNet18 when deployed on the edge device for damage detection. By removing 84% and 79% of the convolution kernels from VGG16 and ResNet18, the inference time for crack detection decreases from 279.7 (sec) to 31.6 (sec) for VGG16 and 36.8 (sec) to 8.9 (sec) for ResNet18. For the corrosion dataset, the inference time decreases from 275.7 (sec) to 30.6 (sec) and from 34.1 (sec) to 9.0 (sec) for VGG16 and ResNet18, respectively. As expected, ResNet18 is more efficient with respect to the inference time (approximately 3.55 times faster than VGG16 at the end of network pruning). The ResNet18 size decreases from 44 (MB) to 2 (MB), which is a 95% reduction compared to the original network size. Notice that VGG16 achieves a 89% reduction in inference time and 80% reduction in memory while ResNet18 achieves a 76% reduction in inference time and 95% reduction in memory demands. This means that through network pruning, the gain for VGG16 is more prominent with respect to the inference time while the gain for ResNet18 is more prominent with respect to memory requirement.

Although ResNet18 is more efficient than VGG16 in terms of inference time and memory costs, VGG16 has higher adaptability for new classification problems due to being a larger network. As shown in Figures 4.6 and 4.13, the detection performance of VGG16 drops approximately from 10% to 20% each time the network is pruned.

For instance, in Figure 4.6(a), the accuracy is 98.4% after fine-tuning when 24% of filters are removed, and the accuracy is 81.7% before fine-tuning when 36% of filters are removed. There is a 16.7% drop in accuracy because of pruning. Once the fine-tuning is applied, the accuracy reaches 97% although 36% of the filters are removed. However, the accuracy of ResNet18 drops significantly, i.e., more than 30%, in each pruning iteration. Such result indicates that ResNet18 is more sensitive to changes in the network.



(a)

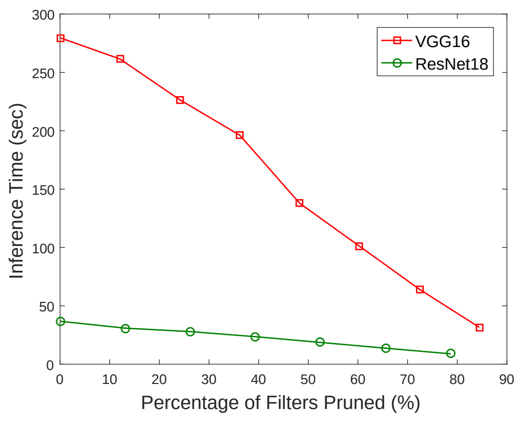


(b)

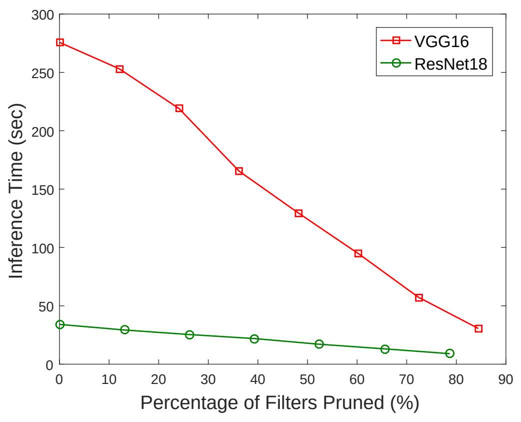
Fig. 4.13.: Detection performance of pruned ResNet18 versus percentage of pruned filters: (a) accuracy of $8,887$ test image patches from the crack dataset, and (b) accuracy of $10,078$ test image patches from the corrosion dataset.

Cross-validation and stopping criterion for pruning

As discussed in Section 4.3, network pruning may terminate when the detection performance on test datasets starts to decrease. In this section, pruning is conducted on VGG16 for both crack and corrosion datasets, and it is stopped if the detection accuracy after fine-tuning drops more than 3%. Five-fold cross-validation with the whole dataset is used to show the statistics of the detection performance and therefore



(a)



(b)

Fig. 4.14.: Inference time required for 3,720 image patches operated on Jetson TX2 GPU for VGG16 and ResNet18 networks: (a) crack dataset, and (b) corrosion dataset.

eliminates the bias induced by fine-tuning and testing on a particular training/testing dataset. In other words, five repeated trials are conducted for both crack and corrosion datasets. Each trial uses different training data, and the test datasets in the five trials are completely independent. Figure 4.15 reports the mean of the damage detection accuracy for both crack and corrosion data from five-fold cross-validation. Tables 4.2 and 4.3 report the mean (μ) and standard deviation (σ) of the accuracy before and after fine-tuning for crack and corrosion datasets, respectively. Up to 84% filters being pruned, the mean detection accuracy after fine-tuning for crack and corrosion is approximately 99% with a standard deviation below 0.5%. This demonstrates the robustness of the proposed approach as the variations in the performance are extremely small when the pruned network still has the capacity to deal with the detection task. When 97% of the filters are removed (i.e., only 128 filters left in VGG16), the mean accuracy of crack detection drops to 84.7% with a standard deviation 19.86%, and the mean accuracy of corrosion detection drops to 96.0% with a standard deviation 0.95%. This indicates that the pruning should terminate due to the increasing variation and decreasing accuracy in the detection performance. For the accuracy before fine-tuning, the mean exhibits a monotonic decreasing behavior as the pruning proceeds. The standard deviation before fine-tuning is quite high, which is due to the quality of random optimization for the remaining parameters in the network. For instance, if the values of the remaining filters are not fine-tuned very well, the accuracy before fine-tuning may drop more significantly, compared to the situation in which the remaining parameters are fine-tuned better. This explains why the accuracy before fine-tuning shown in Figure 4.6 exhibits an oscillation behavior. Notice that the variation in the accuracy before fine-tuning is very small when 97% of filters are removed, since the mean accuracy is 50%, meaning that the network performance is equal to random guesses. Lastly, while conventional transfer learning usually fine-tunes only the FC layers when there is no sufficient training data, the authors observe that fine-tuning the whole network leads to better detection performance for new tasks very different from the original task. It is noted that cross-validation is not necessary for deep learning if there exists sufficient available data. However, as transfer learning-based approaches usually use a small amount of data, cross-validation may enhance the reliability in the capability of generalization.

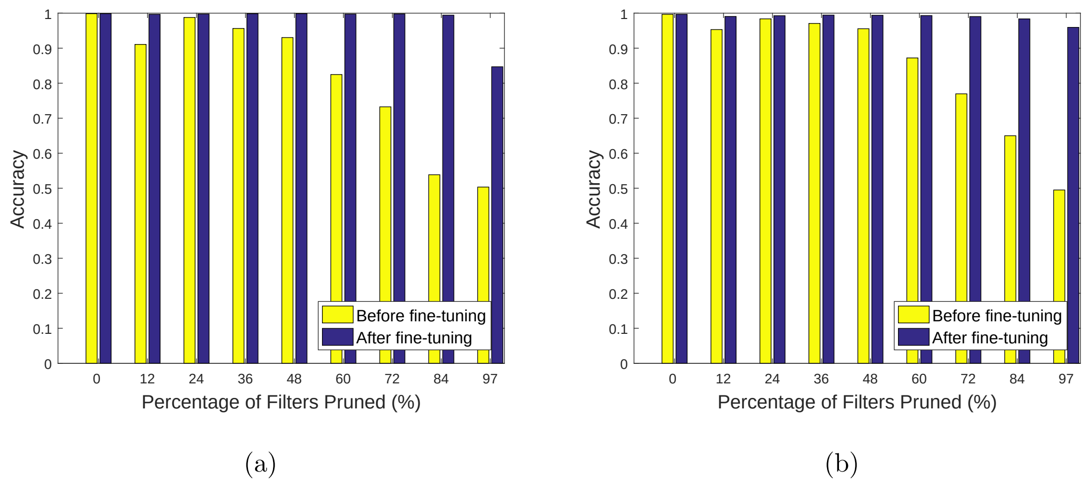


Fig. 4.15.: Mean detection accuracy from 5-fold cross-validation of pruned VGG16 versus percentage of pruned filters: (a) crack dataset, and (b) corrosion dataset.

Table 4.2.: Mean (μ) and standard deviation (σ) of detection accuracy before and after network fine-tuning - VGG16 with crack datasets

Pruned	μ (%)		σ (%)	
filters (%)	before	after	before	after
0	99.9	99.9	0.11	0.11
12	91.1	99.7	2.43	0.12
24	98.8	99.7	1.33	0.11
36	95.6	99.8	4.05	0.06
48	93.0	99.9	8.95	0.05
60	82.5	99.7	9.12	0.09
72	73.3	99.8	19.67	0.10
84	53.8	99.4	4.59	0.19
97	50.3	84.7	0.52	19.86

4.4.3 Optimization for VGG16 feature extraction

To further reduce the inference time, the optimization approach described in Section 4.3.2 is implemented to eliminate the repetitive computation due to the overlapping regions between adjacent sliding windows. For demonstration purposes, the pruned VGG16 network where 84% of kernels are removed is selected for crack detection, and the server setup is used as the computing platform. Since the pruned network is trained using 224×224 image patches as input, it is necessary to re-train the network by removing zero-paddings in the convolution layers for consistency. The reason is that the optimized approach takes the whole image as input, and the patches in the image must be no zero-padding. As shown in Section 4.4.2, it takes 4.04 (sec)

Table 4.3.: Mean (μ) and standard deviation (σ) of detection accuracy before and after network fine-tuning - VGG16 with corrosion datasets

Pruned	μ (%)		σ (%)	
filters (%)	before	after	before	after
0	99.7	99.7	0.28	0.28
12	95.3	99.1	1.76	0.12
24	98.4	99.3	0.27	0.21
36	97.1	99.5	1.40	0.11
48	95.6	99.4	1.20	0.05
60	87.2	99.3	6.26	0.13
72	77.0	99.0	16.98	0.20
84	65.0	98.4	10.20	0.31
97	49.5	96.0	0.74	0.95

to process $3,720$ patches for the pruned VGG16 network when $3,720$ patches are passed to the network separately. By using the whole image as input and extracting features based on the proposed optimization approach, it takes only 0.62 (sec), which is 6.5 times faster than the approach without optimization.

4.5 Concluding Remarks

This study presents an approach for developing efficient DCNNs for damage detection in SHM. This is critical since an effective inspection system requires rapid inference and on-board deployment on mobile inspection devices, especially for large-scale civil infrastructures. The results in Section 4.4.1 show that the conventional transfer learning-based approach alone is not adequate for rapid inference as the resulting network spends the majority of time on computing the features. Section 4.4.2

demonstrates that by network pruning, the inference time of the pruned VGG16 network is nine times faster than the original network, and the memory demand is reduced by 80% without trading the detection performance. To reduce the pruning time spent on the server machine, a sensitivity analysis is conducted. The analysis indicates that using a smaller number of fine-tuning epochs during pruning reduces the pruning computation time and does not lead to a significant drop in detection accuracy. In addition, pruning ResNet18 reduces the inference time on the edge device to 8.9 (sec), which is 3.55 times faster than VGG16. Results from five-fold cross-validation demonstrate the robustness of the proposed approach, as the mean detection accuracy is 99% with a standard deviation below 0.5%. Network pruning should stop when 97% of the filters are removed from VGG16. Finally, Section 4.4.3 shows that the inference time is reduced by a factor of 6.5 through the proposed optimization approach for feature computing.

As data availability and resource efficiency are critical issues for field applications, when designing a DCNN to solve a specific domain problem, one should consider whether it is necessary to construct a large network. By exploiting transfer learning and network pruning, one can construct DCNNs without the need of a huge amount of training data. The efficiency in memory requirement and inference time is achieved without losing performance in damage detection through the reduction of the network size. It is worth mentioning that in some applications, such as nuclear power plant domes, structural and non-structural cracks are both critical since the non-structural cracks can lead to deterioration of underneath surfaces due to infiltration of water and air. For some other applications, however, such as earthquake reconnaissance where the non-structural cracks are not important, one needs to develop approaches to differentiate between structural and non-structural cracks.

5. DESIGN OF ONE-DIMENSIONAL ACOUSTIC METAMATERIALS USING MACHINE LEARNING AND CELL CONCATENATION

5.1 Introduction

5.1.1 Motivation and Relevant Works

Acoustic metamaterials are artificial functional materials that offer unique dynamic properties. Either acoustic or elastic waves propagating through a metamaterial could experience effects including, just to name a few, frequency band gaps [320–323], anomalous refraction [324–327], lensing [328], cloaking [329–332], robust wave-guiding [333–336]. These effects are typically not achievable in ordinary (i.e., non-engineered) materials. Acoustic metamaterials are typically assembled based on a combination of spatially distributed subwavelength scatterers that can span a variety of materials, shapes, and dimensions [337]. Most classes of metamaterials typically exhibit spatial periodicity, i.e., translational symmetry in space, and can be designed based on their frequency-wavevector band structure. Acoustic metamaterials can also be non-periodic. Examples include functionally graded materials [329], graded index phononic crystals [328], and acoustic black holes [338, 339]. The selection of optimal values of the different design parameters to ensure the material exhibit selected dynamic properties (e.g. band gaps in a given operational frequency range) is a very challenging task given that many possible combinations of parameters could potentially result in similar performance; this is a well-known issue in inverse problems due to the non-uniqueness of the solution. In material design, this non-uniqueness is often due to the fact that the dynamic response is not an explicit function of the design parameters, hence the inverse design problems often have no closed-form solutions.

Since the emergence of the concept of metamaterials, several approaches have been proposed in order to efficiently optimize design parameters. The initial, and probably still the most widely adopted, design methodology for metamaterial systems consisted in a physics-driven approach where analytical and numerical tools were used, in an iterative (trial-and-error) fashion, to synthesize materials exhibiting prescribed dynamic properties such as phase and group velocities, wave polarization, and band gaps [327, 328, 340–344]. Although an effective method to understand the underlying physical behavior and to predict performance, this approach heavily depends on designers' insights about the system and on a trial and error process. Such an approach quickly breaks down when a large number of material and geometric parameters are involved or when multiple functionalities are required. More recently, systematic optimization approaches have been used in metamaterial design problems. Among them, topology optimization combined with level set methods has been widely used in optimizing geometrical configurations [345–348]. However, the level set function is usually formulated in the physical space that defines the geometry, such method is not appropriate for the optimization of properties in a transformed space such as the frequency-wavenumber spectrum. In acoustic metamaterial design, level set methods were limited to optimizing the geometry only for a selected target frequency [349]. Besides the level set based approaches, topology optimization [350–355] combined with conventional gradient-based optimizers [356, 357] or genetic algorithm (GA) [358–364] were also explored. The target properties in a transformed space such as group velocity, equi-frequency contours, and frequency band gaps, can be formulated with cost functions for optimization. Having higher chance to reach the global optimum, GA surpasses the performance of gradient-based optimizer which is very sensitive to initial guesses. However, GA algorithms are quite computationally intensive since the evolution of generations scales exponentially with the size of the design space.

Therefore, the developments of new design tools are certainly a critical need for the metamaterial community.

Recently, advances in artificial intelligence (AI) and machine learning (ML) algorithms have significantly impacted numerous fields of research and stimulated many interdisciplinary applications. In the frame of metamaterial design, an increasing number of studies have demonstrated the potential of these ML-based approaches [365–368]. The use of ML techniques leads to more accurate estimates and reduces computation time in forward inferences after training. For instance, the multi-layer perceptron (MLP) or equivalently, the fully-connected neural network (NN), has been shown to be capable of approximating any non-linear function [12,172,174,249]. Based on sensor measurements, the NN learns the mapping between the inputs and the outputs implicitly and serves as a surrogate model of the precise physical model [369]. Compared to purely physics-based approaches, the incorporation of a data-driven model brings more flexibility to design frameworks since an exact model is usually difficult to obtain in real practices.

In addition to NN-based approaches, reinforcement learning (RL) is another category of ML techniques that differs from both the supervised and unsupervised ML algorithms. Inspired by how humans learn new tasks, RL attempts to determine the optimal decision strategy by interacting with the environment [370]. In other words, an agent is trained to find the decision policy that maximizes the cumulative future rewards. In recent years, there have been many successful applications in robotic navigation, control, and AI gaming that demonstrate the potential of RL-based approaches [215, 371–374]. During the training stage, the agent receives a reward assigned from the environment immediately after choosing an action at the current state. The agent is allowed to select the greedy action (i.e., exploitation) or the exploratory action in order to evaluate more possibilities in the state-space for better decision making. In the context of designing metamaterials, conventional optimization-based approaches may reach local solutions since the design parameter space is extremely vast. The incorporation of RL into the metamaterial design framework is expected to reduce computation efforts due to the reward mechanism and its ability in exploration.

This study presents two ML-based frameworks that enable the design of one-dimensional periodic and non-periodic metamaterials. For periodic metamaterials, a RL-based approach is proposed in order to design the materials capable of achieving user-desired frequency band gaps. As the major computational burden lies in the forward simulation of the physical model, it is crucial to evaluate how many executions of the forward simulation are required in a design process. The GA algorithm serves as a baseline reference for the proposed approach since GA avoids the computation of gradients and can be formulated with target properties defined in the frequency domain. For non-periodic metamaterials, a NN-based approach that unifies the simulation of the physical behavior and the optimization of the design parameters is proposed. First, the NN unit learns the behavior of the material unit cell, then the dynamic response of the entire metamaterial is predicted by the concatenation of identical NN units. This is the first demonstration that uses NN as the numerical model with the concatenated networks serving as the discretized elements in the physical domain to solve structural mechanics problems. The design parameters of each individual material element are then determined through standard optimization algorithms.

It is worth mentioning that the proposed NN-based approach requires only a one-time network training to model the entire metamaterial. Both design techniques will be validated via numerical simulations.

5.1.2 Contribution and Scope

This work develops two novel frameworks for the design of periodic and non-periodic metamaterials. For periodic metamaterials, the proposed RL-based approach leverages a reward mechanism and the exploration over the design space and therefore leads to a more efficient search for optimal designs. Moreover, the proposed approach can be applied to the efficient solution of various inverse design problems since it does not require the evaluation of gradients. For non-periodic metamaterials, the proposed NN-based approach builds a surrogate model of the metamaterial with the concatenation of identical NN units trained to describe the dynamic behavior of the material unit. It is worth mentioning that this is the first demonstration of NN being used in the discretization of a continuous media. Without the demand of a precise physical model, the proposed surrogate model is able to accurately predict the dynamic response of the metamaterial with only one-time network training, and the design of each material unit is accomplished through standard optimization algorithms.

The structure of this paper is described as follows. Section 5.2 introduces the two design problems considered in this study along with some fundamental physical details. In Section 5.3, the two proposed frameworks for the periodic and non-periodic metamaterial design are introduced. For each design problem, a numerical example and the corresponding formulations are discussed in detail. Section 5.4 presents the results as well as the discussion about the metamaterial design. A summary and future work are provided in Section 5.5.

5.2 Problem Description

5.2.1 1D Diatomic Mechanical Lattice

The classical configuration of an acoustic metamaterial includes a fundamental unit cell that is periodically repeated in space to form the final structure. The dynamic behavior of the periodic medium can be obtained based on the fundamental response of the unit, that is exploiting the knowledge of the underlying periodicity of both material and geometric properties. This characteristic makes the design and simulation of periodic metamaterials more systematic and computationally efficient. A key feature for the dynamic characterization of the unit cell is the frequency-wavenumber (or wavevector in higher dimensions) dispersion relation. In metamaterial design, this is typically referred to as band structure. Based on this dispersion structure, quantities such as the passing bands and the band gaps (i.e., ranges of frequencies in which propagating waves are not supported) are known, and the phase and group velocities can be derived. In this section, we briefly describe the mathematical model of a simple diatomic one-dimensional mechanical chain and the corresponding optimization problem to be solved.

Consider a 1D diatomic mechanical lattice assembled by a periodic distribution of masses m_1 and m_2 connected by springs with stiffness constant k, as shown in Figure 5.1. The fundamental unit cell contains two masses and two springs, and has a lattice constant a.

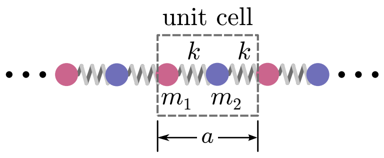


Fig. 5.1.: Schematic illustration of the 1D diatomic mechanical lattice.

Solving the equations of motion with Bloch boundary conditions [375] yields the frequency $\omega_\pm$ as a function of the wavenumber K:

$$\omega_\pm \quad = \quad \left(k\left(\frac{1}{m_1}+\frac{1}{m_2}\right) \pm k\left(\left(\frac{1}{m_1}+\frac{1}{m_2}\right)^2 - \frac{4\sin^2(Ka)}{m_1 m_2}\right)^{1/2}\right)^{1/2} \quad . \tag{5.1}$$

The frequency is a periodic function of K with period π/a, and has two branches denoted by subscriptions + and −. They are separated by a band gap delimited by the frequency values ω_1 to ω_2 where:

$$\omega_{1,2} = \sqrt{2k/m_{1,2}}\ . \tag{5.2}$$

An example of the dispersion curves in the extended zone scheme with the band gap is plotted in Figure 5.2. In this example, $m_2/m_1 = 2$.

A classical design problem in mechanical lattices is to find the value of the constitutive parameters to achieve specific band gap ranges. From Eq. (5.2), it is easily found that $m_{1,2} = 2k/\omega_{1,2}^2$. To this end, in Section 5.3.1, a proposed approach based

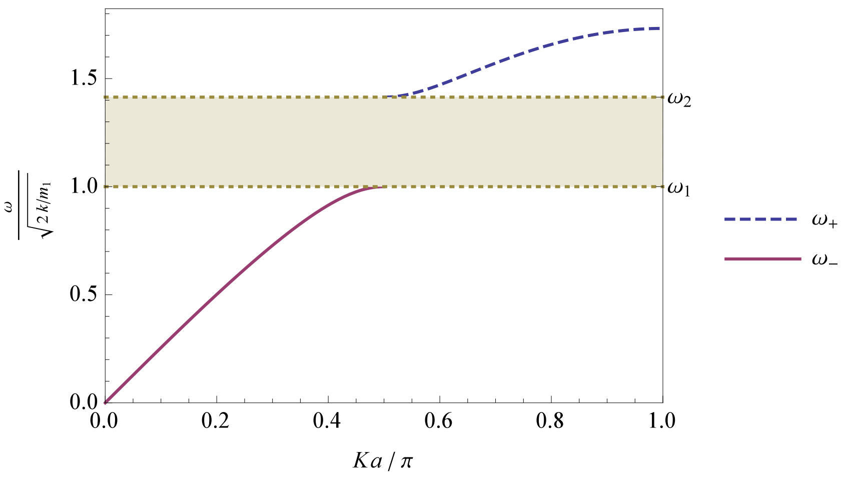


Fig. 5.2.: Dispersion of the 1D diatomic mechanical lattice in the extended zone scheme.

on RL is used to optimize m_1 and m_2 to attain the desired band gap range ω_1 and ω_2, and the estimated values are compared with the exact answer to evaluate the performance of the proposed approach. Although in this study the method is developed for simple lattice structures, this approach could be generalized to optimize the reciprocal space properties in complex phononic structures.

5.2.2 1D Continuous Bar

In this section, we illustrate the mathematical model to design and simulate an elastic periodic bar under longitudinal vibrations. In its most general form, the bar is made of multiple materials (assumed isotropic), therefore it has a piece-wise variable Young's modulus E, mass density ρ, and cross-sectional area A along its longitudinal axis x, as shown in Figure 5.3(a). Suppose the bar is designed to forbid the propagation of the wave in the forward direction (positive x-direction), the corresponding constraint would impose a null displacement at the right end. Clearly, depending on users' specified performance, different constraints may be imposed. In this design example, the main objective is to identify the optimal values of these material parameters (e.g., A or E) in order to satisfy given vibration constraints. A transfer matrix model [376] is adopted for the approximate numerical solution of the wave equation in the bar. The bar is divided into n elements (element number from 1 to n) with $n+1$ associated nodes (node number from 0 to n), as shown in Figure 5.3(b). Since the cross-sectional area A varies smoothly with range much larger than the thickness, Saint-Venant's principle [377] suggests that the normal stress distributes uniformly and can be represented as a normal internal force P_i, at the i-th node. Under harmonic vibrations, the equilibrium of the i-th element yields the transfer matrix equation relating the states at the $(i-1)$-th and the i-th nodes,

$$\begin{Bmatrix} u \\ P \end{Bmatrix}_i = \begin{pmatrix} 1-\omega^2(\Delta x_i)^2(\frac{\rho}{E})_i & -(\frac{\Delta x}{EA})_i \\ \omega^2(\rho A \Delta x)_i & 1 \end{pmatrix} \begin{Bmatrix} u \\ P \end{Bmatrix}_{i-1}, \tag{5.3}$$

where u_i is the x-displacement of the i-th node; E_i, ρ_i, A_i are the discretized parameters of the i-th element (see Figure 5.3(c)), ω is the circular frequency (in rad/s), and $\Delta x_i = x_i - x_{i-1}$.

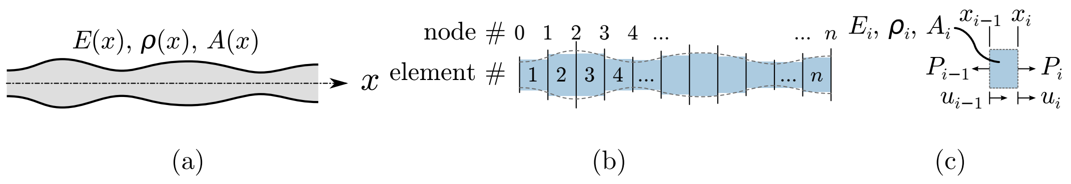

$E(x)$, $\rho(x)$, $A(x)$
x
node # 0 1 2 3 4 ... ... n
element # 1 2 3 4 ... ... n
E_i, ρ_i, A_i
x_{i-1} x_i
P_{i-1}
P_i
u_{i-1}
u_i
(a)
(b)
(c)

Fig. 5.3.: Schematic representation of the lumped transfer matrix model of a 1-D continuous bar: (a) Material properties are functions of the x-coordinate, (b) the bar is discretized into n elements, (c) the free body diagram of the n-th element.

Assume the bar is made of aluminum with material constants $E = 69\text{GPa}$ and $\rho = 2700\text{kg/m}^3$. We divide the bar into 20 units of equal length and assign the target cross-sectional area A_i for each unit as:

$$A_i = [2,4,6,4,2,2,4,6,4,2,2,4,6,4,2,2,4,6,4,2] \times 10^{-3}\ (\text{m}^2),\ i = 1 \sim 20 \tag{5.4}$$

Then, a harmonic axial load is applied at the left end of the bar while the right end is set to be either (a) fixed or (b) free. These two conditions are equivalent to a statement of continuity with the adjacent end having either infinite or zero mechanical impedance, respectively. Of course, any other boundary condition can be considered as a linear combinations of these two states. The model can be solved for the steady state response. Specifically, we solve it for the low, mid, and high frequency ranges, ultimately producing six responses.

With the dynamics of the medium being fully characterized by the transfer matrix model, the design problem can now be formulated. Such problem can be stated as follows: given the six responses in terms of degrees of freedom at the left end and right end (i.e., (u_l, P_l) and (u_r, P_r)), and the corresponding actuation frequencies, we want to determine the twenty cross-sectional areas A_i that allow satisfying (in an optimal sense) the target dynamic conditions. A further extension of this design problem would also consider the determination of these cross-sectional areas in order to achieve the target dynamics while using a minimum number of different sections.

The above two questions identify, in a simple setting, the typical problem that arises in the design of metamaterials and structures having desired frequency dependent characteristics.

5.3 Methodology

5.3.1 Framework for Periodic Metamaterial Design

The proposed framework for the design of periodic metamaterials is schematically illustrated in Figure 5.4. Given a user-defined target dynamic behavior specified in the reciprocal-space (e.g. a frequency-wavenumber dispersion relation or band structure), the proposed approach uses reinforcement learning to estimate the material properties capable of achieving the target behavior. Starting with an initial guess for the material properties, the RL algorithm iteratively searches for the optimal properties based on the rewards estimated on the basis of the physical response calculated via numerical models (e.g., finite element analysis). During the design process, the RL agent selects the design parameters and interacts with the environment (i.e., the physical model in the case of metamaterial design). A reward is then assigned to the agent based on the selected parameters. The process is implemented iteratively and the agent will continue exploring the design space until the termination condition is reached. The design process is completed once the dynamic behavior of the synthesized metamaterial satisfies a user-defined tolerance.

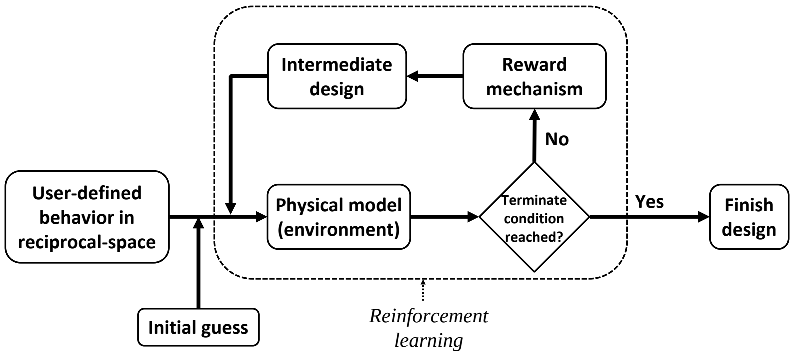

Intermediate design
Reward mechanism
No
User-defined behavior in reciprocal-space
Physical model (environment)
Terminate condition reached?
Yes
Finish design
Initial guess
Reinforcement learning

Fig. 5.4.: Flowchart presenting a schematic view of the design framework for periodic metamaterials.

Application to the Design of a 1D Diatomic Mechanical Lattice

The design framework was validated on a simple case study consisting in the design of a 1D diatomic lattice (as described in Section 5.2.1). The objective was to determine the value of the masses m_1 and m_2 to achieve a specified dynamic behavior in the frequency domain:

$$\Delta\omega = \omega_2 - \omega_1 \tag{5.5}$$

$$\omega_2 = \sqrt{\frac{2k}{m_2}} \tag{5.6}$$

$$\omega_1 = \sqrt{\frac{2k}{m_1}} \tag{5.7}$$

where $\Delta\omega$ is the user-defined frequency difference between the diatomic resonant frequencies. The proposed RL approach starts with an initial guess of (m_1, m_2), or equivalently, an initial state S_0, and attempts to find the optimal state S_{opt} by selecting actions to increase or decrease the values of (m_1, m_2). In this context, the state-action pairs are formulated as:

$$\boldsymbol{S} = \{(m_1, m_2) \mid \alpha_1 \leq m_1 \leq \beta_1, \alpha_2 \leq m_2 \leq \beta_2\} \tag{5.8}$$

$$\begin{aligned}\boldsymbol{A} = \{a_j \in \{m_1 + d, m_1 - d, m_2 + d, m_2 - d, \phi\}\} \\ , j = 1 \sim 5, d > 0\end{aligned} \tag{5.9}$$

where $\alpha_1, \alpha_2, \beta_1$, and β_2 are the lower and the upper bounds of the mass values m_1 and m_2. d is the increment, that is the step size that determines the precision of the estimation. In the action space $\boldsymbol{A}$, we consider five actions, where the first four actions allow the adjustment of the mass values and the fifth action stands for "do nothing," or equivalently, "stay at the current state." The ultimate goal is to determine the best action given a current state value. To this end, the proposed approach implements the Q-learning algorithm to update the Q-score of each state-action pair using the bellman equation [378]:

$$\begin{aligned}Q(S_i, a_j) \leftarrow (1 - \alpha^{(n)})Q(S_i, a_j) + \alpha^{(n)}[R(S_i, a_j, S_{i+1}) \\ -\rho^{(n)}t(S_i, a_j, S_{i+1}) + \gamma \max_{b \in \boldsymbol{A}(S_{i+1})} Q(S_{i+1}, b)]\end{aligned} \tag{5.10}$$

where $Q(S_i, a_j)$ is the Q-score of picking action a_j at state S_i, $R(S_i, a_j, S_{i+1})$ is the immediate reward after choosing action a_j at state S_i, $t(S_i, a_j, S_{i+1})$ is the transition time from state S_i to S_{i+1}, $\max_{b \in \boldsymbol{A}(S_{i+1})} Q(S_{i+1}, b)$ is the maximum Q-score among all the actions $b \in \boldsymbol{A}(S_{i+1})$ in the next state S_{i+1}, n is the iteration number, $\alpha^{(n)}$ is the learning rate at iteration n, $\rho^{(n)}$ is the average reward at iteration n, and γ is the discount factor for the expected future rewards. In this study, the immediate reward is defined as a function of the difference between the vibration frequencies computed using the intermediate design parameters and the target frequencies:

$$\Delta(\%) = \left| \frac{\hat{\omega} - \omega_t}{\omega_t} \right| \times 100\% \tag{5.11}$$

$$R(S_i, a_j, S_{i+1}) = \begin{cases} \frac{1}{\Delta}, & \text{if } \Delta \neq 0 \\ 1, & \text{if } \Delta = 0 \end{cases} \tag{5.12}$$

where $\hat{\omega}$ is the frequencies of the diatomic computed using Eqs. (5.6) and (5.7) with the intermediate values of (m_1, m_2), and $\omega_t = (\omega_{1t}, \omega_{2t})$ is the target frequencies. Note that the lower the Δ value the higher the reward.

During the learning process, the agent is allowed to select the greedy action (exploitation) that has the maximum Q-score at current state, or to pick the exploratory actions (exploration) to visit the states that have not been visited. This is essential since the agent could be trapped in local loops without having the chance to explore an action that leads to a higher expected future reward if the agent always picks the greedy action. The exploration probability should be high enough at the beginning for the agent to explore sufficiently, and decay with the iteration number so that the agent selects the greedy actions at the end of learning. After sufficient number of learning iterations, the best action at each state S_i can be determined by picking the action with the highest Q-score. The optimal policy corresponds to the set of best actions for all the states.

For demonstration purpose, the target frequency difference $\Delta\omega$ is set to 0.2. Assuming the spring constant $k = 1$ and the target frequency of the diatomic is $\omega_t = (\omega_{1t}, \omega_{2t}) = (1.3, 1.5)$, the corresponding target design parameters $(m_{1t}, m_{2t}) = (1.1834, 0.8889)$. The lower and the upper bounds of the mass values are $\alpha_1 = \alpha_2 = 0.5$ and $\beta_1 = \beta_2 = 1.5$, respectively. In terms of the hyper-parameters of the Q-learning algorithm, the learning rate $\alpha^{(n)}$ at iteration n is $log(n)/n$, the discount factor $\gamma = 0.99$, and the transition time $t(S_i, a_j, S_{i+1})$ is set to unity. Initially, the

Q-scores $Q(S_i, a_j)$ are set to zero, and the exploration probability is 1 with a decay rate of 0.999.

For the searching increment d, it is noted that the selection of d requires careful considerations. A small value of d may lead to higher precision in the estimation while increases the chance of reaching local solutions. A large d value is more likely to achieve the global solution while losing the precision of the estimations. To address these issues, an adaptive searching scheme is proposed to explore the design space more efficiently. The searching scheme consists of three epochs with decreasing d values. In the first epoch, d is set to 0.1 with $100,000$ iterations used to update the Q-scores. For the second and the third epoch, the d value is divided by a factor of 10 and 100, respectively, to enhance the precision in the estimation of the design parameters. The number of iteration is $10,000$ for both the second and the third epoch. It is noted that the selection of d can be tied with the user-defined tolerance in the estimates of the design parameters. To demonstrate the robustness of the proposed approach against the selection of initial estimates, the initial state is assumed to be $S_0 = (0.5, 0.5)$, which is far away from the target design values $(1.1834, 0.8889)$. The design results are discussed in Section 5.4.1.

5.3.2 Framework for Non-periodic Metamaterial Design

The proposed framework for the design of non-periodic 1D metamaterials is illustrated in Figure 5.5. The proposed approach aims at designing an inhomogeneous material consisting of multiple, interconnected, and potentially dissimilar unit cells capable of achieving user-defined dynamic performance. In addition, we assume that the dynamics of the different units can be described by the same neural network, as shown in Figure 5.5(a). Then, the entire assembly is obtained by combining together (an operation labeled here below "concatenation") multiple version of the same NN. A critical advantage of such an approach is that the NN does not require to undergo a new training phase every time the dynamic response of a new system (i.e., a new sequence of elements) is sought. The latter statement, of course, assumes that the system can be assembled from the available pre-trained elements.

Note also that, in this approach, instead of using a mathematical model to simulate the dynamics of the unit cell, a NN is previously trained to learn the complex mapping between the input and output responses of a class of unit cells. This is particularly useful when only the vibration measurements of the unit cell is available, or the physical model of the unit cell is not well understood. During the training phase, the NN is trained with the input and output responses, as well as the material properties of the unit cell (Figure 5.5(b)). The training data can be obtained from either numerical simulations, provided that the physical model of the cell is available, or the experimental data. Once the training is completed, the NN is capable of modeling the dynamic behavior of the unit cell, and the NN is ready for applications to prediction and design.

In the design phase, the user specify a desired output for a given input and the network assembles a sequence of cells to form an inhomogeneous structure capable of achieving the desired output. The proposed approach synthesizes the structure of the inhomogeneous system simply based on the concatenation of multiple NN units belonging to the same class of pre-trained NN discussed above. The term class is used to indicate that the NN represents the same dynamics of a specific type of structural elements (e.g. longitudinal waves in thin rods). Within the same class, different geometric parameters describing the finer details of the class are still possible and can be used to represent the dynamics of different elements (e.g. rods with different cross-sectional areas, and material properties).

Given that each NN receives the input from the output of its preceding element (i.e., NN), this condition dictates that input and output parameters must be consistent with each other, that is they must represent the same physical quantities. As shown in Figure 5.5(c), the number of required material units as well as the properties of the unit cells are not known *a priori* and must be determined by means of optimization algorithms. More specifically, the approach starts from a reduced number of units and keep adding new units until a prescribed user-defined tolerance on the target response is satisfied.

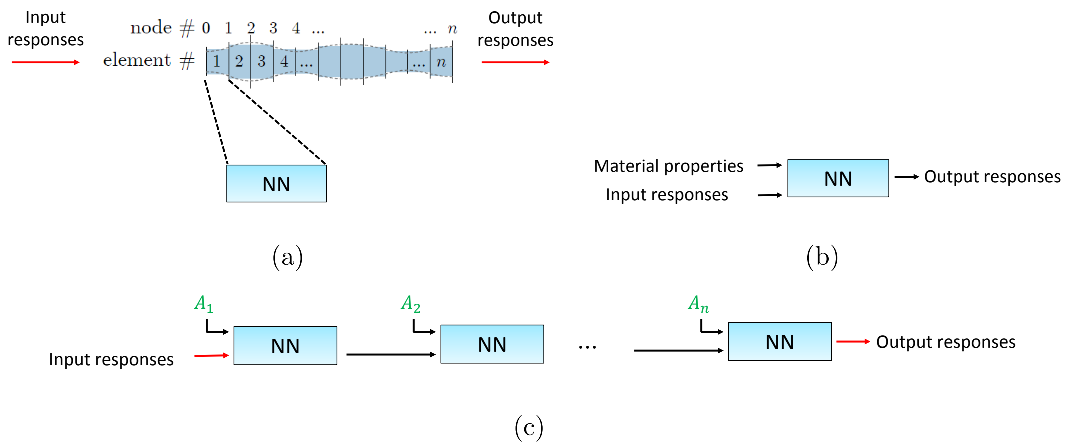


Fig. 5.5.: Schematic of the design framework for non-periodic metamaterial: (a) train a NN to learn the behavior of a generic unit cell (defined as a class of unit cells), (b) NN training inputs and outputs, and (c) design of the material properties (e.g. cross-sectional area) using network concatenation.

Application to the Design of a 1D Continuous Bar

The proposed framework was tested on the design of the 1D continuous bar described in Section 5.2.2. Given a user-defined set of input and output responses of the bar, the objective is to design the material properties of each unit shown in Figure 5.3. The Young's modulus, mass density, and the element length are assumed fixed to the values $E = 69\text{GPa}$, $\rho = 2700\text{kg/m}^3$, and $\Delta x = 0.01\text{m}$, respectively. In this example, the only design parameters are the cross-sectional areas A of each material unit.

In order to build a NN capable of modeling the dynamic behavior of the individual unit, training samples are generated using Eq. 5.3, where the upper and lower bounds of the parameters are listed in Table 5.1. The NN is expected to take the cross-sectional area A of the unit cell, the vibration frequency ω, the displacement u, and the force P as inputs, and estimate the response in terms of output displacement and force. A total of $10,000$ equally spaced values are first generated within the interval for each parameter, and each parameter array is randomly shuffled to obtain $10,000$ pairs of input arrays $[A, \omega, u, P]$ to the network. The input arrays are passed into Eq. 5.3 to generate the target output arrays $[u, P]$. As a result, instead of using a total of $10^4 \times 10^4 \times 10^4 \times 10^4 = 10^{16}$ sample permutations, the NN is trained with 85% samples randomly chosen within the set of $10,000$ samples, which greatly reduces the computation cost. The remaining 15% samples are used as validation dataset to ensure the generalization capability of the network. The NN consists of four layers, i.e., one input layer, two hidden layers with 15 and 10 nodes, and one output layer. The network training is achieved through standard back-propagation algorithm [263].

Once the NN representing the material unit class is established, the 1D continuous bar is designed via a network concatenation strategy. Assuming the bar consists of n units, given the user-defined input responses at the left end of the bar, the vibration frequency, and the output responses at the right end of the bar, the design parameters (i.e., the cross-sectional area of each unit in this example) are obtained by solving the following equation:

$$\arg\min_{A_i} \sum_j \left\| y_j(A_1, ..., A_i, ..., A_n) - y_{t,j} \right\|^2, \; i = 1 \sim n \tag{5.13}$$

where A_i is the cross-sectional area of the i^{th} material unit, $y_j(A_1, A_2, ..., A_n)$ is the estimated response of the right end of the bar from the concatenated NNs in the j^{th} vibration constraint, and $y_{t,j}$ is the target response defined by the user in the j^{th} vibration constraint. The interior-point algorithm [379] is used to solve Eq. (5.13). In this algorithm, there are two main steps in each iteration. The first step computes the Hessian of the Lagrangian of the cost function. If the Hessian is not positive definite, the algorithm proceeds with the second step by optimizing a quadratic approximation to the problem in a trust region.

Table 5.1.: Parameter ranges for the training samples.

Parameter	Unit	Lower bound	Upper bound
A	(m^2)	0.0015	0.009
ω	(MHz)	0.015	0.1
u	(nm)	-1	1
P	(N)	-10	10

In Section 5.4.2, the proposed approach is tested through the design of a bar with $n = 20$ material units. Furthermore, we show that the design of the bar can be achieved by incrementally adding material units, without needing to define the number of units n in advance. This is particularly beneficial when a less complicated design is critical to the user, as the desirable behavior of the bar may be achieved by using less number of units.

5.4 Results and Discussion

5.4.1 Design of the 1D Diatomic Mechanical Lattice

Table 5.2 shows the design results for the 1D diatomic lattice. The target mass values, as described in Section 5.3.1, are $(m_{1t}, m_{2t}) = (1.1834, 0.8889)$ to achieve the user desired frequency band gap width $\Delta\omega = 0.2$. The design error between the optimal (achieved via the RL-based approach) and the target designs is computed as:

$$Err = \frac{\|\hat{m} - m_t\|}{\|m_t\|} \times 100\% \tag{5.14}$$

where $\hat{m}$ is the mass estimates vector obtained from the proposed approach, and m_t is a vector representing the target mass values of the diatomic lattice. As indicated in Table 5.2, the proposed approach achieves a design error of 1.35%, when using a search increment of $d = 0.1$. The design estimates improve while decreasing the search increment d, resulting in smaller design errors 0.87% and 0.56% for $d = 0.01$

and $d = 0.001$, respectively. This demonstrates the effectiveness of the proposed adaptive searching scheme to enhance the precision of the estimates as well as reduce the chance of being trapped in local minima.

Due to the nature of the random walk in the Q-learning algorithm, we conduct three repeated trials to investigate the robustness of the proposed approach. Table 5.3 shows the design estimates and the design errors of the three repeated trials compared with the target values. Although trial 2 has a slightly higher design error of 2.44%, the design estimates of the three trials are very close to the target values. All the three trials achieve design errors below 3%, which again demonstrates the reliability of the proposed framework for periodic metamaterial design.

Furthermore, the GA algorithm [358] is implemented to serve as a baseline reference for the proposed approach. As previously mentioned, the computation cost highly depends on the number of executions at the forward simulation since the computation scales rapidly with the complexity of the physical model (e.g., finite element analysis). In this design example, both the GA and the proposed approach achieve comparable performances for designing the mass parameters. However, the proposed RL-based approach requires only 691 executions of the forward simulation. This is 97% less than the $22,600$ executions required by the GA algorithm. The population size and the maximum of generations are both set to 200 according to the suggestions provided in [358].

Table 5.2.: Results for the design of the diatomic lattice.

Increment	Initial		RL estimates		Target		Error
d	m_1	m_2	m_1	m_2	m_1	m_2	(%)
0.1	0.5000	0.5000	1.2000	0.9000	1.1834	0.8889	1.35
0.01	1.2000	0.9000	1.1900	0.9000	1.1834	0.8889	0.87
0.001	1.1900	0.9000	1.1810	0.8810	1.1834	0.8889	0.56

Table 5.3.: Repeated trials for the design of the diatomic lattice.

Trial	RL estimates		Target		Error
#	m_1	m_2	m_1	m_2	(%)
1	1.1810	0.8810	1.1834	0.8889	0.56
2	1.1490	0.9000	1.1834	0.8889	2.44
3	1.2010	0.8890	1.1834	0.8889	1.19

5.4.2 Design of 1D Continuous Bar

To validate the proposed framework for the design of a non-periodic metamaterial (see Section 5.3.2), we consider the design of a 1D continuous bar with 20 material units, i.e., $n = 20$, as shown in Figure 5.3(b). The vibration response of the bar is dependent on external constraints (e.g. frequency of excitation and boundary conditions) determined by the user. A total of six cases, including three vibration frequencies (i.e., $\omega = 0.02, 0.05, 0.09$ (MHz)) and two boundary conditions (the right end of the bar is fixed or free), are considered in this design example. The target design parameters are the cross-sectional areas of each material unit given in Eq. (5.4). Using the information about the actuation frequency, the boundary conditions (BC), and the cross-sectional areas, the responses at the boundaries of the bar can be obtained analytically.

Before proceeding to the design stage, it is essential to investigate the performance of the network in terms of its ability to represent the dynamics of the material unit. When used in a concatenated form small errors can propagate and get amplified hence leading to a rapid drop in accuracy. To this end, the theoretical displacement and force responses at each node of the bar are first computed by passing the left end responses (shown in Table 5.4) to the transfer matrix (shown in Eq. (5.3)) where the cross-sectional area of each unit is given in Eq. (5.4). Next, following the training scheme described in Section 5.3.2, the NN is established to model the dynamic behavior of

Table 5.4.: Six cases of different continuous bars and their corresponding dynamic response at the boundaries.

Case	ω	BC	u_l	P_l	u_r	P_r
#	(MHz)	at right end	(nm)	(N)	(nm)	(N)
1	0.02	fixed	−0.1257	1	0	1.2087
2	0.02	free	0.1588	1	0.2354	0
3	0.05	fixed	0.1295	1	0	0.0530
4	0.05	free	0.1298	1	0.0053	0
5	0.09	fixed	−0.1277	1	0	1.1049
6	0.09	free	−0.0737	1	0.0489	0

the material unit. By concatenating the same NN 20 times, the vibration responses of the bar at each node are estimated by passing the left end responses, the vibration frequency, and the cross-sectional area of the units to the concatenated NNs. Figures 5.6 to 5.11 show the comparison of the analytical response at each node with the response estimated from the proposed approach. Cases 1 through 6 are shown. For all cases, the estimations from the concatenated NNs match well with the target responses both in terms of displacement and force profiles. Tables 5.5 to 5.10 report the estimated response values from the network as well as the target response values obtained from the physical model at each node for Cases 1 to 6, respectively. The error between the network estimation and the target value is computed using the following equation to avoid the division by zero:

$$error = \frac{\hat{r} - r_t}{max(|r_{t,i}|)} \times 100\% \tag{5.15}$$

where $\hat{r}$ is the estimation from the network, r_t is the target response, and $max(|r_{t,i}|)$ is the maximum of the absolute values of the target responses in the i^{th} case, $i = 1 \sim 6$. According to Tables 5.5 to 5.10, the estimation errors are within $\pm 0.5\%$ where a significant portion of the errors are below $\pm 0.1\%$. Moreover, the estimation errors are not accumulated or amplified during the concatenation process, meaning that there is no limitation with regard to the number of material units that can be used. Table 5.11 shows the normalized root-mean-square (RMS) error of the displacement and force estimations for all cases. The largest RMS error occurs in Case 3 and 4 where the RMS values are 0.00090 and 0.00114 for the displacement and force estimations, respectively. This result demonstrates the robustness of the proposed approach in estimating the vibration responses of the bar using network concatenation.

In the design stage, the cross-sectional area of each material unit is treated as a design parameter. Assume that the objective is to design a bar made of $n = 20$ units and capable of achieving the right-end responses identical to Cases 1, 3 and, 5 (Table 5.4). This case is equivalent to designing the bar with boundary condition fixed at right end. In other words, the desirable right-end responses of the bar would be $[u_r, P_r] = [0, 1.2087]$, $[0, 0.0530]$ and $[0, 1.1049]$, when the bar receives inputs $[u_l, P_l] = [-0.1257, 1]$, $[0.1295, 1]$ and [-0.1277,1], respectively. The corresponding operating frequencies for these three cases are $\omega = 0.02, 0.05$ and 0.09 (MHz), and the associated target values of cross-sectional areas are shown in Eq. (5.4). At this point, 20 identical NNs are concatenated to model the dynamic behavior of the bar, and the cross-sectional areas are determined by solving Eq. (5.13). In this example, $j = 3$ is the number of vibration constraints defined by the user. Figure 5.12 depicts both the target and the design values of the 20 cross-sectional areas of the bar, when considering two boundary conditions (i.e., fixed or free at the right end). The design results do not necessarily match well with the target values for both cases, since there exist non-unique solutions for the cross-sectional areas capable to provide the desirable responses. However, compared to the target cross-sectional areas which are distributed periodically and thus having a band gap around 0.05 (MHz) (resulting in strong attenuation along $+x$ direction in both Cases 3 and 4, see Figures 5.8 and 5.9), the target design exhibit similar periodicity in terms of design parameter values. To investigate whether the design values indeed lead to the user-defined responses, the optimal parameters (i.e., the cross-sectional area values) are used into the physical model (i.e., Eq. (5.3)) to generate the dynamic response of the bar. As indicated in Table 5.12 and 5.13, the fourth column is the response at the right end estimated from the concatenated NNs using the designed material, and the fifth column is the response at the right end obtained from the physical model using the design parameters. Note that the responses from the network are very close to the target responses. According to Table 5.12, the exact design errors for fixed boundary condition at the right end, which are the differences between the responses of the physical model and the target, are extremely small compared to the maximum absolute target values, i.e., $|u_{max}| = 0.21$ (nm) and $|P_{max}| = 3.60$ (N) among Cases 1, 3, and 5. The maximum difference in the estimation of right-end displacement is 0.0012 (nm), while the maximum difference in the estimation of right-end force is 0.0139 (N). Moreover, for free boundary condition at the right end (Table 5.13), the maximum difference in the estimation of right-end displacement is 0.0002 (nm), while the maximum difference in the estimation of right-end force is 0.0019 (N). Compared to the maximum absolute target values $|u_{max}| = 0.26$ (nm) and $|P_{max}| = 2.14$ (N) among Case 2, 4, and 6, the estimation errors are negligible. This set of results demonstrates the robustness of the proposed approach for non-periodic metamaterial design.

Furthermore, the proposed approach is employed to design metamaterials by using less number of material units which, in turns, reduces the complexity of the resulting material and of the associated manufacturing. Without determining the number of material units in advance, the design of metamaterial can be achieved by incrementally adding material units to satisfy a user-defined tolerance in the target responses. For the proposed approach, we start with one NN that represents one material unit, and concatenate one additional identical NN if the design parameters cannot achieve the target responses within a user-defined tolerance (e.g., the value of the objective function shown in Eq. (5.13) is required to be less than 0.001). Similarly, assuming a user would like to design a bar with the least number of units to achieve the right-end responses identical to Cases 1, 3, and 5 shown in Table 5.4. By incrementally adding

NNs, the target right-end responses can be achieved by using only 14 units rather than 20 units to satisfy the tolerance of 0.001. The designed cross-sectional areas of the 14 units are listed as follows:

$$A = [1.9, 8.7, 8.6, 7.4, 1.6, 1.5, 1.6, 8.3, 8.5, 7.3, 3.3, 1.7, 5.7, 8.3] \times 10^{-3}\ (\mathrm{m}^2)$$

Table 5.14 shows the responses at the right end estimated from the concatenated NNs using the designed material, and the responses obtained from the physical model using the designed material. Comparing the fifth column with the target responses, the maximum difference in the estimation of right-end displacement is 0.0027 (nm), while the maximum difference in the estimation of right-end force is 0.0144 (N). These estimation errors are quite small with respect to the maximum absolute values in the target responses. According to Table 5.15, consistent observations are made when the boundary condition at the right end is changed to free. By incrementally adding NNs, the target right-end responses can be achieved with negligible errors by using only 15 units. The maximum difference in the estimation of right-end displacement is 0.0067 (nm), while the maximum difference in the estimation of right-end force is 0.0099 (N). The study presented above clearly indicates that the proposed approach is capable of achieving accurate and less complicated metamaterial design.

5.5 Concluding Remarks

This study presents two frameworks based on machine learning algorithms for the efficient design of acoustic metamaterials. The two frameworks are conceived to achieve user-defined dynamic properties (either in terms of dispersion or responses) and target the design of periodic and non-periodic acoustic metamaterials. The first framework leverages a reinforcement learning (RL) approach in order to optimize the frequency band gap of a phononic lattice. In particular, it is applied to a 1D diatomic chain where the value of the two masses has to be determined to achieve prescribed band gap properties. The proposed RL-based approach leverages a reward mechanism to efficiently explore the space of design parameters and obtain the optimal design.

Table 5.5.: The displacement (u) and force (P) responses at each node for Case 1. Prediction: the output from the network; target: the output from the physical model.

Case 1	u (nm)			P (N)		
node	prediction	target	error(%)	prediction	target	error(%)
0	-0.1257	-0.1257	0.00	1.0000	1.0000	0.00
1	-0.1904	-0.1904	-0.01	0.8927	0.8928	-0.01
2	-0.2110	-0.2110	-0.01	0.5678	0.5682	-0.03
3	-0.2116	-0.2116	0.00	0.0282	0.0285	-0.02
4	-0.1996	-0.1996	0.00	-0.3329	-0.3325	-0.04
5	-0.1632	-0.1632	0.01	-0.5033	-0.5027	-0.05
6	-0.1166	-0.1167	0.02	-0.6425	-0.6418	-0.06
7	-0.0861	-0.0862	0.03	-0.8416	-0.8408	-0.07
8	-0.0605	-0.0606	0.04	-1.0619	-1.0613	-0.05
9	-0.0183	-0.0184	0.04	-1.1653	-1.1646	-0.05
10	0.0673	0.0672	0.05	-1.1809	-1.1803	-0.05
11	0.1487	0.1485	0.06	-1.1234	-1.1230	-0.03
12	0.1802	0.1800	0.06	-0.8701	-0.8697	-0.04
13	0.1901	0.1899	0.06	-0.4092	-0.4091	-0.01
14	0.1931	0.1930	0.06	-0.0853	-0.0852	-0.01
15	0.1874	0.1873	0.04	0.0796	0.0794	0.01
16	0.1700	0.1699	0.03	0.2395	0.2391	0.03
17	0.1508	0.1508	0.02	0.5291	0.5289	0.02
18	0.1287	0.1287	0.01	0.9150	0.9146	0.03
19	0.0876	0.0876	0.00	1.1343	1.1341	0.02
20	0.0000	0.0000	-0.02	1.2090	1.2087	0.02

Table 5.6.: The displacement (u) and force (P) responses at each node for Case 2. Prediction: the output from the network; target: the output from the physical model.

Case 2		u (nm)			P (N)	
node	prediction	target	error(%)	prediction	target	error(%)
0	0.1588	0.1588	0.00	1.0000	1.0000	0.00
1	0.0765	0.0765	-0.01	1.1355	1.1354	0.01
2	0.0306	0.0307	-0.01	1.2658	1.2660	-0.01
3	-0.0018	-0.0018	-0.01	1.3443	1.3444	-0.01
4	-0.0504	-0.0504	-0.01	1.3411	1.3414	-0.02
5	-0.1445	-0.1445	-0.01	1.2980	1.2984	-0.03
6	-0.2297	-0.2296	0.00	1.1746	1.1752	-0.04
7	-0.2580	-0.2580	0.01	0.7828	0.7835	-0.05
8	-0.2610	-0.2610	0.01	0.1228	0.1234	-0.04
9	-0.2493	-0.2494	0.02	-0.3224	-0.3217	-0.05
10	-0.2106	-0.2106	0.03	-0.5352	-0.5344	-0.06
11	-0.1588	-0.1589	0.05	-0.7150	-0.7140	-0.07
12	-0.1231	-0.1232	0.06	-0.9859	-0.9850	-0.06
13	-0.0916	-0.0918	0.06	-1.3007	-1.3002	-0.03
14	-0.0389	-0.0390	0.06	-1.4571	-1.4568	-0.03
15	0.0691	0.0689	0.06	-1.4903	-1.4900	-0.02
16	0.1728	0.1727	0.06	-1.4313	-1.4312	0.00
17	0.2140	0.2138	0.05	-1.1368	-1.1368	0.00
18	0.2282	0.2281	0.05	-0.5894	-0.5897	0.02
19	0.2355	0.2354	0.04	-0.2004	-0.2007	0.02
20	0.2354	0.2354	0.02	0.0006	0.0000	0.04

Table 5.7.: The displacement (u) and force (P) responses at each node for Case 3. Prediction: the output from the network; target: the output from the physical model.

Case 3		u (nm)			P (N)	
node	prediction	target	error(%)	prediction	target	error(%)
0	0.1295	0.1295	0.00	1.0000	1.0000	0.00
1	0.0070	0.0070	0.00	1.6908	1.6902	0.03
2	-0.0569	-0.0569	-0.02	1.7658	1.7651	0.04
3	-0.0776	-0.0776	-0.02	0.8551	0.8549	0.01
4	-0.0786	-0.0786	-0.02	0.0281	0.0280	0.00
5	-0.0503	-0.0503	-0.01	-0.3901	-0.3908	0.04
6	-0.0026	-0.0025	-0.05	-0.6573	-0.6588	0.09
7	0.0222	0.0223	-0.07	-0.6848	-0.6858	0.06
8	0.0302	0.0303	-0.06	-0.3297	-0.3290	-0.04
9	0.0305	0.0305	-0.02	-0.0079	-0.0065	-0.08
10	0.0193	0.0192	0.06	0.1552	0.1560	-0.04
11	0.0006	0.0005	0.08	0.2587	0.2583	0.02
12	-0.0090	-0.0091	0.04	0.2650	0.2633	0.10
13	-0.0119	-0.0119	-0.01	0.1207	0.1183	0.14
14	-0.0117	-0.0116	-0.07	-0.0064	-0.0088	0.14
15	-0.0067	-0.0065	-0.18	-0.0679	-0.0707	0.16
16	0.0008	0.0011	-0.27	-0.1030	-0.1052	0.13
17	0.0042	0.0045	-0.23	-0.0943	-0.0931	-0.07
18	0.0049	0.0050	-0.12	-0.0270	-0.0209	-0.34
19	0.0040	0.0038	0.10	0.0251	0.0326	-0.42
20	0.0006	0.0000	0.48	0.0470	0.0530	-0.34

Table 5.8.: The displacement (u) and force (P) responses at each node for Case 4. Prediction: the output from the network; target: the output from the physical model.

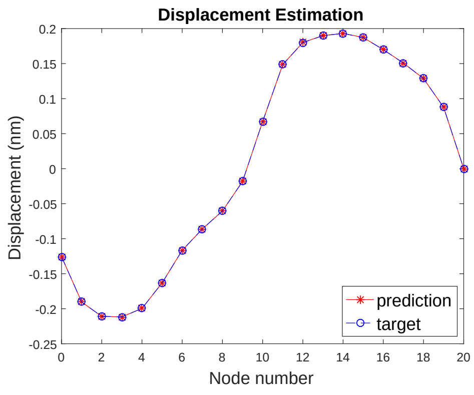
Case 4		u (nm)			P (N)	
node	prediction	target	error(%)	prediction	target	error(%)
0	0.1298	0.1298	0.00	1.0000	1.0000	0.00
1	0.0072	0.0072	0.00	1.6923	1.6917	0.03
2	-0.0569	-0.0569	-0.02	1.7691	1.7684	0.04
3	-0.0777	-0.0776	-0.02	0.8592	0.8591	0.01
4	-0.0788	-0.0788	-0.02	0.0317	0.0316	0.00
5	-0.0507	-0.0506	-0.01	-0.3874	-0.3882	0.04
6	-0.0030	-0.0030	-0.05	-0.6566	-0.6581	0.09
7	0.0219	0.0220	-0.07	-0.6886	-0.6896	0.06
8	0.0301	0.0302	-0.06	-0.3380	-0.3374	-0.04
9	0.0307	0.0307	-0.02	-0.0171	-0.0157	-0.08
10	0.0201	0.0200	0.06	0.1474	0.1482	-0.04
11	0.0017	0.0015	0.08	0.2552	0.2548	0.02
12	-0.0082	-0.0083	0.04	0.2730	0.2713	0.10
13	-0.0116	-0.0116	-0.01	0.1412	0.1388	0.14
14	-0.0123	-0.0122	-0.07	0.0172	0.0148	0.14
15	-0.0088	-0.0085	-0.18	-0.0473	-0.0501	0.16
16	-0.0020	-0.0016	-0.27	-0.0933	-0.0956	0.13
17	0.0022	0.0025	-0.23	-0.1141	-0.1128	-0.07
18	0.0041	0.0042	-0.12	-0.0793	-0.0733	-0.34
19	0.0054	0.0053	0.10	-0.0355	-0.0281	-0.42
20	0.0059	0.0053	0.48	-0.0061	0.0000	-0.34

Table 5.9.: The displacement (u) and force (P) responses at each node for Case 5. Prediction: the output from the network; target: the output from the physical model.

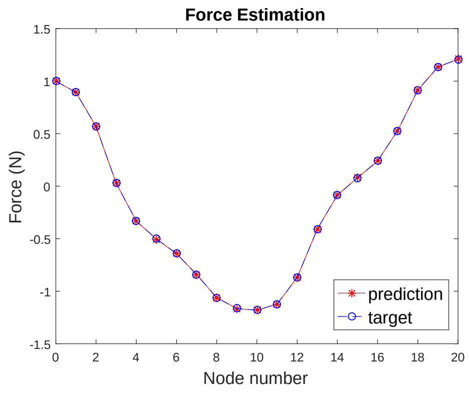
Case 5	u (nm)			P (N)		
node	prediction	target	error(%)	prediction	target	error(%)
0	-0.1277	-0.1277	0.00	1.0000	1.0000	0.00
1	-0.0404	-0.0404	0.00	-1.2049	-1.2050	0.00
2	0.0538	0.0538	0.00	-2.5996	-2.5994	-0.01
3	0.0493	0.0493	0.01	0.1873	0.1879	-0.02
4	-0.0192	-0.0192	0.01	1.8887	1.8894	-0.02
5	-0.1320	-0.1321	0.04	1.5575	1.5580	-0.01
6	-0.0797	-0.0797	0.02	-0.7224	-0.7229	0.01
7	0.0462	0.0462	-0.02	-3.4747	-3.4756	0.03
8	0.0723	0.0723	-0.01	-1.0817	-1.0812	-0.01
9	0.0210	0.0210	0.01	1.4156	1.4170	-0.04
10	-0.1079	-0.1080	0.07	1.7783	1.7796	-0.03
11	-0.1018	-0.1018	0.05	-0.0842	-0.0846	0.01
12	0.0286	0.0287	-0.03	-3.5988	-3.6012	0.07
13	0.0797	0.0798	-0.04	-2.1164	-2.1168	0.01
14	0.0566	0.0566	0.00	0.6370	0.6386	-0.05
15	-0.0604	-0.0605	0.09	1.6148	1.6168	-0.05
16	-0.1018	-0.1020	0.09	0.5720	0.5719	0.00
17	0.0049	0.0049	-0.03	-2.9455	-2.9491	0.10
18	0.0699	0.0700	-0.06	-2.6936	-2.6953	0.05
19	0.0800	0.0801	-0.03	-0.2790	-0.2777	-0.04
20	0.0001	0.0000	0.08	1.1024	1.1049	-0.07

Table 5.10.: The displacement (u) and force (P) responses at each node for Case 6. Prediction: the output from the network; target: the output from the physical model.

Case 6		u (nm)			P (N)	
node	prediction	target	error(%)	prediction	target	error(%)
0	-0.0737	-0.0737	0.00	1.0000	1.0000	0.00
1	-0.0540	-0.0540	0.00	-0.2719	-0.2719	0.00
2	0.0234	0.0234	0.00	-2.1356	-2.1352	-0.02
3	0.0457	0.0457	0.01	-0.9232	-0.9225	-0.03
4	0.0220	0.0219	0.02	0.6549	0.6555	-0.03
5	-0.0530	-0.0530	0.06	1.0339	1.0344	-0.03
6	-0.0616	-0.0616	0.04	0.1191	0.1190	0.01
7	0.0112	0.0112	-0.02	-2.0089	-2.0097	0.04
8	0.0457	0.0457	-0.02	-1.4308	-1.4306	-0.01
9	0.0403	0.0403	0.01	0.1478	0.1489	-0.05
10	-0.0208	-0.0209	0.10	0.8443	0.8454	-0.05
11	-0.0559	-0.0560	0.09	0.4842	0.4841	0.00
12	-0.0035	-0.0035	-0.03	-1.4482	-1.4501	0.09
13	0.0359	0.0359	-0.05	-1.6290	-1.6297	0.03
14	0.0500	0.0500	-0.03	-0.3910	-0.3899	-0.05
15	0.0158	0.0157	0.11	0.4723	0.4739	-0.08
16	-0.0382	-0.0383	0.13	0.7444	0.7447	-0.02
17	-0.0174	-0.0174	-0.02	-0.5747	-0.5773	0.12
18	0.0183	0.0183	-0.08	-1.4751	-1.4768	0.08
19	0.0489	0.0489	-0.07	-0.8452	-0.8445	-0.03
20	0.0490	0.0489	0.08	-0.0018	0.0000	-0.09

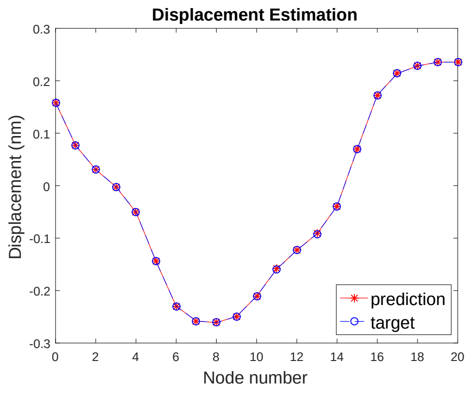


(a)

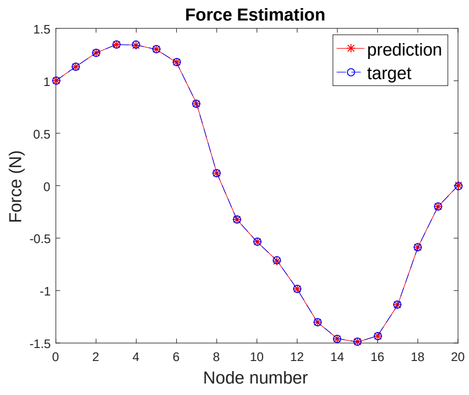


(b)

Fig. 5.6.: Network estimations at each node for Case 1: (a) displacement and (b) force.



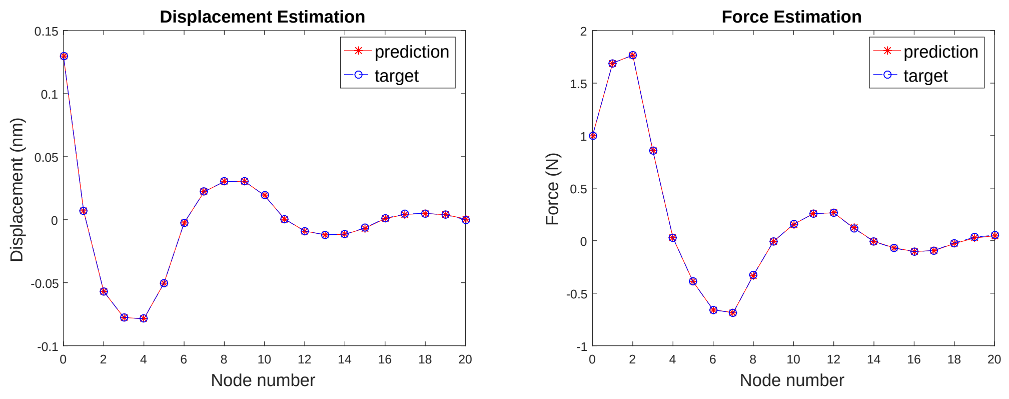
(a)



(b)

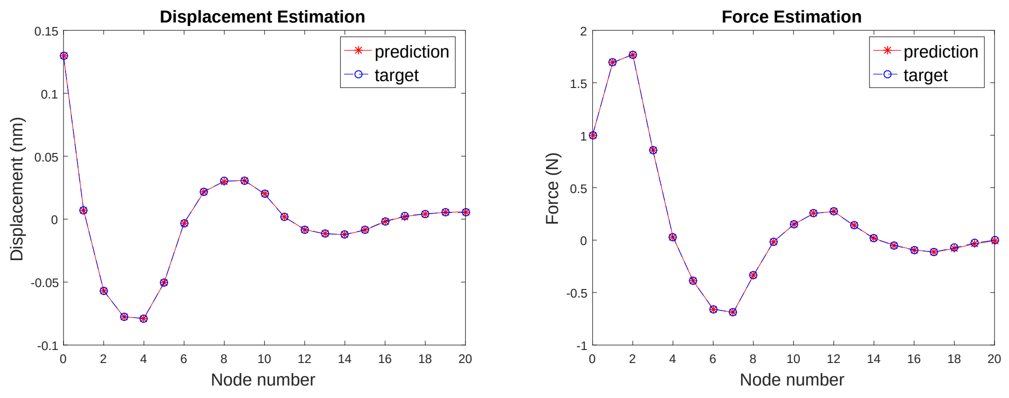
Fig. 5.7.: Network estimations at each node for Case 2: (a) displacement and (b) force.

Direct comparison with the exact solution showed the very good performance of the design approach which was capable of identifying the value of the parameters within a margin of error of 0.56%. Three repeated trials were conducted to address the randomness in the exploration of RL algorithm. All the design errors were found to be below 3%, which indicated the reliability of the proposed approach. In order to



(a) (b)

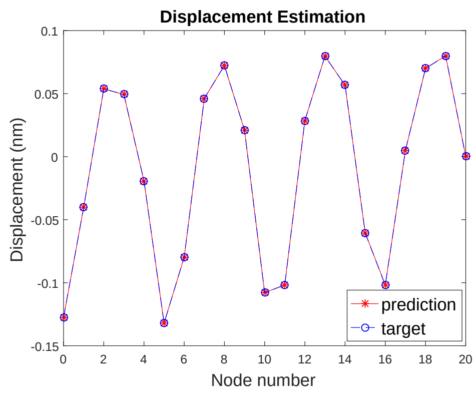
Fig. 5.8.: Network estimations at each node for Case 3: (a) displacement and (b) force.



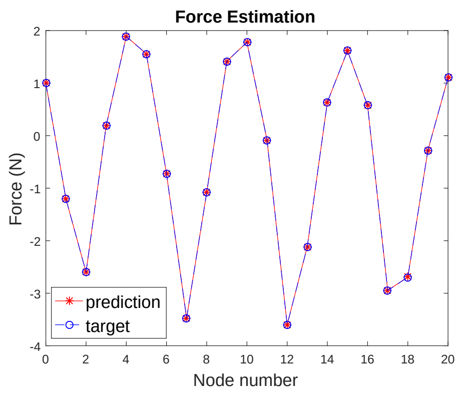
(a) (b)

Fig. 5.9.: Network estimations at each node for Case 4: (a) displacement and (b) force.

put the proposed methodology in perspective with other established techniques, a comparison with the GA algorithm was performed. The comparison showed that the proposed approach achieves a 97% reduction in computational cost, mostly driven by the large reduction in forward simulations.

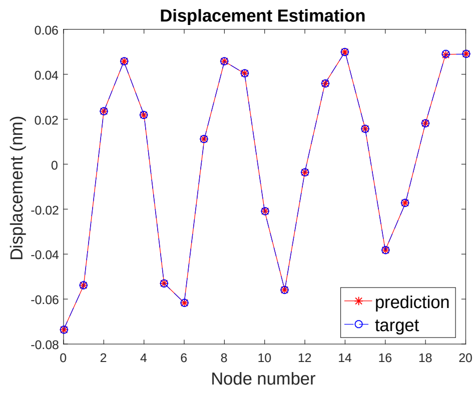


(a)

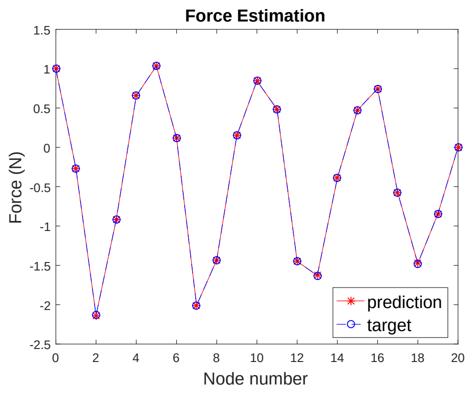


(b)

Fig. 5.10.: Network estimations at each node for Case 5: (a) displacement and (b) force.



(a)



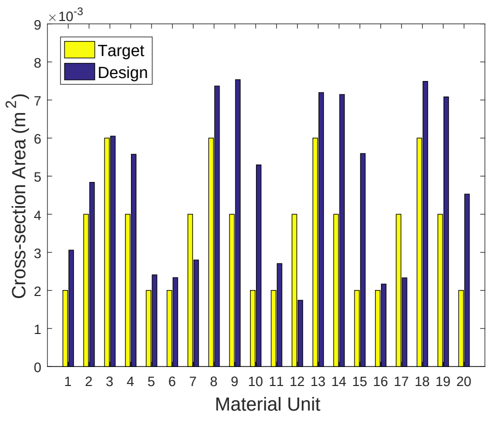
(b)

Fig. 5.11.: Network estimations at each node for Case 6: (a) displacement and (b) force.

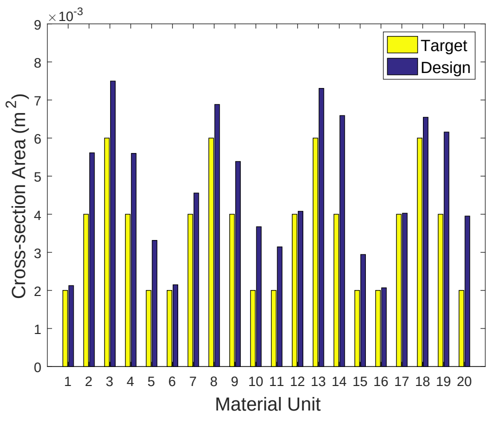
The second design framework is developed for non-periodic metamaterials. In this case, a machine learning based approach was used to learn the dynamic behavior of a class of material units so that the prediction of the dynamic response for any unit in such class could be obtained without the need for a numerical model. This kind of approach enabled also the unique concept of network concatenation in which complex

Table 5.11.: The normalized RMS error for displacement (u) and force (P) estimations.

Case	u	P
1	0.00018	0.00018
2	0.00020	0.00018
3	0.00090	0.00114
4	0.00090	0.00114
5	0.00027	0.00026
6	0.00036	0.00035



(a)



(b)

Fig. 5.12.: Material design results using the concatenation of 20 NNs, assuming the objective is to design a metamaterial to achieve: (a) the responses of Case 1, 3 and 5 (right end fixed), and (b) the responses of Case 2, 4 and 6 (right end free).

systems can be assembled and simulated by simply combining pre-trained networks. The most remarkable characteristic of this approach lies in the ability to build predictive models of complex dynamical systems from the use of basic (previously trained) networks that capture the dynamics of a single class of units. To the best of the authors' knowledge, this is the first demonstration that uses NN as surrogate models

Table 5.12.: The metamaterial design results from the concatenation of NNs, compared with the responses of the physical model and the target responses. (1): Responses of the network using the designed metamaterial, and (2) Responses of the physical model using the designed metamaterial. (Assuming the objective is to design a metamaterial to achieve the responses of Case 1, 3 and 5, boundary condition fixed at the right end)

Case	Parameter	Unit	(1)	(2)	Target responses
1	u_r	(nm)	0.0000	0.0000	0
	P_r	(N)	1.2087	1.2090	1.2087
3	u_r	(nm)	0.0016	0.0012	0
	P_r	(N)	0.0530	0.0669	0.0530
5	u_r	(nm)	0.0000	-0.0002	0
	P_r	(N)	1.1049	1.1069	1.1049

of individual components that can be assembled, by concatenation, to form complex structures. In this approach, the NN replaces the concept of element in conventional discretization methods, such as the finite elements or finite differences, used to solve differential equations. Once the NNs are concatenated, the response of the whole system can be obtained following assigned input conditions without the need to retrain the network with specific material configurations. This approach was tested on a system consisting in a 1D continuous bar. A fully-connected NN was used to model the dynamic behavior of the specific class of the material unit, that is the longitudinal dynamics of a thin bar. The NN accepts the material properties, the actuation frequency, and the external loads as inputs, and estimates the corresponding dynamic response as output. After an initial training for the class of structural elements, the NNs were concatenated to form a chain capable of modeling a 1D metamaterial. The units do not require to be periodic and the material properties are treated as design parameters. The main objective was to determine the necessary configuration and

Table 5.13.: The metamaterial design results from the concatenation of NNs, compared with the responses of the physical model and the target responses. (1): Responses of the network using the designed metamaterial, and (2) Responses of the physical model using the designed metamaterial. (Assuming the objective is to design a metamaterial to achieve the responses of Case 2, 4 and 6, boundary condition free at the right end)

Case	Parameter	Unit	(1)	(2)	Target responses
2	u_r	(nm)	0.2354	0.2354	0.2354
	P_r	(N)	0.0000	0.0002	0
4	u_r	(nm)	0.0055	0.0055	0.0053
	P_r	(N)	0.0000	0.0019	0
6	u_r	(nm)	0.0489	0.0489	0.0489
	P_r	(N)	0.0000	0.0005	0

number of units to achieve a user-defined dynamic response. Numerical simulations showed that with only one-time network training, the concatenated NNs strategy accurately estimated the dynamic response of the non-periodic material while never requiring any additional training (beyond the initial one necessary to characterize the class of units). This result also highlights the potential of the approach to explore large design spaces as well as the propensity to produce highly scalable framework.

Although this study considered only 1D systems as benchmark examples, the proposed design frameworks are completely general and can be extended to higher dimensional systems. Indeed, it is expected that the advantages put forward by this design methodology over existing methods will become even more pronounced when approaching increasingly more complex systems for which the computational burden scales up very rapidly.

Table 5.14.: The incremental design results from the concatenation of NNs, compared with the responses of the physical model and the target responses. (1): Responses of the network using the designed metamaterial, and (2) Responses of the physical model using the designed metamaterial. The incremental design leads to a metamaterial with only 14 units instead of 20 units. (Assuming the objective is to design a metamaterial to achieve the responses of Case 1, 3 and 5, boundary condition fixed at the right end)

Case	Parameter	Unit	(1)	(2)	Target responses
1	u_r	(nm)	0.0000	0.0000	0
	P_r	(N)	1.2087	1.2082	1.2087
3	u_r	(nm)	-0.0027	-0.0027	0
	P_r	(N)	0.0530	0.0545	0.0530
5	u_r	(nm)	-0.0001	-0.0004	0
	P_r	(N)	1.1049	1.1193	1.1049

Table 5.15.: The incremental design results from the concatenation of NNs, compared with the responses of the physical model and the target responses. (1): Responses of the network using the designed metamaterial, and (2) Responses of the physical model using the designed metamaterial. The incremental design leads to a metamaterial with only 15 units instead of 20 units. (Assuming the objective is to design a metamaterial to achieve the responses of Case 2, 4 and 6, boundary condition free at the right end)

Case	Parameter	Unit	(1)	(2)	Target responses
2	u_r	(nm)	0.2354	0.2354	0.2354
	P_r	(N)	0.0000	0.0007	0
4	u_r	(nm)	-0.0020	-0.0014	0.0053
	P_r	(N)	-0.0005	-0.0099	0
6	u_r	(nm)	0.0489	0.0491	0.0489
	P_r	(N)	0.0000	-0.0022	0

6. A PHYSICS-CONSTRAINED DEEP LEARNING BASED APPROACH FOR MULTI-OBJECTIVE INVERSE DESIGN OF ACOUSTIC WAVE SCATTERING

6.1 Introduction

6.1.1 Motivation and Relevant Works

Manipulation and design of acoustic and electromagnetic wave fields is the study of controlling amplitude, phase, and propagation direction of waves. The common approach to the design of wave fields is to use artificially designed material systems, known as metamaterials and metasurfaces [380–386]. Manipulation of wave via metamaterials introduced different features, that are unobtainable via a regular material, such as minimizing wave scattering by an object to make it invisible to the incident wave (cloaking) [381, 387], wave focusing [382, 383], and aligning the propagation direction of the wave (collimation) [384, 385]. While some of these properties are the result of a multi-material domain [388], some researchers utilize a tailored array of scattering objects to design the wave propagation behavior in the material [384, 389] and fulfil the target properties. The traditional procedure of designing the scatterers inside the material is performed by an optimization process [390–392]. This process typically requires the iteration over a large parameter space to determine the optimum material parameters, e.g location and geometry of scatterers inside the material, and hence repeatedly solving the wave equation numerically. Thus, the design process generally require a large amount of computational resources and time. Additionally, as the complexity and dimension of the design parameter space increase, such as a domain with an array of many scatterers, the chance of convergence to the optimal solution for the conventional methodologies reduces. Therefore, it appears that novel, efficient and reliable design approaches for acoustic wave control are still needed to address these design challenges.

Recent advances in the capability of generating large datasets have opened the era of Big Data. The idea of leveraging abundant information with the aid of artificial intelligence (AI) techniques have introduced more opportunities to develop novel solutions for many application domains. In the field of material design, machine learning (ML) algorithms such as support vector machine [159, 160], artificial neural network [12], decision tree [393], and the naïve Bayes [41] algorithms have been widely used in material property predictions and new material discoveries [394]. For instance, ML-based approaches for the material synthesis of layered double hydroxide, battery materials, and thermoelectric materials are reviewed in [395]. In these studies, the structures of materials are determined to satisfy the target material reactions. However, the selection of engineered features is necessary to transform the target responses into a set of quantitative descriptors for processing, which limits the generalization of these approaches. More recently, a lot of attention has been given to research efforts dedicated to deep learning (DL) based approaches. One major advantage of these approaches is the ability to extract meaningful features from the raw data without human intervention, as opposed to the conventional ML-base approaches. A well-known example is the development of deep convolution neural networks (CNN) that has led to breakthroughs in computer vision and natural language processing [161, 162]. Another merit of deep neural networks (DNN) is the relieving of computation burdens. Compared to optimization-based approaches which take a prohibitive amount of time for every query, the design inference of DL-based approaches is quasi-instantaneous once the training process is completed. Moreover, generative models such as auto-encoder [396], variational auto-encoder (VAE) [397], and deep auto-encoder (DAE) [398] have enabled the learning of latent representations of the data. The latent representations are a form of abstract features derived through training a network to reconstruct the inputs. Therefore, these representations approximate the distribution of the input data, and they can be employed to a variety of inference tasks. A review of DL-based relevant works in material design is provided as follows.

In nanophotonics research, DNN-based approaches have been proposed to design the materials that target to generate specific optical responses. The geometry of the plasmonic nanostructures is determined based on a desirable far field optical spectrum [399]. Similarly, parameters that control the geometry of the chiral metamaterial unit are estimated to satisfy a target optical reflection spectrum [366]. Using an adaptive DNN, the thickness of each nanostructure layer is determined to achieve the given optical absorbance spectrum [400]. The topology pattern of a 2D integrated nanophotonic Devices can be designed with DNN as well [401]. In addition to deterministic approaches, the incorporation of generative models into the design framework has revealed a great potential for material inverse design. An example is the use of VAE to learn the latent distribution of the material geometries where the latent features are employed to train the mapping between the target optical responses and the material geometries [367]. Also, the geometry of the metasurface unit cell can be retrieved based on generative models to satisfy the target optical transmission spectra [402]. In terms of the manipulation of acoustic waves, a DAE-based approach is proposed to design the geometry of phononic crystal that results in an anticipated wave dispersion behavior [368].

In this work, a DAE-based approach is proposed to design the geometry of acoustic wave scatterers that achieve the target downstream pressure fields at single or multiple wave frequencies. Compared to the existing studies which only aim at 1D signal of either optical responses or band gap distributions, the proposed approach tackles a more sophisticated inverse design of the scatterers satisfying 2D target responses. Instead of building separate networks, the proposed approach jointly optimizes a geometry estimator with a DAE that provides the estimator with physics constraints during training. The DAE attempts to reconstruct the input target pressure field while the geometry estimator leverages the latent representations of the target responses to design the scatterer geometry. Moreover, the existing studies solve the inverse design that achieves only one objective (e.g., a specific optical response or dispersion curve) at a time. The proposed approach, however, is able to design with multiple objectives since the target downstream pressure fields of the same scatterer geometry vary with the associated wave frequencies. This is referred to the situation where a user specifies different preferences of target response at different wave frequencies. In other words, multiple target responses exist when considering more than one wave frequency. It is worth mentioning that this is the first demonstration of the multi-objective inverse design of wave scatterers using DAE-based approaches. The proposed approach is validated with numerical simulations, and design examples are provided to further demonstrate the robustness of the proposed approach.

6.1.2 Contribution and Scope

This study presents a physics-constrained DAE-based approach to design the topological structures of materials based on the user-defined target downstream acoustic pressure fields. The proposed approach integrates a DAE and a topology estimator to solve the inverse design with aid of the latent representations of the target responses learned by the DAE. By treating the full 2D target pressure fields as an input image, the proposed approach builds a network that designs the topology of wave scatterers and reconstructs the input pressure fields simultaneously. Once the network is trained, individual material units can be designed with different geometries, and the design phase is quasi-instantaneous. The proposed multi-objective design methodology is applicable to numerous state of the art problems including but not limited to holographic tweezers [403]and acoustic trap displays [404].

The rest of the paper is organized as follows. Section 6.2 describes the inverse design problems interested in this study. Section 6.3 discusses the proposed methodology as well as the details of dataset generation and network training. Section 6.4 reports the results and the associated discussions. The concluding remarks and future works are addressed in Section 6.5.

6.2 Problem Description

6.2.1 Acoustic Wave Scattering

Consider an acoustic source generating a plane wave that passes through a set of wave scatterers, the objective is to design the topological structure of the scatterers which leads to the user-defined downstream pressure fields. The target pressure fields can be determined by the users based on the application of interest such as wave focusing and cloaking. This falls into the domain of inverse design problems, in which the problem involves a forward simulation and an inverse inference, as indicated in Figure 6.1. The forward process takes a set of parameters such as material properties, geometry or boundary conditions as input, and generates the output responses using a mathematical model that typically is a finite element simulation. Given a set of target responses such as vibration control, wave guide or impact reduction, the corresponding input parameters can be achieved by solving the backward inference.

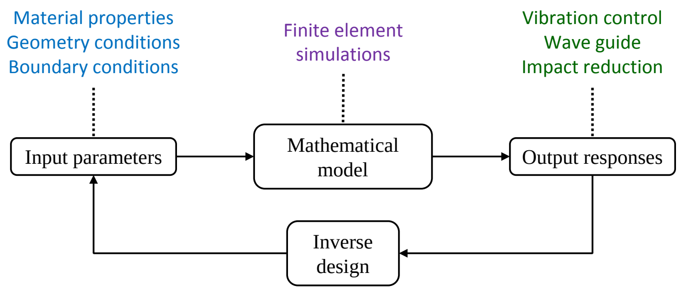


Fig. 6.1.: Inverse design problems.

Figure 6.2 shows two samples demonstrating the resulting downstream pressure fields after the incident acoustic wave is scattered by the scatterers. The topological structure of the material (i.e incident wave propagation medium) is defined as the geometry of the wave scatterers. Given the user-defined target downstream pressure fields, indicated by the dashed line region, the objective is to design the corresponding shapes of the scatterer units that lead to the target downstream responses. In this study the proposed DAE-based material design methodology is applied to three different design scenarios:

1. One scatterer with single frequency: This scenario considers one scatterer, and the acoustic source generates waves with one frequency. The design parameter is one weight factor that controls the scatterer shape varying from a diamond to a rectangle.

2. Quadruple scatterer with single frequency: This scenario increases the number of scatterer from one to four, and the acoustic source generates waves with one frequency. The design parameters are the geometric choices of each scatterer.

3. Quadruple scatterer with multiple frequencies: In this scenario, the acoustic source is allowed to generate waves with different frequencies. Therefore, this inverse problem is a multi-objective design task in which the geometry of the scatterers must lead to all the target downstream responses induced by the waves generated with different frequencies at the same time. The design parameters are the geometric choices of each scatterer.

These scenarios are designed to demonstrate the generalizability and robustness of the proposed approach. While Scenario 1 targets the inverse problem of single scattering, the complexity of the inverse design problem gradually increases with multiple scatterers and wave frequencies being considered in Scenario 2 and 3.

6.2.2 Dataset Generation

Scenario 1

In this scenario, an incident plane wave with wavelength λ_0 traveling in horizontal direction x, is scattered by the a single scatterer. The center of this scatterer is located at point $(0,0)$. The shapes are generated using non-uniform rational B-spline (NURBS) [405] (see Appendix for the detailed formulation of NURBS) with quadratic

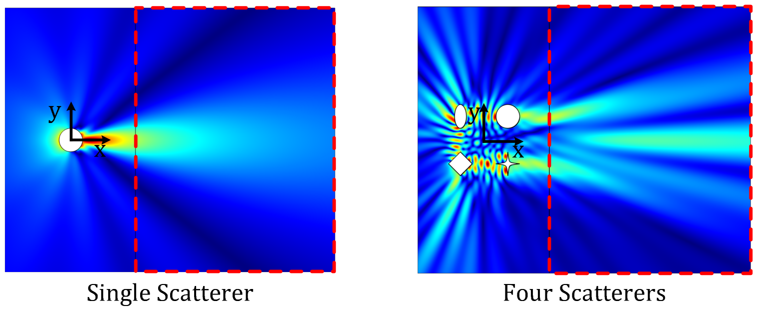


Fig. 6.2.: The inverse design problems considered in this study. Given the downstream pressure fields indicated in dash lines, the objective is to retrieve the geometry of the scatterers.

B-spline basis functions and knot vector $(0, 0, 0, 0.25, 0.25, 0.5, 0.5, .75, .75, 1, 1, 1)$. The coordinates of control points 1 to 9 (Figure 6.3) are $x = [1, 1, 0, -1, -1, -1, 0, 1, 1] \times \lambda_0$ and $y = [0, 1, 1, 1, 0, -1, -1, -1, 0] \times \lambda_0$, where the first and the last point have the same coordinates. The weights of the control points 1 to 9 are defined by the vector $[1, w, 1, w, 1, w, 1, w, 1]$. Hence, the weight factors of control points 1, 3, 5, 7, and 9 are fixed and set to 1. The value of w, which is the weight factor corresponding to control points 2, 4, 6, and 8, ranges from 0 to $\frac{6}{\sqrt{2}}$ and the associated shape gradually varies from a diamond (Figure 6.3a), obtained by $w = 0$, to square which is obtained by setting $w = \frac{6}{\sqrt{2}} = 4.2426$.

The scatterer length scale l_0 is defined as the distance between the shape center point $(x, y) = (0, 0)$ and the control point 3. This control point is fixed and consequently all of the shapes in this scenario have the same length scale equal to incident wave wavelength, i.e $l_0 = \lambda_0$. It should be noted that, the parameter that controls the scattering regime is the dimensionless size parameter β defined by:

$$\beta = l_0 k = l_0 \frac{2\pi f}{c} = l_0 \frac{2\pi}{\lambda_0} \tag{6.1}$$

where k is the wavenumber, f is the frequency, and c is the speed of the incident wave. Since in Scenario 1 it is assumed that $l_0 = \lambda_0$, for any given incident wave speed and frequency the value of β is fixed and equal to:

$$\beta = l_0 \frac{2\pi}{\lambda_0} = \lambda_0 \frac{2\pi}{\lambda_0} = 2\pi \tag{6.2}$$

Thus, in this scenario the scattering regime is the same for all the scatterer shapes. The values of λ_0 and l_0 are determined based on an arbitrary incident wave propagation domain properties and frequency. Without loss of generality, in this study it is assumed that the incident wave is traveling with 5000 (Hz) frequency in air medium in which the wave speed and density are 343.21 (m/s) and 1.24 (kg/m^3), respectively. Using the assumed frequency and wave speed, the wavelength is calculated as $\lambda_0 = l_0 = 0.0686$ (m).

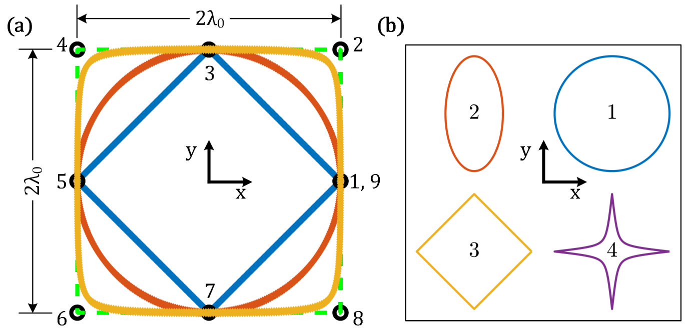


Fig. 6.3.: (a) Samples of scatterer shapes (solid line curves). Circular markers show the NURBS control points and are connected with the dashed line. In Scenario 1, the weight factor of control points 2,4,6 and 8 are changed to vary the scatterer shape from diamond to square with round corners. (b) Shapes used in scenarios 2 and 3 at scatterer locations 1 to 4. The scatterer geometry shown in this figure corresponds to a configuration labeled as [1,2,3,4].

To generate the dataset that consists of different scatterer shapes, the weight factor $w \in [0, 4.2426]$ is uniformly divided into 42,000 steps and the corresponding shapes are generated via NURBS approach. For each scatterer shape, the wave scattering is simulated with the finite element method (FEM) in COMSOL Multiphysics software. The real and imaginary parts of the downstream scattered wave field in the window $5.5\lambda_0 \leq x \leq 22.5\lambda_0$ and $-11.25\lambda_0 \leq y \leq 11.25\lambda_0$ (illustrated by the dashed lines in Figure 6.2) are saved using a 2D grid of size 210×270. Note that the selection of grid size depends on the wavelength. In this study, the grid size is chosen to ensure that at least 10 data points per incoming wave wavelength in each x and y directions exist to avoid any aliasing. Same logic applies to the data recording in Scenario 3 when considering multiple wave frequencies. The training dataset consists of 85% randomly selected samples from the aforementioned dataset, and the remaining 15% samples are used for testing.

Scenario 2

In this scenario, four scatterers exist in the field at locations 1 to 4 shown in Figure 6.3(b). Discrete choices of scatterer geometry at each location are defined using four basic shapes and three scaling factors. The basic scatterer shapes (i.e., circle, diamond, ellipse and star) are plotted in Figure 6.3(b), where the circle and diamond are obtained by using $w = \frac{1}{\sqrt{2}}$ and $w = 0$ in the shape configuration described in Scenario 1. The ellipse is defined by horizontally scaling the circle by a factor of 0.5. The star is obtained using the knot vector defined in scenario 1 using the control points $x = [1, 0.1, 0, -0.1, -1, -0.1, 0, 0.1, 1] \times \lambda_0$, $y = [0, 0.1, 1, 0.1, 0, -0.1, -1, -0.1, 0] \times \lambda_0$, and $w = \frac{1}{\sqrt{2}}$. All the basic objects have length scale λ_0 and their centers are located at a distance of $4\lambda_0$ from the origin in both x and y directions.

In this scenario, three scale factors $s_0 = 1.0$, $s_1 = 1.25$, and $s_2 = 0.8$ are used to scale the basic scatterer shapes, leading to 12 different shapes. By incorporating the option of no scatterer, a total of 13 choices, labeled from 1 to 13, can be placed at each scatterer location. The definition of labels and their corresponding shapes are presented in Table 6.1. The wave scattering is simulated in COMSOL Multiphysics using the domain parameters defined in Scenario 1. Also, the dataset is recorded with the same scheme in Scenario 1. 85% of the total $13^4 = 28,561$ scatterer geometry configurations are used for network training, and the rest 15% are employed for testing.

Table 6.1.: Shape labels defined in Scenario 2 and 3 and their corresponding scale factors. Label 5 refers to the condition in which no object exists at a scatterer location.

Scale factor	1.0	1.25	0.8
Circle	1	6	10
Ellipse	2	7	11
Diamond	3	8	12
Star	4	9	13

Scenario 3

In this scenario, all the conditions are the same as Scenario 2, except that the incident wave frequency is allowed to have four different choices, leading to a multi-objective inverse design problem. The frequencies are selected to cover from ultra-low to ultra-high frequency ranges. Using $l_0 = 0.0686$, that is the length scale of basic shapes with unity scale factor, the size parameter β of the four frequencies are set as $\beta = 0.2513$ (ultra-low), $\beta = 1$, $\beta = 2\pi$, and $\beta = 5\pi$ (ultra-high). Using air properties as the wave propagation medium, the corresponding incident wave frequencies are 200 (Hz), 796 (Hz), 5000 (Hz), and 12500 (Hz), respectively. The wave scattering is simulated for all the 13^4 scatterer configurations at all the four frequencies to generate the dataset. The simulation procedure and the recording of downstream pressure fields are identical to the previous scenario. Note that the scattered field distributions shown in Figure 6.4 are significantly different among the four frequencies. These variations in the pressure fields highly increase the complexity of the material design problem. It is worth mentioning that these four frequencies are selected to cover both low and high values of length scales for the purpose of demonstration. The proposed approach in this study applies to any wave frequency (i.e., size parameter β) combinations for other applications. Table 6.2 reports the numbers of training, testing and total samples in design scenarios 1, 2 and 3.

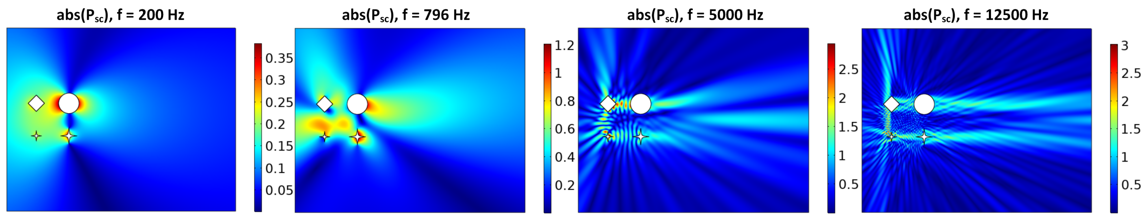


Fig. 6.4.: The amplitudes of the scattered pressure fields at the four frequencies in Scenario 3 for a scatterer configuration.

Table 6.2.: The number of training, testing, and total samples in each design scenario.

Scenario	Training	Testing	Total
1	35,700	6,300	42,000
2	24,277	4,284	28,561
3	24,277	4,284	28,561

6.3 Methodology

6.3.1 The Proposed Approach

Given a target downstream pressure field to estimate the corresponding geometry of the scatterers, the objective is to train an inverse model that consists of a geometry estimator and a DAE. As shown in Figure 6.5, the DAE consists of an encoder and a decoder, and it learns to reconstruct the input target pressure field [398]. The geometry estimator and the decoder share the latent representations produced by the encoder, and all the three components are optimized jointly during training. Denote

$\boldsymbol{W_e}$, $\boldsymbol{W_g}$ and $\boldsymbol{W_d}$ as the parameter sets of the encoder, the geometry estimator and the decoder, respectively. The training is achieved by solving the following equation:

$$\underset{\boldsymbol{W_e},\boldsymbol{W_g},\boldsymbol{W_d}}{\arg\min}\ J_1\left(\hat{y}_s(\boldsymbol{W_e},\boldsymbol{W_g},y_p),y_s\right)+\alpha\cdot J_2\left(\hat{y}_p(\boldsymbol{W_e},\boldsymbol{W_d},y_p),y_p\right) \tag{6.3}$$

where $\hat{y}_s$ denotes the predicted scatterer geometry that is a function of $\boldsymbol{W_e}$ and $\boldsymbol{W_g}$, y_s is the true value of the scatterer geometry, $\hat{y}_p$ denotes the reconstructed downstream pressure fields that is a function of $\boldsymbol{W_e}$ and $\boldsymbol{W_d}$, y_p is the input downstream pressure field, $J_1(\cdot)$ and $J_2(\cdot)$ are the loss functions used for the geometry estimator and the decoder, respectively, and α is a hyperparameter that controls the weighting between $J_1(\cdot)$ and $J_2(\cdot)$ during joint training. It is worth mentioning that the training of such a delicate network requires a thoughtful implementation in order to optimize the geometry estimator and the DAE jointly. The encoder and decoder are implemented using fully-convolutional network (FCN) [406, 407], and the geometry estimator is implemented with fully-connected neural networks [12].

Compared to training a network with the encoder and the geometry estimator alone, the proposed hybrid network learns the inverse inference with the reinforcement of the encoder that generates the latent representations able to produce the forward inference of the downstream pressure fields. In other words, the training of the geometry estimator is guided by the constraints provided from the DAE. These constraints are imposed based on the physics of the target pressure field. Although the proposed approach is validated through the material design with acoustic waves, it can be applied to the broad domain of other inverse design problems. The details of network architecture and training are provided in the next section.

6.3.2 Network Training

Network input and output

A series of preliminary experiments have indicated that using purely the amplitude of the downstream pressure fields as the input for the proposed hybrid network fails

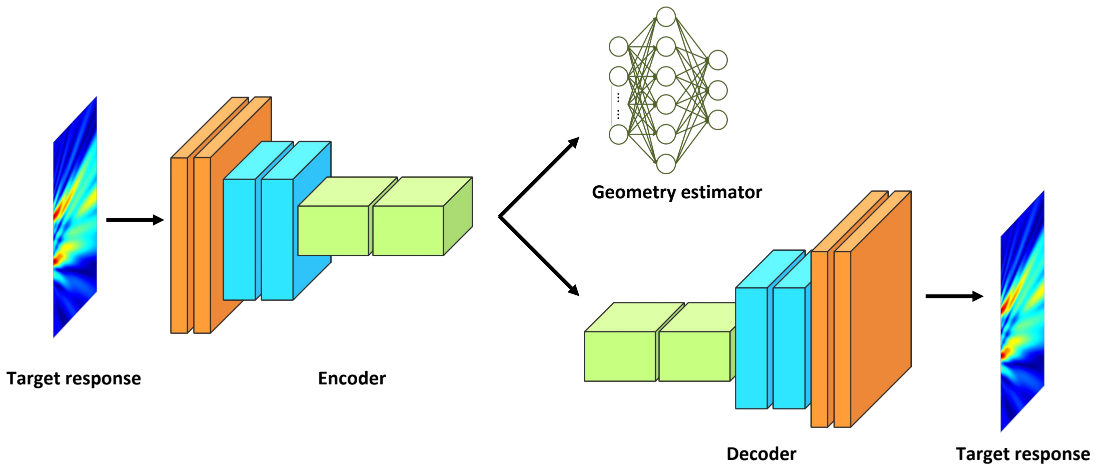


Fig. 6.5.: The proposed DAE-based approach trains the encoder, the decoder and the geometry estimator jointly using the loss values computed from true labels of scatterer geometries and input target responses.

to converge during training. Therefore, the inputs to the network are the real and imaginary part of the pressure fields, which contain the information of both phase and amplitude. Prior to training, the real and imaginary parts are scaled linearly to $[0, 1]$ according to their maximum and minimum values among all the pressure fields. For Scenario 3, the pressure fields from the same wave frequency are scaled to $[0, 1]$ according to their maximum and minimum values in the associated frequency. The resulting inputs to the network are two image channels of size $270 \times 210 \times 2$ for design scenarios 1 and 2. In design scenario 3, the inputs are eight image channels with size $270 \times 210 \times 8$ corresponding to 200 (Hz), 796 (Hz), 5000 (Hz) and 12500 (Hz) wave frequencies. The network output in design scenario 1 is a weight factor that controls the shape of the single scatterer ranging from 0 to 4.2426. Note that the weight factor is scaled to $[0, 1]$ before training. In the testing phase, the network output is scaled back accordingly for error quantification. For design scenarios 2 and 3, the geometry estimator produces four categorical outputs that characterize the shapes of the four scatterers. The dimension of the decoder output is identical to the input dimension since the decoder attempts to reconstruct the input.

Network architecture

Table 6.3 reports the FCN architectures of the encoder and the decoder used in this study. The encoder consists of five convolution layers and two pooling layers. Accordingly, the decoder has five convolution transpose layers and two upsampling layers. The convolution transpose layer applies an approximate deconvolution operation, and the upsampling layer uses bilinear interpolation to enlarge the feature maps by a factor of 2 as opposed to the pooling operation. Each convolution or convolution transpose layer is followed by the nonlinear *ReLU* activation layer. Depending on the number of image channels of the target inputs, the number of kernel in the last layer of the decoder is adjusted appropriately to reconstruct the inputs. In Scenario 3, the architectures of the encoder and the decoder are slightly tailored to account for the large variations in the pressure fields generated with different wave frequencies. Supported by a series of heuristic experiments, the number of kernels in Layer 1 of the encoder and Layer 6 of the decoder has been increased from 64 to 256, allowing a larger capacity for network learning.

The architectures of the geometry estimator are dependent on the geometric parameters to be estimated. For Scenario 1, the geometry estimator is a fully-connected neural network that has two hidden layers with 70 and 60 hidden nodes in each hidden layer. The output layer is one node followed by the Sigmoid activation function to estimate the weight factor. In Scenario 2 and 3, the geometry estimator has four fully-connected neural networks that characterize the shape of the four scatterers separately. Each fully-connected neural network consists of two hidden layers with 18 and 15 hidden nodes, and the output layer has 13 nodes to estimate the probability of each class out of the 13 choices of shapes as shown in Table 6.1.

To train the whole network jointly, it is noted that the hyperparameter α, shown in Eq. (6.3), must be selected appropriately to control the weighting between the loss values contributed from the geometry estimator and the decoder. For Scenario 1, the loss functions $J_1(\cdot)$ and $J_2(\cdot)$ are the same and the mean square error (MSE) function is used to compute the loss. For Scenario 2 and 3, the loss function of the geometry estimator $J_1(\cdot)$ is the cross entropy loss function while the loss function of the decoder $J_2(\cdot)$ is the MSE function. To determine the α value for each design scenario, a series of experiments have been conducted by tuning α with a factor of 10 to ensure the convergence of both the geometry estimator and the decoder. The α value is set to 100 for Scenario 1 and 20 for Scenario 2 and 3. During training, the stochastic gradient descent algorithm [308] is used to optimize the parameters of the network with a learning rate 0.0001 and momentum 0.9. The training proceeds until the loss values converge. The proposed network is implemented in Python 2 using PyTorch [309] version 0.2 with CUDA 8.0, cuDNN 6.0.21, and Ubuntu 16.04. The NVIDIA Titan X GPU, which has 3584 CUDA cores at a base clock rate of 1417 MHz and 12 GB GDDR5X memory, is used for network training.

Table 6.3.: The network architectures of the encoder and decoder. (Conv.: convolution; Conv. Trans.: convolution transpose.)

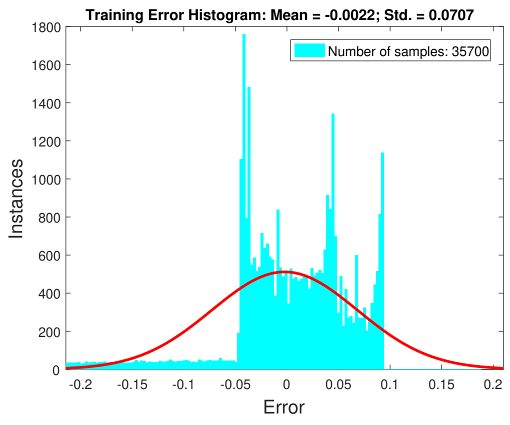
	Encoder			Decoder		
Layer #	Layer	Kernel size	# of kernel	Layer	Kernel size	# of kernel
1	Conv.	3×3	64	Conv. Trans.	3×3	16
2	Conv.	3×3	64	Upsampling	-	-
3	Pooling	2×2	-	Conv. Trans.	3×3	32
4	Conv.	3×3	32	Conv. Trans.	3×3	64
5	Conv.	3×3	16	Upsampling	-	-
6	Pooling	2×2	-	Conv. Trans.	3×3	64
7	Conv.	3×3	8	Conv. Trans.	3×3	2 or 8

6.4 Results and Discussions

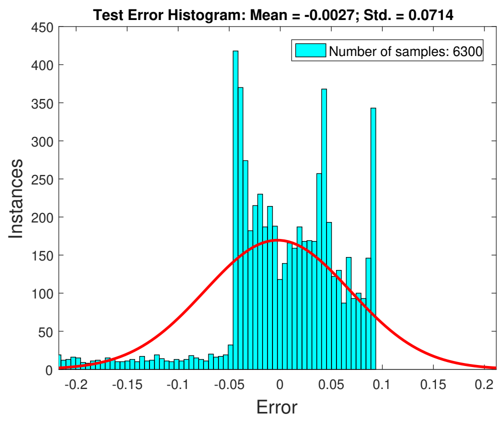
6.4.1 Scenario 1: One Scatterer with Single Frequency

Figure 6.6 shows the histogram of the errors between the predicted weight factors and the target weight factors for the training and testing dataset. The mean and standard deviation of the training errors are -0.0022 and 0.0707, respectively. For testing errors, the mean and standard deviation are -0.0027 and 0.0714, respectively. The similarity between the training and testing performance indicates that the network learns to generalize well on the unseen data. Another interesting phenomenon observed in Figure 6.6 is that both the training and testing error histograms have a small flat region in the left tail of the distribution. This means that the network has a slight tendency to underestimate the weight factor since the shapes of the scatterer vary much more when being generated with weight factors in $[0, 2.8425]$ than in $[2.8425, 4.2426]$, as shown in Figure 6.7. Given the weight factor ranges from 0 to 4.2426, these error statistics suggest that more than 99.7% of the predictions from the network achieve an estimation error within $\pm 3 \times 0.07 \pm 0.21$, since the error distributions are more centered compared to the normal distributions shown with solid lines. The normalized root mean square errors (RMSE) for training and testing data are 1.67% and 1.68%, respectively. These results confirm the outstanding performance of the proposed approach in solving the inverse design.

Figure 6.8 depicts an example of the network inputs and the corresponding reconstructed outputs by the proposed DAE. The reconstructed real and imaginary parts of the downstream pressure fields match well with the network inputs. Using the real and imaginary parts, the amplitudes of the pressure fields are computed and shown in Figure 6.9, demonstrating a decent input field reconstruction performance achieved by the DAE.



(a)



(b)

Fig. 6.6.: Error histograms of (a) training dataset, and (b) testing dataset for estimating the weight factors that control the shape of the scatterer in Scenario 1. (Std.: standard deviation.). The solid line is the fitted normal distribution.

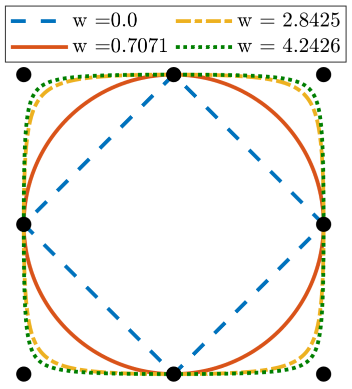


Fig. 6.7.: Variations in the scatterer shape versus the weight factor w in Scenario 1. The markers shows the NURBS control points. For $w \geqslant 2.8425$, the scatterer shapes change much less than the shapes generated in the range $0 \leqslant w < 2.8425$.

6.4.2 Scenario 2: Quadruple Scatterer with Single Frequency

In Scenario 2, the proposed approach predicts the shapes of four scatterers given a target downstream pressure field. To evaluate the performance of the network, a successful prediction is considered only when all the shapes of the four scatterers

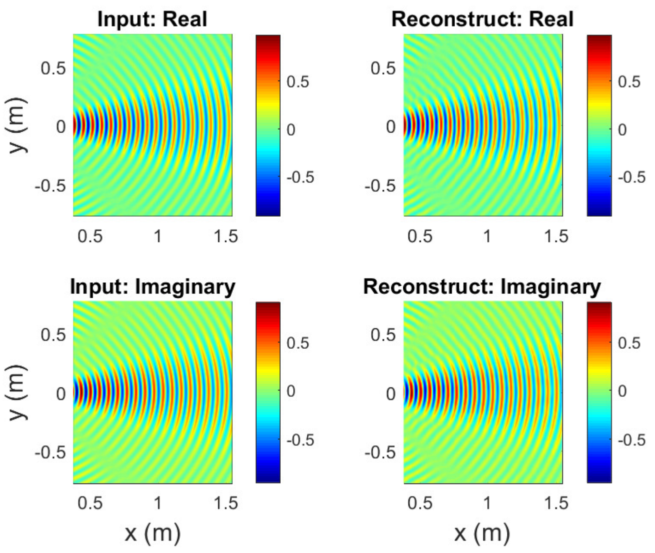


Fig. 6.8.: Real and imaginary parts of a sample network input and the corresponding reconstructed real and imaginary parts of the input pressure field in Scenario 1. (Unit: Pa)

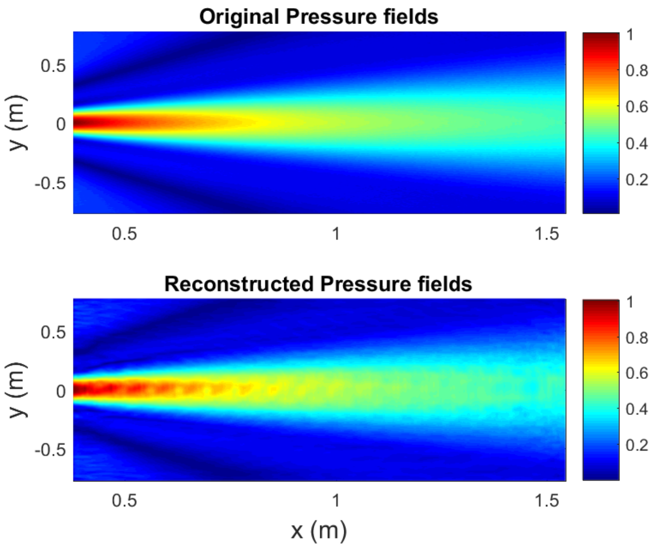


Fig. 6.9.: An sample of the input pressure field amplitude and its corresponding reconstructed pressure field amplitude in Scenario 1. (Unit: Pa)

are predicted correctly by the network. In other words, even if only one of the predicted scatterer shapes is predicted incorrectly, the prediction would be treated as a failed design. Figure 6.10 shows the network inputs and the reconstructed outputs obtained from the proposed DAE for a test sample. The reconstructed real and imaginary parts of the pressure fields match well with the target inputs, and the corresponding amplitudes of the pressure fields shown in Figure 6.11 also demonstrate a high similarity between the original and the reconstructed pressure fields. Note that the network is trained to reconstruct the real and imaginary parts of the pressure field, not its amplitude. A small phase inconsistency between the two fields introduces perturbations into the pressure amplitude fields. The proposed network achieves 99.1% and 95.5% accuracy in 24,277 training samples and 4,284 testing samples, respectively.

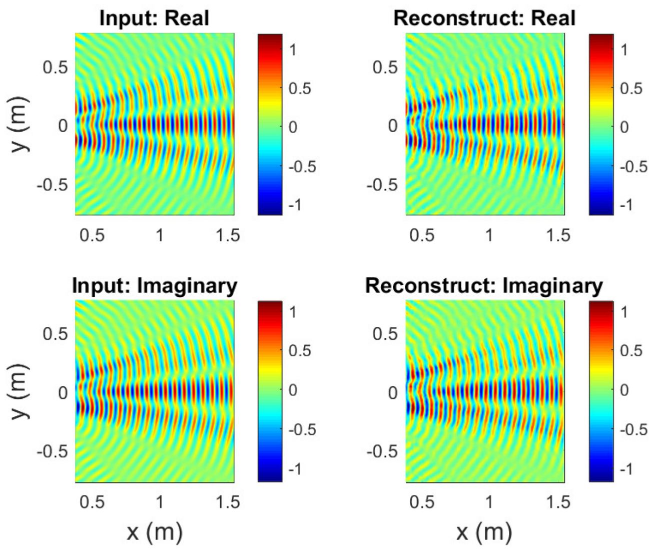


Fig. 6.10.: Real and imaginary parts of a sample network input and the corresponding reconstructed real and imaginary parts of the input pressure field in Scenario 2. (Unit: Pa)

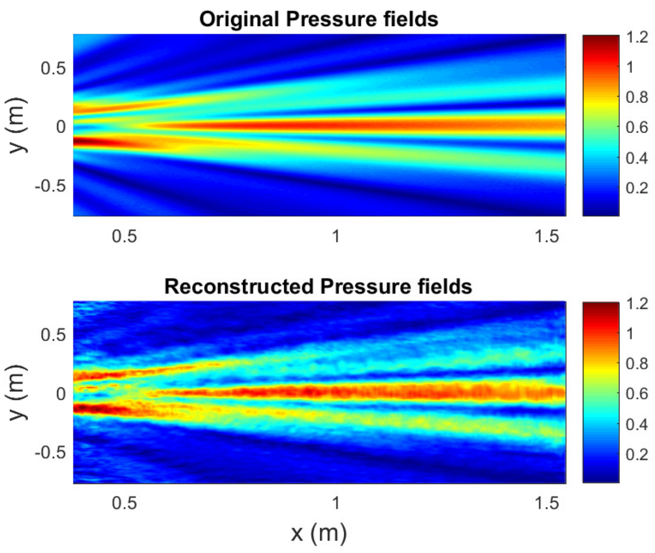


Fig. 6.11.: An example of the amplitude of input pressure field and the reconstructed pressure field from a test sample in Scenario 2. (Unit: Pa)

6.4.3 Scenario 3: Quadruple Scatterer with Multiple Frequencies

In this section, a multi-objective inverse design problem is considered by predicting the shapes of the four scatterers that satisfy the target downstream responses generated with four different wave frequencies (i.e., 200 (Hz), 796 (Hz), 5000 (Hz) and 12500 (Hz)) all at once. Figure 6.12 shows the network inputs and the reconstructed outputs from the DAE using a test sample. The reconstructed real and imaginary parts of the pressure field match well with the inputs even though the patterns of data vary significantly from ultra-low to ultra-high wave frequencies. Figure 6.13 presents a respectable result of the reconstructed amplitudes of the pressure fields. For all the four frequencies, the DAE captures the high amplitude signatures quite well. However, small disturbances exist in low amplitude regions due to minor errors in the reconstructed real and imaginary parts of the pressure field. Using the same evaluation method described in Scenario 2, the proposed network achieves 100% and 99.9% accuracy in 24,277 training and 4,284 testing samples, respectively. Notice that the proposed approach achieves higher accuracy in Scenario 3 than Scenario 2 since the network received more information contributed from more than one wave frequency during training. In terms of costs in computation time after training, it takes approximately 0.01 (hr) for the network to finish 30 design queries in CPU mode. However, the genetic algorithm (GA), which is commonly used in conventional design approaches, spends approximately 1.95 (hr) to finish the same 30 design queries. Compared to GA, the proposed approach reduces the computation time by a factor of 195 in design stage. For both GA and the proposed network, the computation time is reported using a Intel Xeon processor E52620 CPU, 2.1 GHz with 16 GB RAM. The maximum value of generations and the population size for GA are both set to 200 according to the suggestions provided in [358]. The results demonstrate an outstanding performance of the network in predicting the topology of the wave scatterers accurately and quasi-instantaneously.

6.4.4 Utilization of the Trained Network in Material Design Problem

In this section, application of the trained networks for material design problems are discussed. Although the inverse design is demonstrated using only the network trained in Scenario 3, same procedure applies when conducting material design in other scenarios.

In the design phase, suppose a user attempts to design the shapes of the scatterers that satisfy multiple given target responses, such as the multi-objective problem considered in Scenario 3. It is true that the user may or may not be aware of whether the multiple target responses are resulted from a consistent scatterer configuration or not. For instance, the desirable four target responses at different frequencies may come from different scatterer configurations, e.g., $[1, 2, 3, 4]$, $[5, 6, 7, 8]$, $[9, 10, 11, 12]$, and $[13, 1, 2, 3]$ in Scenario 3. Such a case is then defined as "no physics-informed user demands" since a consistent solution to the scatterer configuration might be physically impossible. The proposed network, which is trained with consistent scatterer configurations, is therefore established with target demands that are physics-informed. The

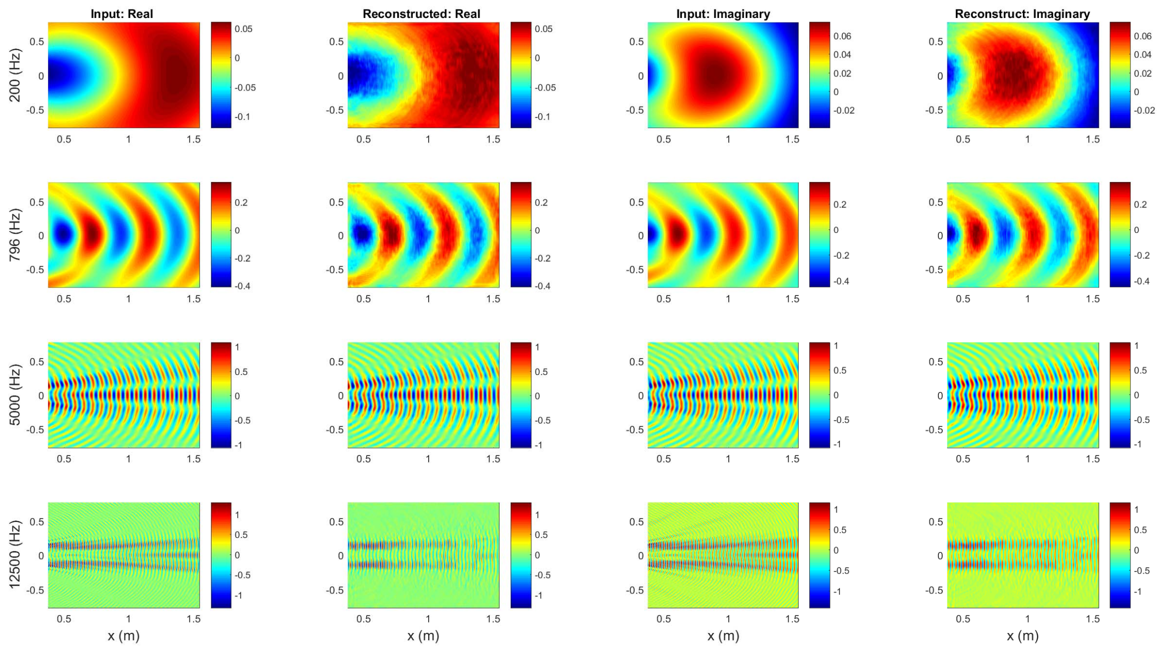


Fig. 6.12.: Real and imaginary parts of a sample network input and the corresponding reconstructed real and imaginary parts of the input pressure field in Scenario 3. The first and the third column are the real and imaginary parts of the input, respectively. The second and the fourth column are the reconstructed real and imaginary parts of the input, respectively. (Unit: Pa)

design capability of the proposed approach is demonstrated with these two types of user demands by considering Scenario 3, which is the most sophisticated case, in the next two sections.

Design with physics-informed user demands

To further evaluate how the proposed approach performs on the unseen data, a new test dataset is created by varying the scale factors described in Section 6.2.2 to generate new samples. Two scale factors are changed as $s_1 = 1.11$ and $s_2 = 0.9$, resulting in eight new choices of shapes out of the original 13 choices. Using these new shapes, a total of 1,000 target downstream pressure fields that have consistent

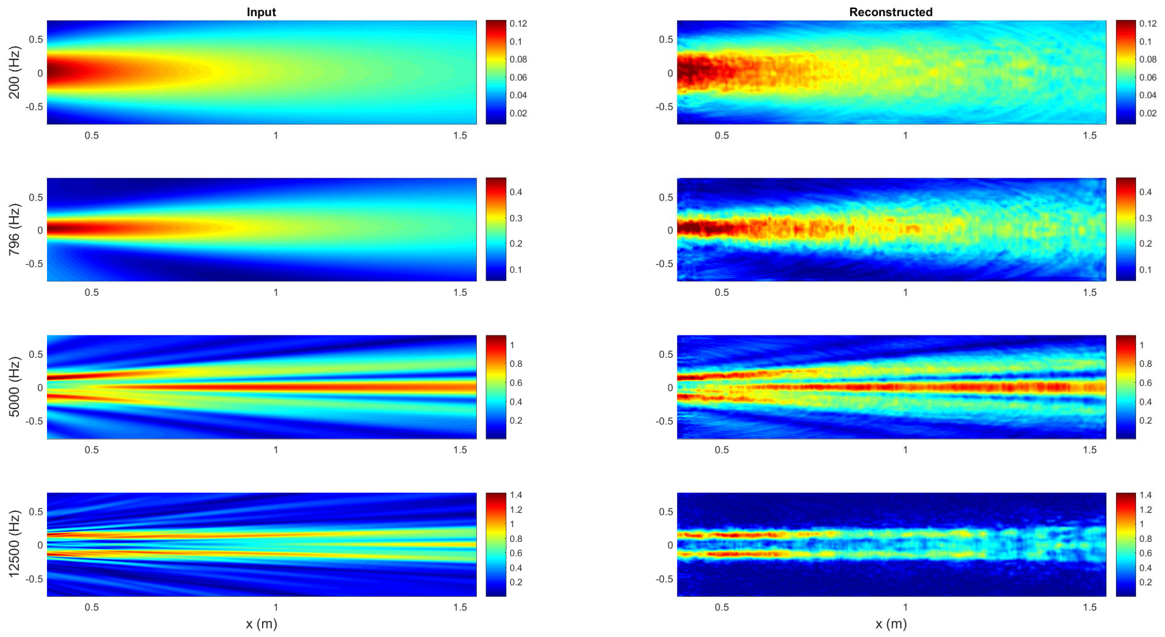
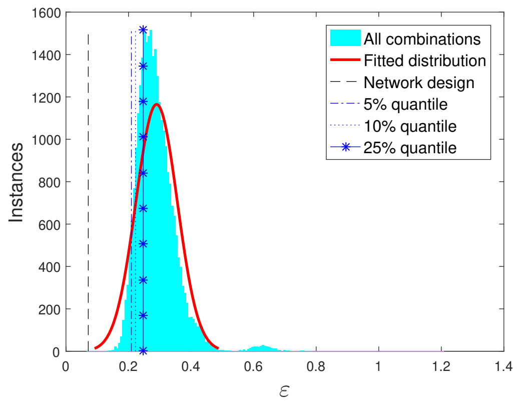


Fig. 6.13.: An example of the amplitudes of the input pressure field and the reconstructed pressure field from a test sample in Scenario 3. (Unit: Pa)

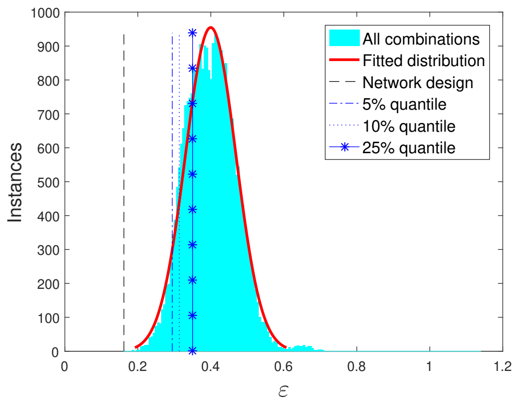
solutions to the scatterer configuration are employed to evaluate the established network discussed in Section 6.4.3. Noticing that the network is trained to predict the original choices of shapes, it is anticipated that the scatterer configuration returned by the network will be a near-optimal solution. This is analogous to real-world design examples in which a factory only has a fixed choices of molds and the manufacturer attempts to find the best choice to approximate the responses resulted from an unknown mold. Using the predicted scatterer configuration from the network to generate the downstream pressure fields, the prediction error ε is computed by summing up the RMSE between the predicted and the target amplitudes of pressure fields at the four frequencies. The confidence of the network performance is quantified by comparing with the exhaustive search, or equivalently, the prediction errors calculated using the total 28,561 combinations of the original scatterer configurations . Figure 6.14 shows four samples of the error distributions computed with exhaustive search in four design cases. The solid line indicates the fitted normal distribution. In Figure 6.14(a) and 6.14(b), the network returns the optimal scatterer configuration among all the 28,561 choices, leading to a minimum prediction error for the pressure fields. In Figure 6.14(c) and 6.14(d), two mediocre designs and the corresponding prediction errors are illustrated with errors located in between the 10% and 25% error quantiles. By locating the prediction error obtained from the network design in the distribution, the confidence of the proposed network is evaluated by measuring how many times the network leads to an error that is smaller than the 5%, 10% and 25% error quantiles in all design cases. Among the 1,000 design cases, there are 991, 998 and 1,000 times that the network achieves a prediction error less than the 5%, 10% and 25% quantiles, respectively. This implies that there is a 99.1%, 99.8% and 100% probability that the prediction errors from the network are smaller than the 5%, 10% and 25% error quantiles, respectively, when the network deals with new and unseen datasets. This demonstrates the generalization capability of the proposed approach when the user demands are physics-informed.

Design with no physics-informed user demands

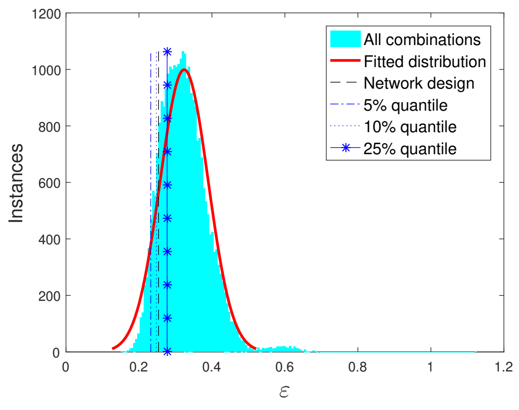
In this section, target responses originated from inconsistent scatterer configurations are used to test the established network in Scenario 3. Although there exists no physically possible solution to perfectly match with the target responses, the proposed network could still manage to return a scatterer configuration that leads to superior approximation in the predicted pressure fields. The target downstream pressure field at each frequency is generated by using a randomly selected scatterer configuration out of the total 28,561 combinations. Therefore, the target pressure fields at the four frequencies correspond to four different scatterer configurations. Using the same evaluation method described in the previous section, the prediction errors from the network in 1,000 design cases are compared with the exhaustive search over the total 28,561 scatterer configurations. Among the 1,000 design cases, there are 345, 519 and 779 times that the network achieves prediction errors below the 5%, 10% and



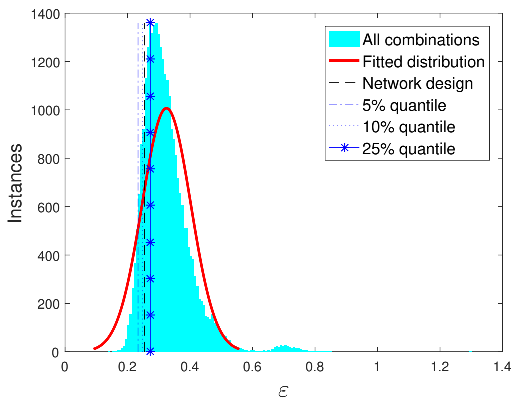
(a)



(b)



(c)



(d)

Fig. 6.14.: Error distributions computed using the total 28,561 original scatterer configurations for the four sample design cases: (a) and (b) are samples of scatterer configuration returned by the network that have led to minimum prediction error ε; (c) and (d) are samples of the designed scatterer configuration where the associated prediction error lies between 10% and 25% error quantile.

25% error quantiles, respectively. This suggests that although the network is trained with the target responses generated with consistent scatterer configurations, there is a 34.5%, 51.9% and 77.9% probability that the prediction errors from the network are smaller than the 5%, 10% and 25% error quantiles, respectively, when receiving random targets as inputs. These evidences further demonstrate the robustness and flexibility of the proposed DAE-based approach.

6.4.5 Interpretation of the Trained Network

The interpretation of deep neural networks has been recognized as a challenging task [279]. This section provides an insight into the functionality of the convolution layers after the training is finished. Compared to the early layers in the network, the deeper layers usually generate more abstract representations that are difficult to interpret. Therefore, the first layer of the encoder and the last layer of the decoder are selected to be investigated since the inputs of the former and the outputs of the latter are known. In general, all the layers in the encoder and the decoder operate with the principle of convolution:

$$\begin{aligned} F_i(x,y) &= \sum_{j=1}^{\mathrm{N_I}} K_{ij}(u,v) * I_j(x,y) \\ &= \sum_{j=1}^{\mathrm{N_I}} \sum_{u=-n}^{n} \sum_{v=-n}^{n} K_{ij}(u,v) I_j(x-u,y-v) \end{aligned} \tag{6.4}$$

where $F_i(x,y)$ is the i^{th} feature map generated from the layer, $\mathrm{N_I}$ is the number of input channels of the layer, $I_j(x,y)$ is the j^{th} channel of the inputs, $K_{ij}(u,v)$ is the j^{th} channel of the i^{th} kernel in the layer, x and y are the variables of the spatial coordinates, and n is a positive integer determined from the size of the kernel. Conceptually, each resulting feature map of the layer has an associated kernel. The feature map is obtained from the channel-wise convolution between the inputs and the kernel followed by a summation. For the first layer of the encoder, the kernels in the convolution layer are not flipped:

$$\begin{aligned} F_i(x,y) &= \sum_{j=1}^{\mathrm{N_I}} K_{ij}(-u,-v) * I_j(x,y) \\ &= \sum_{j=1}^{\mathrm{N_I}} \sum_{u=-n}^{n} \sum_{v=-n}^{n} K_{ij}(-u,-v) I_j(x-u,y-v) \end{aligned} \tag{6.5}$$

However, for the last layer of the decoder, the kernels in the convolution transpose layer are flipped and hence lead to a valid convolution operation given in Eq. (6.4). Using the network trained in Scenario 3, the interpretation of these two layers is discussed below based on Eq. (6.4) and (6.5).

As shown in Figure 6.12, the first layer of the encoder takes eight channels of inputs. A total of 256 feature maps are generated by 256 kernels, and each kernel has eight channels of size 3×3. Therefore, for this layer, $\mathrm{N_I} = 8$ and $n = 1$. According to Eq. (6.5), the frequency response of the i^{th} feature map, $\mathcal{F}\{F_i(x, y)\}$, can be computed using $\sum_{j=1}^{\mathrm{N_I}} \mathcal{F}\{K_{ij}(-u, -v)\}\mathcal{F}\{I_j(x, y)\}$. By investigating the frequency response of the input and the kernel channel by channel, the resulting patterns of the feature map can be inferred. Figure 6.15 shows the 43^{rd}, 81^{st} and 183^{rd} feature map generated from the first layer of the encoder using a test sample. These feature maps exhibit patterns very similar to the 796 (Hz), 5000 (Hz) and 12500 (Hz) frequency input (i.e., shown in Figure 6.10), respectively. This indicates that the associated kernels must extract primarily the pressure fields of a specific wave frequency. Figure 6.16 depicts the frequency responses of the eight channels of the input and the 43^{rd}, 81^{st} and the 183^{rd} kernel. The top and the bottom row of Figure 6.16(b) to 6.16(d) are associated with the real and imaginary input channels, respectively. The frequency spectrum is presented in wavenumber coordinates $\mathrm{k_x}$ and $\mathrm{k_y}$, with $\mathrm{k_x} = \mathrm{k_y} = \omega/c = 2\pi f/c$, where $c = 343.21$ (m/s) is the wave speed. Note that Figure 6.16(a) is plotted in log-amplitudes for better visualization. According to Figure 6.16(a) and 6.16(b), the 43^{rd} kernel mainly extracts the wavenumber signature possessed by the real part of the 796 (Hz) input, leading to a feature map that is dominated by the 796 (Hz) pressure fields. On the other hand, as shown in Figure 6.16(c) and 6.16(d), the 81^{st} and 183^{rd} kernel dominantly extract the real part of the 5000 (Hz) input and the imaginary part of the 12500 (Hz) input, respectively. This demonstrates that after training, in some cases, the kernels act as wavenumber band-pass filters to generate feature maps composed of various wavenumber patterns.

For the last layer of the decoder, it is expected that the layer outputs would be similar to the eight input channels of the encoder since the decoder is trained for reconstruction. In this layer, there are 256 input feature maps and eight kernels responsible for the resulting eight outputs. Each kernel has 256 channels of size 3×3. Therefore, for this layer, $\mathrm{N_I} = 256$ and $n = 1$. Figure 6.17 shows the frequency responses of the outputs that reconstruct the real part of the pressure fields for the four wave frequencies. It is noted that the exhibited wavenumber signatures are similar to the ones shown in the top row of Figure 6.16(a). To generate an output that contains a specific wavenumber signature, an intuitive assumption would be that every channel of the associated kernel learns to extract the particular wavenumber pattern of the corresponding wave frequency from the 256 input feature maps. However, this hypothesis turns out to be invalid after careful scrutiny. Instead, some channels of the kernel extract the patterns that do not belong to the associated wave frequency, and these irrelevant wavenumber patterns are cancelled out due to the coupling effects of phase difference in the summation process. This phenomenon is demonstrated via the frequency response of outputs computed with varying number of channels considered in the kernel. By rearranging the order of the channels of the kernel based on their maximum amplitudes in frequency spectrum, the output with varying number of dominant channels is computed using:

$$\hat{F}_i(x, y) = \sum_{j=1}^{\mathrm{N_d}} K_{ij}(u, v) * I_j(x, y) \tag{6.6}$$

where $\mathrm{N_d}$ is the number of dominant channels being considered, and $\hat{F}_i(x, y)$ is the associated intermediate output computed using the N_d dominant channels out of the total 256 channels of the kernel. It is noted that the computing of $\hat{F}_i(x, y)$ requires the retrieval of $K_{ij}(u, v)$ and $I_j(x, y)$ from the network, and $\hat{F}_i(x, y)$ should be identical to the decoder output when $\mathrm{N_d}$ is set to 256. Figure 6.18 illustrates the frequency responses of $\hat{F}_i(x, y)$ computed with the number of dominant channels varied at an increment of 32, using the kernel associated with the real part of 200 (Hz), 796 (Hz), 5000 (Hz) and 12500 (Hz) pressure fields, respectively. According to

Figure 6.18, the irrelevant wavenumber signatures are reduced and the corresponding wavenumber signature are more prominent when computing $\hat{F}_i(x, y)$ with increasing $\mathrm{N_d}$. The most visually recognizable one is the real part of the 12500 (Hz) pressure fields shown in Figure 6.18(d), where other low frequency wavenumber signatures are reducing and the 12500 (Hz) signature emerges more as $\mathrm{N_d}$ is increased. For all the four frequencies, the frequency response of $\hat{F}_i(x, y)$ becomes almost identical to the ones shown in Figure 6.17 when $N_d \geq 160$. As a result, after training, the kernels in the last layer of decoder learn to preserve the significant wave patterns corresponding to each wave frequency while eliminating the irrelevant wavenumber signatures. Although the discussion here focuses on the real part of the pressure fields, consistent observations are made for the outputs of imaginary parts as well.

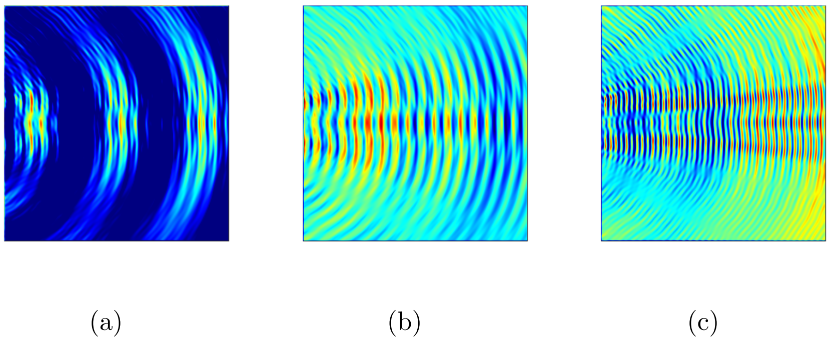
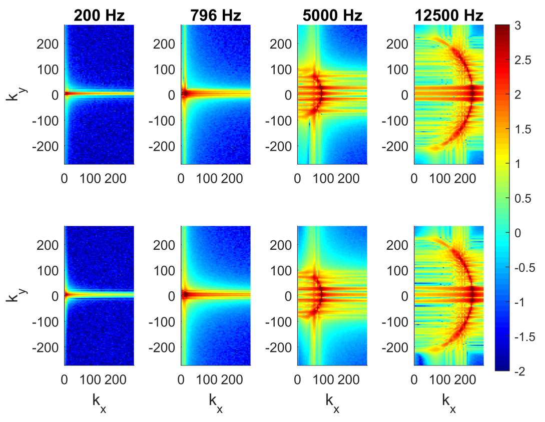
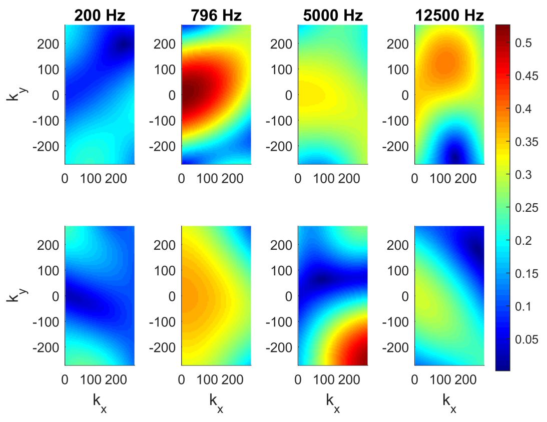


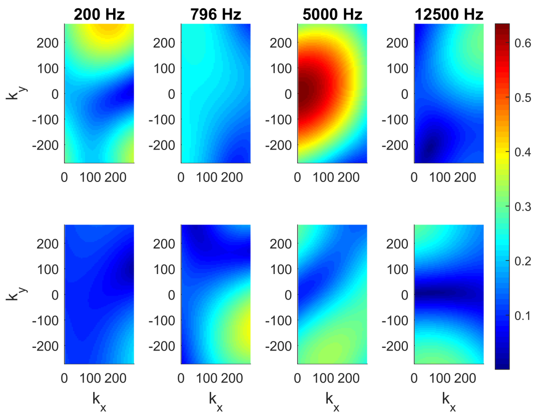
Fig. 6.15.: (a) The 43rd, (b) the 81st and (c) the 183rd feature map generated from the first layer of the encoder.



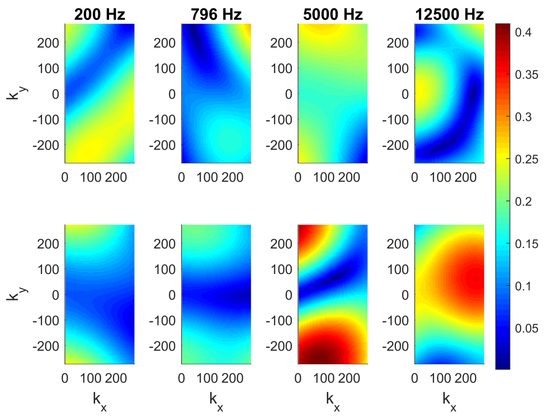
(a)



(b)



(c)



(d)

Fig. 6.16.: The frequency responses of the eight channels of (a) input, (b) the 43^{rd} kernel, (c) the 81^{st} kernel and (d) the 183^{rd} kernel, in the first layer of the encoder. The top and the bottom row of (b) to (d) are associated with the real and imaginary input channels shown in the top and the bottom row of (a), respectively. The frequency responses of the channels of the kernel indicate the dominant wavenumber patterns extracted from the input.

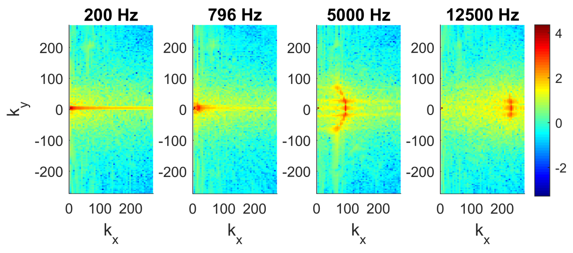


Fig. 6.17.: The frequency responses of the decoder outputs associated with the reconstructed real part of the pressure fields in the last layer of the decoder.

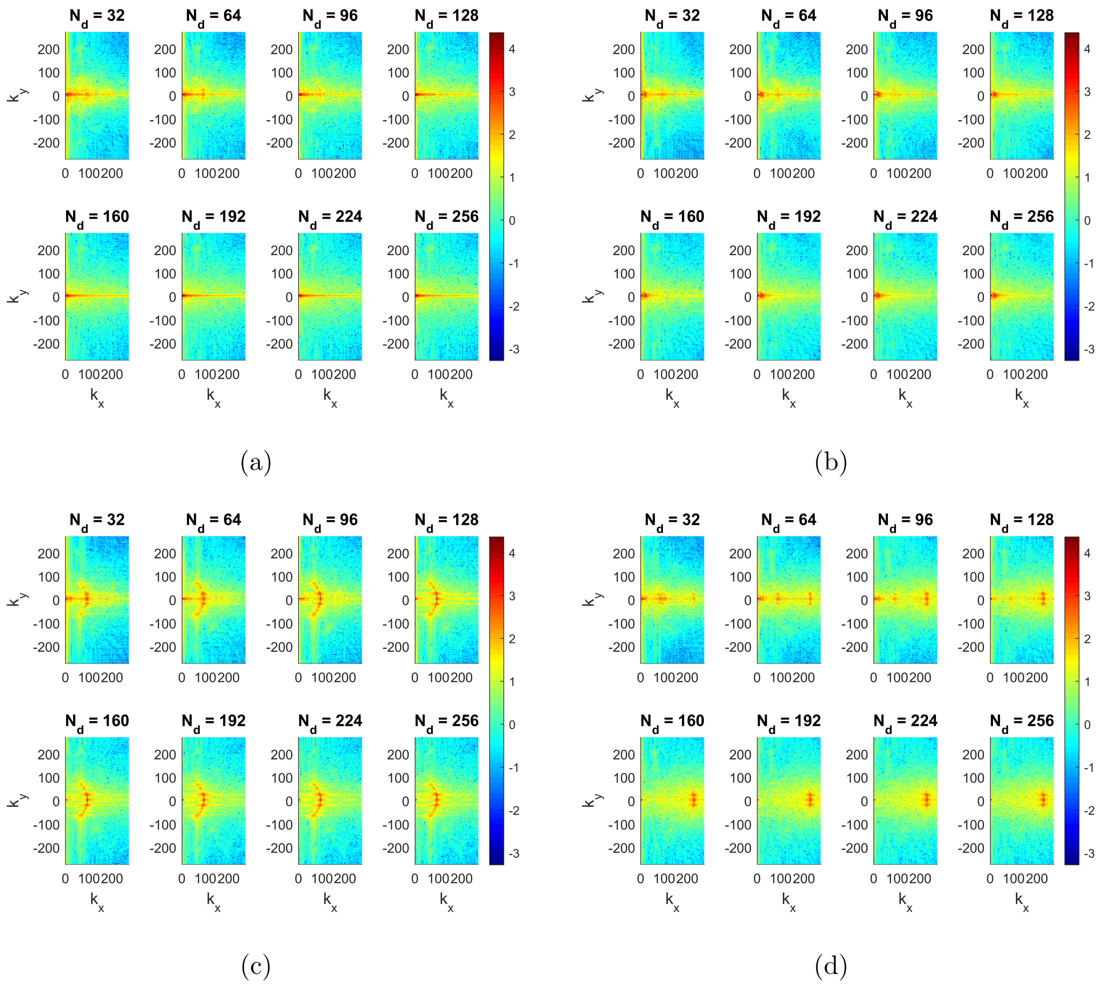


Fig. 6.18.: The frequency responses of $\hat{F}_i(x, y)$ computed with varying number of dominant channels being considered, using the kernel associated with the real part of (a) 200 (Hz), (b) 796 (Hz), (c) 5000 (Hz) and (d) 12500 (Hz) pressure fields.

6.5 Concluding Remarks

This study presents a physics-constrained DAE-based approach for multi-objective inverse design of acoustic wave scatterers that lead to the target downstream pressure fields. As the first demonstration of multi-objective inverse wave scattering application, the proposed approach is validated through three design scenarios with varying numbers of wave scatterers and wave frequencies. The proposed network succeeds in decent training and testing performance for all the three scenarios. To further evaluate how the network generalizes on the unseen data, a dataset generated with new shapes of wave scatterers is used to produce additional design cases. Compared with exhaustive search in the design space, the design confidence of the proposed approach is quantified with statistic measures, and an outstanding performance is achieved by the proposed network. Moreover, the proposed approach enables the user to retrieve the scatterer geometry even when the target demands have no physically consistent solutions. The flexibility and the efficiency of the proposed approach lead to a promising future in the community of acoustic wave scattering. Lastly, the proposed DAE-based approach can be extended to other research domains with little efforts. Future work will be dedicated to expanding the proposed approach with the ability to design arbitrary geometries of the wave scatterers.

Appendix A: NURBS Description:

A Non-uniform rational basis spline (NURBS) curve $C(t)$ is defined by

$$C(t) = \sum_{i=1}^{n} R_{i,q}\mathbf{P_i}; \quad R_{i,q} = \frac{\omega_i N_{i,q}(t)}{\sum_{i=1}^{n} \omega_i N_{i,q}(t)} \tag{6.7}$$

where ω_i are the weight factors, $\mathbf{P}_i \in \mathbf{P} = \{\mathbf{P}_0, \mathbf{P}_1, ..., \mathbf{P}_n\}$ are the control points with predefined (x, y) coordinates, q is the order, $R_{i,q}$ is called the rational basis function, and $N_{i,q}(t)$ is the B-spline basis function of order q given by

$$\begin{aligned} N_{i,0}(t) &= \begin{cases} 1 & t_i \leq t < t_{i+1} \text{ and } t_i < t_{i+1} \\ 0 & \text{otherwise} \end{cases} \\ N_{i,j}(t) &= \frac{t - t_i}{t_{i+j} - t_i} N_{i,j-1}(t) + \frac{t_i + j + 1 - t}{t_{i+j+1} - t_i + 1} N_{i+1,j-1}(t), \quad j = 0, 1, 2, ..., q \end{aligned} \tag{6.8}$$

where $t_i \in [0, 1]$ is the *i-th* member of a non-descending vector called knot vector $\mathbf{T} = \{t_0, t_1, ..., t_m\}$, and the order $q = m - n - 1$. In order to find the coordinates of N points that forms a specific shape, the range $[t_0, t_m]$ is divided into N points and substituted as t in Eq. 6.7.

7. SUMMARY AND FUTURE WORKS

7.1 Summary

This study proposes AI-based approaches for SHM applications and metamaterial design. In Chapter 2, a comprehensive review for the state-of-the-art data fusion and ML techniques are provided. The theoretical basis of each algorithm is reviewed, followed by applications in SHM discussed in aspects of data source, underlying assumptions, and limitations. Challenges of each algorithm are addressed, and a roadmap is provided for future research.

In Chapter 3, a deep CNN-based approach is proposed for structural dynamic response estimation and system identification. Three case studies including a linear SDOF system, a nonlinear SDOF system and a full-scale three-story MDOF steel frame, are considered. Various cases of noise-contaminated signals are used in this study, and the conventional MLP algorithm serves as a reference for the proposed CNN approach. According to both the results from numerical simulations and experimental data, the proposed CNN approach is able to predict the structural responses accurately, and it is more robust against noisy data compared to the MLP algorithm. Moreover, the physical interpretation of CNN model is discussed in the context of structural dynamics. It is demonstrated that in some special cases the convolution kernel has the capability of approximating the numerical integration operator, and the convolution layers attempt to extract the dominant frequency signature observed in the ideal target signal while eliminating the irrelevant information during the training process.

Chapter 4 introduces a vision-based damage detection approach based on network pruning and deep transfer learning for efficient inference in edge devices. Results from comprehensive experiments on two pre-trained networks (i.e., VGG16 and ResNet18) and two types of prevalent surface defects (i.e., crack and corrosion) are presented and discussed in details with respect to damage detection performance, memory demands, and the inference time for damage detection. The experiments show that by using the NVIDIA Jetson TX2 GPU, simulating the on-board computing platform, the approach achieves an inference time which is nine and four times faster than the original VGG16 and ResNet18 networks, respectively. Furthermore, the network size is reduced by 80% and 95% for the VGG16 and ResNet18 networks, respectively. Cross-validation and stopping criterion for pruning are addressed, and an optimization scheme is proposed for VGG16 feature extraction. Results demonstrate that the proposed approach significantly enhances resource efficiency without decreasing damage detection performance.

Chapter 5 proposes two design frameworks for periodic and non-periodic materials. Conventional material design processes are computationally intensive which severely limits the possibility to explore the vast design space offered by engineered materials. For the periodic material design, a RL-based framework is developed to determine the material properties of a 1-dimensional diatomic lattice, given a user-defined vibration characteristics in the frequency domain. Results from numerical simulation show that the design error of the proposed approach is 0.56% compared with the target values. Also, to address the randomness in the RL exploration, three repeated trials are conducted where the achieved design errors are below 3%. Compared to a conventional GA algorithm that requires $22,600$ executions of the forward simulation, the proposed approach reaches the solution with only 691 executions, which reduces 97% of the computation efforts. For the non-periodic material design, a fully-connected NN is used to model the dynamic behavior the material unit. After training, the same networks are concatenated to form a chain model of the metamaterial, and the material properties are treated as design parameters determined by minimizing the differences between the estimated responses and the user-defined target responses. This is the first demonstration of NN being used in the discretization of a continuous media. With only one-time network training, the concatenated networks accurately estimate the right-end responses of a bar consists of 20 units without error propagation, indicating the scalability of the proposed approach when designing a metamaterial made of numerous units. Moreover, the cross-section areas of each material unit are designed to achieve a user-specified left-end and right-end responses of the bar. Also, the proposed approach is demonstrated to design metamaterials with less units by incrementally adding material units to satisfy the desirable user tolerance in the response estimation.

In Chapter 6, a physics-constrained DAE based approach is developed to design the geometry of acoustic wave scatterers that achieve user-defined target downstream 2D pressure fields. This is the first demonstration of the multi-objective inverse design of wave scatterers using DAE-based approaches. By joint learning, the proposed network leverages the latent representations of the target pressure fields to strengthen the estimation of the scatterer geometry. In the most sophisticated design scenario (i.e., Scenario 3), the proposed approach achieves 99.91% accuracy in testing samples. To further evaluate how the network generalizes on the unseen dataset, a new dataset with new shapes of wave scatterers is generated. Compared with the exhaustive search over all choices of scatterer configuration, the proposed approach achieves 99.1%, 99.8% and 100.0% probability that the design returned by the network is superior to 95%, 90% and 75% choices of all configurations. Furthermore, the kernels of DAE are interpreted mathematically using the principle of convolution. In the first layer of the encoder, it is found that the kernels in the convolution layer serve as wavenumber bandpass filters to extract the characteristics of different wave frequency inputs. In the last layer of the decoder, the kernels in the convolution transpose layer attempt to reconstruct the target pressure fields. As a result, the kernels learn to preserve the significant wave patterns corresponding to each wave frequency while eliminating the irrelevant wavenumber signatures.

7.2 Future Works

In this dissertation, novel solutions to applications in SHM and metamaterial design are developed based on the state-of-the-art AI and ML techniques. Although the proposed approaches are demonstrated to be more robust against conventional tools, there still exists interesting and yet challenging problems that need to be addressed in future research. In this section, a roadmap for future works that may benefit the SHM and material design research communities is presented.

First of all, one major concern for data-driven approaches is whether these approaches are indeed able to generalize well and conduct prediction using new data. In SHM applications, it is common that the ML models are trained with a predefined damage scenarios and evaluated using data that belong to these scenarios. However, the damage patterns could vary case by case in practice. It is therefore reasonable to question the performance of the trained models when receiving new data that come from a damage scenario not considered during training. In other words, how the established models can be effectively augmented with these data remains an open question. Moreover, one should consider where the knowledge in physics stands in a decision making framework or how the physics reasoning can potentially be beneficial for AI-based approaches. Since the training of a deep network relies on high dimensional optimization that typically has no unique solution to the training parameters, a physics-guided optimization method can generate a better network. Also, the incorporation of physics models into data-driven based approaches may be favored in prognostic tasks such as the estimation of remaining service life of civil infrastructure. A hybrid approach that integrates data-enabled and physics-informed approaches needs further investigation. In addition, the metamaterial design approaches developed in this dissertation are demonstrated with one dimensional metamaterials. Extensions of the proposed approach are necessary to account for more complex cases and geometric conditions. Lastly, although this dissertation manages to provide an interpretation of deep neural networks, the discussions and mathematical derivations are limited to one or two layers. The interpretation of intermediate layers or deep layers still remains challenging due to the lack of more powerful mathematical tools to express the interconnecting relationships between the layers. As a result, more explorations in deep neural networks is another future intriguing research topic.

REFERENCES

REFERENCES

[1] M. R. Jahanshahi, "Vision-based studies for structural health monitoring and condition assessment," Ph.D. dissertation, University of Southern California, 2011.

[2] ASCE, "2017 report card for america's infrastructure," 2017.

[3] M. H. Rafiei, W. Khushefati, R. Demirboga, and H. Adeli, "Supervised deep restricted boltzmann machine for estimation of concrete compressive strength," *ACI Materials Journal*, vol. 114, no. 2, pp. 237–244, 2017.

[4] M. H. Rafiei and H. Adeli, "A novel machine learning based algorithm to detect damage in highrise building structures," *The Structural Design of Tall and Special Buildings*, vol. 26, no. 18, p. DOI: 10.1002/tal.1400, 2017.

[5] X. Kong and J. Li, "Vision-based fatigue crack detection of steel structures using video feature tracking," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 9, p. 783–799, 2018.

[6] H. Qarib and H. Adeli, "Recent advances in health monitoring of civil structures," *Scientia Iranica - Transaction A: Civil Engineering*, vol. 21, no. 6, pp. 1733–1742, 2014.

[7] S. W. Park, H. S. Park, J. Kim, and H. Adeli, "3d displacement measurement model for health monitoring of structures using a motion capture system," *Measurement*, vol. 59, pp. 352–362, 2015.

[8] J. Amezquita-Sanchez and H. Adeli, "Synchrosqueezed wavelet transform-fractality model for locating, detecting, and quantifying damage in smart highrise building structures," *Smart Materials and Structures*, vol. 24, p. 065034, 2015.

[9] Z. Li, H. Park, and H. Adeli, "New method for modal identification and health monitoring of superhighrise building structures using discretized synchrosqueezed wavelet and hilbert transforms," *The Structural Design of Tall and Special Buildings*, vol. 26, no. 3, p. DOI: 10.1002/tal.1312, 2017.

[10] B. K. Oh, K. Kim, Y. Kim, H. S. Park, and H. Adeli, "Evolutionary learning based sustainable strain sensing model for structural health monitoring of highrise buildings," *Applied Soft Computing*, vol. 58, pp. 576–585, 2017.

[11] M. Abdelbarr, Y. L. Chen, M. R. Jahanshahi, S. F. Masri, W.-M. Shen, and U. A. Qidwai, "3d dynamic displacement-field measurement for structural health monitoring using inexpensive RGB-d based sensor," *Smart Materials and Structures*, vol. 26, no. 12, p. 125016, nov 2017.

[12] I. A. Basheer and M. Hajmeer, "Artificial neural networks: fundamentals, computing, design, and application," *Journal of Microbiological Methods*, vol. 43, no. 1, pp. 3–31, 2000.

[13] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time-series," *in The Handbook of Brain Theory and Neural Networks*, 1995.

[14] R. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction.* The MIT Press, 1998.

[15] S. F. Masri, M. Nakamura, A. G. Chassiakos, and T. K. Caughey, "Neural network approach to detection of changes in structural parameters," *JOURNAL OF ENGINEERING MECHANICS*, vol. 122, no. 4, pp. 350–360, 1996.

[16] J. Tang, D. Sun, S. Liu, and J. L. Gaudiot, "Enabling deep learning on iot devices," *Computer*, vol. 50, no. 10, pp. 92–96, 2017.

[17] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, 2016.

[18] D. C. Verma and G. D. Mel, "Measures of network centricity for edge deployment of iot applications," *2017 IEEE International Conference on Big Data (Big Data)*, no. 10.1109/BigData.2017.8258505, 2017.

[19] M. R. Jahanshahi, W.-M. Shen, T. G. Mondal, M. Abdelbarr, S. F. Masri, and U. A. Qidwai, "Reconfigurable swarm robots for structural health monitoring: a brief review," *International Journal of Intelligent Robotics and Applications*, vol. 1, no. 3, pp. 287–305, 2017.

[20] M. R. Jahanshahi, F.-C. Chen, A. Ansar, C. Padgett, D. Clouse, and D. Bayard, "Accurate and robust scene reconstruction in the presence of misassociated features for aerial sensing," *Journal of Computing in Civil Engineering*, vol. 31, 11 2017.

[21] T. G. Mondal and M. R. Jahanshahi, "Autonomous vision-based damage chronology for spatiotemporal condition assessment of civil infrastructure using unmanned aerial vehicle," *Smart Structures and Systems*, vol. 25, pp. 733–749, 06 2020.

[22] T. G. Mondal, M. R. Jahanshahi, R.-T. Wu, and Z. Y. Wu, "Deep learning-based multi-class damage detection for autonomous post-disaster reconnaissance," *Structural Control and Health Monitoring*, vol. 27, no. 4, p. e2507, 2020, e2507 stc.2507.

[23] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz, "Pruning convolutional neural networks for resource efficient inference," *arXiv preprint*, no. arXiv:1611.06440v2, 2017.

[24] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint*, no. arXiv:1409.1556, 2014.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv preprint*, no. arXiv:1512.03385, 2015.

[26] R.-T. Wu, A. Singla, M. R. Jahanshahi, E. Bertino, B. J. Ko, and D. Verma, "Pruning deep convolutional neural networks for efficient edge computing in condition assessment of infrastructures," *Computer-Aided Civil and Infrastructure Engineering*, vol. 34, no. 9, pp. 774–789, 2019.

[27] R.-T. WU, A. SINGLA, M. Jahanshahi, and E. Bertino, "Pruning deep neural networks for efficient edge computing in internet of things: A structural health monitoring case study," 11 2019.

[28] C. R. Farrar and K. Worden, "An introduction to structural health monitoring," *Philosophical Transactions of the Royal Society, A*, vol. 365, pp. 303–315, 2007.

[29] M. R. Jahanshahi, F. Jazizadeh, S. F. Masri, and B. Becerik-Gerber, "Unsupervised approach for autonomous pavement-defect detection and quantification using an inexpensive depth sensor," *Journal of Computing in Civil Engineering*, vol. 27, no. 6, pp. 743–754, November 2013.

[30] Y. Mizuno, M. Abe, Y. Fujino, and M. Abe, "Development of interactive support system for visual inspection of bridges," *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 4337, pp. 155–166, March 2001.

[31] A. Mahmoudzadeh, A. Golroo, M. R. Jahanshahi, and S. F. Yeganeh, "Estimating pavement roughness by fusing color and depth data obtained from an inexpensive rgb-d sensor," *Sensors*, vol. 19, no. 7, 2019.

[32] S. F. Yeganeh, A. Golroo, and M. R. Jahanshahi, "Automated rutting measurement using an inexpensive rgb-d sensor fusion approach," *Journal of Transportation Engineering, Part B: Pavements*, vol. 145, no. 1, p. 04018061, 2019.

[33] D. L. Hall and J. Llinas, "An introduction to multi-sensor data fusion," *Proc. IEEE*, vol. 85, pp. 6–23, 1997.

[34] E. Waltz and J. Llinas, "Multisensor data fusion (vol. 685)," *Boston: Artech house*, 1990.

[35] M. E. Liggins, D. L. Hall, and J. Llinas, *Multisensor data fusion: Theory and practice.*, 2nd ed. Boca Raton, FL: CRC Press., 2009.

[36] N.-E. E. Faouzi, H. Leung, and A. Kurian, "Data fusion in intelligent transportation systems: Progress and challenges-a survey," *Information Fusion*, vol. 12, pp. 4–10, 2011.

[37] D. L. Hall and J. Llinas, "Handbook of multisensor data fusion," *CRC Press LLC*, 2001.

[38] I. Lopez and N. Sarigul-Klijn, "A review of uncertainty in flight vehicle structural damage monitoring, diagnosis and control: Challenges and opportunities," *Progress in Aerospace Sciences*, vol. 46, pp. 247–273, 2010.

[39] J. Llinas and D. L. Hall, "An introduction to multi-sensor data fusion," *Circuits and Systems, 1998. ISCAS '98. Proceedings of the 1998 IEEE International Symposium on*, vol. 6, 1998.

[40] J. Hu, *Data Fusion: A First Step in Decision Informatics.* Proquest, Umi Dissertation Publishing, 2011.

[41] L. A. Klein, *Sensor and Data Fusion: A Tool for Information Assessment and Decision Making*, 2nd ed. SPIE Press, 2012.

[42] M. N. Chatzis, E. N. Chatzi, and A. W. Smyth, "On the observability and identifiability of nonlinear structural and mechanical systems," *Structural Control and Health Monitoring*, vol. 22, no. 3, p. 574–593, 2015.

[43] M. G. Kogan, W.-Y. Kim, Y. Bock, and A. W. Smyth, "Load response on a large suspension bridge during the nyc marathon revealed by gps and accelerometers," *Seismological Research Letters*, vol. 79, no. 1, p. 12–19, 2008.

[44] E. N. Chatzi and A. W. Smyth, "The unscented kalman filter and particle filter methods for nonlinear structural system identification with non-collocated heterogeneous sensing," *Structural Control and Health Monitoring*, vol. 16, p. 99–123, 2009.

[45] D. D. Freedman and P. A. Smyton, "Overview of data fusion activities," *American Control Conference, IEEE*, 1993.

[46] R. C. Gonzalez and R. E. Woods, "Digital image processing," *Addison-Wesley, Reading*, vol. 302, 1993.

[47] L. G. Brown, "A survey of image registration techniques," *ACM Computing Surveys*, vol. 24, no. 4, p. 325–376, 1992.

[48] Z. Liu, D. S. Forsyth, J. P. Komorowski, K. Hanasaki, and T. Kirubarajan, "Survey: State of the art in nde data fusion techniques," *Instrumentation and Measurement*, vol. 56, no. 6, p. 2435–2451, 2007.

[49] M. E. Liggins, D. L. Hall, and J. Llinas, "Handbook of multisensor data fusion: theory and practice," *CRC Press*, 2017.

[50] F. C. Chen and M. R. Jahanshahi, "Nb-cnn: Deep learning-based crack detection using convolutional neural network and naïve bayes data fusion," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 5, pp. 4392–4400, 2018.

[51] J. Bernardo and A. Smith, *Bayesian Theory.* Wiley, Chichester, 1994.

[52] J. K. Ghosh, M. Delampady, and T. Samanta, *An Introduction to Bayesian Analysis Theory and Methods.* Springer, 2006.

[53] K.-V. Yuen, *Bayesian Methods for Structural Dynamics and Civil Engineering.* John Wiley & Sons, 2010.

[54] F. Jalayer, R. D. Risi, and G. Manfredi, "Bayesian cloud analysis: efficient structural fragility assessment using linear regression," *Bulletin of Earthquake Engineering*, vol. 13, p. 1183–1203, 2015.

[55] J. L. Beck and L. S. Katafygiotis, "Updating models and their uncertainties. i: Bayesian statistical framework," *Journal of Engineering Mechanics*, vol. 124, no. 4, p. 455–461, 1998.

[56] D. V. Lindley, "Introduction to probability and statistics from a bayesian viewpoint," *Cambridge University Press*, 1965.

[57] J. L. Beck and S.-K. Au, "Bayesian updating of structural models and reliability using markov chain monte carlo simulation," *Journal of Engineering Mechanics*, vol. 128, no. 4, p. 380–391, 2002.

[58] J. Ching, M. Muto, and J. L. Beck, "Bayesian linear structural model updating using gibbs sampler with modal data," *Proceedings of the 9th International Conference on Structural Safety and Reliability*, pp. 2609–2616, 2005.

[59] ——, "Structural model updating and health monitoring with incomplete modal data using gibbs sampler," *Comput.-Aided Civ. Inf.*, vol. 21, no. 4, p. 242–257, 2006.

[60] J. L. Beck, "Bayesian system identification based on probability logic," *Structural Control and Health Monitoring*, vol. 17, p. 825–847, 2010.

[61] M. W. Vanik, J. L. Beck, and S. K. Au, "Bayesian probabilistic approach to structural health monitoring," *Journal of Engineering Mechanics*, vol. 126, no. 7, pp. 738–745, 2000.

[62] J. L. Beck, S. K. Au, and M. W. Vanik, "Monitoring structural health using a probabilistic measure," *Computer-Aided Civil and Infrastructure Engineering*, vol. 16, pp. 1–11, 2001.

[63] C. Papadimitriou, J. L. Beck, and L. S. Katafygiotis, "Updating robust reliability using structural test data," *Probabilistic Engineering Mechanics*, vol. 16, pp. 103–113, 2001.

[64] K.-V. Yuen, S. K. Au, and J. L. Beck, "Two-stage structural health monitoring approach for phase i benchmark studies," *Journal of Engineering Mechanics*, vol. 130, no. 1, pp. 16–33, 2004.

[65] K.-V. Yuen, J. L. Beck, and L. S. Katafygiotis, "Efficient model updating and health monitoring methodology using incomplete modal data without mode matching," *Structural Control and Health Monitoring*, vol. 13, p. 91–107, 2006.

[66] M. Muto and J. L. Beck, "Bayesian updating and model class selection for hysteretic structural models using stochastic simulation," *Journal of Vibration and Control*, vol. 14, no. 1-2, p. 7–34, 2006.

[67] E. Ntotsios, C. Papadimitriou, P. Panetsos, G. Karaiskos, K. Perros, and P. C. Perdikaris, "Bridge health monitoring system based on vibration measurements," *Bulletin of Earthquake Engineering*, vol. 7, p. 469–483, 2009.

[68] C. Papadimitriou and E. Ntotsios, "Bayesian methodology for structural damage identification and reliability assessment," *3rd International Operational Modal Analysis Conference*, 2009.

[69] T.-K. Lin, L.-C. Yu, C.-H. Ku, K.-C. Chang, and A. Kiremidjian, "Implementation of a bio-inspired two-mode structural health monitoring system," *Smart Structures and Systems*, vol. 8, no. 1, p. 119–137, 2011.

[70] B. A. Zárate, J. M. Caicedo, J. Yu, and P. Ziehl, "Bayesian model updating and prognosis of fatigue crack growth," *Engineering Structures*, vol. 45, pp. 53–61, 2012.

[71] M. Gobbato, J. P. Conte, J. B. Kosmatka, and C. R. Farrar, "A reliability-based framework for fatigue damage prognosis of composite aircraft structures," *Probabilistic Engineering Mechanics*, vol. 29, pp. 176–188, 2012.

[72] M. Gobbato, J. B. Kosmatka, and J. P. Conte, "A recursive bayesian approach for fatigue damage prognosis: An experimental validation at the reliability component level," *Mechanical Systems and Signal Processing*, vol. 45, pp. 448–467, 2014.

[73] V. H. Jaramilloa, J. R. Ottewilla, R. Dudek, D. Lepiarczyk, and P. Pawlik, "Condition monitoring of distributed systems using two-stage bayesian inference data fusion," *Mechanical Systems and Signal Processing*, vol. 87, pp. 91–110, 2016.

[74] M. Rabiei and M. Modarres, "A recursive bayesian framework for structural health management using online monitoring and periodic inspections," *Reliability Engineering and System Safety*, vol. 112, pp. 154–164, 2013.

[75] D. Zonta, F. Bruschetta, R. Zandonini, M. Pozzi, M. Wang, B. Glisic, D. Inaudi, D. Posenato, and Y. Zhao, "Sensor fusion on structural monitoring data analysis: Application to a cable-stayed bridge," *Key Engineering Materials*, vol. 569-570, pp. 812–819, 2013.

[76] H. F. Lam, H. Y. Peng, and S. K. Au, "Development of a practical algorithm for bayesian model updating of a coupled slab system utilizing field test data," *Engineering Structures*, vol. 79, p. 182–194, 2014.

[77] L. F. Ramos, T. Miranda, M. Mishra, F. M. Fernandes, and E. Manning, "A bayesian approach for ndt data fusion: The saint torcato church case study," *Engineering Structures*, vol. 84, pp. 120–129, 2015.

[78] G. Yan and J. Tang, "A bayesian approach for localization of acoustic emission source in plate-like structures," *Mathematical Problems in Engineering*, no. 247839, 2015.

[79] W. Wang, A. Joshi, N. Tirpankar, P. Erickson, M. Cline, P. Thangaraj, and T. C. Henderson, "Bayesian computational sensor networks: Small-scale structural health monitoring," *ICCS 2015 International Conference On Computational Science*, vol. 51, pp. 2603–2612, 2015.

[80] H. Sun and R. Betti, "A hybrid optimization algorithm with bayesian inference for probabilistic model updating," *Computer-Aided Civil and Infrastructure Engineering*, vol. 30, p. 602–619, 2015.

[81] H. Sun and O. Büyüköztürk, "Identification of traffic-induced nodal excitations of truss bridges through heterogeneous data fusion," *Smart Materials and Structures*, vol. 24, p. 075032, 2015.

[82] S. K. Au, "Fast bayesian fft method for ambient modal identification with separated modes," *Journal of Engineering Mechanics*, vol. 137, no. 3, pp. 214–226, 2011.

[83] K. V. Yuen and L. S. Katafygiotis, "Bayesian fast fourier transform approach for modal updating using ambient data," *Advances in Structural Engineering*, vol. 6, no. 2, p. 81–95, 2003.

[84] S. K. Au and F. L. Zhang, "Fast bayesian ambient modal identification incorporating multiple setups," *Journal of Engineering Mechanics*, vol. 138, no. 7, pp. 800–815, 2012.

[85] F.-L. Zhang, S.-K. Au, and H.-F. Lam, "Assessing uncertainty in operational modal analysis incorporating multiple setups using a bayesian approach," *Structural Control and Health Monitoring*, vol. 22, p. 395–416, 2015.

[86] F.-L. Zhang and S.-K. Au, "Fundamental two-stage formulation for bayesian system identification, part ii: Application to ambient vibration data," *Mechanical Systems and Signal Processing*, vol. 66-67, p. 43–61, 2016.

[87] S. K. Au and Y.-C. Ni, "Fast bayesian modal identification of structures using known single-input forced vibration data," *Structural Control and Health Monitoring*, vol. 21, p. 381–402, 2014.

[88] W.-J. Yan and L. S. Katafygiotis, "Application of transmissibility matrix and random matrix to bayesian system identification with response measurements only," *Smart Materials and Structures*, vol. 25, no. 105017, 2016.

[89] F. C. Chen, M. R. Jahanshahi, R. T. Wu, and C. Joffe, "A texture-based video processing methodology using bayesian data fusion for autonomous crack detection on metallic surfaces," *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, p. 271–287, 2017.

[90] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, p. 971–987, 2002.

[91] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, p. 273–297, 1995.

[92] Y. Huang, J. L. Beck, and H. Li, "Bayesian system identification based on hierarchical sparse bayesian learning and gibbs sampling with application to structural damage assessment," *Comput. Methods Appl. Mech. Engrg.*, vol. 318, p. 382–411, 2017.

[93] S. Geman and D. Geman, "Stochastic relaxation, gibbs distribution and the bayesian restoration of images," *IEEE Trans. Pattern Anal.*, vol. 6, p. 721–741, 1984.

[94] S. E. Minson, S. Wu, J. L. Beck, and T. H. Heaton, "Combining multiple earthquake models in real time for earthquake early warning," *Bulletin of the Seismological Society of America*, vol. 107, no. 4, p. 1868–1882, 2017.

[95] X. E. Gross, "NDT data fusion," 1997.

[96] N. Brierley, T. Tippetts, and P. Cawley, "Data fusion for automated nondestructive inspection," *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 470, no. 2167, 2014.

[97] BSSC, *NEHRP guidelines for the seismic rehabilitation of buildings.* Washington, D.C.: FEMA-273, 1997.

[98] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *J. Basic Eng.*, vol. 82, no. 1, pp. 35–45, 1960.

[99] R. Faragher, “Understanding the basis of the kalman filter via a simple and intuitive derivation,” *IEEE Signal Process. Mag.*, vol. 29, no. 5, pp. 128–132, 2012.

[100] A. H. Jazwinski, “Stochastic processes and filtering theory,” *San Diego, CA: Academic*, 1970.

[101] E. H. W. Sorenson, “Kalman filtering: Theory and application,” *Piscataway, NJ: IEEE*, 1985.

[102] S. J. Julier and J. K. Uhlmann, “Unscented filtering and nonlinear estimation,” *PROCEEDINGS OF THE IEEE*, vol. 92, no. 3, p. 401–422, 2004.

[103] A. W. Smyth, T. Kontoroupi, and P. T. Brewick, “Efficient data fusion and practical considerations for structural identification,” *CISM International Centre for Mechanical Sciences*, vol. 567, pp. DOI: 10.1007/978–3–319–32 077–9_2, 2016.

[104] S. J. Julier and J. K. Uhlmann, “A new extension of the kalman filter to nonlinear systems,” *Proceedings of the International Symposium on Aerospace/Defense Sensing, Simulation and Controls*, p. 32, 1991.

[105] P. D. Moral, “Non linear filtering: Interacting particle solution,” *Markov Processes and Related Fields*, vol. 2, no. 4, p. 555–580, 1996.

[106] J. S. Liu and R. Chen, “Sequential monte carlo methods for dynamic systems,” *Journal of the American Statistical Association*, vol. 93, no. 443, p. 1032–1044, 1998.

[107] T. Khan, P. Ramuhalli, and S. C. Dass, “Particle-filter-based multisensor fusion for solving low-frequency electromagnetic nde inverse problems,” *Instrumentation and Measurement, IEEE Transactions*, vol. 60, no. 6, p. 2142–2153, 2011.

[108] P.-T. Liu, F. Li, and H. Xiao, “A state decoupling approach to estimate unobservable tracking systems,” *IEEE J. Oceanic Eng.*, vol. 21, no. 3, pp. 256–259, 1996.

[109] M. N. Chatzis, E. N. Chatzi, and S. Triantafyllou, “A discontinuous extended Kalman filter for non-smooth dynamic problems,” *Mechanical Systems and Signal Processing*, vol. 92, pp. 13–29, 2017.

[110] M. N. Chatzis and E. N. Chatzi, “A discontinuous unscented Kalman filter for non-smooth dynamic problems,” *Front. Built Environ.*, vol. 3, no. 56, pp. 1–15, 2017.

[111] A. Smyth and M. Wu, “Multi-rate kalman filtering for the data fusion of displacement and acceleration response measurements in dynamic system monitoring,” *Mechanical Systems and Signal Processing*, vol. 21, no. 2, pp. 706–723, 2007.

[112] H. Tan, A. M. Wilson, and J. Lowe, “Measurement of stride parameters using a wearable gps and inertial measurement unit,” *Journal of Biomechanics*, vol. 41, no. 7, p. 1398–1406, 2008.

[113] F. Caron, E. Duflos, E. Pomorski, and P. Vanheeghe, "Gps/imu data fusion using mutli-sensor kalman filtering: introduction of contextual aspects," *Information Fusion*, vol. 7, p. 221–230, 2006.

[114] C. C. Chang and X. H. Xiao, "Accurate displacement measurement from fusion of vision-based displacement and acceleration with Kalman filter," *20th Int. Conf. on Adaptive Structures and Technologies*, 2009.

[115] J. N. Yang, S. Lin, H. Huang, and L. Zhou, "An adaptive extended kalman filter for structural damage identification," *Structural Control and Health Monitoring*, vol. 13, no. 4, p. 849–867, 2006.

[116] J. N. Yang, S. Pan, and H. Huang, "An adaptive extended kalman filter for structural damage identifications ii: unknown inputs," *Structural Control and Health Monitoring*, vol. 14, no. 3, p. 497–521, 2007.

[117] M. Wu and A. W. Smyth, "Application of the unscented Kalman filter for real-time nonlinear structural system identification," *Structural Control and Health Monitoring*, vol. 14, p. 971–990, 2007.

[118] Y. K. Wen, "Method for random vibration of hysteretic systems," *Journal of the Engineering Mechanics Division*, vol. 102, no. 2, pp. 249–263, 1976.

[119] R. van der Merwe and E. Wan, "Gaussian mixture sigma-point particle filters for sequential probabilistic inference in dynamic state-space models," *In Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2003.

[120] Y. Bock, D. Melgar, and B. W. Crowell, "Real-time strong-motion broadband displacements from collocated GPS and accelerometers," *Bulletin of the Seismological Society of America*, vol. 101, no. 6, p. 2904–2925, 2011.

[121] J. A. Rodger, "Toward reducing failure risk in an integrated vehicle health maintenance system: A fuzzy multi-sensor data fusion kalman filter approach for ivhms," *Expert Systems with Applications*, vol. 39, pp. 9821–9836, 2012.

[122] J. Kim, K. Kim, and H. Sohn, "Autonomous dynamic displacement estimation from data fusion of acceleration and intermittent displacement measurements," *Mechanical Systems and Signal Processing*, vol. 42, pp. 194–205, 2014.

[123] S. E. Azam, E. Chatzi, and C. Papadimitriou, "A dual Kalman filter approach for state estimation via output-only acceleration measurements," *Mechanical Systems and Signal Processing*, vol. 60-61, pp. 866–886, 2015.

[124] E. Lourens, E. Reynders, G. DeRoeck, G. Degrande, and G. Lombaert, "An augmented kalman filter for force identification in structural dynamics," *Mechanical Systems and Signal Processing*, vol. 27, pp. 446–460, 2012.

[125] S. Gillijns and B. DeMoor, "Unbiased minimum-variance input and state estimation for linear discrete-time systems with direct feedthrough," *Automatica*, vol. 43, p. 934–937, 2007.

[126] F. Naets, J. Cuadrado, and W. Desmet, "Stable force identification in structural dynamics using kalman filtering and dummy-measurements," *Mechanical Systems and Signal Processing*, vol. 50-51, p. 235–248, 2015.

[127] S. Cho, J.-W. Park, R. P. Palanisamy, and S.-H. Sim, "Reference-free displacement estimation of bridges using kalman filter-based multimetric data fusion," *Journal of Sensors*, vol. 2016, no. 3791856, 2016.

[128] X.-H. Huang, S. Dyke, Z. Sun, and Z.-D. Xu, "Simultaneous identification of stiffness, mass, and damping using an on-line model updating approach," *Struct. Control Health Monit.*, vol. 24, 2017.

[129] K. Kim and H. Sohn, "Dynamic displacement estimation by fusing ldv and lidar measurements via smoothing based kalman filtering," *Mechanical Systems and Signal Processing*, vol. 82, pp. 339–355, 2017.

[130] K. Erazo and E. M. Hernandez, "Bayesian model-data fusion for mechanistic postearthquake damage assessment of building structures," *Journal of Engineering Mechanics*, vol. 142, no. 9, p. 04016062, 2016.

[131] G. Evensen, "The ensemble kalman filter: Theoretical formulation and practical implementation," *Ocean Dyn.*, vol. 53, no. 4, p. 343–367, 2003.

[132] E. N. Chatzi and C. Fuggini, "Structural identification of a super-tall tower by GPS and accelerometer data fusion using a multi-rate Kalman filter," *Life-Cycle and Sustainability of Civil Infrastructure Systems: Proceedings of the Third International Symposium on Life-Cycle Civil Engineering (IALCCE'12)*, 2012.

[133] M. N. Chatzis, E. N. Chatzi, and A. W. Smyth, "An experimental validation of time domain system identification methods with fusion of heterogeneous data," *Earthquake Engineering & Structural Dynamics*, vol. 44, p. 523–547, 2015.

[134] O. P. Van and M. B. De, "Subspace algorithms for the identification of combined deterministic-stochastic systems," *Automatica*, vol. 30, no. 1, pp. 75–93, 1994.

[135] E. Asgarieh, B. Moaveni, A. R. Barbosa, and E. Chatzi, "Nonlinear model calibration of a shear wall building using time and frequency data features," *Mechanical Systems and Signal Processing*, vol. 85, p. 236–251, 2017.

[136] M. N. Chatzis and E. N. Chatzi, "Online bayesian identification of non-smooth systems," *Procedia Engineering*, vol. 199, pp. 918–923, 2017.

[137] G. Shafer, "A mathematical theory of evidence," *Princeton University Press*, 1976.

[138] H. Wu, "Sensor data fusion for context-aware computing using dempster-shafer theory," Ph.D. dissertation, The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 2003.

[139] R. R. Yager, J. Kacprzyk, and M. Fedrizzi, Eds., *Advances in the Dempster-Shafer Theory of Evidence.* John Wiley & Sons, 1994.

[140] T. Inagaki, "Interdependence between safety-control policy and multiple-sensor schemes via dempster-shafer theory," *IEEE Transactions on Reliability*, vol. 40, no. 2, pp. 182–188, 1991.

[141] Y. Bao, H. Li, Y. An, and J. Ou, "Dempster-shafer evidence theory approach to structural damage detection," *Structural Health Monitoring*, vol. 11, no. 1, pp. 13–26, 2011.

[142] E. Zervas, A. Mpimpoudis, C. Anagnostopoulos, O. Sekkas, and S. Hadjiefthymiades, “Multisensor data fusion for fire detection,” *Information Fusion*, vol. 12, pp. 150–159, 2011.

[143] X. Zhao, R. Wang, H. Gu, G. Song, and Y. L. Mo, “Innovative data fusion enabled structural health monitoring approach,” *Mathematical Problems in Engineering*, no. 369540, 2014.

[144] Q. Zhou, H. Zhou, Q. Zhou, F. Yang, L. Luo, and T. Li, “Structural damage detection based on posteriori probability support vector machine and Dempster-Shafer evidence theory,” *Applied Soft Computing*, vol. 36, pp. 368–374, 2015.

[145] A. Moosavian, M. Khazaee, G. Najafi, M. Kettner, and R. Mamat, “Spark plug fault recognition based on sensor fusion and classifier combination using Dempster-Shafer evidence theory,” *Applied Acoustics*, vol. 93, pp. 120–129, 2015.

[146] L. Hou and N. W. Bergmann, “Novel industrial wireless sensor networks for machine condition monitoring and fault diagnosis,” *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 10, pp. 2787–2798, 2012.

[147] E. Grande and M. Imbimbo, “A multi-stage approach for damage detection in structural systems based on flexibility,” *Mechanical Systems and Signal Processing*, vol. 76-77, pp. 455–475, 2016.

[148] J. Huang and W. Liu, “A pavement crack detection method combining 2D with 3D information based on Dempster-Shafer theory,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 29, pp. 299–313, 2014.

[149] C. Völker and P. Shokouhi, “Multi sensor data fusion approach for automatic honeycomb detection in concrete,” *NDT & E International*, vol. 71, pp. 54–60, 2015.

[150] F. Zhang, “The schur complement and its applications,” *Science + Business Media, Inc, Springer*, 2005.

[151] L. A. Zadeh, “Fuzzy logic,” *IEEE Computer*, pp. 83–93, 1988.

[152] S.-F. Jiang, C.-M. Zhang, and S. Zhang, “Two-stage structural damage detection using fuzzy neural networks and data fusion techniques,” *Expert Systems with Applications*, vol. 38, pp. 511–519, 2011.

[153] R. Heideklang and P. Shokouhi, “Multi-sensor image fusion at signal level for improved near-surface crack detection,” *NDT and E International*, vol. 71, pp. 16–22, 2015.

[154] X. Lou and K. A. Loparo, “Bearing fault diagnosis based on wavelet transform and fuzzy inference,” *Mechanical Systems and Signal Processing*, vol. 18, pp. 1077–1095, 2004.

[155] M. Sugeno, “Fuzzy measures and fuzzy integrals: a survey,” *In: Gupta M. M., Saridis G. N., Gains B. R., editors. Fuzzy automata and decision processes. Amsterdam: NorthHollan*, pp. 89–102, 1977.

[156] K. Amolins, Y. Zhang, and P. Dare, “Wavelet based image fusion techniques—an introduction, review and comparison,” *ISPRS J Photogramm Remote Sens*, vol. 62, pp. 249–263, 2007.

[157] J. Teng, W. Lu, R. Wen, and T. Zhang, “Instrumentation on structural health monitoring systems to real world structures,” *Smart Structures and Systems*, vol. 15, no. 1, pp. 151–167, 2015.

[158] P. J. Dempsey and S. Sheng, “Investigation of data fusion applied to health monitoring of wind turbine drivetrain components,” *Wind Energy*, vol. 16, pp. 479–489, 2013.

[159] B. E. Boser, I. Guyon, and V. Vapnik, “A training algorithm for optimal margin classifiers,” *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pp. 144–152, 1992.

[160] V. N. Vapnik, *The Nature of Statistical Learning Theory.* New York: Springer-Verlag, 1995.

[161] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, p. 436–444, 2015.

[162] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *NIPS*, p. 1106–1114, 2012.

[163] R. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, and H. Seung, “Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit,” *Nature*, vol. 405, p. 947–951, 2000.

[164] G. Hinton and R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science*, p. 504–507, 2006.

[165] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, “Greedy layer-wise training of deep networks,” *Advances in neural information processing systems*, vol. 19, no. 153, 2007.

[166] Y.-Y. Liu, Y.-F. Ju, C.-D. Duan, and X.-F. Zhao, “Structure damage diagnosis using neural network and feature fusion,” *Engineering Applications of Artificial Intelligence*, vol. 24, no. 1, pp. 87–92, 2011.

[167] R. Ogden, *Essential Wavelets for Statistical Applications and Data Analysis.* Boston: Birkhäuser, 1997.

[168] R. R. Coifman and M. Wickerhauser, “Entropy-based algorithms for best basis selection,” *IEEE Trans. Inform. Theory*, vol. 38, pp. 713–718, 1992.

[169] J.-S. Pei, J. P. Wright, and A. W. Smyth, “Neural network initialization with prototypes - a case study in function approximation,” *Proceedings of International Joint Conference on Neural Networks*, July 31 - August 4, Montreal, Canada 2005.

[170] J.-S. Pei and A. W. Smyth, “New approach to designing multilayer feedforward neural network architecture for modeling nonlinear restoring forces. i: Formulation,” *JOURNAL OF ENGINEERING MECHANICS*, vol. 132, no. 12, pp. 1290–1300, 2006.

[171] ——, "New approach to designing multilayer feedforward neural network architecture for modeling nonlinear restoring forces. ii: Applications," *JOURNAL OF ENGINEERING MECHANICS*, vol. 132, no. 12, pp. 1301–1312, 2006.

[172] J.-S. Pei, E. C. Mai, J. P. Wright, and A. W. Smyth, "Neural network initialization with prototypes - function approximation in engineering mechanics applications," *Proceedings of International Joint Conference on Neural Networks*, August 12-17, Orlando, Florida, USA 2007.

[173] J.-S. Pei and E. C. Mai, "Constructing multilayer feedforward neural networks to approximate nonlinear functions in engineering mechanics applications," *Journal of Applied Mechanics*, vol. 75, no. 061002, 2008.

[174] J.-S. Pei, J. P. Wright, S. F. Masri, E. C. Mai, and A. W. Smyth, "Toward constructive methods for sigmoidal neural networks - function approximation in engineering mechanics applications," *Proceedings of International Joint Conference on Neural Networks*, July, San Jose, USA 2011.

[175] J.-S. Pei, E. C. Mai, J. P. Wright, and S. F. Masri, "Mapping some basic functions and operations to multilayer feedforward neural networks for modeling nonlinear dynamical systems and beyond," *Nonlinear Dyn*, vol. 71, p. 371–399, 2013.

[176] J.-S. Pei and S. F. Masri, "Demonstration and validation of constructive initialization method for neural networks to approximate nonlinear functions in engineering mechanics applications," *Nonlinear Dyn*, vol. 79, p. 2099–2119, 2015.

[177] R. Ghiasi, P. Torkzadeh, and M. Noori, "Structural damage detection using artificial neural networks and least square support vector machine with particle swarm harmony search algorithm," *Int. J. Sustain. Mater. Struct. Syst.*, vol. 1, no. 4, pp. 303–320, 2014.

[178] J. A. Suykens, J. D. Brabanter, L. Lukas, and J. Vandewalle, "Weighted least squares support vector machines: robustness and sparse approximation," *Neurocomputing*, vol. 48, pp. 85–105, 2002.

[179] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," *In Proceedings of the Sixth International Symposium on Micro Machine and Human Science, MHS'95, IEEE*, p. 39–43, 1995.

[180] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: harmony search," *Simulation*, vol. 76, no. 2, p. 60–68, 2001.

[181] S. Stramondo, C. Bignami, M. Chini, N. Pierdicca, and A. Tertulliani, "Satellite radar and optical remote sensing for earthquake damage detection: results from different case studies," *International Journal of Remote Sensing*, vol. 27, no. 20, pp. 4433–4447, 2006.

[182] N. Longbotham, T. Glenn, A. Zare, M. Volpi, D. Tuia, E. Cristophe, J. Michel, J. Inglada, J. Chanussot, Q. Du, and F. Pacifici, "Multi-modal change detection, application to the detection of flooded areas: Outcome of the 2009-2010 data fusion contest," *IEEE J. Sel. Topics Appl. Earth Observ.*, vol. 5, no. 1, pp. 1–12, 2012.

[183] S. Khazaeli, A. G. Ravandi, S. Banerji, and A. Bagchi, "The application of data mining and cloud computing techniques in data-driven models for structural health monitoring," *Health Monitoring of Structural and Biological Systems*, p. 98052M, 2016.

[184] T. Khuc and F. N. Catbas, "Structural identification using computer vision–based bridge health monitoring," *Journal of Structural Engineering, ASCE*, vol. 144, no. 2, p. 04017202, 2018.

[185] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, p. 119–139, 1997.

[186] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *Proceeding of Computer Vision and Pattern Recognition*, vol. 511, 2001.

[187] D. Sun, V. C. Lee, and Y. Lu, "An intelligent data fusion framework for structural health monitoring," *2016 IEEE 11th Conference on Industrial Electronics and Applications (ICIEA)*, p. DOI: 10.1109/ICIEA.2016.7603550, 2016.

[188] S. Mohanty, B. Jagielo, C. B. Bhan, S. Majumdar, and K. Natesan, "Online stress corrosion crack monitoring in nuclear reactor components using active ultrasonic sensor networks and nonlinear system identification - data fusion based big data analytics approach," *Proceedings of the ASME 2015 Pressure Vessels and Piping Conference*, July 19-23, Boston, USA 2015.

[189] R. A. Osornio-Rios, J. P. Amezquita-Sanchez, R. J. Romero-Troncoso, and A. Garcia-Perez, "Music-ann analysis for locating structural damages in a truss-type structure by means of vibrations," *Computer-Aided Civil and Infrastructure Engineering*, vol. 27, pp. 687–698, 2012.

[190] K. N. Kesavan and A. S. Kiremidjian, "A wavelet-based damage diagnosis algorithm using principal component analysis," *Struct. Control Health Monit.*, vol. 19, p. 672–685, 2012.

[191] S. Safari, F. Shabani, and D. Simon, "Multirate multisensor data fusion for linear systems using Kalman filters and a neural network," *Aerospace Science and Technology*, vol. 39, p. 465–471, 2014.

[192] L. P. Yan, B. S. Liu, and D. H. Zhou, "Multirate multisensor data fusion for linear systems using Kalman filters and a neural network," *Aerospace Science and Technology*, vol. 39, p. 465–471, 2006.

[193] A. Derkevorkian, M. Hernandez-Garcia, H.-B. Yun, S. F. Masri, and P. Li, "Nonlinear data-driven computational models for response prediction and change detection," *STRUCTURAL CONTROL AND HEALTH MONITORING*, vol. 22, p. 273–288, 2015.

[194] S. Mokhtari, L. Wu, and H.-B. Yun, "Comparison of supervised classification techniques for vision-based pavement crack detection," *Journal of the Transportation Research Board*, vol. 2595, pp. 119–127, 2016.

[195] S. Terzi, "Terzi, s. modeling for pavement roughness using the anfis approach," *Advances in Engineering Software*, vol. 57, p. 59–64, 2013.

[196] S. D. Vanraj and B. Pabla, "Hybrid data fusion approach for fault diagnosis of fixed-axis gearbox," *Structural Health Monitoring*, vol. 00, no. 0, pp. 1–10, 2017.

[197] C. Völker, S. Kruschwitz, C. Boller, and H. Wiggenhauser, "Feasibility study on adapting a machine learning based multi-sensor data fusion approach for honeycomb detection in concrete," *NDE/NDT for Highways & Bridges: SMT 2016*, 2016.

[198] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, 1996.

[199] Y.-J. Cha, W. Choi, and O. Büyüköztürk, "Deep learning-based crack damage detection using convolutional neural networks," *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, p. 361–378, 2017.

[200] O. Abdeljaber, O. Avci, S. Kiranyaz, M. Gabbouj, and D. J. Inman, "Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks," *Journal of Sound and Vibration*, vol. 388, p. 154–170, 2017.

[201] F.-C. Chen and M. R. Jahanshahi, "Arf-crack: rotation invariant deep fully convolutional network for pixel-level crack detection," *Machine Vision and Applications*, vol. 31, p. 47, 07 2020.

[202] ——, "Nb-fcn: Real-time accurate crack detection in inspection videos using deep fully convolutional network and parametric data fusion," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 8, pp. 5325–5334, 2020.

[203] A. Akintayo, K. G. Lore, S. Sarkar, and S. Sarkar, "Prognostics of combustion instabilities from hi-speed flame video using a deep convolutional selective autoencoder," *International Journal of Prognostics and Health Management*, vol. 7, 2016.

[204] S. Sarkar, K. K. Reddy, M. Giering, and M. R. Gurvich, "Deep learning for structural health monitoring: A damage characterization application," *Annual Conference of The Prognosis and Health Management society*, 2016.

[205] K. K. Reddy, S. Sarkar, V. Venugopalan, and M. Giering, "Anomaly detection and fault disambiguation in large flight data: A multi-modal deep auto-encoder approach," *Annual Conference of The Prognosis and Health Management society*, 2016.

[206] W. Yan and L. Yu, "On accurate and reliable anomaly detection for gas turbine combustors: A deep learning approach," *Annual Conference of The Prognosis and Health Management society*, 2015.

[207] K. J. Arrow, "A difficulty in the concept of social welfare," *The Journal of Political Economy*, vol. 58, no. 4, p. 328–346, 1950.

[208] G. A. Hazelrigg, "Fundamentals of decision making for engineering design and systems engineering," *ISBN: 978-0-9849976-0-2*, 2012.

[209] Y. Lu and J. E. Michaels, “Feature extraction and sensor fusion for ultrasonic structural health monitoring under changing environmental conditions,” *Sensors Journal, IEEE*, vol. 9, no. 11, p. 1462–1471, 2009.

[210] J. E. Michaels and T. E. Michaels, “Guided wave signal processing and image fusion for in situ damage localization in plates,” *Wave Motion*, vol. 44, p. 482–492, 2007.

[211] X. Guan, J. Zhang, S. K. Zhou, E. Rasselkorde, and W. Abbasi, “Post-processing of phased-array ultrasonic inspection data with parallel computing for nondestructive evaluation,” *Journal of Nondestructive Evaluation*, vol. 33, no. 3, p. 342–351, 2014.

[212] R. Bai, X. Song, M. Radzieński, M. Cao, W. Ostachowicz, and S. Wang, “Crack location in beams by data fusion of fractal dimension features of laser-measured operating deflection shapes,” *Smart Structures and Systems*, vol. 13, no. 6, pp. 975–991, 2014.

[213] M. Katz, “Fractals and the analysis of waveforms,” *Comput. Biol. Med.*, vol. 18, no. 3, pp. 145–156, 1988.

[214] R. Heideklang and P. Shokouhi, “Application of data fusion in nondestructive testing (NDT),” *In: Proceedings of 16th international conference on information fusion*, 2013.

[215] J. Zhu, L. Wang, and H. Jiang, “Study on damage diagnosis indices fusion for long-span suspension bridges under ambient vibration,” *Mechanics of Structures and Materials: Advancements and Challenges*, pp. ISBN 978–1–138–02 993–4, 2016.

[216] C. Haynes and M. Todd, “Enhanced damage localization for complex structures through statistical modeling and sensor fusion,” *Mechanical Systems and Signal Processing*, vol. 54-55, pp. 195–209, 2015.

[217] C. Haynes, M. D. Todd, E. Flynn, and A. Croxford, “Statistically-based damage detection in geometrically-complex structures using ultrasonic interrogation,” *Structural Health Monitoring*, vol. 12, no. 2, pp. 141–152, 2012.

[218] R. A. Fisher, “The use of multiple measurements in taxonomic problems,” *Ann. Eugenics*, vol. 7, pp. 111–132, 1936.

[219] K. Salahshoor, M. Kordestani, and M. S. Khoshro, “Fault detection and diagnosis of an industrial steam turbine using fusion of svm (support vector machine) and anfis (adaptive neuro-fuzzy inference system) classifiers,” *Energy*, vol. 35, pp. 5472–5482, 2010.

[220] J.-S. Jang, “Anfis: adaptive network-based fuzzy inference systems,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. 23, no. 3, pp. 665–685, 1993.

[221] Y. L. Chen, M. Abdelbarr, M. R. Jahanshahi, and S. F. Masri, “Color and depth data fusion using an rgb-d sensor for inexpensive and contactless dynamic displacement-field measurement,” *Structural Control and Health Monitoring*, vol. 24, no. 11, p. e2000, 2017, e2000 stc.2000.

[222] A. Geiger, F. Moosmann, O. Car, and B. Schuster, "Automatic camera and range sensor calibration using a single shot," *in 2012 IEEE Int. Conf. Rob. Autom.*, p. 3936–3943, 2012.

[223] G. Bradski and A. Kaehler, "Learning opencv: Computer vision with the opencv library," *OReilly Media Inc, Sebastopol*, 2008.

[224] Y. L. Chen, M. R. Jahanshahi, P. Manjunatha, W. Gan, M. Abdelbarr, S. F. Masri, B. Becerik-Gerber, and J. P. Caffrey, "Inexpensive multimodal sensor fusion system for autonomous data acquisition of road surface conditions," *IEEE SENSORS JOURNAL*, vol. 16, no. 21, pp. 7731–7743, 2016.

[225] J. W. Park, S. H. Sim, and H. J. Jung, "Displacement estimation using multimetric data fusion," *IEEE/ASME Transactions On Mechatronics*, vol. 18, no. 6, pp. 1675–1682, 2013.

[226] H. S. Lee, Y. H. Hong, and H. W. Park, "Design of an fir filter for the displacement reconstruction using measured acceleration in low-frequency dominant structures," *Int. J. Numer. Methods Eng.*, vol. 82, no. 4, p. 403–434, 2010.

[227] S. Shin, S.-U. Lee, Y. Kim, and N.-S. Kim, "Estimation of bridge displacement responses using fbg sensors and theoretical mode shapes," *Struct. Eng. Mech.*, vol. 42, no. 2, p. 229–245, 2012.

[228] J. W. Park, S. H. Sim, and H. J. Jung, "Wireless displacement sensing system for bridges using multi-sensor fusion," *Smart Mater. Struct.*, vol. 23, p. 045022, 2014.

[229] S. H. Sim, "Estimation of flexibility matrix of beam structures using multisensor fusion," *Journal of Structural Integrity and Maintenance*, vol. 1, no. 2, p. 60–64, 2016.

[230] D. Bernal, "Extracting flexibility matrices from state-space realizations," *In COST F3 Conference*, vol. 1, p. 127–135, 2000.

[231] H. Kim, S. Cho, and S.-H. Sim, "Data fusion of acceleration and angular velocity for improved model updating," *Measurement*, vol. 91, p. 239–250, 2016.

[232] J.-W. Park, K.-C. Lee, S.-H. Sim, H.-J. Jung, and B. F. S. Jr., "Traffic safety evaluation for railway bridges using expanded multisensor data fusion," *Computer-Aided Civil and Infrastructure Engineering*, vol. 31, pp. 749–760, 2016.

[233] R. Heideklang and P. Shokouhi, "Decision-level fusion of spatially scattered multi-modal data for nondestructive inspection of surface defects," *Sensors*, vol. 16, no. 105, 2016.

[234] E. Parzen, "On estimation of a probability density function and mode," *Ann. Math. Stat.*, vol. 33, p. 1065–1076, 1962.

[235] J.-W. Park, D. Moon, H. Yoon, F. Gomez, B. F. S. Jr., and J. R. Kim, "Visual–inertial displacement sensing using data fusion of vision-based displacement with acceleration," *Struct Control Health Monit.*, 2017.

[236] B. Khaleghi, A. Khamis, F. O. Karray, and S. N. Razavi, "Multisensor data fusion: A review of the state-of-the-art," *Information Fusion*, vol. 14, no. 1, pp. 28–44, 2013.

[237] G. E. Hinton, "A practical guide to training restricted boltzmann machines," 2010.

[238] P. Tamilselvan and P. Wang, "Failure diagnosis using deep belief learning based health state classification," *Reliability Engineering and System Safety*, vol. 115, pp. 124–135, 2013.

[239] A. Y. Ng and M. I. Jordan, "On discriminative versus generative classifiers: a comparison of logistic regression and naïve bayes." *Neural Inform. Process. Syst.*, vol. 14, p. 605–610, 2001.

[240] S. Hochreiter and J. Schmidhuber, "Long short term memory," *Neural computation*, vol. 9, no. 8, p. 1735–1780, 1997.

[241] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, "Lstm-based encoder-decoder for multi-sensor anomaly detection," *ICML 2016 Anomaly Detection Workshop*, 2016.

[242] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," *31st Conference on Neural Information Processing Systems*, 2017.

[243] R. E. Bellman, *Dynamic Programming.* Princeton University Press, 1957.

[244] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[245] G. H. Golub and C. Reinsch, "Singular value decomposition and least squares solutions," *Numer. Math.*, vol. 14, no. 5, p. 403–420, 1970.

[246] M. Steinbach, L. Ertöz, and V. Kumar, "The challenges of clustering high dimensional data," *New Directions in Statistical Physics*, pp. 273–309, 2004.

[247] C. Batini and M. Scannapieco, "Data and information quality," *Springer*, 2016.

[248] L. M. Bruce and D. Reynolds, "Game theory based data fusion for precision agriculture applications," *Geoscience and Remote Sensing Symposium (IGARSS), 2016 IEEE International*, 2016.

[249] S. F. Masri, A. W. Smyth, A. G. Chassiakos, T. K. Caughey, and N. F. Hunter, "Application of neural networks for detection of changes in nonlinear systems," *JOURNAL OF ENGINEERING MECHANICS*, vol. 126, no. 7, pp. 666–676, 2000.

[250] J.-S. Pei, J. P. Wright, and A. W. Smyth, "Mapping polynomial fitting into feedforward neural networks for modeling nonlinear dynamic systems and beyond," *Comput. Methods Appl. Mech. Engrg.*, vol. 194, no. 42–44, p. 4481–4505, 2005.

[251] B. Xu, A. Gong, J. He, and S. F. Masri, “A novel time-domain structural parametric identification methodology based on the equivalency of neural networks and arma model,” *Proceedings of the 5th international conference on Emerging intelligent computing technology and applications*, vol. 79, pp. 888–897, September, Ulsan, South Korea 2009.

[252] B.-S. Wang, “Identifying damage of the benchmark structure by using artificial neural network methods,” *Advanced Materials Research*, vol. 219-220, pp. 312–317, 2011.

[253] A. Cury and C. Crémona, “Pattern recognition of structural behaviors based on learning algorithms and symbolic data concepts,” *STRUCTURAL CONTROL AND HEALTH MONITORING*, vol. 19, pp. 161–186, 2012.

[254] S. Arangio and J. L. Beck, “Bayesian neural networks for bridge integrity assessment,” *STRUCTURAL CONTROL AND HEALTH MONITORING*, vol. 19, pp. 3–21, 2012.

[255] S. Suresh, S. Narasimhan, and S. Nagarajaiah, “Direct adaptive neural controller for the active control of earthquake-excited nonlinear base-isolated buildings,” *STRUCTURAL CONTROL AND HEALTH MONITORING*, vol. 19, p. 370–384, 2012.

[256] C.-Y. Kao and C.-H. Loh, “Monitoring of long-term static deformation data of fei-tsui arch dam using artificial neural network-based approaches,” *STRUCTURAL CONTROL AND HEALTH MONITORING*, vol. 20, p. 282–303, 2013.

[257] S. V. Razavi, M. Jumaat, and H. E.-S. Ahmed, “Load-deflection analysis of cfrp strengthened rc slab using focused feed-forward time delay neural network,” *Concrete Research Letters*, vol. 5, no. 3, p. 858–872, 2014.

[258] M. Längkvist, L. Karlsson, and A. Loutfi, “A review of unsupervised feature learning and deep learning for time-series modeling,” *Pattern Recognition Letters*, vol. 42, p. 11–24, 2014.

[259] J. Tompson, K. Schlachter, P. Sprechmann, and K. Perlin, “Accelerating eulerian fluid simulation with convolutional networks,” *Proceedings of the 34 th International Conference on Machine Learning*, June, Sydney, Australia 2017.

[260] D. J. Atha and M. R. Jahanshahi, “Evaluation of deep learning approaches based on convolutional neural networks for corrosion detection,” *Structural Health Monitoring*, vol. 17, no. 5, pp. 1110–1128, 2018.

[261] K. Levenberg, “A method for the solution of certain non-linear problems in least squares,” *Quarterly of Applied Mathematics*, p. 164–168, 1944.

[262] D. Marquardt, “An algorithm for least-squares estimation of nonlinear parameters,” *SIAM Journal on Applied Mathematics*, vol. 11, no. 2, p. 431–441, 1963.

[263] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE 86*, vol. 11, p. 2278–2324, 1998.

[264] N. M. Newmark, “A method of computation for structural dynamics,” *Journal of the Engineering Mechanics Division*, vol. 85, no. 3, 1959.

[265] A. Vedaldi and K. Lenc, "Matconvnet – convolutional neural networks for matlab," in *Proceeding of the ACM Int. Conf. on Multimedia*, 2015.

[266] M. R. Jahanshahi and S. Masri, "Adaptive vision-based crack detection using 3d scene reconstruction for condition assessment of structures," *Automation in Construction*, vol. 22, pp. 567–576, 03 2012.

[267] M. R. Jahanshahi and S. F. Masri, "Parametric performance evaluation of wavelet-based corrosion detection algorithms for condition assessment of civil infrastructure systems," *Journal of Computing in Civil Engineering*, vol. 27, no. 4, pp. 345–357, 2013.

[268] M. R. Jahanshahi, F. Jazizadeh, S. F. Masri, and B. Becerik-Gerber, "Unsupervised approach for autonomous pavement-defect detection and quantification using an inexpensive depth sensor," *Journal of Computing in Civil Engineering*, vol. 27, no. 6, pp. 743–754, 2013.

[269] M. R. Jahanshahi, F. J. Karimi, S. F. Masri, and B. Becerik-Gerber, "Autonomous pavement condition assessment," U.S. Patent 9 196 048, Nov. 2015.

[270] M. H. Rafiei and H. Adeli, "A novel unsupervised deep learning model for global and local health condition assessment of structures," *Engineering Structures*, vol. 156, no. 1, pp. 598–607, 2018.

[271] K. Park, M. Torbol, and S. Kim, "Vision-based natural frequency identification using laser speckle imaging and parallel computing," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 1, pp. 51–63, 2018.

[272] J. Choi, C. M. Yeum, S. J. Dyke, and M. R. Jahanshahi, "Computer-aided approach for rapid post-event visual evaluation of a building facade," *Sensors*, vol. 18, no. 9, p. doi: 10.3390/s18093017, 2018.

[273] R.-T. Wu and M. R. Jahanshahi, "Data fusion approaches for structural health monitoring and system identification: Past, present, and future," *Structural Health Monitoring*, vol. 19, no. 2, pp. 552–586, 2020.

[274] E. Bertino and M. R. Jahanshahi, "Adaptive and cost-effective collection of high-quality data for critical infrastructure and emergency management in smart cities—framework and challenges," *Journal of Data and Information Quality*, vol. 10, pp. 1–6, 05 2018.

[275] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, "Swarm robotics: a review from the swarm engineering perspective," *Swarm Intelligence*, vol. 7, no. 1, pp. 1–41, Mar 2013. [Online]. Available: https://doi.org/10.1007/s11721-012-0075-2

[276] C. M. Yeum, S. Dyke, J. Ramirez, and B. Benes, "Big visual data analytics for damage classification in civil engineering," *Proceedings of the International Conference on Smart Infrastructure and Construction*, 2016.

[277] C. M. Yeum, S. J. Dyke, and J. Ramirez, "Visual data classification in post-event building reconnaissance," *Engineering Structures*, vol. 155, p. 16–24, 2018.

[278] S. S. Kumar, D. M. Abraham, M. R. Jahanshahi, T. Iseley, and J. Starr, "Automated defect classification in sewer closed circuit television inspections using deep convolutional neural networks," *Automation in Construction*, vol. 91, pp. 273–283, 2018.

[279] R. T. Wu and M. R. Jahanshahi, "Deep convolutional neural network for structural dynamic response estimation and system identification," *Journal of Engineering Mechanics (ASCE)*, vol. 145, no. 1, pp. DOI: 10.1061/(ASCE)EM.1943–7889.0 001 556, 2019.

[280] Y. Z. Lin, Z. H. Nie, and H. W. Ma, "Structural damage detection with automatic feature-extraction through deep learning," *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, no. 12, pp. 1025–1046, 2017.

[281] A. Zhang, K. C. P. Wang, B. Li, E. Yang, X. Dai, Y. Peng, Y. Fei, Y. Liu, J. Q. Li, and C. Chen, "Automated pixel-level pavement crack detection on 3d asphalt surfaces using a deep-learning network," *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, no. 10, pp. 805–819, 2017.

[282] Y. D. Xue and Y. Li, "A fast detection method via region-based fully convolutional neural networks for shield tunnel lining defects," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 8, pp. 638–654, 2018.

[283] Y. Cha, W. Choi, G. Suh, S. Mahmoudkhani, and O. Büyüköztürk, "Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 9, pp. 731–747, 2018.

[284] Y. Gao and K. Mosalam, "Deep transfer learning for image-based structural damage recognition," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 9, pp. 748–768, 2018.

[285] S. S. Kumar, M. Wang, D. M. Abraham, M. R. Jahanshahi, T. Iseley, and J. C. P. Cheng, "Deep learning&#x2013;based automated detection of sewer defects in cctv videos," *Journal of Computing in Civil Engineering*, vol. 34, no. 1, p. 04019047, 2020.

[286] S. Han, J. Pool, J. Tran, and W. J. Dally, "Learning both weights and connections for efficient neural networks," *arXiv preprint*, no. arXiv:1506.02626v3, 2015.

[287] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, Jul. 2002.

[288] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, Jun. 2005, pp. 886–893 vol. 1.

[289] Y. Guo, A. Yao, and Y. Chen, "Dynamic network surgery for efficient dnns," *arXiv preprint*, no. arXiv:1608.04493v2, 2016.

[290] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," *arXiv preprint*, no. arXiv:1510.00149v5, 2016.

[291] A. Aghasi, A. Abdi, N. Nguyen, and J. Romberg, "Net-trim: Convex pruning of deep neural networks with performance guarantee," *arXiv preprint*, no. arXiv:1611.05162v4, 2017.

[292] A. Zhou, A. Yao, Y. Guo, L. Xu, and Y. Chen, "Incremental network quantization: Towards lossless cnns with low-precision weights," *arXiv preprint*, no. arXiv:1702.03044v2, 2017.

[293] J. Yu, A. Lukefahr, D. Palframan, G. Dasika, R. Das, and S. Mahlke, "Scalpel: Customizing DNN pruning to the underlying hardware parallelism," in *Proceedings of the 44th Annual International Symposium on Computer Architecture*, ser. ISCA '17. New York, NY, USA: ACM, 2017, pp. 548–560. [Online]. Available: http://doi.acm.org/10.1145/3079856.3080215

[294] J.-H. Luo, J. Wu, and W. Lin, "Thinet: A filter level pruning method for deep neural network compression," *arXiv preprint*, no. arXiv:1707.06342v1, 2017.

[295] A. Mallya and S. Lazebnik, "Packnet: Adding multiple tasks to a single network by iterative pruning," *arXiv preprint*, no. arXiv:1711.05769v2, 2018.

[296] Z. Liu, M. Sun, T. Zhou, G. Huang, and T. Darrell, "Rethinking the value of network pruning," *arXiv preprint*, no. arXiv:1810.05270v1, 2018.

[297] H. Hu, R. Peng, Y.-W. Tai, and C.-K. Tang, "Network trimming: A data-driven neuron pruning approach towards efficient deep architectures," *arXiv preprint*, no. arXiv:1607.03250v1, 2018.

[298] Y. He, J. Lin, Z. Liu, H. Wang, L.-J. Li, and S. Han, "AMC: AutoML for model compression and acceleration on mobile devices," *arXiv preprint*, no. arXiv:1802.03494v4, 2019.

[299] M. R. Jahanshahi, F.-C. Chen, C. Joffe, and S. F. Masri, "Vision-based quantitative assessment of microcracks on reactor internal components of nuclear power plants," *Structure and Infrastructure Engineering*, vol. 13, no. 8, pp. 1013–1026, 2017.

[300] "Nvidia jetson solutions for drones and uavs," https://www.nvidia.com/en-us/autonomous-machines/uavs-drones-technology/, accessed: 2018-09-27.

[301] M. A. Akbar, U. Qidwai, and M. R. Jahanshahi, "An evaluation of image-based structural health monitoring using integrated unmanned aerial vehicle platform," *Structural Control and Health Monitoring*, vol. 26, no. 1, p. e2276, 2019, e2276 STC-17-0270.R1.

[302] Y. LeCun, J. S. Denker, S. Solla, R. E. Howard, and L. D. Jackel, "Optimal brain damage," *Advances in Neural Information Processing Systems (NIPS)*, 1990.

[303] S. Han, X. Liu, H. Mao, J. Pu, A. Pedram, M. A. Horowitz, and W. J. Dally, "Eie: Efficient inference engine on compressed deep neural network," *Proceedings of the 43rd International Symposium on Computer Architecture*, pp. 243–254, 2016.

[304] H. Hu, R. Peng, Y.-W. Tai, and C.-K. Tang, "Eie: Efficient inference engine on compressed deep neural network," *arXiv preprint*, no. arXiv:1607.03250, 2016.

[305] H. Zhou, J. M. Alvarez, and F. Porikli, "Less is more: Towards compact cnns," *European Conference on Computer Vision*, p. 662–677, 2016.

[306] J. M. Alvarez and M. Salzmann, "Learning the number of neurons in deep networks," *Advances in Neural Information Processing Systems*, vol. 29, p. 2262–2270, 2016.

[307] V. Lebedev and V. Lempitsky, "Fast convnets using group-wise brain damage," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, p. 2554–2564, 2016.

[308] L. Bottou, "Large-scale machine learning with stochastic gradient descent," *Proceedings of COMPSTAT' 2010*, pp. 177–186, 2010.

[309] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *NIPS-W*, 2017.

[310] K. Gopalakrishnan, H. Gholami, A. Vidyadharan, A. Choudhary, and A. Agrawal, "Crack damage detection in unmanned aerial vehicle images of civil infrastructure using pre-trained deep learning model," *International Journal for Traffic and Transport Engineering*, vol. 8, no. 1, pp. 1–14, 2018.

[311] Q. D. Cao and Y. Choe, "Deep learning based damage detection on post-hurricane satellite imagery," *arXiv preprint*, no. arXiv:1807.01688v1, 2018.

[312] H. Maeda, Y. Sekimoto, T. Seto, T. Kashiyama, and H. Omata, "Road damage detection using deep neural networks with images captured through a smartphone," *arXiv preprint*, no. arXiv:1801.09454v2, 2018.

[313] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *arXiv preprint*, no. arXiv:1406.4729v4, 2015.

[314] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[315] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[316] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin., "Liblinear: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[317] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning." in *OSDI*, vol. 16, 2016, pp. 265–283.

[318] F. Chollet, "keras," *GitHub*, 2015, https://github.com/fchollet/keras.

[319] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[320] J. Vasseur, P. A. Deymier, B. Chenni, B. Djafari-Rouhani, L. Dobrzynski, and D. Prevost, "Experimental and theoretical evidence for the existence of absolute acoustic band gaps in two-dimensional solid phononic crystals," *Physical Review Letters*, vol. 86, no. 14, p. 3012, 2001.

[321] T.-T. Wu, L.-C. Wu, and Z.-G. Huang, "Frequency band-gap measurement of two-dimensional air/silicon phononic crystals using layered slanted finger interdigital transducers," *Journal of Applied Physics*, vol. 97, no. 9, p. 094916, 2005.

[322] M. Oudich, M. Senesi, M. B. Assouar, M. Ruzenne, J.-H. Sun, B. Vincent, Z. Hou, and T.-T. Wu, "Experimental evidence of locally resonant sonic band gap in two-dimensional phononic stubbed plates," *Physical Review B*, vol. 84, no. 16, p. 165136, 2011.

[323] T.-W. Liu, Y.-C. Lin, Y.-C. Tsai, T. Ono, S. Tanaka, and T.-T. Wu, "Evidence of a love wave bandgap in a quartz substrate coated with a phononic thin layer," *Applied Physics Letters*, vol. 104, no. 18, p. 181905, 2014.

[324] A. Sukhovich, B. Merheb, K. Muralidharan, J. Vasseur, Y. Pennec, P. A. Deymier, and J. Page, "Experimental and theoretical evidence for subwavelength imaging in phononic crystals," *Physical review letters*, vol. 102, no. 15, p. 154301, 2009.

[325] X. Zhang and Z. Liu, "Negative refraction of acoustic waves in two-dimensional phononic crystals," *Applied Physics Letters*, vol. 85, no. 2, pp. 341–343, 2004.

[326] Y. Xie, W. Wang, H. Chen, A. Konneker, B.-I. Popa, and S. A. Cummer, "Wavefront modulation and subwavelength diffractive acoustics with an acoustic metasurface," *Nature communications*, vol. 5, p. 5553, 2014.

[327] H. Zhu and F. Semperlotti, "Anomalous refraction of acoustic guided waves in solids with geometrically tapered metasurfaces," *Physical review letters*, vol. 117, no. 3, p. 034302, 2016.

[328] S.-C. S. Lin, T. J. Huang, J.-H. Sun, and T.-T. Wu, "Gradient-index phononic crystals," *Physical Review B*, vol. 79, no. 9, p. 094302, 2009.

[329] S. A. Cummer and D. Schurig, "One path to acoustic cloaking," *New Journal of Physics*, vol. 9, no. 3, p. 45, 2007.

[330] H. Chen and C. Chan, "Acoustic cloaking in three dimensions using acoustic metamaterials," *Applied physics letters*, vol. 91, no. 18, p. 183518, 2007.

[331] B.-I. Popa, L. Zigoneanu, and S. A. Cummer, "Experimental acoustic ground cloak in air," *Physical review letters*, vol. 106, no. 25, p. 253901, 2011.

[332] H. Zhu and F. Semperlotti, “Double-zero-index structural phononic waveguides,” *Physical Review Applied*, vol. 8, no. 6, p. 064031, 2017.

[333] P. Wang, L. Lu, and K. Bertoldi, “Topological phononic crystals with one-way elastic edge waves,” *Physical review letters*, vol. 115, no. 10, p. 104302, 2015.

[334] C. He, X. Ni, H. Ge, X.-C. Sun, Y.-B. Chen, M.-H. Lu, X.-P. Liu, and Y.-F. Chen, “Acoustic topological insulator and robust one-way sound transport,” *Nature Physics*, vol. 12, no. 12, p. 1124, 2016.

[335] S. H. Mousavi, A. B. Khanikaev, and Z. Wang, “Topologically protected elastic waves in phononic metamaterials,” *Nature communications*, vol. 6, p. 8682, 2015.

[336] T.-W. Liu and F. Semperlotti, “Experimental evidence of robust acoustic valley hall edge states in a nonresonant topological elastic waveguide,” *Physical Review Applied*, vol. 11, no. 1, p. 014040, 2019.

[337] H. Huang, C. Sun, and G. Huang, “On the negative effective mass density in acoustic metamaterials,” *International Journal of Engineering Science*, vol. 47, no. 4, pp. 610–617, 2009.

[338] L. Zhao, S. C. Conlon, and F. Semperlotti, “Broadband energy harvesting using acoustic black hole structural tailoring,” *Smart materials and structures*, vol. 23, no. 6, p. 065021, 2014.

[339] L. Zhao and F. Semperlotti, “Embedded acoustic black holes for semi-passive broadband vibration attenuation in thin-walled structures,” *Journal of Sound and Vibration*, vol. 388, pp. 42 – 52, 2017.

[340] K. T. Tan, H. Huang, and C. Sun, “Optimizing the band gap of effective mass negativity in acoustic metamaterials,” *Applied Physics Letters*, vol. 101, no. 24, p. 241902, 2012.

[341] T.-T. Wu, Y.-T. Chen, J.-H. Sun, S.-C. S. Lin, and T. J. Huang, “Focusing of the lowest antisymmetric lamb wave in a gradient-index phononic crystal plate,” *Applied Physics Letters*, vol. 98, no. 17, p. 171911, 2011.

[342] T.-W. Liu, Y.-C. Tsai, Y.-C. Lin, T. Ono, S. Tanaka, and T.-T. Wu, “Design and fabrication of a phononic-crystal-based love wave resonator in ghz range,” *AIP Advances*, vol. 4, no. 12, p. 124201, 2014.

[343] P. A. Feurtado, S. C. Conlon, and F. Semperlotti, “A normalized wave number variation parameter for acoustic black hole design,” *The Journal of the Acoustical Society of America*, vol. 136, no. 2, pp. EL148–EL152, 2014.

[344] C.-M. Lin, J.-C. Hsu, D. G. Senesky, and A. P. Pisano, “Anchor loss reduction in aln lamb wave resonators using phononic crystal strip tethers,” in *2014 IEEE International Frequency Control Symposium (FCS)*. IEEE, 2014, pp. 1–5.

[345] M. Y. Wang, X. Wang, and D. Guo, “A level set method for structural topology optimization,” *Computer methods in applied mechanics and engineering*, vol. 192, no. 1-2, pp. 227–246, 2003.

[346] G. Allaire, F. Jouve, and A.-M. Toader, "A level-set method for shape optimization," *Comptes Rendus Mathematique*, vol. 334, no. 12, pp. 1125–1130, 2002.

[347] ——, "Structural optimization using sensitivity analysis and a level-set method," *Journal of computational physics*, vol. 194, no. 1, pp. 363–393, 2004.

[348] Y. Wang, Z. Luo, N. Zhang, and Z. Kang, "Topological shape optimization of microstructural metamaterials using a level set method," *Computational Materials Science*, vol. 87, pp. 178–186, 2014.

[349] L. Lu, T. Yamamoto, M. Otomori, T. Yamada, K. Izui, and S. Nishiwaki, "Topology optimization of an acoustic metamaterial with negative bulk modulus using local resonance," *Finite Elements in Analysis and Design*, vol. 72, pp. 1–12, 2013.

[350] O. Sigmund and J. Søndergaard Jensen, "Systematic design of phononic band–gap materials and structures by topology optimization," *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 361, no. 1806, pp. 1001–1019, 2003.

[351] C. J. Rupp, A. Evgrafov, K. Maute, and M. L. Dunn, "Design of phononic materials/structures for surface wave devices using topology optimization," *Structural and Multidisciplinary Optimization*, vol. 34, no. 2, pp. 111–121, 2007.

[352] H.-W. Dong, S.-D. Zhao, Y.-S. Wang, and C. Zhang, "Topology optimization of anisotropic broadband double-negative elastic metamaterials," *Journal of the Mechanics and Physics of Solids*, vol. 105, pp. 54–80, 2017.

[353] J. H. Oh, Y. K. Ahn, and Y. Y. Kim, "Maximization of operating frequency ranges of hyperbolic elastic metamaterials by topology optimization," *Structural and Multidisciplinary Optimization*, vol. 52, no. 6, pp. 1023–1040, 2015.

[354] J. S. Jensen and O. Sigmund, "Phononic band gap structures as optimal designs," in *IUTAM symposium on asymptotics, singularities and homogenisation in problems of mechanics.* Springer, 2003, pp. 73–83.

[355] G. Yi and B. D. Youn, "A comprehensive survey on topology optimization of phononic crystals," *Structural and Multidisciplinary Optimization*, vol. 54, no. 5, pp. 1315–1344, 2016.

[356] M. Wormser, F. Wein, M. Stingl, and C. Körner, "Design and additive manufacturing of 3d phononic band gap structures based on gradient based optimization," *Materials*, vol. 10, no. 10, p. 1125, 2017.

[357] J. H. Park, P. S. Ma, and Y. Y. Kim, "Design of phononic crystals for self-collimation of elastic waves using topology optimization method," *Structural and Multidisciplinary Optimization*, vol. 51, no. 6, pp. 1199–1209, 2015.

[358] G. A. Gazonas, D. S. Weile, R. Wildman, and A. Mohan, "Genetic algorithm optimization of phononic bandgap structures," *International journal of solids and structures*, vol. 43, no. 18-19, pp. 5851–5866, 2006.

[359] H.-W. Dong, X.-X. Su, Y.-S. Wang, and C. Zhang, "Topological optimization of two-dimensional phononic crystals based on the finite element method and genetic algorithm," *Structural and Multidisciplinary Optimization*, vol. 50, no. 4, pp. 593–604, 2014.

[360] H. Meng, J. Wen, H. Zhao, and X. Wen, "Optimization of locally resonant acoustic metamaterials on underwater sound absorption characteristics," *Journal of Sound and Vibration*, vol. 331, no. 20, pp. 4406–4416, 2012.

[361] Z.-f. Liu, B. Wu, and C.-f. He, "Band-gap optimization of two-dimensional phononic crystals based on genetic algorithm and fpwe," *Waves in Random and Complex Media*, vol. 24, no. 3, pp. 286–305, 2014.

[362] O. R. Bilal and M. I. Hussein, "Ultrawide phononic band gap for combined in-plane and out-of-plane waves," *Physical Review E*, vol. 84, no. 6, p. 065701, 2011.

[363] D. Li, L. Zigoneanu, B.-I. Popa, and S. A. Cummer, "Design of an acoustic metamaterial lens using genetic algorithms," *The Journal of the Acoustical Society of America*, vol. 132, no. 4, pp. 2823–2833, 2012.

[364] M. I. Hussein, K. Hamza, G. M. Hulbert, and K. Saitou, "Optimal synthesis of 2d phononic crystals for broadband frequency isolation," *Waves in Random and Complex Media*, vol. 17, no. 4, pp. 491–510, 2007.

[365] A. Bacigalupo, G. Gnecco, M. Lepidi, and L. Gambarotta, "Machine-learning techniques for the optimal design of acoustic metamaterials," *Journal of Optimization Theory and Applications*, 2019.

[366] W. Ma, F. Cheng, and Y. Liu, "Deep-learning-enabled on-demand design of chiral metamaterials," *ACS Nano*, vol. 12, no. 6, pp. 6326–6334, 2018, pMID: 29856595. [Online]. Available: https://doi.org/10.1021/acsnano.8b03569

[367] W. Ma, F. Cheng, Y. Xu, Q. Wen, and Y. Liu, "Probabilistic representation and inverse design of metamaterials based on a deep generative model with semi-supervised learning strategy," *Advanced Materials*, vol. 31, no. 35, p. 1901111, 2019.

[368] X. Li, S. Ning, Z. Liu, Z. Yan, C. Luo, and Z. Zhuang, "Designing phononic crystal with anticipated band gap through a deep learning based data-driven method," *Computer Methods in Applied Mechanics and Engineering*, vol. 361, p. 112737, 2020.

[369] D. A. White, W. J. Arrighi, J. Kudo, and S. E. Watts, "Multiscale topology optimization using neural network surrogate models," *Computer Methods in Applied Mechanics and Engineering*, vol. 346, pp. 1118 – 1135, 2019.

[370] A. Gosavi, *Simulation-Based Optimization: Parametric Optimization Techniques and Reinforcement Learning*. New York: Springer, 2015.

[371] H. van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," *arXiv preprint*, no. arXiv:1509.06461v3, 2015.

[372] Z. Wang, T. Schaul, M. Hessel, H. van Hasselt, M. Lanctot, and N. de Freitas, "Dueling network architectures for deep reinforcement learning," *arXiv preprint*, no. arXiv:1511.06581v3, 2016.

[373] G. Lample and D. S. Chaplot, "Playing fps games with deep reinforcement learning," *arXiv preprint*, no. arXiv:1609.05521v2, 2018.

[374] K. Adil, F. Jiang, S. Liu, A. Grigorev, B. Gupta, and S. Rho, "Training an agent for fps doom game using visual reinforcement learning and vizdoom." in *International Journal of Advanced Computer Science and Applications, IJACSA 2017, Vol. 8, No. 12*, 2017.

[375] C. Kittel, P. McEuen, and P. McEuen, *Introduction to solid state physics.* Wiley New York, 1996, vol. 8.

[376] L. Meirovitch, *Analytical methods in vibrations*, ser. Macmillan series in advanced mathematics and theoretical physics. Macmillan, 1967.

[377] S. Chiriţă and R. Quintanilla, "On saint-venant's principle in linear elastodynamics," *Journal of elasticity*, vol. 42, no. 3, pp. 201–215, 1996.

[378] R. Bellman, "On the theory of dynamic programming," *Proceedings of the National Academy of Sciences*, vol. 38, no. 8, pp. 716–719, 1952.

[379] R. H. Byrd, M. E. Hribar, and J. Nocedal, "An interior point algorithm for large-scale nonlinear programming," *SIAM Journal on Optimization*, vol. 9, no. 4, p. 877–900, 1999.

[380] H. Zhu and F. Semperlotti, "Anomalous refraction of acoustic guided waves in solids with geometrically tapered metasurfaces," *Physical review letters*, vol. 117, no. 3, p. 034302, 2016.

[381] X. Ni, Z. J. Wong, M. Mrejen, Y. Wang, and X. Zhang, "An ultrathin invisibility skin cloak for visible light," *Science*, vol. 349, no. 6254, pp. 1310–1314, 2015.

[382] Y. Li, B. Liang, X. Tao, X.-f. Zhu, X.-y. Zou, and J.-c. Cheng, "Acoustic focusing by coiling up space," *Applied Physics Letters*, vol. 101, no. 23, p. 233508, 2012.

[383] S. Guenneau, A. Movchan, G. Pétursson, and S. A. Ramakrishna, "Acoustic metamaterials for sound focusing and confinement," *New Journal of physics*, vol. 9, no. 11, p. 399, 2007.

[384] H. Zhu and F. Semperlotti, "Two-dimensional structure-embedded acoustic lenses based on periodic acoustic black holes," *Journal of Applied Physics*, vol. 122, no. 6, p. 065104, 2017.

[385] O. A. Kaya, A. Cicek, and B. Ulug, "Self-collimated slow sound in sonic crystals," *Journal of Physics D: Applied Physics*, vol. 45, no. 36, p. 365101, 2012.

[386] J. Bucay, E. Roussel, J. Vasseur, P. A. Deymier, A. Hladky-Hennion, Y. Pennec, K. Muralidharan, B. Djafari-Rouhani, and B. Dubus, "Positive, negative, zero refraction, and beam splitting in a solid/air phononic crystal: Theoretical and experimental study," *Physical Review B*, vol. 79, no. 21, p. 214305, 2009.

[387] H. Chen and C. Chan, "Acoustic cloaking in three dimensions using acoustic metamaterials," *Applied physics letters*, vol. 91, no. 18, p. 183518, 2007.

[388] T. J. Cui, D. R. Smith, and R. Liu, *Metamaterials.* Springer, 2010.

[389] M. Dubois, C. Shi, X. Zhu, Y. Wang, and X. Zhang, "Observation of acoustic dirac-like cone and double zero refractive index," *Nature communications*, vol. 8, no. 1, pp. 1–6, 2017.

[390] M. V. Zhelyeznyakov, A. Zhan, and A. Majumdar, "Design and optimization of ellipsoid scatterer-based metasurfaces via the inverse t-matrix method," *OSA Continuum*, vol. 3, no. 1, pp. 89–103, 2020.

[391] P. Packo, A. N. Norris, and D. Torrent, "Inverse grating problem: Efficient design of anomalous flexural wave reflectors and refractors," *Physical Review Applied*, vol. 11, no. 1, p. 014023, 2019.

[392] Z. Lu, L. Sanchis, J. Wen, L. Cai, Y. Bi, and J. Sánchez-Dehesa, "Acoustic cloak based on bézier scatterers," *Scientific reports*, vol. 8, no. 1, pp. 1–10, 2018.

[393] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 21, no. 3, pp. 660–674, 1991.

[394] Y. Liu, T. Zhao, W. Ju, and S. Shi, "Materials discovery and design using machine learning," *Journal of Materiomics*, vol. 3, no. 3, pp. 159 – 177, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2352847817300515

[395] W. Lu, R. Xiao, J. Yang, H. Li, and W. Zhang, "Data mining-aided materials discovery and optimization," *Journal of Materiomics*, vol. 3, no. 3, pp. 191 – 201, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2352847817300618

[396] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, 1986.

[397] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," *arXiv e-prints*, p. arXiv:1312.6114, Dec. 2013.

[398] N. Zeng, H. Zhang, B. Song, W. Liu, Y. Li, and A. M. Dobaie, "Facial expression recognition via learning deep sparse autoencoders," *Neurocomputing*, vol. 273, pp. 643 – 649, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231217314649

[399] I. Malkiel, M. Mrejen, A. Nagler, U. Arieli, L. Wolf, and H. Suchowski, "Plasmonic nanostructure design and characterization via deep learning," *Light: Science and Applications*, vol. 7, no. 60, 2018.

[400] Y. Chen, J. Zhu, Y. Xie, N. Feng, and Q. H. Liu, "Smart inverse design of graphene-based photonic metamaterials by an adaptive artificial neural network," *Nanoscale*, vol. 11, pp. 9749–9755, 2019. [Online]. Available: http://dx.doi.org/10.1039/C9NR01315F

[401] M. H. Tahersima, K. Kojima, T. Koike-Akino, D. Jha, B. Wang, C. Lin, and K. Parsons, "Deep Neural Network Inverse Design of Integrated Photonic Power Splitters," *Scientific Reports*, vol. 9, p. 1368, Feb. 2019.

[402] Z. Liu, D. Zhu, S. P. Rodrigues, K.-T. Lee, and W. Cai, "Generative model for the inverse design of metasurfaces," *Nano Letters*, vol. 18, no. 10, pp. 6570–6576, 2018, pMID: 30207735. [Online]. Available: https://doi.org/10.1021/acs.nanolett.8b03171

[403] A. Marzo and B. W. Drinkwater, "Holographic acoustic tweezers," *Proceedings of the National Academy of Sciences*, vol. 116, no. 1, pp. 84–89, 2019.

[404] R. Hirayama, D. M. Plasencia, N. Masuda, and S. Subramanian, "A volumetric display for visual, tactile and audio presentation using acoustic trapping," *Nature*, vol. 575, no. 7782, pp. 320–323, 2019.

[405] L. Piegl and W. Tiller, *The NURBS book.* Springer Science & Business Media, 2012.

[406] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[407] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," *arXiv e-prints*, p. arXiv:1603.07285, Mar. 2016.