Condition Assessment of Civil Infrastructure and Materials Using Deep Learning

Fangyu Liu

Dissertation submitted to the faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
In
Civil Engineering

Linbing Wang, Chair
Alexander Brand
Susan Hotle
Liqing Zhang

July 21, 2022
Blacksburg, Virginia

Keywords: Deep learning; Pavement; Concrete; Infrared thermography; Vibration

Copyright © 2022 by Fangyu Liu

Condition Assessment of Civil Infrastructure and Materials Using Deep Learning

Fangyu Liu

ABSTRACT

The abilities of powerful regression and multi-type data processing allow deep learning to effectively and accurately complete multi-tasks, which is the need of civil engineering. More cases showed that deep learning has become a greatly powerful and increasingly popular tool for civil engineering. Based on these, this dissertation developed deep learning studies for the condition assessment of civil infrastructure and materials. This dissertation included five main works: (1) Deep learning and infrared thermography for asphalt pavement crack severity classification. This work focused on longitudinal or transverse cracking. This work first built a dataset with four severity levels (no, low-severity, medium-severity, and high-severity) and three image types (visible, infrared, and fusion). Then this work applied the convolutional neural network (CNN) to classify the crack severity based on two strategies deep learning from scratch and transfer learning). This work also investigated the effect of image types on the accuracy of these two strategies and on the classification of different severity levels. (2) Asphalt pavement crack detection based on convolutional neural network and infrared thermography. This work first built an open dataset with three image types (visible, infrared, and fusion) and different conditions (single, multi, thin, and thick cracks; clean, rough, light, and dark backgrounds) and periods (morning, noon, and dusk). Then this work evaluated the performance of the CNN model based on the accuracy and complexity (computational and model). (3) An artificial neural network model on tensile behavior of hybrid steel-PVA fiber reinforced concrete containing fly ash and slag powder. This work considered a total of 23 factors for predicting the tensile behavior of hybrid fiber reinforced concrete (HFRC), including fibers' characteristics, mechanical properties of plain concrete, and concrete composition. Then this work compared the performance of the artificial neural network (ANN) method and the traditional equation-based method in terms of predicting the tensile stress, tensile strength, and strain corresponding to tensile strength. (4) Deep transfer learning-based vehicle classification by asphalt pavement vibration. This work first applied the pavement vibration IoT monitoring system to collect raw vibration signals and performed the wavelet transform to obtain denoised vibration signals. Then this work represented the vibration signals in two different ways, including the time-domain graph and the time-frequency graph. Finally, this work proposed two deep transfer learning-based vehicle classification methods according to these two representations of vibration signals. (5) Physical-informed long short-term memory (PI-LSTM) network for data-driven structural response modeling. This work first applied the single-degree-of-freedom (SDOF) system to investigate the performance of the proposed PI-LSTM network compared with the existing methods. Then this work further investigated and validated the proposed PI-LSTM network in terms of the experimental results of one six-story building and the numerical simulation results of another six-story building.

Condition Assessment of Civil Infrastructure and Materials Using Deep Learning

Fangyu Liu

GENERAL AUDIENCE ABSTRACT

With the development of technologies, deep learning has been applied to numerous fields to improve accuracy and efficiency. More work shows that deep learning has become a greatly powerful and increasingly popular tool for civil engineering. Since civil infrastructure and materials play a dominant role in civil engineering, this dissertation applied deep learning to the condition assessment of civil infrastructure and materials. Deep learning methods were applied to detect cracks in asphalt pavements. The mechanical properties of fiber reinforced concrete were investigated by deep learning methods. Based on the asphalt pavement vibration, the type of vehicles was classified by deep learning methods. Deep learning methods were also used to investigate the structural response.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my utmost gratitude and appreciation to my advisor, Dr. Linbing Wang, for accepting me as his Ph.D. student and for his guidance, advice, encouragement, and support during my Ph.D. studies. The research road is so long and full of difficulties and challenges. His dedication and sincerity to science and engineering have always inspired and motivated me to keep going on this road. The research freedom, advice, encouragement, and support he has provided enabled me to follow my heart to do the research I like, making me feel so excited, happy, and passionate to continue on this road. It's my honor to become his Ph.D. student.

I would like to sincerely thank my committee members, Dr. Alexander Brand, Dr. Susan Hotle, and Dr. Liqing Zhang, for serving on my committee and for their review, advice, suggestions, and comments on my dissertation.

In addition, I am grateful to my nice colleagues and friends at Virginia Tech for their encouragement and help. I would like to thank Jian Liu for his generous help in my research and graduate life.

Last but not least, I would like to express my warmest and deepest gratitude and thankfulness to my parents, my grandparents, and my parents-in-law for their unconditional love and support. I would like to express my biggest thanks to my wife, Yanran Xu, for all her love, encouragement, support, understanding, and patience. Words cannot express my thanks. She always encourages me, gives me advice, takes care of me, understands, and supports me so that I could overcome all kinds of difficulties and keep going forward.

TABLE OF CONTENT

Chapter 1 INTRODUCTION 1

1.1 Deep learning in static setups 3

1.2 Deep learning in dynamic setups 5

1.3 Objective and Outline 7

References 7

Chapter 2 DEEP LEARNING AND INFRARED THERMOGRAPHY FOR ASPHALT PAVEMENT CRACK SEVERITY CLASSIFICATION 12

2.1 Introduction 12

2.2 Methods 14

2.2.1 Deep learning from scratch 14

2.2.2 Transfer learning 14

2.2.3 Evaluation metrics 15

2.2.4 Hyperparameters of models 17

2.3 Dataset 17

2.4 Results 19

2.4.1 Deep learning from scratch 19

2.4.2 Transfer learning vs. deep learning from scratch 24

2.5 Discussion 26

2.5.1 Effects of image types on deep learning from scratch 26

2.5.2 Effects of image types on transfer learning 28

2.6 Conclusion 31

2.7 Reprint and Permission 32

References 32

Chapter 3 ASPHALT PAVEMENT CRACK DETECTION BASED ON CONVOLUTIONAL NEURAL NETWORK AND INFRARED THERMOGRAPHY 35

3.1 Introduction 35

3.2 Methods 36

3.2.1 Infrared Camera 36

3.2.2 Segmentation Models 37
3.2.3 Evaluation Metrics 38
3.2.3 Hyperparameters of Models 39
3.3 Dataset 39
3.4 Results and Discussion 43
3.4.1 Loss 43
3.4.2 Evaluation Metrics 44
3.4.3 Prediction 51
3.5 Conclusion 56
3.6 Reprint and Permission 56
References 57
Chapter 4 AN ARTIFICIAL NEURAL NETWORK MODEL ON TENSILE BEHAVIOR OF HYBRID STEEL-PVA FIBER REINFORCED CONCRETE CONTAINING FLY ASH AND SLAG POWDER 60
4.1 Introduction 60
4.2 Framework of the ANN model 61
4.2.1 ANN Approach 61
4.2.2 Procedures of the ANN model 63
4.3 Database of the tensile behavior of HFRC 66
4.4 Performance of the ANN model 69
4.4.1 Indices for evaluating model 69
4.4.2 ANN model training and evaluating 70
4.4.3 Equation-based model 71
4.4.4 Model performance analysis 73
4.5 Sensitivity Analysis and Discussion 75
4.5.1 Sensitivity Analysis 75
4.5.2 Discussion 77
4.6 Conclusions 78
4.7 Reprint and Permision 79
References 79

Chapter 5 DEEP TRANSFER LEARNING-BASED VEHICLE CLASSIFICATION BY ASPHALT PAVEMENT VIBRATION 82
5.1 Introduction 82
5.2 Methods 84
5.2.1 Monitoring pavement vibration 85
5.2.2 Denoising vibration signals 86
5.2.3 Classifying vehicles 87
5.2.3.4 Image preprocessing and hyperparameters 91
5.3 Results 91
5.3.1 Denoised vibration signals 91
5.3.2 Datasets for deep transfer learning 93
5.3.3 Accuracy metrics of vehicle classification 95
5.4 Discussion 99
5.4.1 Difference in input types 99
5.4.2 Difference in accuracy metrics 100
5.4.3 Difference in application prospects 103
5.5 Conclusion 104
5.6 Reprint and Permision 104
References 104
Chapter 6 PI-LSTM: PHYSICS-INFORMED LONG SHORT-TERM MEMORY NETWORK FOR DATA-DRIVEN STRUCTURAL RESPONSE MODELING 108
6.1 Introduction 108
6.2 Physics-informed long short-term memory (PI-LSTM) 109
6.3 SDOF numerical validation 112
6.3.1 Comparison with the existing model 112
6.3.2 Case study 113
6.4 Experimental and numeral validation 119
6.4.1 Experimental validation 119
6.4.2 Numerical validation 122

6.5 Conclusion 125
References 126
Chapter 7 CONCLUSION 129
7.1 Overview 129
7.2 Major findings 129
7.3 Recommendation for future work 130

LIST OF FIGURES

Figure 1-1 Scheme of deep learning in static and dynamic setups. (a) Scheme of static setups and (b) Scheme of dynamic setups 2
Figure 1-2 Scheme of different tasks for structural damage detection 4
Figure 2-1 Framework of deep learning and infrared thermography for asphalt pavement crack severity classification 14
Figure 2-2 Scheme of transfer learning with CNN for asphalt pavement severity classification 15
Figure 2-3 Examples of the visible image, infrared image, and fusion image in the dataset. (a) No crack, (b) low crack, (c) medium crack, and (d) high crack 18
Figure 2-4 Loss curves of the test set for deep learning from scratch. (a) Visible image, (b) infrared image, and (c) fusion image. 19
Figure 2-5 Accuracy metrics of the test set on the visible image for deep learning from scratch. (a) Accuracy, (b) macro F_1 score, and (c) weighted F_1 score. 21
Figure 2-6 Accuracy metrics of the test set on the infrared image for deep learning from scratch. (a) Accuracy, (b) macro F_1 score, and (c) weighted F_1 score. 21
Figure 2-7 Accuracy metrics of the test set on the fusion image for deep learning from scratch. (a) Accuracy, (b) macro F_1 score, and (c) weighted F_1 score. 21
Figure 2-8 Accuracy metrics of CNN models on three types of images. (a) Accuracy, (b) macro F_1 score, and (c) weighted F_1 score. 23
Figure 2-9 Loss curves of the test set for deep learning from scratch and transfer learning. (a) Visible image, (b) infrared image, and (c) fusion image. 24
Figure 2-10 Accuracy metrics of the test set on the visible image for deep learning from scratch and transfer learning. (a) Accuracy, (b) macro F_1 score, and (c) weighted F_1 score. 25
Figure 2-11 Accuracy metrics of the test set on the infrared image for deep learning from scratch and transfer learning. (a) Accuracy, (b) macro F_1 score, and (c) weighted F_1 score. 25
Figure 2-12 Accuracy metrics of the test set on the fusion image for deep learning from scratch and transfer learning. (a) Accuracy, (b) macro F_1 score, and (c) weighted F_1 score. 25
Figure 2-13 Normalized confusion matrix on the visible image. (a) EfficientNet-B3 for deep learning from scratch and (b) EfficientNet-B3 for transfer learning. It only displays values greater than 0.05. 27
Figure 2-14 Normalized confusion matrix on the infrared image. (a) EfficientNet-B3 for deep learning from scratch and (b) EfficientNet-B3 for transfer learning. It only displays values greater than 0.05. 27
Figure 2-15 Normalized confusion matrix on the fusion image. (a) EfficientNet-B3 for deep learning from scratch and (b) EfficientNet-B3 for transfer learning. It only displays values greater than 0.05. 28
Figure 2-16 Examples of the visible image, infrared image, and fusion image in the dataset. (a) and (b) are medium crack, and (c) and (d) are high crack. 28
Figure 2-17 Accuracy of CNN models on three types of images as the pre-trained convolutional layers are fixed. 30

Figure 2-18 RGB distribution of the examples of the visible image, infrared image, and fusion image in the dataset. (a) No crack, (b) low crack, (c) medium crack, and (d) high crack. Images were used according to Figure 2-3. 31
Figure 3-1 Examples of (a) the visible image and (b) infrared image 36
Figure 3-2 Overview of the infrared camera and software. (a) The front of the infrared camera (including the visible light camera lens and infrared camera lens), (b) the back of the infrared camera, and (c) the software used to process images (including temperature information and blending level of images). 37
Figure 3-3 Examples of the visible image, infrared image, and fusion image at the same location in three periods. (a) Morning, (b) noon, and (c) dusk. 40
Figure 3-4 Examples of the visible image, infrared image, fusion image, and ground truth in the dataset. (a) Single crack, (b) multi cracks, (c) thin crack, (d) thick crack, (e) clean background, (f) rough background, (g) light background, and (h) dark background. 42
Figure 3-5 Loss curves of the test set. (a) Visible images, (b) Infrared images, (c) Fusion images 43
Figure 3-6 Accuracy metrics of the visible image. (a) F_1 score, (b) PA, (c) MPA, and (d) MIoU 45
Figure 3-7 Accuracy metrics of the infrared image. (a) F_1 score, (b) PA, (c) MPA, and (d) MIoU 45
Figure 3-8 Accuracy metrics of the fusion image. (a) F_1 score, (b) PA, (c) MPA, and (d) MIoU 46
Figure 3-9 Accuracy metrics of segmentation models for three types of images. (a) F_1 score, (b) PA, (c) MPA, and (d) MIoU. 48
Figure 3-10 Ball chart reporting the accuracy metrics (MIoU and MPA) vs. computational complexity (floating-point operations (FLOPs), G-FLOPs=$1x10^9$ FLOPs) with same thresholds. The size of each ball corresponds to the number of parameters for each model. (a) MIoU and (b) MPA for the visible image, (c) MIoU and (d) MPA for the infrared image, (e) MIoU and (f) MPA for the fusion image. 50
Figure 3-11 Predictions of three segmentation models for different cracks based on three types of images. (a) Single crack, (b) multi cracks, (c) thin cracks, and (d) thick cracks. 52
Figure 3-12 Predictions of three segmentation models for different backgrounds based on three types of images. (a) Clean background, (b) rough background, (c) light background, and (d) dark background. 53
Figure 3-13 Predictions of three segmentation models for different periods based on three types of images. (a) Morning, (b) noon, and (c) dusk. 54
Figure 3-14 Predictions of three segmentation models for different epochs based on three types of images. (a) FPN, (b) DeepLabv3, and (c) UNet-VGG19. 55
Figure 4-1 5-fold cross-validation 62
Figure 4-2 Scheme of the ANN approach in this study 62
Figure 4-3 Framework of the ANN model in this study 63
Figure 4-4 Tensile stress-strain curves of (a) $S_{0.5}P_{0.5}$, (b) $S_{0.5}P_{1.0}$, (c) $S_{0.5}P_{1.5}$, (d) $S_{1.0}P_{0.5}$, (e) $S_{1.0}P_{1.0}$, (f) $S_{1.0}P_{1.5}$, (g) $S_{1.5}P_{0.5}$, (h) $S_{1.5}P_{1.0}$, (i) $S_{1.5}P_{1.5}$ 68
Figure 4-5 R^2 of the training and validation dataset for (a) the 2-hidden-layer ANN model with ReLU and Sigmoid respectively, (b) the 3-hidden-layer ANN model with ReLU and Sigmoid respectively, (c) R^2 of the test dataset for the 2-hidden-layer ANN model with ReLU and Sigmoid respectively, the 3-

hidden-layer ANN model with ReLU and Sigmoid respectively, SVM, Random Forests, Gaussian Processes. 70
Figure 4-6 Training and evaluating results of the ANN model: (a) Loss, (b) R^2, and (c) MAE. 71
Figure 4-7 Back-fitting results of key parameters in the equation-based model: (a) tensile strength, ft, H; (b) SCTS, $\varepsilon t, H$ and (c) descending shape parameter, αt. 73
Figure 4-8 Comparison of prediction results and experimental results of the tensile stress-strain curves: (a) $S_{0.5}P_{1.0}$, (b) $S_{0.5}P_{1.5}$, (c) $S_{1.0}P_{1.0}$, and (d) $S_{1.0}P_{1.5}$. 74
Figure 4-9 The relative error of the ANN model and equation-based model in terms of (a) the tensile strength and (b) SCTS. 75
Figure 4-10 Contributions of each input feature to the tensile strength and strain corresponding to tensile strength of HFRC: (a) tensile strength; (b) SCTS 77
Figure 4-11 Contribution of steel fibers, PVA fibers, mechanical properties of plain concrete, and components of HFRC to the tensile strength and strain corresponding to tensile strength of HFRC: (a) tensile strength as the fiber volume of steel fibers is constant; (b) tensile strength as the fiber volume of PVA fibers is constant;(c) SCTS as the fiber volume of steel fibers is constant; (d) SCTS as the fiber volume of PVA fibers is constant 77
Figure 5-1 Framework of deep transfer learning-based vehicle classification by asphalt pavement vibration 85
Figure 5-2 The field layout of the pavement vibration IoT monitoring system. (a) Scheme and (b) image of the on-site monitoring. 86
Figure 5-3 Wavelet functions used in this work. (a) Biorthogonal 6.8 (bior6.8), (b) Symlets 6 (sym6), and (c) Coiflets 3 (coif3). 87
Figure 5-4 Overview of (a) the time-domain graph (a) and (b) the time-frequency graph. 88
Figure 5-5 Scheme of deep transfer learning applied to CNN 89
Figure 5-6 Comparison of raw signals and denoised signals. (a) Raw signals and denoised signals of one acceleration sensing node and (b) raw signals and denoised signals of seven acceleration sensing nodes. 92
Figure 5-7 Denoised signals of seven acceleration sensing nodes and their corresponding time-frequency graphs. (a) Denoised signals of seven acceleration sensing nodes and (b) time-frequency graphs of Node-3, Node-6, and Node-7. 94
Figure 5-8 The number of acceleration sensing nodes that received effective vibration signals (defined in [8, 13]). 0: motorcycle, 1: hatchback, 2: sedan, 3: SUV, 4: mini VAN, 5: VAN, 6: pickup truck, 7: mini truck, 8: truck, and 9: big truck. 95
Figure 5-9 Image of the dataset for two methods. (a) Method I (Time-domain & TL) and (b) Method II (Time-frequency & TL). 95
Figure 5-10 Loss of two methods. (a) Method I (Time-domain & TL) and (b) Method II (Time-frequency & TL). 97
Figure 5-11 Accuracy metrics of two methods, Method I (Time-domain & TL) and Method II (Time-frequency & TL). (a) Accuracy, (b) macro F_1 score, and (c) weighted F_1 score. 98

Figure 5-12 Accuracy metrics of the CNN models for two methods, Method I (Time-domain & TL) and Method II (Time-frequency & TL). (a) Accuracy, (b) macro F_1 score, and (c) weighted F_1 score. 101
Figure 5-13 Normalized confusion matrix of EfficientNet-B0 and -B3 for two methods. (a) Method I (Time-domain & TL) and (b) Method II (Time-frequency & TL). Note: 0: motorcycle, 1: hatchback, 2: sedan, 3: SUV, 4: mini VAN, 5: VAN, 6: pickup truck, 7: mini truck, 8: truck, and 9: big truck. 102
Figure 5-14 Comparison of time-domain graphs and time-frequency graphs 103
Figure 6-1 Scheme of physics-informed long short-term memory (PI-LSTM) network. The input includes the ground acceleration (xg) and the previous structural response (z). The output is the structural response (z). The total loss consists of the data loss from the difference between the prediction and ground truth and the physics loss from the physics constraint. Both αD and αP are set to 1 in this work. 111
Figure 6-2 Performance of PhyCNN and PI-LSTM for predicting displacement, velocity, and restoring force. 113
Figure 6-3 Data for the ground acceleration, structural displacement, structural velocity, and structural acceleration 114
Figure 6-4 Performance of LSTM and PI-LSTM for predicting displacement and acceleration 115
Figure 6-5 Performance of LSTM and PI-LSTM for predicting displacement 116
Figure 6-6 Performance of LSTM and PI-LSTM for predicting acceleration 117
Figure 6-7 Scheme of the modified PI-LSTM network for non-linear analysis. 118
Figure 6-8 Performance of LSTM and PI-LSTM for predicting acceleration (non-linear analysis) 119
Figure 6-9 Data for the ground acceleration, structural displacement, structural velocity, and structural acceleration 121
Figure 6-10 Performance of PhyCNN and PI-LSTM for predicting the structural response in terms of the (a) third floor and (b) roof. 122
Figure 6-11 Prediction performance of PhyCNN and PI-LSTM in terms of the (a) third floor and (b) roof. 122
Figure 6-12 Elevation of the six-story building (from the literature [34]) 124
Figure 6-13 Data for the ground acceleration, structural displacement, structural velocity, and structural acceleration 124
Figure 6-14 Performance of PhyCNN and PI-LSTM for predicting the structural response in terms of the (a) third floor and (b) roof. 125
Figure 6-15 Prediction performance of LSTM and PI-LSTM in terms of the (a) third floor and (b) roof. 125

LIST OF TABLES

Table 1-1 Summary of static and dynamic setups [8] 2
Table 1-2 Deep learning for structural damage detection 4
Table 1-3 Deep learning for structural vibration analysis 6
Table 2-1 Summary of the dataset (image size: 640×480) 19
Table 2-2 Summary of accuracy metrics on the visible image 22
Table 2-3 Summary of accuracy metrics on the infrared image 22
Table 2-4 Summary of accuracy metrics on the fusion image 23
Table 2-5 Summary of accuracy for deep learning from scratch and transfer learning 26
Table 2-6 Summary of the statistical characteristics of ImageNet, visible images, infrared images, and fusion images 30
Table 3-1 Summary of the dataset 43
Table 3-2 Performance of different models for the visible image 49
Table 3-3 Performance of different models for the infrared image 49
Table 3-4 Performance of different models for the fusion image 49
Table 4-1 Features of the input data 63
Table 4-2 The steel-PVA fiber hybridization of HFRC from previous literature 66
Table 4-3 Summary of the dataset for the ANN model (Size=9801) 68
Table 4-4 MAE, R^2 of the ANN model and equation-based model for predicting the tensile stress-strain curve of HFRC 74
Table 5-1 Evaluation metrics of the wavelet transform for signal denoising 92
Table 5-2 Summary of the dataset 95
Table 5-3 Accuracy metrics of image preprocessing methods 96
Table 5-4 Accuracy metrics of Method I (Time-domain & TL) 98
Table 5-5 Accuracy metrics of Method II (Time-frequency & TL) 99
Table 6-1 Summary of the comparison between PhyCNN and PI-LSTM 112
Table 6-2 Summary of the comparison between LSTM and PI-LSTM for predicting displacement and acceleration 115
Table 6-3 Summary of the comparison between LSTM and PI-LSTM for predicting displacement 116
Table 6-4 Summary of the comparison between LSTM and PI-LSTM for predicting acceleration 117
Table 6-5 Summary of the comparison between LSTM and PI-LSTM for predicting acceleration (non-linear analysis) 118
Table 6-6 Summary of the comparison between PhyCNN and PI-LSTM for predicting acceleration 121

Chapter 1 INTRODUCTION

Machine learning (ML) is an application of artificial intelligence (AI) that could automatically learn and improve through experience and data without being explicitly programmed [1]. Deep learning (DL) is a branch of machine learning based on artificial neural networks (ANN) and it could transform the input data into a more abstract and composite representation by using representation learning [2, 3]. Nowadays, deep learning has been widely used in autonomous driving [4], image recognition [2], natural language processing [5], speech recognition [6], recommendation systems [7], and so on.

Deep learning automatically and implicitly build the relationship between input and output. Implicit learning makes it difficult to interpret deep learning. To better understand how deep learning correlates input and output, it needs to understand the whole process of deep learning. Starting with the raw input, deep learning applied non-linear modules to obtain a higher and slightly more abstract representation, and then deep learning figure out the representation needed for detection or classification based on multiple levels of representation [2]. For example, in image classification tasks, the raw input is the matrix of pixels and deep learning utilizes modules to obtain multiple levels of representation: (1) the first layer of representation indicates the presence or absence of edges in the specific directions and locations in the image, (2) the second layer of representation detects patterns by finding specific edge alignments without considering small changes in edge position, (3) the third layer of representation combine the aforementioned patterns into larger combinations that belong to parts of objects and subsequent layers would figure out objects [2]. In these representations, higher layers of representation focus on the discriminative parts of the input and suppress the irrelevant variations [2]. Finally, deep learning models label the image as a class.

What are the input and output for deep learning? The input includes the available information for describing problems and the output includes the solutions to problems [8]. Based on the input and output, deep learning could be separated into two groups: supervised learning and unsupervised learning [2, 3]. Supervised learning contains both the input and output, namely input-out pairs, but unsupervised learning only contains the input. For supervised learning, it includes classification and regression based on the type of variables, discrete and continuous. For unsupervised learning, it has to self-discover the patterns of data and common examples include clustering and principal component analysis.

Meanwhile, deep learning could also be divided into two categories, static and dynamic setups, based on the input and output [8]. Table 1-1 summarizes the summary of static and dynamic setups. For static setups, the input and output are not related to the variation of time. Figure 1-1a shows the scheme of deep learning in static setups. In the image classification task, the input is an image and the output is a label, such as ‘crack’. Another example is the semantic segmentation tasks, both the input and output are images. On the contrary, for dynamic setups, both the input and output could be a function of time. Figure 1-1b illustrates the scheme of deep learning in dynamic setups. In the structural damage detection task, the input is the vibration of structures that is related to the loading and the output is the structural conditions that vary as the loading time increases.

The abilities of powerful regression and multi-type data processing allow deep learning to effectively and accurately complete multi-tasks, which is the need of civil engineering. This is because civil engineering is a subject that usually integrates several areas of knowledge simultaneously. Besides, heavy reliance on manpower and human knowledge might hinder the progress of civil engineering towards mechanization and automation. The ability to figure out patterns from big data could enable deep learning to reduce human intervention in civil engineering. More cases showed that deep learning has become a greatly powerful and increasingly popular tool for civil engineering.

Table 1-1 Summary of static and dynamic setups [8]

	Static		**Dynamic**	
	Deterministic	**Stochastic**	**Deterministic**	**Stochastic**
Ingredients	$x \in \mathbb{R}^n$ $y \in \mathbb{R}^d$	$x \in L^p(\Omega, \mathbb{R}^n)$ $y \in L^p(\Omega, \mathbb{R}^d)$	$x \in (\mathbb{R}^n)^{\mathbb{Z}_-}$ $y \in (\mathbb{R}^d)^{\mathbb{Z}_-}$	$x \in L^p(\Omega, (\mathbb{R}^n)^{\mathbb{Z}_-})$ $y \in L^p(\Omega, (\mathbb{R}^d)^{\mathbb{Z}_-})$
Problem to be solved	$y = f(x)$ f is measurable	$E[y\|x]$	$y(\cdot) = F(x(\cdot))$	$E[y(\cdot)\|x(\cdot)]$
Applications	• Translators • Transcription • Calibration of financial models	• Image classification • Semantic segmentation	• Structural mechanics • Vibration analysis • Robotics • Memory tasks	• Time series forecasting • Volatility filtering

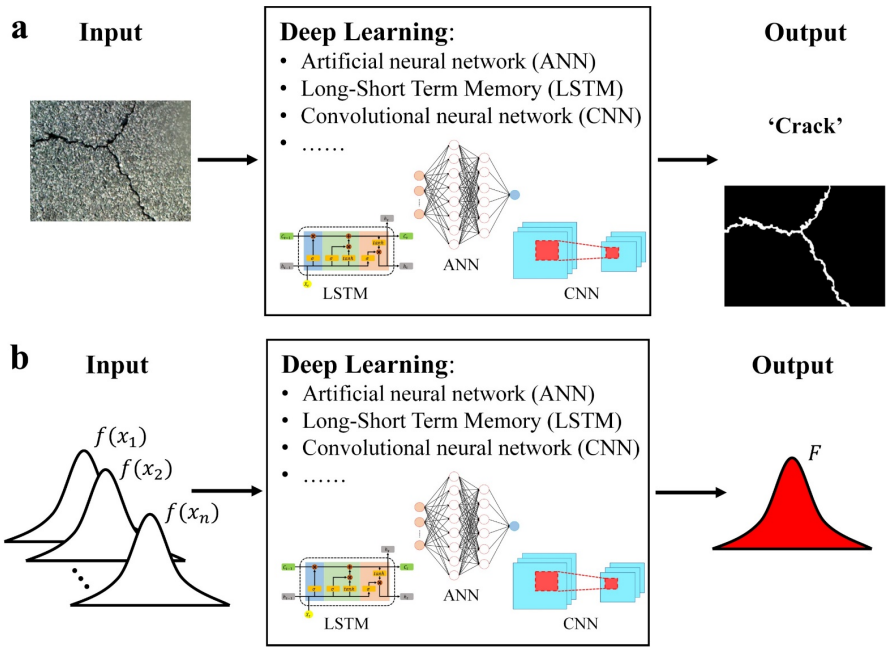

Figure 1-1 Scheme of deep learning in static and dynamic setups. (a) Scheme of static setups and (b) Scheme of dynamic setups

1.1 Deep learning in static setups

As shown in Table 1-1, the common tasks of civil engineering for deep learning in static setups are strongly related to images and the typical example is structural damage detection. Structural damage detection could be mainly separated into three subsets: classification, segmentation, and object detection, which is shown in Figure 1-2. Classification tasks would give the input a label. For example, as shown in Figure 1-2, the input is an image with cracks and the output is a label, namely 'crack'. Segmentation tasks would label each pixel of the input one class and their output is usually an image. For example, the input is an image with cracks and the output is a binary image where white ('1') often represents 'crack' and black ('0') means 'non-crack', which is shown in Figure 1-2. Object detection tasks would point out the location and class of objects. For example, the input is an image or every frame of a video and the output is also an image with several rectangles and labels that illustrate the location and class of objects.

Table 1-2 summarizes some research that applied deep learning for structural damage detection. Zhang et al. [9] proposed a fully convolutional network (FCN) based on dilated convolution for concrete crack segmentation. They used the residual network and the dilated convolutions to extract feature maps and then applied the stacked deconvolution and the softmax function to get the final prediction. Xu et al. [10] proposed a modified faster region-based convolutional neural network (Faster R-CNN) to identify multiple damages to reinforced concrete columns. They built a dataset of 400 images with different damages and rough backgrounds. Gopalakrishnan et al. [11] transferred an ImageNet pre-trained CNN model to detect cracks in Hot-Mix Asphalt (HMA) and Portland Cement Concrete (PCC) surfaced pavement images. They used CNN to extract feature maps from the input images and a new classifier (neural network, support vector machine, random forest, and so on) to predict the labels. Maeda et al. [12] collected 9053 road damage images taken by a smartphone and used Single Shot MultiBox Detector (SSD) to detect distresses. Based on their datasets and models, they developed a smartphone application for the public to detect distress on their smartphones. Majidifard et al. [13] built a pavement image dataset from two camera views, namely a wide view and a top-down view. The former one was used to identify distresses and the latter one was used to calculate the distress density. For the wide view group dataset, pavement images were labeled with 9 types of distresses, including 8 types of crack and pothole. Then they employed You Only Look Once (YOLO v2) and Faster R-CNN to make a benchmark on this dataset. Ayele et al. [14] adapted an Unmanned Aerial Vehicle (UAV) to collect the images of bridges and then built 3D models for navigation and control purposes. Finally, they employed Mask R-CNN to detect cracks in UAV images and evaluate bridge damage conditions. Liang [15] proposed a three-level image-based inspection method for the reinforced concrete bridge using deep learning. Firstly, at the system level, he used VGG-16 to determine whether the bridge has a system-level major failure or not. Then, at the component level, he employed Faster R-CNN to detect and localize the bridge structural components. Finally, at the local damage level, he applied SegNet for damage semantic pixel-wise segmentation. Xu and Yang [16] used Mask R-CNN to detect cracks on the tunnel images collected through ten cameras. Meanwhile, they combined the deep learning-based crack identification method and the robust modeling that was used for the deformation analysis. Gao et al. [17] proposed a deep learning FCN-RCNN method for detecting multiple tunnel defects. They used a Shield

tunnel automatic inspection vehicle to collect images, then employed Faster R-CNN for multiple defects classification and localization, and finally adapted FCN for defect segmentation.

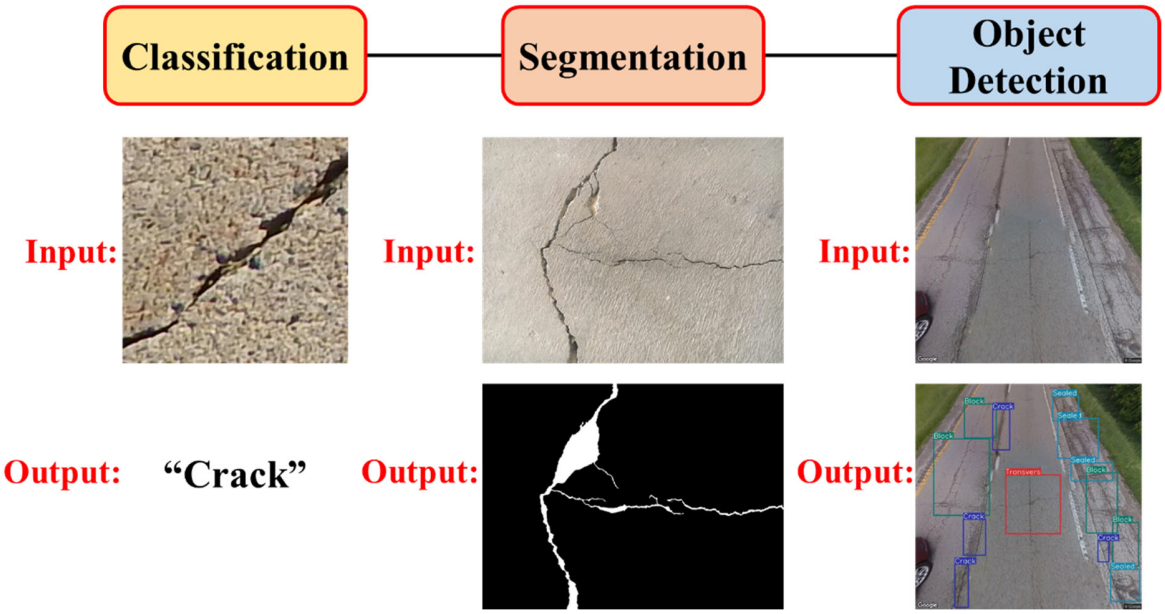

Figure 1-2 Scheme of different tasks for structural damage detection

Table 1-2 Deep learning for structural damage detection

Structure type	Application	Task	Method	Ref.
Building	Crack detection	Classification	VGG-16	[18]
		Segmentation	FCN	[9]
		Object detection	YOLO-v3	[19]
		Object detection + Segmentation	AlexNet + Transfer Learning	[20]
	Multiple damages detection	Classification	VGG-16 + Transfer learning	[21]
		Segmentation	FCN	[22]
		Object detection	Faster R-CNN	[10]
Pavement	Crack detection	Classification	VGG-16 + Transfer learning	[11]
		Segmentation	U-Net	[23]
		Object detection	YOLO-v2 + Transfer learning	[24]
			SSD + Inception-v2 / MobileNet	[12]
	Distress (Multiple damages) detection	Classification	CNN	[25]
		Segmentation	GAN	[26]
		Object detection	YOLO-v2 / Faster R-CNN	[13]
Bridge	Crack detection	Classification	VGG-16 + Transfer learning	[27]

		Segmentation	Mask R-CNN + UAV	[14]
		Object detection	YOLO-v3	[28]
	Multiple damages detection	Classification	DNN	[29]
			CNN + UAV	[30]
		Segmentation	FCN	[31]
		Object detection	YOLO-v3	[32]
		Classification + Segmentation + Object detection	VGG-16 + Faster R-CNN + SegNet	[15]
	Loosened bolt detection	Segmentation	R-CNN	[33]
Tunnel	Crack detection	Segmentation	FCN	[34]
		Object detection	Faster R-CNN	[35]
		Object detection + Segmentation	Mask R-CNN	[16]
	Multiple damages detection	Segmentation	FCN	[36]
		Object detection	Faster RCNN	[37]
		Object detection + Segmentation	Faster R-CNN + FCN	[17]
	Weak interlayers identification	Segmentation	FCN	[38]
	Moisture marks identification	Segmentation	Mask R-CNN + FCN	[39]

1.2 Deep learning in dynamic setups

As shown in Table 1-1, the common tasks of civil engineering for deep learning in dynamic setups are strongly related to the time-series data and the typical example is structural vibration. The structural vibration is a kind of time-series data and it varies as the loading, structural properties, environment factors (e.g. earthquake), and other factors change as well as the time. Researchers measured the structural vibration to determine the structure damage, structural properties, structure conditions, and so on.

Table 1-3 summarizes some research that applied deep learning for structural vibration. Lu et al. [40] adapted CNN for evaluating the regional post-event seismic damage condition. They collected the ground motion of buildings and their corresponding damage levels and then obtained the time-frequency distribution graphs of the ground motions that were regarded as the input of CNN. Based on these datasets, they applied CNN to predict the damage states. Abdeljaber et al. [41] proposed a 1D-CNN method to perform vibration-based damage detection and localization of the damage. They found that 1D-CNN with

low-cost running requirements helped them to achieve real-time damage detection. Ye et al. [42] collected the pavement vibration and then employed an ANN and the k-means++ cluster analysis to classify vehicle types. Jeong et al. [43] applied CNN and multi-index vehicle dynamics data (e.g. driving speed, accelerometer) to evaluate the pavement roughness. Li et al. [44] proposed a data fusion-based method for pavement distress evaluation. Firstly, they applied YOLO v3 to detect and localize the pavement distress, and then they used the wavelet transform to extract the acceleration effectuated by the defects using vibration data. Finally, they computed a comfort evaluation index based on the fusion of the results of image and vibration data. Zhang et al. [45] developed a CNN method to detect tiny local structural stiffness and mass changes for steel bridges. They collected the acceleration data from a T-shaped steel beam, a short steel girder bridge, and a long steel girder bridge. Gulgec et al. [46] utilized LSTM to process the acceleration data to estimate the strain responses. Subsequently, they used the strain information to do fatigue analysis and determine the remaining useful life. Li et al. [47] derived a general form of time-dependent ordinary differential equation to describe vortex-induced vibrations dynamics. Then guided by this formulation and field observation, they proposed a recurrent neural network (RNN) method to identify vortex-induced vibrations dynamics. Liu et al. [48] used LSTM and CNN for vibration-based working face ground identification. They collected the tunnel boring machine (TBM) vibration data from field monitoring and then used LSTM and CNN to process the time-series data and time-frequency graphs, respectively. Chen et al. [49] used a deep belief network (DBN) method to predict the collapse sections with the field-collected data.

Table 1-3 Deep learning for structural vibration analysis

Structure type	Task	Method	Ref.
Building	Regional seismic damage assessment	LSTM	[50]
	Regional post-event seismic damage assessment	CNN	[40]
	Structural damage detection	1D-CNN	[41]
	Structural dynamic response estimation and system identification	CNN	[51]
	Wind-induced response estimation	CNN	[52]
	Health condition assessment	Unsupervised deep Boltzmann machine	[53]
Pavement	Vehicle classification	ANN	[42]
	Pavement roughness assessment	CNN	[43]
	Pavement distress evaluation	CNN+Wavelet transform	[44]
	Pavement transverse cracks detection	CNN	[54]
Bridge	Tiny local structural stiffness and mass change detection	CNN	[45]
	Vibration category classification	CNN	[55]
	Strain estimation for fatigue assessment	LSTM	[46]
	Vortex-induced vibration identification	RNN	[47]

	Data pattern identification	1D-CNN	[56]
Tunnel	Ground identification of working faces	LSTM+CNN	[48]
	Shield machine slurry pressure prediction	LSTM	[57]
	Tunnel collapse section diagnosis	DBN	[49]

1.3 Objective and Outline

This work aims to investigate how deep learning would be applied to the condition assessment of civil infrastructure and materials. The outline of this work is:

- Chapter 2 is about deep learning and infrared thermography for asphalt pavement crack severity identification.
- Chapter 3 is about asphalt pavement crack segmentation based on the convolutional neural network and infrared thermography.
- Chapter 4 is about the artificial neural network model on tensile behavior of hybrid steel-PVA fiber reinforced concrete containing fly ash and slag powder.
- Chapter 5 is about deep transfer learning-based vehicle classification by asphalt pavement vibration.
- Chapter 6 is about the physics-informed deep learning model for data-driven structural response modeling

References

[1] T.M. Mitchell, Machine learning, (1997).

[2] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, nature 521(7553) (2015) 436-444.

[3] Y. Bengio, A. Courville, P. Vincent, Representation learning: A review and new perspectives, IEEE transactions on pattern analysis and machine intelligence 35(8) (2013) 1798-1828.

[4] S. Grigorescu, B. Trasnea, T. Cocias, G. Macesanu, A survey of deep learning techniques for autonomous driving, Journal of Field Robotics 37(3) (2020) 362-386.

[5] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, P. Kuksa, Natural language processing (almost) from scratch, Journal of machine learning research 12(ARTICLE) (2011) 2493− 2537.

[6] G. Hinton, L. Deng, D. Yu, G.E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T.N. Sainath, Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups, IEEE Signal processing magazine 29(6) (2012) 82-97.

[7] A.M. Elkahky, Y. Song, X. He, A multi-view deep learning approach for cross domain user modeling in recommendation systems, Proceedings of the 24th international conference on world wide web, 2015, pp. 278-288.

[8] J.-P. Ortega, Machine Learning of Dynamic Processes with Applications to Forecasting, (2019).

[9] J. Zhang, C. Lu, J. Wang, L. Wang, X.-G. Yue, Concrete cracks detection based on FCN with dilated convolution, Applied Sciences 9(13) (2019) 2686.

[10] Y. Xu, S. Wei, Y. Bao, H. Li, Automatic seismic damage identification of reinforced concrete columns from images by a region‐based deep convolutional neural network, Structural Control and Health Monitoring 26(3) (2019) e2313.

[11] K. Gopalakrishnan, S.K. Khaitan, A. Choudhary, A. Agrawal, Deep Convolutional Neural Networks with transfer learning for computer vision-based data-driven pavement distress detection, Construction and Building Materials 157 (2017) 322-330.

[12] H. Maeda, Y. Sekimoto, T. Seto, T. Kashiyama, H. Omata, Road damage detection and classification using deep neural networks with smartphone images, Computer‐Aided Civil and Infrastructure Engineering 33(12) (2018) 1127-1141.

[13] H. Majidifard, P. Jin, Y. Adu-Gyamfi, W.G. Buttlar, Pavement image datasets: A new benchmark dataset to classify and densify pavement distresses, Transportation Research Record 2674(2) (2020) 328-339.

[14] Y.Z. Ayele, M. Aliyari, D. Griffiths, E.L. Droguett, Automatic Crack Segmentation for UAV-Assisted Bridge Inspection, Energies 13(23) (2020) 6250.

[15] X. Liang, Image‐based post‐disaster inspection of reinforced concrete bridge systems using deep learning with Bayesian optimization, Computer‐Aided Civil and Infrastructure Engineering 34(5) (2019) 415-430.

[16] X. Xu, H. Yang, Vision measurement of tunnel structures with robust modelling and deep learning algorithms, Sensors 20(17) (2020) 4945.

[17] X. Gao, M. Jian, M. Hu, M. Tanniru, S. Li, Faster multi-defect detection system in shield tunnel using combination of FCN and faster RCNN, Advances in Structural Engineering 22(13) (2019) 2907-2921.

[18] W.R.L.d. Silva, D.S.d. Lucena, Concrete cracks detection based on deep learning image classification, Multidisciplinary digital publishing institute proceedings, 2018, p. 489.

[19] P. Kumar, S. Batchu, S.R. Kota, Real-Time Concrete Damage Detection Using Deep Learning for High Rise Structures, IEEE Access 9 (2021) 112312-112331.

[20] B. Kim, S. Cho, Automated vision-based detection of cracks on concrete surfaces using a deep learning technique, Sensors 18(10) (2018) 3452.

[21] Y. Gao, K.M. Mosalam, Deep transfer learning for image‐based structural damage recognition, Computer‐Aided Civil and Infrastructure Engineering 33(9) (2018) 748-768.

[22] S. Li, X. Zhao, G. Zhou, Automatic pixel‐level multiple damage detection of concrete structure using fully convolutional network, Computer‐Aided Civil and Infrastructure Engineering 34(7) (2019) 616-634.

[23] J. Huyan, W. Li, S. Tighe, Z. Xu, J. Zhai, CrackU‐net: A novel deep convolutional neural network for pixelwise pavement crack detection, Structural Control and Health Monitoring 27(8) (2020) e2551.

[24] V. Mandal, L. Uong, Y. Adu-Gyamfi, Automated road crack detection using deep convolutional neural

networks, 2018 IEEE International Conference on Big Data (Big Data), IEEE, 2018, pp. 5212-5215.
[25] B. Li, K.C. Wang, A. Zhang, E. Yang, G. Wang, Automatic classification of pavement crack using deep convolutional neural network, International Journal of Pavement Engineering 21(4) (2020) 457-463.
[26] D. Mazzini, P. Napoletano, F. Piccoli, R. Schettini, A novel approach to data augmentation for pavement distress segmentation, Computers in Industry 121 (2020) 103225.
[27] C.V. Dung, H. Sekiya, S. Hirano, T. Okatani, C. Miki, A vision-based method for crack detection in gusset plate welded joints of steel bridges using deep convolutional neural networks, Automation in Construction 102 (2019) 217-229.
[28] S.E. Park, S.-H. Eem, H. Jeon, Concrete crack detection and quantification using deep learning and structured light, Construction and Building Materials 252 (2020) 119096.
[29] Y. Bao, Z. Tang, H. Li, Y. Zhang, Computer vision and deep learning–based data anomaly detection method for structural health monitoring, Structural Health Monitoring 18(2) (2019) 401-421.
[30] C.M. Yeum, J. Choi, S.J. Dyke, Automated region-of-interest localization and classification for vision-based visual assessment of civil infrastructure, Structural Health Monitoring 18(3) (2019) 675-689.
[31] J. Rubio Guillamón, Bridge Structural Damage Segmentation Using Fully Convolutional Networks, Universitat Politècnica de Catalunya, 2018.
[32] C. Zhang, C.c. Chang, M. Jamshidi, Concrete bridge surface damage detection using a single‐stage detector, Computer‐Aided Civil and Infrastructure Engineering 35(4) (2020) 389-409.
[33] T.-C. Huynh, J.-H. Park, H.-J. Jung, J.-T. Kim, Quasi-autonomous bolt-loosening detection method using vision-based deep learning and image processing, Automation in Construction 105 (2019) 102844.
[34] Y. Ren, J. Huang, Z. Hong, W. Lu, J. Yin, L. Zou, X. Shen, Image-based concrete crack detection in tunnels using deep fully convolutional networks, Construction and Building Materials 234 (2020) 117367.
[35] C. Li, P. Xu, L. Niu, Y. Chen, L. Sheng, M. Liu, Tunnel crack detection using coarse‐to‐fine region localization and edge detection, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 9(5) (2019) e1308.
[36] H.-w. Huang, Q.-t. Li, D.-m. Zhang, Deep learning based image recognition for crack and leakage defects of metro shield tunnel, Tunnelling and underground space technology 77 (2018) 166-176.
[37] D. Li, Q. Xie, X. Gong, Z. Yu, J. Xu, Y. Sun, J. Wang, Automatic defect detection of metro tunnel surfaces using a vision-based inspection system, Advanced Engineering Informatics 47 (2021) 101206.
[38] J. Chen, D. Zhang, H. Huang, M. Shadabfar, M. Zhou, T. Yang, Image-based segmentation and quantification of weak interlayers in rock tunnel face via deep learning, Automation in Construction 120 (2020) 103371.
[39] S. Zhao, D.M. Zhang, H.W. Huang, Deep learning–based image instance segmentation for moisture marks of shield tunnel lining, Tunnelling and Underground Space Technology 95 (2020) 103156.
[40] X. Lu, Y. Xu, Y. Tian, B. Cetiner, E. Taciroglu, A deep learning approach to rapid regional post‐event seismic damage assessment using time‐frequency distributions of ground motions, Earthquake Engineering & Structural Dynamics 50(6) (2021) 1612-1627.
[41] O. Abdeljaber, O. Avci, S. Kiranyaz, M. Gabbouj, D.J. Inman, Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks, Journal of Sound and Vibration 388 (2017)

154-170.
[42] Z. Ye, H. Xiong, L. Wang, Collecting comprehensive traffic information using pavement vibration monitoring data, Computer‐Aided Civil and Infrastructure Engineering 35(2) (2020) 134-149.
[43] J.H. Jeong, H. Jo, G. Ditzler, Convolutional neural networks for pavement roughness assessment using calibration‐free vehicle dynamics, Computer‐Aided Civil and Infrastructure Engineering 35(11) (2020) 1209-1229.
[44] Y. Li, C. Liu, Y. Du, S. Jiang, A novel evaluation method for pavement distress based on impact of ride comfort, International Journal of Pavement Engineering (2020) 1-13.
[45] Y. Zhang, Y. Miyamori, S. Mikami, T. Saito, Vibration‐based structural state identification by a 1‐dimensional convolutional neural network, Computer‐Aided Civil and Infrastructure Engineering 34(9) (2019) 822-839.
[46] N.S. Gulgec, M. Takáč, S.N. Pakzad, Structural sensing with deep learning: Strain estimation from acceleration data for fatigue assessment, Computer‐Aided Civil and Infrastructure Engineering 35(12) (2020) 1349-1364.
[47] S. Li, S. Laima, H. Li, Physics-guided deep learning framework for predictive modeling of bridge vortex-induced vibrations from field monitoring, Physics of Fluids 33(3) (2021) 037113.
[48] M. Liu, S. Liao, Y. Yang, Y. Men, J. He, Y. Huang, Tunnel boring machine vibration-based deep learning for the ground identification of working faces, Journal of Rock Mechanics and Geotechnical Engineering (2021).
[49] Z. Chen, Y. Zhang, J. Li, X. Li, L. Jing, Diagnosing tunnel collapse sections based on TBM tunneling big data and deep learning: A case study on the Yinsong Project, China, Tunnelling and Underground Space Technology (2020) 103700.
[50] Y. Xu, X. Lu, B. Cetiner, E. Taciroglu, Real‐time regional seismic damage assessment framework based on long short‐term memory neural network, Computer‐Aided Civil and Infrastructure Engineering 36(4) (2021) 504-521.
[51] R.-T. Wu, M.R. Jahanshahi, Deep convolutional neural network for structural dynamic response estimation and system identification, Journal of Engineering Mechanics 145(1) (2019) 04018125.
[52] B.K. Oh, B. Glisic, Y. Kim, H.S. Park, Convolutional neural network‐based wind‐induced response estimation model for tall buildings, Computer‐Aided Civil and Infrastructure Engineering 34(10) (2019) 843-858.
[53] M.H. Rafiei, H. Adeli, A novel unsupervised deep learning model for global and local health condition assessment of structures, Engineering Structures 156 (2018) 598-607.
[54] C. Chen, H. Seo, Y. Zhao, A novel pavement transverse cracks detection model using WT-CNN and STFT-CNN for smartphone data analysis, International Journal of Pavement Engineering (2021) 1-13.
[55] A. Shrestha, J. Dang, Deep learning-based real-time auto classification of smartphone measured bridge vibration data, Sensors 20(9) (2020) 2710.
[56] Y. Zhang, Y. Lei, Data Anomaly Detection of Bridge Structures Using Convolutional Neural Network Based on Structural Vibration Signals, Symmetry 13(7) (2021) 1186.
[57] Q. Wang, X. Xie, I. Shahrour, Y. Huang, Use of deep learning, denoising technic and cross-correlation

analysis for the prediction of the shield machine slurry pressure in mixed ground conditions, Automation in Construction 128 (2021) 103741.

Chapter 2 DEEP LEARNING AND INFRARED THERMOGRAPHY FOR ASPHALT PAVEMENT CRACK SEVERITY CLASSIFICATION[1]

2.1 Introduction

How to automatically classify asphalt pavement crack severity? Crack is an important evaluation indicator for evaluating the pavement condition. When identifying and measuring cracks, it needs to point out severity levels of cracks [1, 2]. Traditional crack classification heavily depends on manual inspection which is laborious, time-consuming, expensive, and sometimes dangerous [3, 4]. Among these issues, efficiency and reliability are two main challenges of manual inspection due to human subjectivity. These two aspects are strongly affected by the skills of inspectors. Experienced inspectors usually require less time than first-timers, while the latter are more likely to document ill-reported damage. Significant variability has been reported in the routine inspection and discrepancy was observed for the crack severity classification [4].

To address these aforementioned drawbacks of manual inspection, the deep learning-based crack detection methods have been attracting more attention with the explosive development of artificial intelligence. The convolutional neural network (CNN) is a typical deep learning (DL) method aimed at computer vision tasks and CNN has achieved state-of-the-art and human-competitive performance in image classification, semantic segmentation, object detection, and so on [5, 6]. Unlike traditional image analysis methods (such as image processing methods), CNN does not necessitate the creation of manual rules and is able to automatically extract multi-level feature representations [7]. Numerous works have demonstrated that CNN has been becoming an increasingly popular and powerful tool for crack detection [4, 8-13]. Gopalakrishnan et al. [8] developed a pre-trained VGG-16 with a single-layer neural network classifier for pavement crack classification, while the types of images were separated into two groups, ‘crack’ and ‘no_crack’. Huyan et al. [9] proposed a modified U-net architecture, named CrackU-net, to achieve pixel-level crack detection by 3000 pavement crack images. Mandal et al. [10] developed a pavement crack detection system based on the YOLO v2 deep learning framework by using a dataset of 9053 road images taken from a vehicle-mounted smartphone. Majidifard et al. [11] proposed a deep learning-based crack detection system where the YOLO deep learning framework was used for distress classification and U-net was applied for crack segmentation. Guan et al. [14] developed an automated pixel-level pavement distress detection method based on deep learning and stereo vision, which integrated the color image, depth image, and color-depth overlapped image. However, few works have been carried out for crack severity classification. Although some researchers have classified images containing cracks, they focus on whether images have cracks or not, namely two types of labels (‘crack’ and ‘no_crack’). They did not label the crack severity which is crucial for quantifying cracks. This highlights a requirement for a method aiming at the asphalt pavement

[1] Part of this chapter is published as journal papers:

Fangyu Liu, Jian Liu, and Linbing Wang. "Deep learning and infrared thermography for asphalt pavement crack severity classification." Automation in Construction 140 (2022): 104383.

crack severity classification.

The low contrast between the edge of cracks and the pavement surface is a big challenge in classifying asphalt pavement crack severity [15-17]. This unclear boundary is mainly caused by the particle textures, while it is also affected by the shadow on the pavement and probable crack degradation [15, 17]. In this case, infrared thermography (IRT) could be a solution alternative as it could observe gray-level information and temperature information on asphalt pavement cracks [15, 16]. IRT, as one nondestructive evaluation (NDE) technique, could provide intuitive images for crack detection in a completely non-contact way [18]. IRT could record the thermal radiation from different objects, which could be used for ground object identification and inversion of surface parameters [19], such as crack detection. IRT has been successfully applied in monitoring civil infrastructures. Omar and Nehdi [20] employed the IRT and unmanned aerial vehicle (UAV) to detect subsurface delaminations in concrete bridge decks, which are non-contact and do not need traffic interruption. Taillade et al. [21] used IRT to detect and analyze the adhesion defects of externally bonded fiber-reinforced polymers (FRP) on concrete structures. Bauer et al. [22] applied IRT to quantitatively measure the extent of damage to cracks in a building façade by evaluating the crack depth and width. The previous works illustrate three main advantages of IRT applied to distress detection, especially crack detection: (1) Compared with manual inspection and sensor-based methods, IRT could reduce the effects of external environments, such as sunlight and smog [19]. IRT could also operate in low-light conditions that conventional optical cameras struggle to cope with [23]. (2) IRT is sensitive to objects and areas with significant infrared thermal characteristics [19]. It will be substantially helpful to find cracks rapidly, even if cracks are similar to the pavement surface. (3) IRT could also be a quantitative technique for measuring the severity of the damage, which could be measured by the number of pixels in the thermogram corresponding to the thermal disturbance [22]. These help IRT to have high detection efficiency and abundant detection information [24], which makes it possible for IRT to become a popular tool for crack detection.

Therefore, this work aims to employ deep learning and infrared thermography to classify asphalt pavement crack severity (Figure 2-1). The key contributions of this work are shown as follows:

- A dataset of asphalt pavement crack severity was built in this work, including four levels of crack severity, no crack, low-severity crack, medium-severity crack, and high-severity crack. This dataset had three types of images, the visible image, infrared image, and the fusion of visible and infrared images.

- For deep learning from scratch, thirteen typical CNN models were trained and evaluated on the aforementioned dataset to make a benchmark.

- For transfer learning, eight pre-trained CNN models trained by ImageNet were trained and evaluated on the dataset built in this work for comparison with deep learning from scratch.

- The effects of image types on the accuracy of deep learning from scratch and transfer learning were investigated, along with the effects of image types on classifying the levels of crack severity.

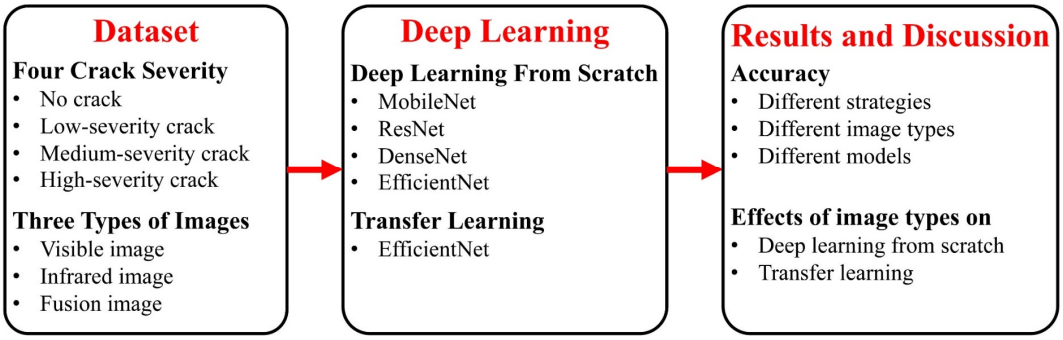

Figure 2-1 Framework of deep learning and infrared thermography for asphalt pavement crack severity classification

2.2 Methods

2.2.1 Deep learning from scratch

When mentioning deep learning, it is more commonly referred to as deep learning from scratch. It means deep learning models are built with the random initialization (random parameters, such as weights and biases). Then deep learning models are trained and evaluated with the assigned dataset. Specifically, when researchers apply CNN for crack detection, they need to first build the dataset of cracks. After developing a CNN model with the random initialization, they train and evaluate their CNN with their dataset. CNN could automatically detect the important features and identify underlying patterns from images, thus achieving state-of-the-art accuracy. During this process, the parameters of CNN are tuned to maximize the performance of CNN. Apart from their dataset and the CNN model with the random initialization, researchers do not need numerous things. Until now, a large number of CNN models have been developed and typical CNN models include ResNet [25], DenseNet [26], EfficientNet [27], and MobileNet [28]. As these CNN models usually have different layers to learn, several models are selected in this work, including ResNet-34 and -152, DenseNet-121 and -161, EfficientNet-B0, -B1, -B2, -B3, -B4, -B5, -B6, and -B7, and MobileNetV2.

2.2.2 Transfer learning

In practical engineering, the dataset used for deep learning is often relatively small as it is difficult and expensive to collect data. But the performance of deep learning models heavily relies on the size of the dataset. How to deal with this issue? Transfer learning (TL) could be a possible solution. Transfer learning focuses on storing and applying knowledge obtained while addressing one problem to a separate but related problem [29]. Given a source domain and learning task, and a target domain and learning task, the goal of transfer learning is the improvement of the target prediction function's learning by using the knowledge from the source domain and learning task [29]. Four main methods are frequently used in transfer learning,

including transferring knowledge of instances, transferring knowledge of feature representations, transferring knowledge of parameters, and transferring relational knowledge [29]. When transfer learning is applied in deep neural networks, there are four main categories: instance-based transfer learning, mapping-based transfer learning, network-based transfer learning, and adversarial-based transfer learning [30]. Transfer learning has been successfully applied in civil engineering [4, 8, 31], especially in areas related to image tasks, such as structural damage detection. These areas are more likely to have limited access to data collection or to require significant labor and resources to collect data. This results in their datasets being exceptionally small. However, the current CNN models tend to be very complex and require a large dataset to avoid overfitting. Transfer learning is becoming a popular method to address these issues for the CNN model applied in civil engineering.

There are two main steps of transfer learning with CNN for asphalt pavement crack severity classification (Figure 2-2). Firstly, for the source domain, the CNN model is trained on very large image datasets, usually, ImageNet [32] which contains 1.2 million images with 1000 categories. The trained CNN model is commonly referred to as the pre-trained CNN model, which is available from online resources [33, 34]. Secondly, the pre-trained CNN model is used as the model for the target domain. The parameters of the convolutional layers (i.e., pre-trained convolutional layers) would be retained, but the fully connected layers (FC layers) need to be adjusted to fit the requirement of the assigned output labels (4 labels in this work). Then the adjusted pre-trained CNN model is fine-tuned (trained and evaluated) with the dataset of this work.

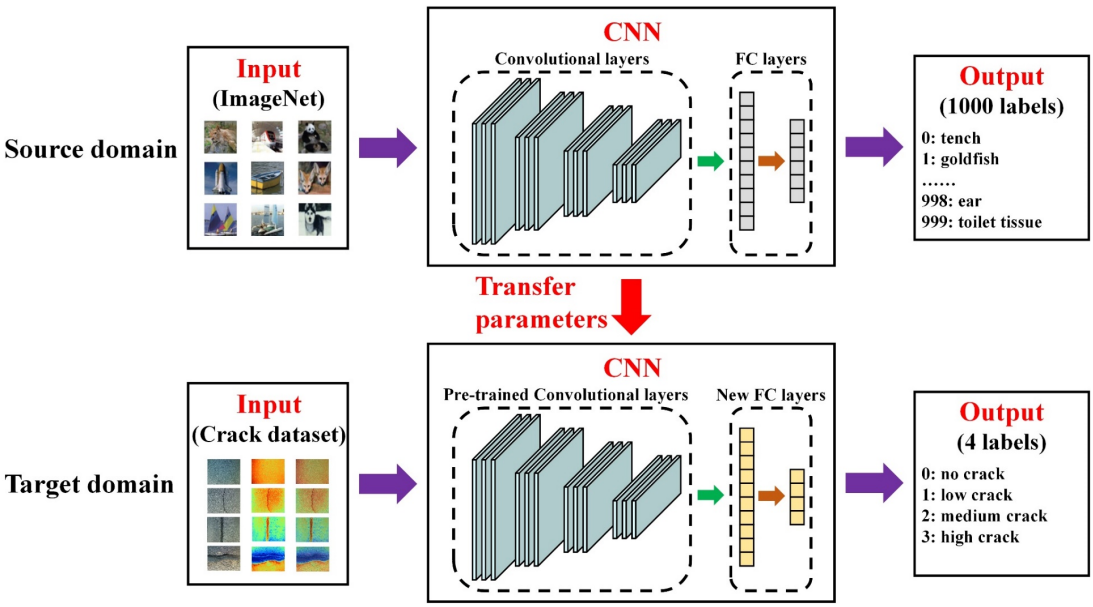

Figure 2-2 Scheme of transfer learning with CNN for asphalt pavement severity classification

2.2.3 Evaluation metrics

In the classification task, the most common accuracy metrics include accuracy, precision, recall, F_1 score,

and confusion matrix. Accuracy (Eq. (2-1)) is the number of correct predictions over all kinds of predictions made by deep learning models. Precision (Eq. (2-2)) is the percentages of relevant samples among retrieved samples, showing the capacity of the classifier to avoid labeling negative samples as positive [35]. Recall (or sensitivity, Eq. (2-3)) is the percentage of relevant samples that were retrieved, illustrating the capacity of the classifier to point out all positive samples [35]. F_1 score (Eq. (2-4)) is the weighted harmonic mean of the precision and recall. The confusion matrix is a specific table that visualizes the performance of deep learning models, whose elements include true positive (TP), true negative (TN), false positive (FP), and false negative (FN). In most cases, the classification task refers to the multiclass classification that classifies instances into one of three or more classes. After considering the effect of multiclass, precision, recall, and F_1 score can be categorized into three types, 'macro', 'micro', and 'weighted'. 'Macro' means it needs to calculate metrics for each class and then get the unweighted mean without taking label imbalance into account [35]. Similar to 'macro', but 'weighted' calculates the weighted mean by considering label imbalance [35]. In comparison, 'micro' calculates metrics by counting all TP, FN, and FP, while the value of micro precision, micro recall, micro F_1 score, and accuracy are identical.

- Accuracy

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2-1}$$

- Precision

$$Precision = \frac{TP}{TP + FP} \tag{2-2}$$

- Recall

$$Recall = \frac{TP}{TP + FN} \tag{2-3}$$

- F_1 score

$$F_1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{2-4}$$

- Macro Precision

$$Precision_{macro} = \frac{\sum_{l \in L} Precision(y_l, \hat{y}_l)}{|L|} \tag{2-5}$$

- Macro Recall

$$Recall_{macro} = \frac{\sum_{l \in L} Recall(y_l, \hat{y}_l)}{|L|} \tag{2-6}$$

- Macro F_1 score

$$F_{1,\ macro} = \frac{\sum_{l \in L} F_1(y_l, \hat{y}_l)}{|L|} \tag{2-7}$$

- Weighted Precision

$$Precision_{weighted} = \frac{\sum_{l \in L} Precision(y_l, \hat{y}_l) \times |y_l|}{\sum_{l \in L} |y_l|} \quad (2\text{-}8)$$

- Weighted Recall

$$Recall_{weighted} = \frac{\sum_{l \in L} Recall(y_l, \hat{y}_l) \times |y_l|}{\sum_{l \in L} |y_l|} \quad (2\text{-}9)$$

- Weighted F_1 score

$$F_{1,\ weighted} = \frac{\sum_{l \in L} F_1(y_l, \hat{y}_l) \times |y_l|}{\sum_{l \in L} |y_l|} \quad (2\text{-}10)$$

where L is the set of labels. y is the set of *true* (*sample, label*) pairs and y_l is the subset of y with label l. $\hat{y}$ is the set of *predicted* (*sample, label*) pairs and $\hat{y}_l$ is the subset of $\hat{y}$ with label l.

2.2.4 Hyperparameters of models

The CNN model was built based on online resources [33, 34]. The loss function is the cross-entropy loss and the optimization algorithm is Adam [36]. The raw image size is 640×480 and after the image preprocessing (Resize, RandomCrop, and Normalize), the input image size becomes 224×224 which is the common input size for the pre-trained CNN model. Other hyperparameters include the learning rate (0.001), training epoch (100), and batch size (32).

2.3 Dataset

Few datasets are available for asphalt pavement crack severity classification and let alone IRT datasets. Therefore, a new dataset needs to be built for asphalt pavement crack severity classification based on infrared thermography. The images of cracks were collected at Virginia Polytechnic Institute and State University (Virginia Tech), especially on the asphalt pavements near the Thomas M. Murray Structures and Materials Laboratory. To capture the visible and infrared images simultaneously, the infrared camera FLUKE TiX580 was used to collect images, which has a visible light camera lens and an infrared camera lens. Apart from the visible image and infrared image, the fusion of the visible image and infrared image (i.e., fusion image) could be acquired with the IR-Fusion™ technology that could combine visible and infrared images together in any ratio to create a single image. This fusion could also be achieved by the software (SmartView). This means that three types of images (visible, infrared, and fusion images) are able to be collected at the same time for the same object. More details about images could be found in the literature [12].

Three types of images (visible, infrared, and fusion images) are included in this dataset to make a comparison (Figure 2-3). It should be noted that in this dataset, the visible images are fully visible, the infrared images are fully infrared, and the fusion images are a combination of 50% visible and 50% infrared images. The equal combination could take full advantage of the visual image and infrared image [19, 37].

This means that for the classification of crack severity, the fusion image could not only obtain abundant detailed information in the visual image, which is mainly highlighted by the physical boundaries of the cracks, but also acquire effective target areas in the infrared image, which could be distinguished by the color contrast between the cracks and environment.

This dataset focuses on longitudinal or transverse cracks for asphalt pavements. According to the literature [2], cracks could be classified into four severity levels based on the width of cracks, including no crack, low-severity crack (i.e., low crack), medium-severity crack (i.e., medium crack), and high-severity crack (i.e., high crack). Therefore, all images in the dataset are separated into four groups, no crack (Figure 2-3a), low crack (Figure 2-3b), medium crack (Figure 2-3c), and high crack (Figure 2-3d). Table 2-1 summarizes the dataset and there are a total of 2316 images (image size: 640×480). The number of images with low, medium, and high cracks is similar, which is larger than that of images without cracks. For training and evaluating the CNN model, the whole dataset is divided into two subsets, the training set, and the test set, while the former occupies 80% of the whole dataset and the left 20% belongs to the latter. According to the size of four groups (no, low, medium, and high), 20% of the corresponding images were randomly selected and combined into the test set (see Table 2-1).

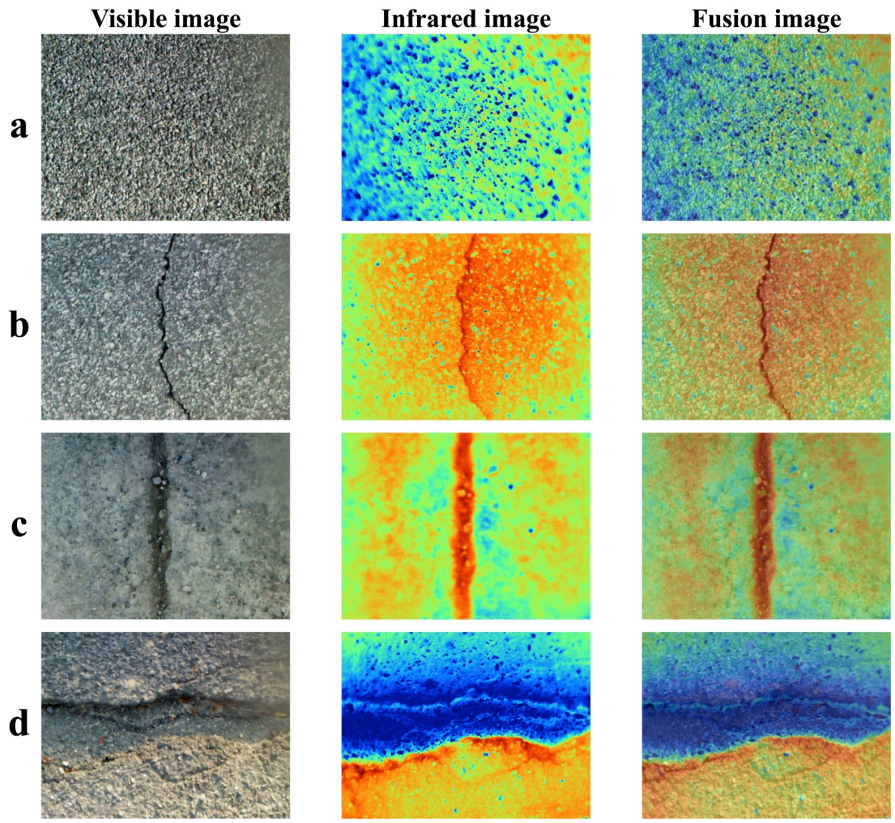

Figure 2-3 Examples of the visible image, infrared image, and fusion image in the dataset. (a) No crack, (b) low crack, (c) medium crack, and (d) high crack

Table 2-1 Summary of the dataset (image size: 640×480)

Dataset	No crack	Low crack	Medium crack	High crack	Total
Train	371	500	499	485	1855
Test	92	124	124	121	461
Total	463	624	623	606	2316

2.4 Results

2.4.1 Deep learning from scratch

2.4.1.1 Loss

Figure 2-4 shows the loss curves of the visible image, infrared image, and fusion image for deep learning from scratch. The visible image and fusion image had similar patterns of the loss curves, while their loss value was lower than that of the infrared image. For the visible image and fusion image, the loss value fluctuated during the first several epochs and then it decreased and tended to converge as the epoch increased (Figure 2-4a and c). In comparison, although the loss curves of the infrared image also had a tendency to converge with the increasing epochs, their value fluctuated more greatly on a larger value than the visible image and fusion image (Figure 2-4b). Among all models, ResNet-152, DenseNet-161, and EfficientNet-B0 and -B3 were selected to show their loss curves. For the loss curves of all three types of images, ResNet-152 was the most volatile, while the other three models had a smoother variation. From the perspective of the loss curve, the fusion image and visible image are similar, while they are better than the infrared image.

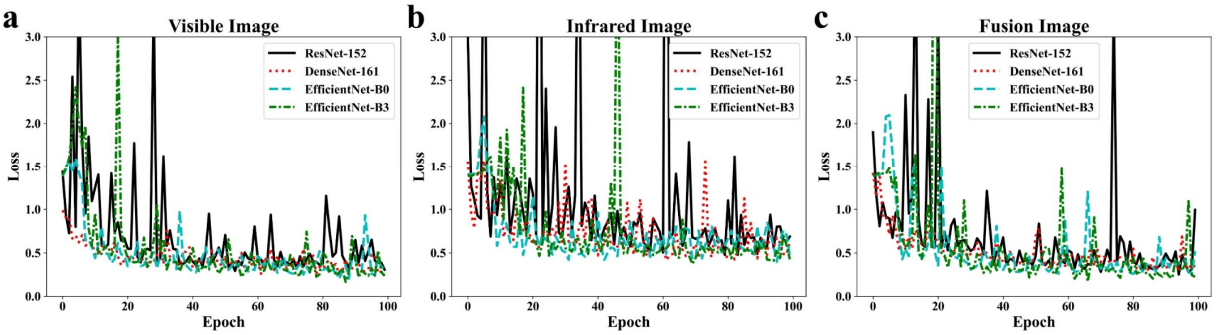

Figure 2-4 Loss curves of the test set for deep learning from scratch. (a) Visible image, (b) infrared image, and (c) fusion image.

2.4.1.2 Accuracy metrics

To show the performance of the CNN model and the types of images, the curves of accuracy metrics (accuracy, macro F_1 score, and weighted F_1 score) are plotted, while the effect of label imbalance is taken into account by the weighted F_1 score. Figure 2-5 shows the accuracy metric curves of the visible image. Throughout the beginning of the whole training process, the accuracy metric values of ResNet-152 and

DenseNet-161 started to be very high, especially DenseNet-161 with an accuracy of 0.568. In contrast, EfficientNet-B0 and -B3 began with relatively low accuracy (0.269). However, the accuracy metrics of EfficientNet-B0 and -B3 improved rapidly as the epoch increased and quickly surpassed that of ResNet-152 and DenseNet-161 at approximately Epoch 10. Finally, DenseNet-161, EfficientNet-B0, and -B3 achieved similar accuracy metrics that were higher and more stable than ResNet-152. Table 2-2 summarizes the accuracy metrics of the visible image. EfficientNet-B3 obtained the highest value in all accuracy metrics, followed by EfficientNet-B0 and -B2. DenseNet-161, EfficientNet-B1, and -B5 had similar accuracy metrics (accuracy = 0.9262), while ResNet had the lowest accuracy metrics.

Figure 2-6 presents the accuracy metric curves of the infrared image. Similar to the visible image, the accuracy metric values of EfficientNet-B0 and -B3 started low, while ResNet-152 and DenseNet-161 had higher initial accuracy. Finally, the accuracy metrics of EfficientNet-B0 and -B3 exceeded ResNet-152 and DenseNet-161 and they were more stable than ResNet-152 and DenseNet-161. Table 2-3 summarizes the accuracy metrics of the infrared image. EfficientNet-B3 still achieved the highest accuracy metrics. The accuracy metrics of EfficientNet were higher than MobileNet, ResNet, and DenseNet.

Figure 2-7 illustrates the accuracy metric curves of the fusion image. The accuracy metrics of the fusion image had similar patterns to the visible image and infrared image. The initial accuracy of ResNet-152 and DenseNet-161 was much higher than EfficientNet-B0 and -B3. However, different from the visible image and infrared image, all four CNN models (ResNet-152, DenseNet-161, EfficientNet-B0, and -B3) remained stable in all accuracy metrics. Table 2-4 summarizes the accuracy metrics of the fusion image. The highest accuracy metrics were still observed in EfficientNet-B3. All EfficientNets had higher accuracy metrics than MobileNet, ResNet, and DenseNet.

According to Figure 2-5-Figure 2-7, the accuracy, macro F_1 score, and weighted F_1 score had similar curve patterns and their corresponding value was also similar (Table 2-2-Table 2-4). It means considering or not considering the effect of the types of cracks has no significant influence on the accuracy metrics. This is because the number of various cracks in the dataset is very similar (Table 2-1). Besides, when dividing the training set and test set, the number of each type of crack was also consistent with the proportion of the whole dataset. Thus, it helps to eliminate the effect of label imbalance and enables the CNN model to fully learn the characteristics of each type of crack instead of focusing on only one or a few kinds of crack. Another thing worth noting is the impact of the CNN model on accuracy metrics. EfficientNet-B3 achieved the highest accuracy metrics in all three types of images. For the visible image, EfficientNet and DenseNet had similar performance, while they performed better than ResNet and MobileNet. For the infrared image and fusion image, EfficientNet was better than other CNN models. For EfficientNet, the accuracy did not improve as the architecture of EfficientNet became more complex. For example, EfficientNet-B1 and -B5 had similar performances on the visible image while EfficientNet-B1 and -B6 performed similarly on the infrared image.

To investigate the effect of the image type on accuracy metrics, Figure 2-8 summarizes the accuracy metrics of the CNN model on three types of images (visible, infrared, and fusion images). For all models, the fusion image was slightly better than the visible image, while both these two types of images were much better than the infrared image. For deep learning from scratch, the fusion image is a better choice as it helps the CNN model to improve accuracy.

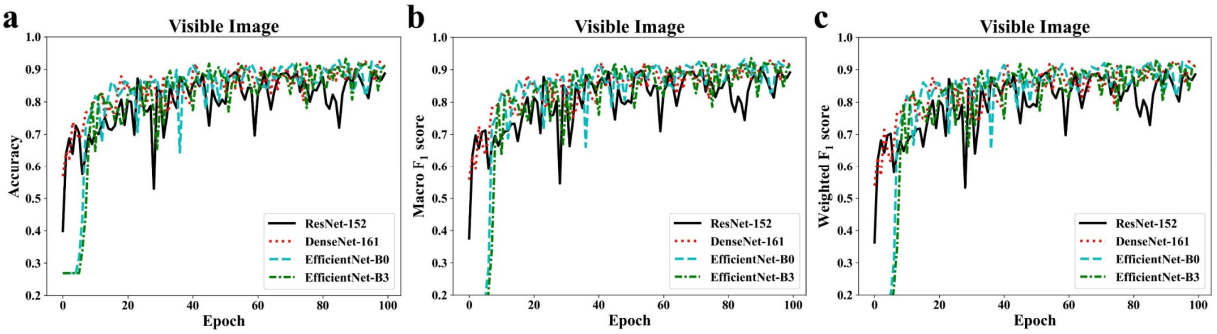

Figure 2-5 Accuracy metrics of the test set on the visible image for deep learning from scratch. (a) Accuracy, (b) macro F_1 score, and (c) weighted F_1 score.

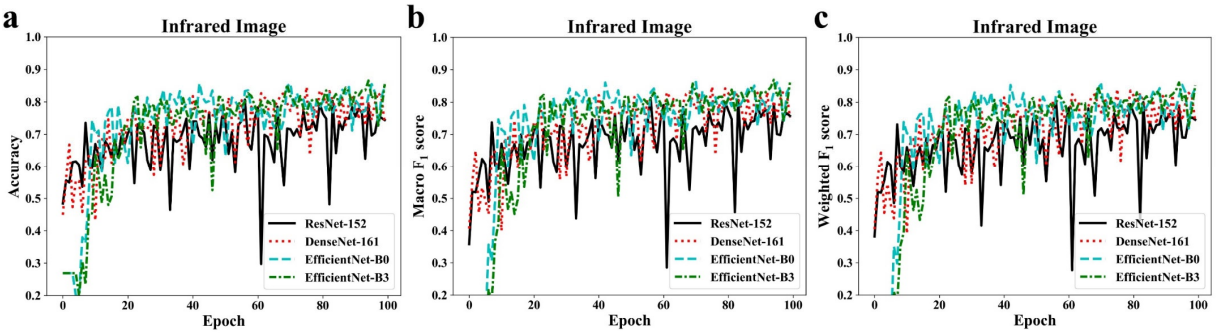

Figure 2-6 Accuracy metrics of the test set on the infrared image for deep learning from scratch. (a) Accuracy, (b) macro F_1 score, and (c) weighted F_1 score.

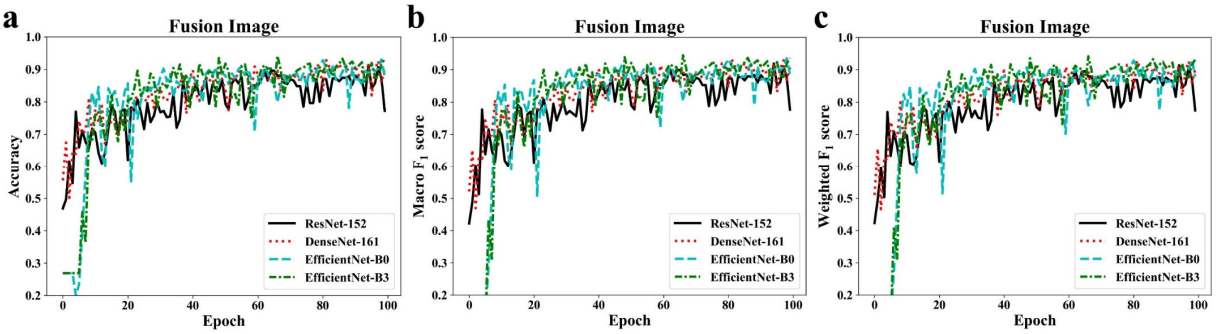

Figure 2-7 Accuracy metrics of the test set on the fusion image for deep learning from scratch. (a) Accuracy, (b) macro F_1 score, and (c) weighted F_1 score.

Table 2-2 Summary of accuracy metrics on the visible image

Model	Accuracy	Precision		Recall		F1 score	
		Macro	Weighted	Macro	Weighted	Macro	Weighted
MobileNetV2	0.9154	0.9216	0.9165	0.9198	0.9154	0.9206	0.9158
ResNet-34	0.9067	0.9081	0.9055	0.9115	0.9067	0.9093	0.9056
ResNet-152	0.9002	0.9076	0.9018	0.9056	0.9002	0.9065	0.9009
DenseNet-121	0.9219	0.9250	0.9214	0.9259	0.9219	0.9253	0.9215
DenseNet-161	0.9262	0.9267	0.9259	0.9301	0.9262	0.9282	0.9259
EfficientNet-B0	0.9284	0.9332	0.9290	0.9323	0.9284	0.9322	0.9281
EfficientNet-B1	0.9262	0.9317	0.9273	0.9293	0.9262	0.9304	0.9267
EfficientNet-B2	0.9284	0.9328	0.9283	0.9320	0.9284	0.9323	0.9283
EfficientNet-B3	**0.9328**	**0.9360**	**0.9325**	**0.9361**	**0.9328**	**0.9360**	**0.9326**
EfficientNet-B4	0.9219	0.9259	0.9219	0.9260	0.9219	0.9258	0.9218
EfficientNet-B5	0.9262	0.9323	0.9282	0.9294	0.9262	0.9306	0.9269
EfficientNet-B6	0.9197	0.9279	0.9235	0.9244	0.9197	0.9247	0.9201
EfficientNet-B7	0.9219	0.9343	0.9306	0.9258	0.9219	0.9254	0.9213

Table 2-3 Summary of accuracy metrics on the infrared image

Model	Accuracy	Precision		Recall		F1 score	
		Macro	Weighted	Macro	Weighted	Macro	Weighted
MobileNetV2	0.8395	0.8465	0.8380	0.8497	0.8395	0.8467	0.8372
ResNet-34	0.8547	0.8618	0.8542	0.8603	0.8547	0.8608	0.8542
ResNet-152	0.8069	0.8098	0.8043	0.8168	0.8069	0.8129	0.8052
DenseNet-121	0.8482	0.8569	0.8500	0.8551	0.8482	0.8550	0.8480
DenseNet-161	0.8460	0.8625	0.8590	0.8556	0.8460	0.8478	0.8405
EfficientNet-B0	0.8568	0.8685	0.8633	0.8655	0.8568	0.8623	0.8551
EfficientNet-B1	0.8547	0.8642	0.8578	0.8628	0.8547	0.8584	0.8508
EfficientNet-B2	0.8568	0.8646	0.8590	0.8660	0.8568	0.8613	0.8536
EfficientNet-B3	**0.8655**	**0.8757**	**0.8693**	**0.8745**	**0.8655**	**0.8689**	**0.8609**
EfficientNet-B4	0.8633	0.8683	0.8627	0.8721	0.8633	0.8670	0.8596
EfficientNet-B5	0.8590	0.8704	0.8637	0.8670	0.8590	0.8641	0.8564
EfficientNet-B6	0.8547	0.8565	0.8513	0.8640	0.8547	0.8578	0.8505
EfficientNet-B7	0.8590	0.8623	0.8570	0.8680	0.8590	0.8620	0.8547

Table 2-4 Summary of accuracy metrics on the fusion image

Model	Accuracy	Precision		Recall		F1 score	
		Macro	Weighted	Macro	Weighted	Macro	Weighted
MobileNetV2	0.9197	0.9278	0.9229	0.9240	0.9197	0.9251	0.9206
ResNet-34	0.9197	0.9232	0.9202	0.9238	0.9197	0.9232	0.9196
ResNet-152	0.9132	0.9209	0.9157	0.9179	0.9132	0.9191	0.9141
DenseNet-121	0.9241	0.9315	0.9269	0.9274	0.9241	0.9290	0.9250
DenseNet-161	0.9284	0.9342	0.9302	0.9323	0.9284	0.9324	0.9284
EfficientNet-B0	0.9306	0.9360	0.9320	0.9341	0.9306	0.9350	0.9312
EfficientNet-B1	0.9371	0.9416	0.9378	0.9401	0.9371	0.9406	0.9372
EfficientNet-B2	0.9393	0.9440	0.9403	0.9420	0.9393	0.9426	0.9393
EfficientNet-B3	**0.9414**	**0.9452**	**0.9417**	**0.9443**	**0.9414**	**0.9447**	**0.9415**
EfficientNet-B4	0.9306	0.9345	0.9303	0.9341	0.9306	0.9342	0.9304
EfficientNet-B5	0.9328	0.9368	0.9327	0.9361	0.9328	0.9364	0.9327
EfficientNet-B6	0.9328	0.9363	0.9329	0.9363	0.9328	0.9362	0.9327
EfficientNet-B7	0.9393	0.9440	0.9410	0.9423	0.9393	0.9428	0.9397

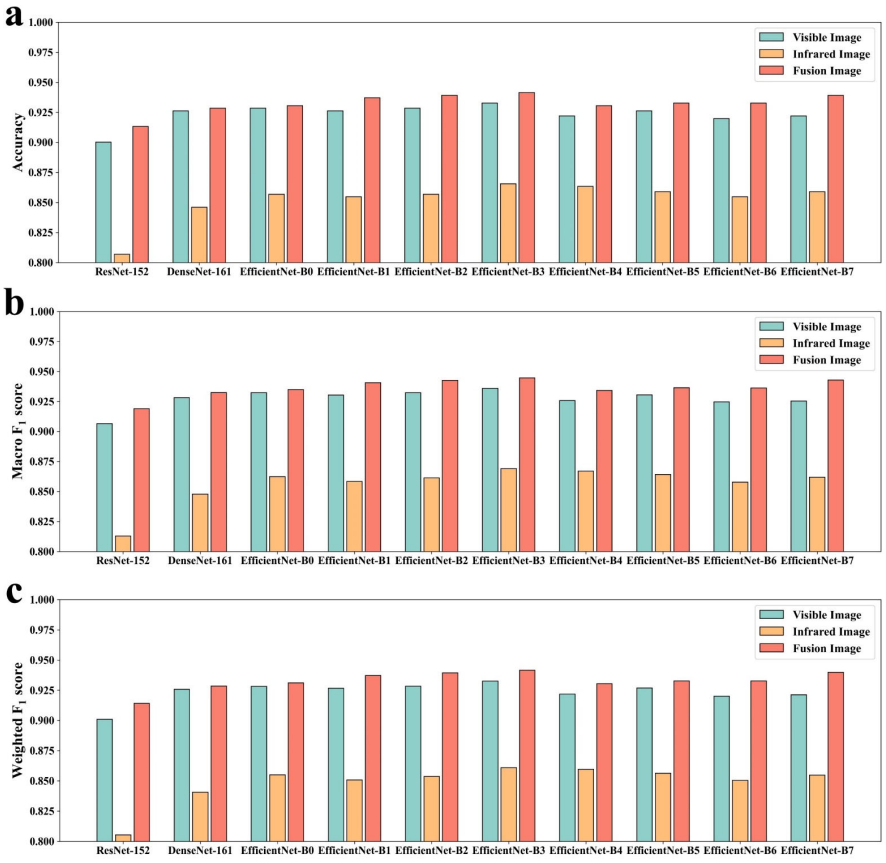

Figure 2-8 Accuracy metrics of CNN models on three types of images. (a) Accuracy, (b) macro F_1 score, and

(c) weighted F_1 score.

2.4.2 Transfer learning vs. deep learning from scratch

2.4.2.1 Loss

Figure 2-9 shows the loss curves for deep learning from scratch and transfer learning. EfficientNet-B0 and -B3 were used to make a comparison for deep learning from scratch and transfer learning. For three types of images, the loss of CNN models for transfer learning reached a very low value before the first five epochs and some CNN models even started with a very low initial loss value. In the latter stage, CNN models for deep learning from scratch and transfer learning performed similarly in terms of the loss value.

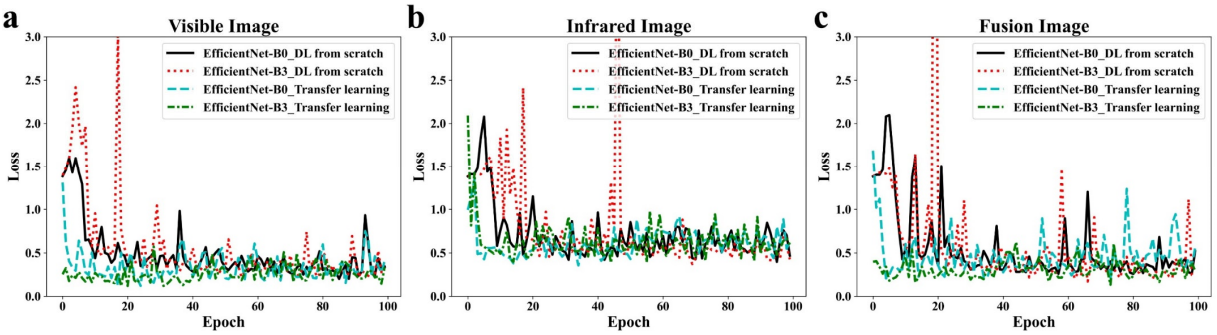

Figure 2-9 Loss curves of the test set for deep learning from scratch and transfer learning. (a) Visible image, (b) infrared image, and (c) fusion image.

2.4.2.2 Accuracy metrics

To be consistent with the results of deep learning from scratch, the curves of accuracy metrics (accuracy, macro F_1 score, and weighted F_1 score) are also plotted for deep learning from scratch and transfer learning. For deep learning from scratch and transfer learning, Figure 2-10-Figure 2-12 show the accuracy metrics of the visible image, infrared image, and fusion image, respectively. These pre-trained CNN models for transfer learning started with very high accuracy (more than 0.5), which was much higher than the corresponding CNN models for deep learning from scratch. In particular, the initial accuracy of pre-trained EfficientNet-B3 exceeded 0.8 on both the visible image and fusion image (Figure 2-10 and Figure 2-12). In the later stages, these pre-trained CNN models for transfer learning performed more consistently and achieved higher accuracy than the corresponding CNN models for deep learning from scratch. The gap between transfer learning and deep learning from scratch was more noticeable for the infrared image than for the visible image and fusion image. For transfer learning, these pre-trained CNN models retained the pre-trained convolutional layers, allowing these pre-trained CNN models to more accurately extract feature representations from raw inputs. This resulted in higher initial accuracy than the untrained CNN models, namely deep learning from scratch. The improvement in accuracy with increasing epochs was caused by fine-tuning the pre-trained CNN models by the target dataset.

Table 2-5 summarizes the accuracy of three types of images for deep learning from scratch and transfer

learning. Compared with deep learning from scratch, the accuracy of all CNN models for transfer learning increased on all three types of images (visible, infrared, and fusion images). The largest increase was observed in the infrared image, followed by the visible image, while the fusion image showed the least improvement. Another thing that needs to point out is that the fusion image had the highest accuracy among all three types of images for deep learning from scratch but the highest accuracy for transfer learning was observed on the visible image. EfficientNet-B3 achieved the highest accuracy for deep learning from scratch and transfer learning on all three types of images. Based on this, EfficientNet-B3 is a good CNN model for asphalt pavement crack severity classification no matter which types of images are used.

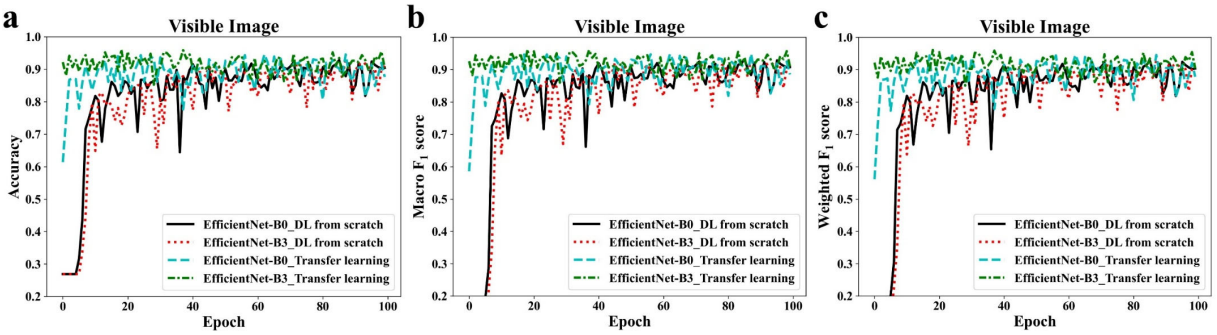

Figure 2-10 Accuracy metrics of the test set on the visible image for deep learning from scratch and transfer learning. (a) Accuracy, (b) macro F_1 score, and (c) weighted F_1 score.

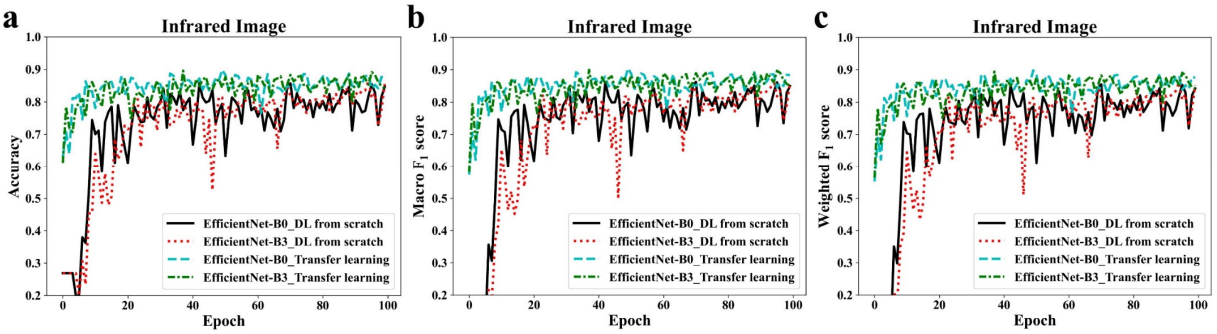

Figure 2-11 Accuracy metrics of the test set on the infrared image for deep learning from scratch and transfer learning. (a) Accuracy, (b) macro F_1 score, and (c) weighted F_1 score.

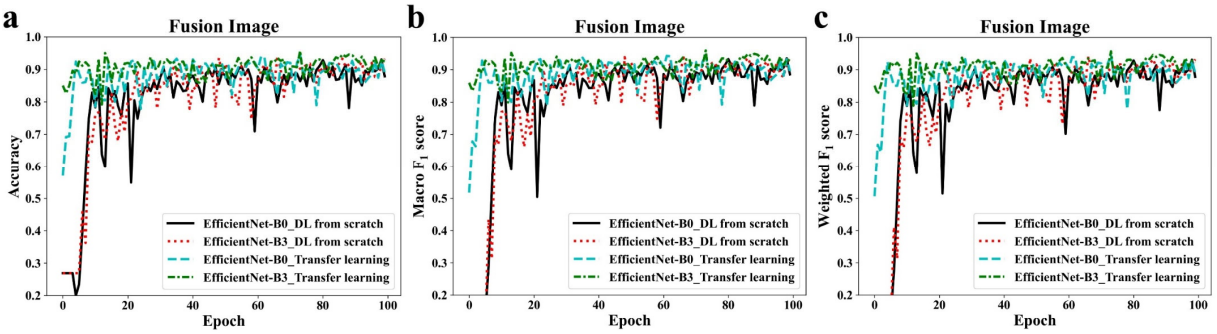

Figure 2-12 Accuracy metrics of the test set on the fusion image for deep learning from scratch and transfer

learning. (a) Accuracy, (b) macro F_1 score, and (c) weighted F_1 score.

Table 2-5 Summary of accuracy for deep learning from scratch and transfer learning

Model	Visible Image			Infrared Image			Fusion Image		
	DL	TL	Increase	DL	TL	Increase	DL	TL	Increase
EfficientNet-B0	0.9284	0.9458	0.0174	0.8568	0.8872	0.0304	0.9306	0.9436	0.0130
EfficientNet-B1	0.9262	0.9523	0.0260	0.8547	0.8894	0.0347	0.9371	0.9458	0.0087
EfficientNet-B2	0.9284	0.9501	0.0217	0.8568	0.8872	0.0304	0.9393	0.9414	0.0022
EfficientNet-B3	**0.9328**	**0.9588**	0.0260	**0.8655**	**0.8959**	0.0304	**0.9414**	**0.9566**	**0.0152**
EfficientNet-B4	0.9219	0.9566	**0.0347**	0.8633	0.8850	0.0217	0.9306	0.9458	**0.0152**
EfficientNet-B5	0.9262	0.9566	0.0304	0.8590	0.8915	0.0325	0.9328	0.9393	0.0065
EfficientNet-B6	0.9197	0.9544	**0.0347**	0.8547	0.8915	**0.0369**	0.9328	0.9414	0.0087
EfficientNet-B7	0.9219	0.9544	0.0325	0.8590	0.8915	0.0325	0.9393	0.9414	0.0022

Note: DL is deep learning from scratch and TL is transfer learning. “Increase” means the value where transfer learning increases over deep learning from scratch.

2.5 Discussion

2.5.1 Effects of image types on deep learning from scratch

The same CNN models had different accuracy on three types of images. How does the type of images affect the accuracy of CNN models? Does the type of images influence the performance of CNN models on four kinds of cracks? The confusion matrix is used to visualize the performance of CNN models when classifying different classes. Figure 2-13-Figure 2-15 show the normalized confusion matrix of the visible image, infrared image, and fusion image, respectively. Through these confusion matrices, CNN models had a very satisfying performance on both the no crack and low crack for all three types of images. But for the medium crack and high crack, three types of images had different influences on the performance of CNN models, while misclassification occurred mainly on the medium crack and high crack for all three types of images. The infrared image had more misclassifications than the visible image and fusion image, although both two types of images also misclassified the medium crack and high crack. The high misclassifications caused the CNN models to be less accurate on the infrared image than on the other two types of images. Why do all three types of images have different performances on the medium crack and high crack? There are two main reasons. Firstly, the temperature contrast of pavement and cracks is sometimes not very obvious, which results in a temperature buffer between cracks and the surrounding area, namely a color gradient area in the infrared image. For example, as shown in Figure 2-16a and c, the boundary between cracks and pavement is very clear in the visible image. However, the low temperature difference causes the enlargement (Figure 2-16a) or reduction (Figure 2-16c) of the displayed crack area on the infrared image. This has the potential to cause misclassification for the infrared image, thus decreasing the accuracy of the infrared image compared to the visible image. However, by combining the visible image and infrared image, the fusion image enhances the contrast between cracks and pavement more than the full infrared image. It allows the fusion image to perform better than the infrared image and to some extent similar to the visible image.

Secondly, the inconspicuous edge between cracks and pavement leads to misclassification for the visible image (Figure 2-16b and d). In contrast, with the help of temperature information, cracks on the infrared image and fusion image are much easier to distinguish (Figure 2-16b and d). Compared with the visible image, this helps to improve the accuracy of the infrared image and so does the fusion image. Throughout the dataset, the first phenomenon is more frequent than the second phenomenon, which might account for the low accuracy of the infrared image. However, by taking advantage of the visible image and infrared image, the fusion image achieved the highest accuracy of the CNN model for deep learning from scratch. When training the CNN model from scratch, the fusion image is a better choice.

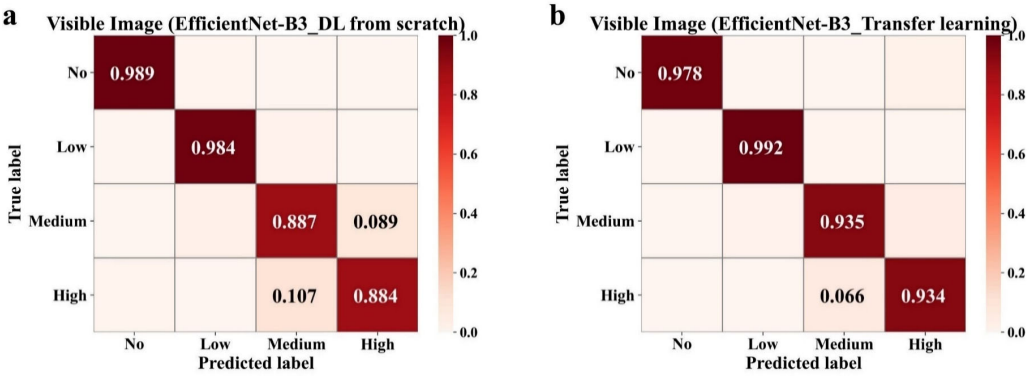

Figure 2-13 Normalized confusion matrix on the visible image. (a) EfficientNet-B3 for deep learning from scratch and (b) EfficientNet-B3 for transfer learning. It only displays values greater than 0.05.

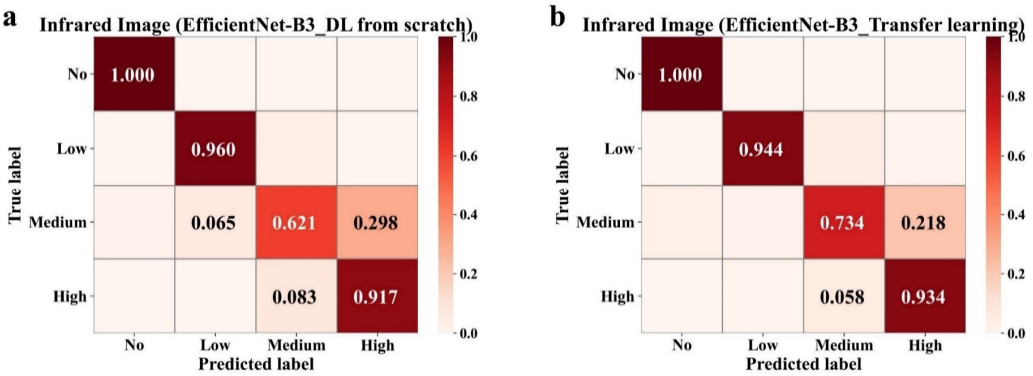

Figure 2-14 Normalized confusion matrix on the infrared image. (a) EfficientNet-B3 for deep learning from scratch and (b) EfficientNet-B3 for transfer learning. It only displays values greater than 0.05.

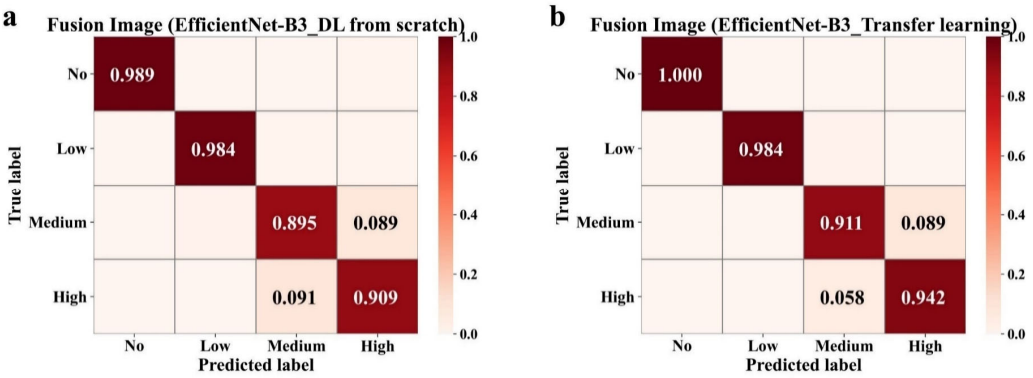

Figure 2-15 Normalized confusion matrix on the fusion image. (a) EfficientNet-B3 for deep learning from scratch and (b) EfficientNet-B3 for transfer learning. It only displays values greater than 0.05.

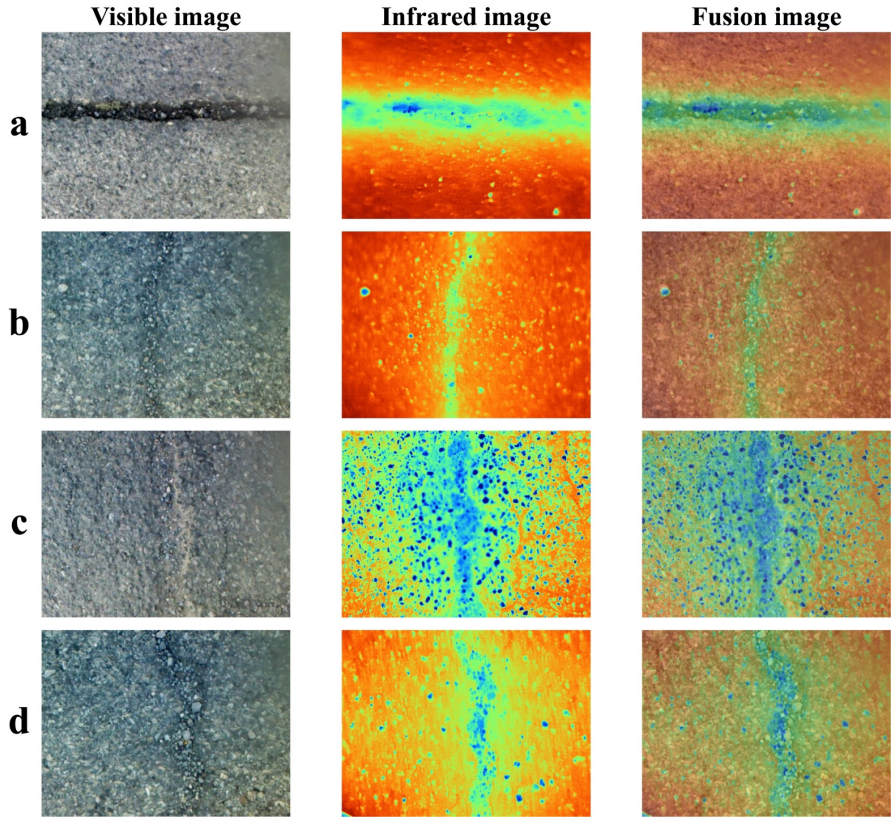

Figure 2-16 Examples of the visible image, infrared image, and fusion image in the dataset. (a) and (b) are medium crack, and (c) and (d) are high crack.

2.5.2 Effects of image types on transfer learning

As shown in Table 2-5, CNN models had the highest accuracy on the fusion image for deep learning from scratch, but the highest accuracy on the visible image for transfer learning. Why does this happen? For deep learning from scratch, the CNN model is completely new with the random initialization before training. In this case, the performance of the same CNN model strongly relies on the quality of the dataset. According

to Table 2-5, the fusion image is the most suitable for deep learning from scratch among the three types of images, which is explained in the last section. However, CNN models achieved the highest accuracy on the visible image for transfer learning. Why does the visible image have the highest accuracy for transfer learning instead of the fusion image?

The convolutional layers are the most important part of CNN and play a dominant role in the accuracy of CNN, while they are also the most difficult part of CNN to be trained. For transfer learning (Figure 2-2), the convolutional layers of the target domain (crack severity classification) directly adopted the parameters and architecture of the convolutional layers of the source domain (ImageNet classification), namely pre-trained convolutional layers. Thus, to a large extent, the pre-trained convolutional layers determine the accuracy of the crack severity classification for transfer learning, although the pre-trained CNN models were fine-tuned by the dataset built in this work. To show how the pre-trained convolutional layers perform on the three types of images, the pre-trained convolutional layers of the pre-trained CNN models are fixed and only the fully connected layers are trained for crack severity classification. The accuracy of CNN models with fixed pre-trained convolutional layers is shown in Figure 2-17. In this case, the visible image achieved the highest accuracy among the three types of images for all CNN models. It means the pre-trained convolutional layers performed best on the visible image, followed by the fusion image and, finally the infrared image (Figure 2-17). The pre-trained convolutional layers are likely to make pre-trained CNN models perform best on the visible image for transfer learning. Besides, as shown in Figure 2-10-Figure 2-12, the initial accuracy of CNN models was the highest on the visible image, followed by the fusion images and, finally the infrared image. This also confirms the aforementioned inference.

Furthermore, when the source domain and target domain are similar, the pre-trained CNN model is likely to perform well on the target domain [38]. It means if the images in the dataset built in this work are similar to the images in ImageNet, the pre-trained CNN model should have a satisfying accuracy. To examine the difference between the source domain and target domain, Table 2-6 summarizes the statistical characteristics of ImageNet, visible images, infrared images, and fusion images and Figure 2-18 shows the RGB distribution of the examples of the visible image, infrared image, and fusion image (used in Figure 2-3). From the perspective of the statistical characteristics of images, the ImageNet is closer to the visible image, followed by the fusion image and, finally the infrared image. It causes that the visible image as the target domain is more similar to the source domain (ImageNet) compared to the infrared image and fusion image. This allows the pre-trained CNN models to perform better on the visible image than on the infrared image and fusion image for transfer learning.

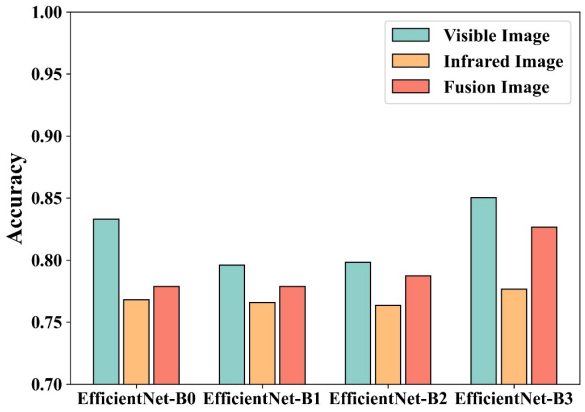

Figure 2-17 Accuracy of CNN models on three types of images as the pre-trained convolutional layers are fixed.

Table 2-6 Summary of the statistical characteristics of ImageNet, visible images, infrared images, and fusion images

Channel	ImageNet		Visible Image		Infrared Image		Fusion Image	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
R	0.485	0.229	0.477	0.129	0.747	0.156	0.618	0.110
G	0.456	0.224	0.509	0.122	0.631	0.178	0.573	0.110
B	0.406	0.225	0.495	0.115	0.256	0.145	0.370	0.094

Note: ‘Std’ is the standard deviation. ‘R’ is red, ‘G’ is green, and ’ B’ is blue.

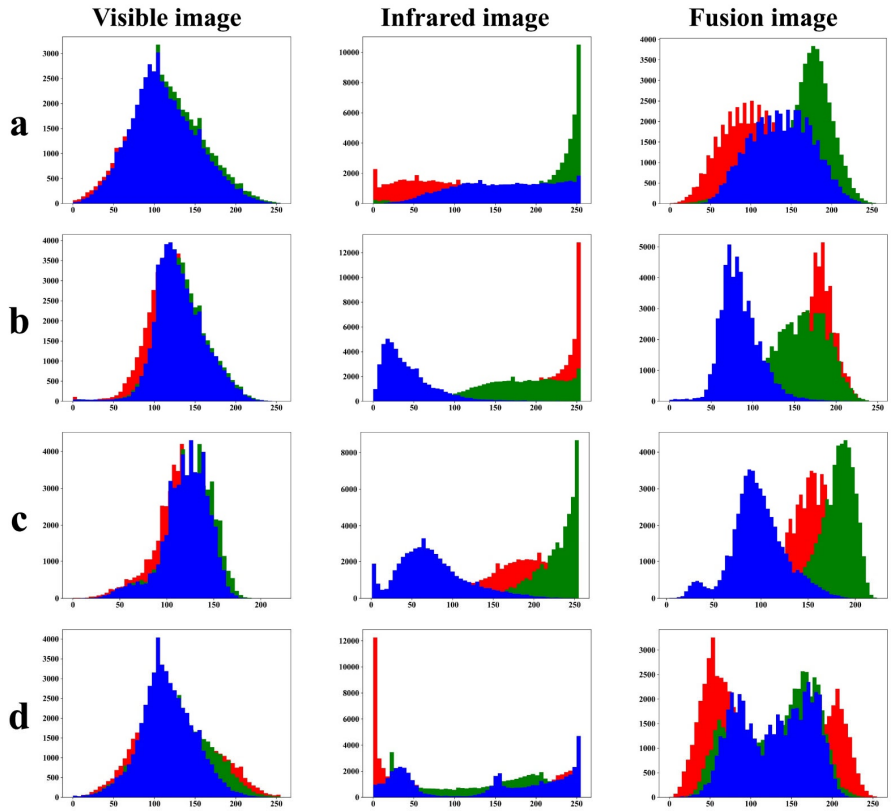

Figure 2-18 RGB distribution of the examples of the visible image, infrared image, and fusion image in the dataset. (a) No crack, (b) low crack, (c) medium crack, and (d) high crack. Images were used according to Figure 2-3.

2.6 Conclusion

CNN is becoming a popular and powerful tool for crack detection. Although some researchers have done some work about crack detection based on CNN, few works have been carried out for crack severity classification. Another issue is the indistinct edge between cracks and asphalt pavement surface. With the help of gray-level information and temperature information, infrared thermography might become a solution alternative. This work aims to apply deep learning and infrared thermography to classify asphalt pavement crack severity. The main contributions of this work include: (1) a dataset of asphalt pavement crack was built in this work. This dataset divided the crack severity into four groups, including no crack, low-severity crack, medium-severity crack, and high-severity crack. Besides, this dataset had three types of images, the visible image, the infrared image, and the fusion of visible and infrared images. (2) For deep learning from scratch, thirteen typical CNN models were trained and evaluated on the dataset built in this work. (3) For transfer learning, eight pre-trained CNN models trained by ImageNet were trained and evaluated on the aforementioned dataset for comparison with deep learning from scratch. (4) The effects of image types on the accuracy of deep learning from scratch and transfer learning were investigated, as well as the effects of image types on the levels of crack severity. The results show that the CNN models had the

highest accuracy on the fusion image for deep learning from scratch, but the highest accuracy on the visible image for transfer learning. The CNN models had a very satisfying performance on both the no crack and low crack for all three types of images. But for the medium crack and high crack, three types of images had different influences on the performance of the CNN models, while misclassification occurs mainly on the medium crack and high crack for all three types of images. EfficientNet-B3 had the highest accuracy on all three types of images for both deep learning from scratch and transfer learning.

In the future, more crack types, such as fatigue cracking, will be investigated to continue to evaluate the performance of deep learning and infrared thermography for asphalt pavement crack severity classification. A larger dataset will be established with more crack types as well as more distresses based on infrared thermography.

2.7 Reprint and Permission

©[2022] Elsevier, from [Fangyu Liu, Jian Liu, and Linbing Wang. "Deep learning and infrared thermography for asphalt pavement crack severity classification." Automation in Construction 140 (2022): 104383.] Please note that, as the author of this Elsevier article, you retain the right to include it in a thesis or dissertation, provided it is not published commercially. Permission is not required.

References

[1] D. ASTM, Standard practice for roads and parking lots pavement condition index surveys, (2011).

[2] J.S. Miller, W.Y. Bellinger, Distress identification manual for the long-term pavement performance program, United States. Federal Highway Administration. Office of Infrastructure Research and Development, 2003.

[3] Q. Yang, W. Shi, J. Chen, W. Lin, Deep convolution neural network-based transfer learning method for civil infrastructure crack detection, Automation in Construction 116 (2020) 103199.

[4] D. Dais, I.E. Bal, E. Smyrou, V. Sarhosis, Automatic crack classification and segmentation on masonry surfaces using convolutional neural networks and transfer learning, Automation in Construction 125 (2021) 103606.

[5] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, nature 521(7553) (2015) 436-444.

[6] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Advances in neural information processing systems 25 (2012) 1097-1105.

[7] Z. Liu, Y. Cao, Y. Wang, W. Wang, Computer vision-based concrete crack detection using U-net fully convolutional networks, Automation in Construction 104 (2019) 129-139.

[8] K. Gopalakrishnan, S.K. Khaitan, A. Choudhary, A. Agrawal, Deep convolutional neural networks with transfer learning for computer vision-based data-driven pavement distress detection, Construction and building materials 157 (2017) 322-330.

[9] J. Huyan, W. Li, S. Tighe, Z. Xu, J. Zhai, CrackU‐net: a novel deep convolutional neural network for pixelwise pavement crack detection, Structural Control and Health Monitoring 27(8) (2020) e2551.

[10] V. Mandal, L. Uong, Y. Adu-Gyamfi, Automated road crack detection using deep convolutional neural

networks, 2018 IEEE International Conference on Big Data (Big Data), IEEE, 2018, pp. 5212-5215.
[11] H. Majidifard, Y. Adu-Gyamfi, W.G. Buttlar, Deep machine learning approach to develop a new asphalt pavement condition index, Construction and building materials 247 (2020) 118513.
[12] F. Liu, J. Liu, L. Wang, Asphalt Pavement Crack Detection Based on Convolutional Neural Network and Infrared Thermography, IEEE Transactions on Intelligent Transportation Systems (2022).
[13] F. Liu, L. Wang, UNet-based model for crack detection integrating visual explanations, Construction and Building Materials 322 (2022) 126265.
[14] J. Guan, X. Yang, L. Ding, X. Cheng, V.C. Lee, C. Jin, Automated pixel-level pavement distress detection based on stereo vision and deep learning, Automation in Construction 129 (2021) 103788.
[15] Y. Du, X. Zhang, F. Li, L. Sun, Detection of Crack Growth in asphalt pavement through Use of infrared imaging, Transportation Research Record 2645(1) (2017) 24-31.
[16] A.A. Oloufa, H.S. Mahgoub, H. Ali, Infrared thermography for asphalt crack imaging and automated detection, Transportation research record 1889(1) (2004) 126-133.
[17] Q. Zou, Y. Cao, Q. Li, Q. Mao, S. Wang, CrackTree: Automatic crack detection from pavement images, Pattern Recognition Letters 33(3) (2012) 227-238.
[18] K. Jang, N. Kim, Y.-K. An, Deep learning–based autonomous concrete crack evaluation through hybrid image scanning, Structural Health Monitoring 18(5-6) (2019) 1722-1737.
[19] X. Jin, Q. Jiang, S. Yao, D. Zhou, R. Nie, J. Hai, K. He, A survey of infrared and visual image fusion methods, Infrared Physics & Technology 85 (2017) 478-501.
[20] T. Omar, M.L. Nehdi, Remote sensing of concrete bridge decks using unmanned aerial vehicle infrared thermography, Automation in Construction 83 (2017) 360-371.
[21] F. Taillade, M. Quiertant, K. Benzarti, C. Aubagnac, Shearography and pulsed stimulated infrared thermography applied to a nondestructive evaluation of FRP strengthening systems bonded on concrete structures, Construction and Building Materials 25(2) (2011) 568-574.
[22] E. Bauer, P.M. Milhomem, L.A.G. Aidar, Evaluating the damage degree of cracking in facades using infrared thermography, Journal of Civil Structural Health Monitoring 8(3) (2018) 517-528.
[23] J.-H. He, D.-P. Liu, C.-H. Chung, H.-H. Huang, Infrared Thermography Measurement for Vibration-Based Structural Health Monitoring in Low-Visibility Harsh Environments, Sensors 20(24) (2020) 7067.
[24] J. Yang, W. Wang, G. Lin, Q. Li, Y. Sun, Y. Sun, Infrared thermal imaging-based crack detection using deep learning, Ieee Access 7 (2019) 182060-182077.
[25] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770-778.
[26] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4700-4708.
[27] M. Tan, Q. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, International Conference on Machine Learning, PMLR, 2019, pp. 6105-6114.
[28] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, Mobilenetv2: Inverted residuals and linear bottlenecks, Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 4510-4520.

[29] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Transactions on knowledge and data engineering 22(10) (2009) 1345-1359.

[30] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, C. Liu, A survey on deep transfer learning, International conference on artificial neural networks, Springer, 2018, pp. 270-279.

[31] Y. Gao, K.M. Mosalam, Deep transfer learning for image‐based structural damage recognition, Computer‐Aided Civil and Infrastructure Engineering 33(9) (2018) 748-768.

[32] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, 2009 IEEE conference on computer vision and pattern recognition, Ieee, 2009, pp. 248-255.

[33] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, Pytorch: An imperative style, high-performance deep learning library, Advances in neural information processing systems 32 (2019) 8026-8037.

[34] L. Melas-Kyriazi, EfficientNet PyTorch, 2021. https://github.com/lukemelas/EfficientNet-PyTorch. (Accessed October 26, 2021).

[35] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, Scikit-learn: Machine learning in Python, the Journal of machine Learning research 12 (2011) 2825-2830.

[36] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).

[37] J. Ma, H. Zhang, Z. Shao, P. Liang, H. Xu, GANMcC: A generative adversarial network with multiclassification constraints for infrared and visible image fusion, IEEE Transactions on Instrumentation and Measurement 70 (2020) 1-14.

[38] B. Wang, J. Mendez, M. Cai, E. Eaton, Transfer learning via minimizing the performance gap between domains, Advances in Neural Information Processing Systems 32 (2019).

Chapter 3 ASPHALT PAVEMENT CRACK DETECTION BASED ON CONVOLUTIONAL NEURAL NETWORK AND INFRARED THERMOGRAPHY[1]

3.1 Introduction

How to more efficiently and accurately detect pavement cracks? Being subjected to heavy and repetitive loading, and undesirable environmental impact (e.g. freeze-thaw), pavements need periodic inspections, assessments, maintenance, and rehabilitation to ensure their safety and prolong their service life [1, 2]. Crack is one of the crucial distresses for pavement because crack is the common symptom of pavement [3]. Different from other civil infrastructures, pavement crack types could be mainly separated into three categories: transverse crack, longitudinal crack, and alligator crack, which may be attributed to different causes [4, 5]. Pavement cracks could reduce the load-spreading and water-resistant capacity of pavement as well as accelerate the degradation process of pavement surface [6]. Pavement crack detection strongly depends on manual inspection and contact- or embedded-sensor-based methods [3, 7]. However, manual inspection is susceptible, unreliable, and time-consuming while it is dangerous in some cases [7]. The application of sensor-based methods is also limited due to environmental disadvantages and vulnerability to temperature and humidity changes [3]. Therefore, a safe, reliable, accurate, non-contact, and efficient method is needed for pavement crack detection.

The convolutional neural network (CNN) could be an ideal technique for crack detection. With the rapid development of deep learning, CNN has become a powerful and efficient tool for computer vision tasks, such as image recognition, semantic segmentation, and object detection. Several studies have illustrated the validity and usefulness of CNN in detecting pavement cracks [2, 8-11]. All these works evaluated their methods and compared their methods with previous methods based on the accuracy criterion (e.g. precision and F_1 score). However, only the accuracy could not fully illustrate the performance of CNN models. The model and computational complexity also play an important role in evaluating CNN [12, 13]. Although some time-consuming CNN models achieve state-of-the-art performance, they could be unaffordable or unnecessary for engineering applications [12]. These problems highlight the importance of balance between accuracy and complexity and urge to find an accurate and efficient CNN model for detecting pavement cracks.

Another issue is the indistinct edges of cracks and asphalt pavement surface [14, 15]. Asphalt pavement is bonded granular materials; randomly distributed particles make the edges of cracks and pavement surface indistinct [14] (Figure 3-1a). Therefore it may cause misidentification of many noises in the background as crack fragments [15]. Infrared thermography (IRT), as a contactless, non-invasive, and non-destructive

[1] Part of this chapter is published as journal papers:

Fangyu Liu, Jian Liu, and Linbing Wang. "Asphalt Pavement Crack Detection Based on Convolutional Neural Network and Infrared Thermography." IEEE Transactions on Intelligent Transportation Systems (2022).

testing method, has been successfully applied in civil engineering, such as assessment of HMA paving and compaction [16], debonding detection in asphalt pavements [17], and defect identification in composite materials [18]. Different from visible images that only provide gray-level information, IRT could generate gray-level information and temperature information [14, 19]. The significant temperature difference between cracks and pavement surfaces [14, 16, 20] could better distinguish cracks in infrared images than in visible images (Figure 3-1), thereby improving the detection accuracy.

Therefore, this work aims to propose a robust crack detection method based on CNN and infrared thermography. The following summarizes the primary contribution of this work:

- Build an open benchmark dataset for crack detection based on three types of images, including visible images, infrared images, and the fusion of visible and infrared images (in short, fusion images).

- The dataset also considers different conditions and periods, including single and multi cracks, thin and thick cracks, clean and rough backgrounds, light and dark backgrounds, and three periods in a day.

- Five classical CNN segmentation models and two UNet-based models are trained and evaluated on the aforementioned dataset to make a benchmark.

- Evaluation metrics (accuracy, computational complexity, and model complexity) are used to have an overall evaluation of CNN segmentation models rather than the only accuracy metrics.

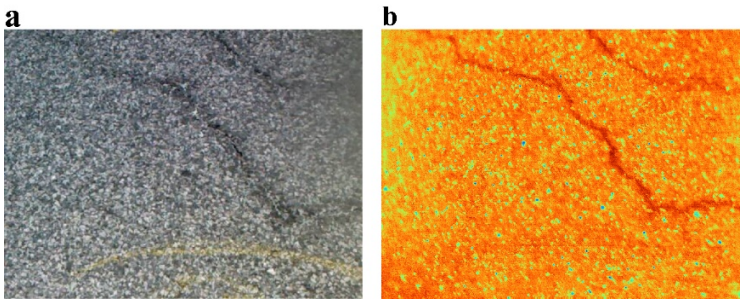

Figure 3-1 Examples of (a) the visible image and (b) infrared image

3.2 Methods

3.2.1 Infrared Camera

The infrared camera is FLUKE TiX580 and has two lenses, including a visible light camera lens and an infrared camera lens (Figure 3-2a). The two lenses enable this infrared camera to take the visible image and infrared image simultaneously. The image size is 640×480. In addition to only visible images and infrared images, the fusion of visible and infrared images could be obtained by using the IR-FusionTM technology to adjust the blending level of visible images and infrared images. The display screen in the back of the

infrared camera (Figure 3-2b) could show the image to help to adjust the camera angle and camera range. In addition to visualization, it could also edit emissivity, enable color alarms and makers, and adjust the blending level of visible and infrared images. After storing images on computers, the software (SmartView) could also adjust the level of visible and infrared light (Figure 3-2c).

Figure 3-2 Overview of the infrared camera and software. (a) The front of the infrared camera (including the visible light camera lens and infrared camera lens), (b) the back of the infrared camera, and (c) the software used to process images (including temperature information and blending level of images).

3.2.2 Segmentation Models

Crack detection could be divided into two main categories: object detection and segmentation. Object detection needs to distinguish objects in an image or video and segmentation needs to label each pixel to certain classes. Specifically for crack detection, object detection often points out the location and types of cracks. Different from object detection, segmentation would label each pixel as 'crack' or 'non-crack' and its outputs are usually binary images, while white ('1') often represents 'crack' and black ('0') means 'non-crack'. This method could be used for qualitative and quantitative analysis, such as the level and direction of cracks. The work presented in this paper would focus on segmentation in terms of crack detection. As an approach for automatically detecting important features without human supervision, CNN has been widely applied in image segmentation. Common CNN models for this application include Pyramid Scene Parsing Network (PSPNet) [21], Feature Pyramid Networks (FPN) [22], Fully Convolutional Networks (FCN) [23], DeepLab [24], and UNet [25]. PSPNet was a multi-scale network and combined up-sampling and concatenation to capture both local and global context information [21]. FPN was mainly proposed for object detection and then was also applied in image segmentation, while it consisted of a bottom-up pathway, a top-down pathway, and lateral connections [22]. FCN, as one of the first CNN models for image segmentation, replaced the fully connected layers of typical CNN architectures with fully convolutional layers and combined semantic information and appearance information by skip connections [23]. DeepLab used the atrous spatial pyramid pooling (ASPP) and combined the cascade and parallel modules of dilated convolution [24]. UNet was initially proposed for the segmentation of biological microscope images and had two parts, encoder (down-sampling) and decoder (up-sampling) [25]. UNet-based models (UNet-

ResNet101 and UNet-VGG19) were built by replacing the encoder of the UNet with typical CNN architectures. The encoders of FCN, FPN, and PSPNet are VGG-19 [26] in this work, while the encoder of DeepLabv3 is ResNet-101 [27]. Models are established based on [28-30].

3.2.3 Evaluation Metrics

To better evaluate the performance of segmentation models and keep a balance between accuracy and complexity, three evaluation metrics are used, including accuracy, computational complexity, and model complexity.

3.2.3.1 Accuracy

Many criteria have been proposed to evaluate the accuracy of segmentation models and pixel accuracy, mean pixel accuracy, and mean intersection over union are the most popular accuracy metrics used to measure the performance of per-pixel labeling methods on image segmentation [13]. Pixel accuracy (PA) calculates the ratio between the number of correctly classified pixels and their total number and mean pixel accuracy (MPA) is an improved PA that considers the per-class basis [13]. Mean intersection over union (MIoU) calculates the ratio between the intersection and the union of two sets (the ground truth and predicted segmentation) and it is the most used metrics because of its representativeness and simplicity [13]. In addition, the F_1 score is the measure of accuracy in binary classification. Suppose there are a total of k+1 classes (2 classes for crack detection, 'crack' and 'non-crack') and p_{ij} refers to the number of pixels of class i inferred to belong to class j. This means p_{ii}, p_{ij}, and p_{ji} represent the number of true positives (*TP*), false positives (*FP*), and false negatives (*FN*), respectively.

- Pixel Accuracy (PA):

$$PA = \frac{\sum_{i=0}^{k} p_{ii}}{\sum_{i=0}^{k} \sum_{j=0}^{k} p_{ij}} \tag{3-1}$$

- Mean Pixel Accuracy (MPA):

$$MPA = \frac{1}{k+1} \sum_{i=0}^{k} \frac{p_{ii}}{\sum_{j=0}^{k} p_{ij}} \tag{3-2}$$

- Mean Intersection over Union (MIoU):

$$MIoU = \frac{1}{k+1} \sum_{i=0}^{k} \frac{p_{ii}}{\sum_{j=0}^{k} p_{ij} + \sum_{j=0}^{k} p_{ji} - p_{ii}} \tag{3-3}$$

- F_1 score:

$$F_1\ score = 2 \times \frac{Precision \cdot Recall}{Precision + Recall} = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \tag{3-4}$$

3.2.3.2 Computational and Model Complexity

The computational complexity of CNN reflects the speed or runtime. It is a very valuable metric as most CNN models must meet strict requirements on how much time can be spent on the inference pass [13]. However, the exact timings might be meaningless because it strongly relies on the hardware. Depending on the architecture of CNN, the floating-point operations (FLOPs) could represent the execution time or computational complexity of CNN [31]. Therefore, FLOPs are used to evaluate the computational complexity of segmentation models in this work.

Another important evaluation metric is the model complexity. This metric heavily depends on the number of the model's parameters. More parameters mean the model is more complex and needs more training data to avoid overfitting problems. As almost all deep learning methods need Graphics Processing Units (GPUs) to speed up, the GPU memory proposes the requirements on the model complexity. Generally speaking, the higher the model complexity, the higher the needed GPU memory. This might be a challenge to a research group. The FLOPs and the number of parameters are computed based on [32].

3.2.3 Hyperparameters of Models

The segmentation models were trained and evaluated on PyTorch [33]. The inputs are RGB images (size: 640×480×3) and the outputs or ground truths are binary images (size: 640×480×1). The loss function is the Binary Cross-Entropy (BCE) loss and the optimization algorithm is Adam [34]. Other hyperparameters include the epoch (100), learning rate (0.0001), and batch size (1, mini-batch). For all epochs, the accuracy metrics of the model were immediately evaluated after training. The model was saved every five epochs.

3.3 Dataset

There is rarely an open benchmark dataset for crack detection based on IRT, although few previous studies provided small datasets for crack detection. Therefore, an open benchmark dataset for IRT-based crack detection was established in this work[2]. There are four types of images in this dataset, including the visible image, infrared image, fusion image, and ground truth. The visible image and infrared image are fully visible and infrared, respectively. The fusion image is the combination of visible and infrared images achieved by IR-Fusion™ technology, which is approximately 50% and 50%, respectively. According to the previous literature [7, 35, 36], the ground truth in the dataset is manually labeled at the pixel level by using

[2] Available at: https://github.com/lfangyu09/IR-Crack-detection

Photoshop software.

As the temperature difference between cracks and pavement surfaces enables IRT to distinguish cracks, the temperature change in the whole day might influence this temperature difference. Therefore, data (images) were acquired at three periods, including morning (8:00 am), noon (12:00 pm), and dusk (5:00 pm). The highest pavement surface temperature is almost consistent with the daily maximal temperature, while the dusk temperature is a little higher than that of the morning. Figure 3-3 shows examples of the visible image, infrared image, and fusion image at the same location in three periods. Although these visible images are greatly similar and even identical, their infrared images and fusion images are significantly different. Firstly, their temperatures are different. Images at noon (Figure 3-3b) have the highest temperature, ranging from 17.1 to 31.0 ℃, while images at dusk (Figure 3-3c) have the second-highest temperature (5.0 to 15.9 ℃) and images in the morning (Figure 3-3a) have the lowest temperature (0.5 to 7.5 ℃). This means the same color in these images represents different temperatures. Secondly, the crack area has the highest temperature but the background has different patterns. The image background at noon approximately has the middle temperature of its temperature range. The image background temperature at dusk is higher than the middle temperature and so does the image background temperature in the morning, while its color is much closer to red (much higher than the middle temperature). Thirdly, the distinction of cracks is different. Some crack areas in images at noon have a similar temperature to the background, causing some misidentification of cracks. The distinction of cracks in images at dusk and in the morning is more clear and more obvious. In addition, shadows caused by guardrails and trees might affect both visible images and infrared images. To eliminate such environmental factors, images were only taken from the pavement without guardrails and trees.

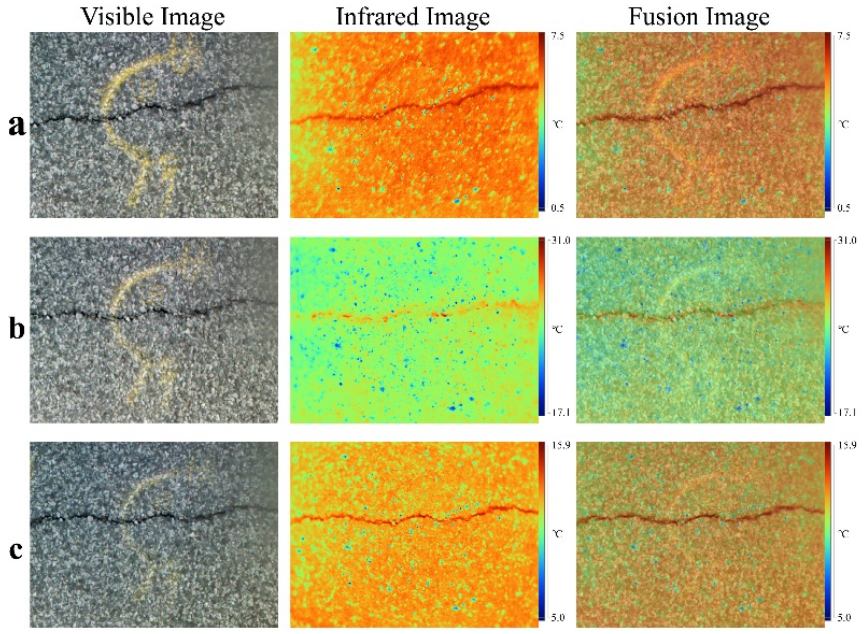

Figure 3-3 Examples of the visible image, infrared image, and fusion image at the same location in three

periods. (a) Morning, (b) noon, and (c) dusk.

Figure 3-4 shows examples of the visible image, infrared image, fusion image, and ground truth in the dataset. There are eight types of conditions, including single and multi cracks, thin and thick cracks, clean and rough backgrounds, and light and dark backgrounds. These different images are used to evaluate the performance of segmentation models under different conditions. It can be observed that in the infrared image or fusion image, it is easy to distinguish cracks when the width of cracks is small (Figure 3-4a-c, g, and h). When the width of cracks increases, the boundary of cracks and the background is not clear and there is a buffer zone, namely color gradient areas, between these two areas (Figure 3-4d and e). Another phenomenon that needs to be figured out is that the wet area generally has a lower temperature (Figure 3-4d and f).

Table 3-1 illustrates the summary of this dataset. There are a total of 448 images for each type (4 types, including the visible image, infrared image, fusion image, and ground truth). 186 images were taken in the morning, 142 at noon, and 120 at dusk. To train and evaluate the segmentation models, the whole dataset is separated into two subsets, the training set, and the test set. The training set has 382 images and the test set has 66 images. In addition, the percentages of crack pixels and non-crack pixels are also summarized in Table 3-1. The crack area only occupies a small part (less than 4%) of the whole image.

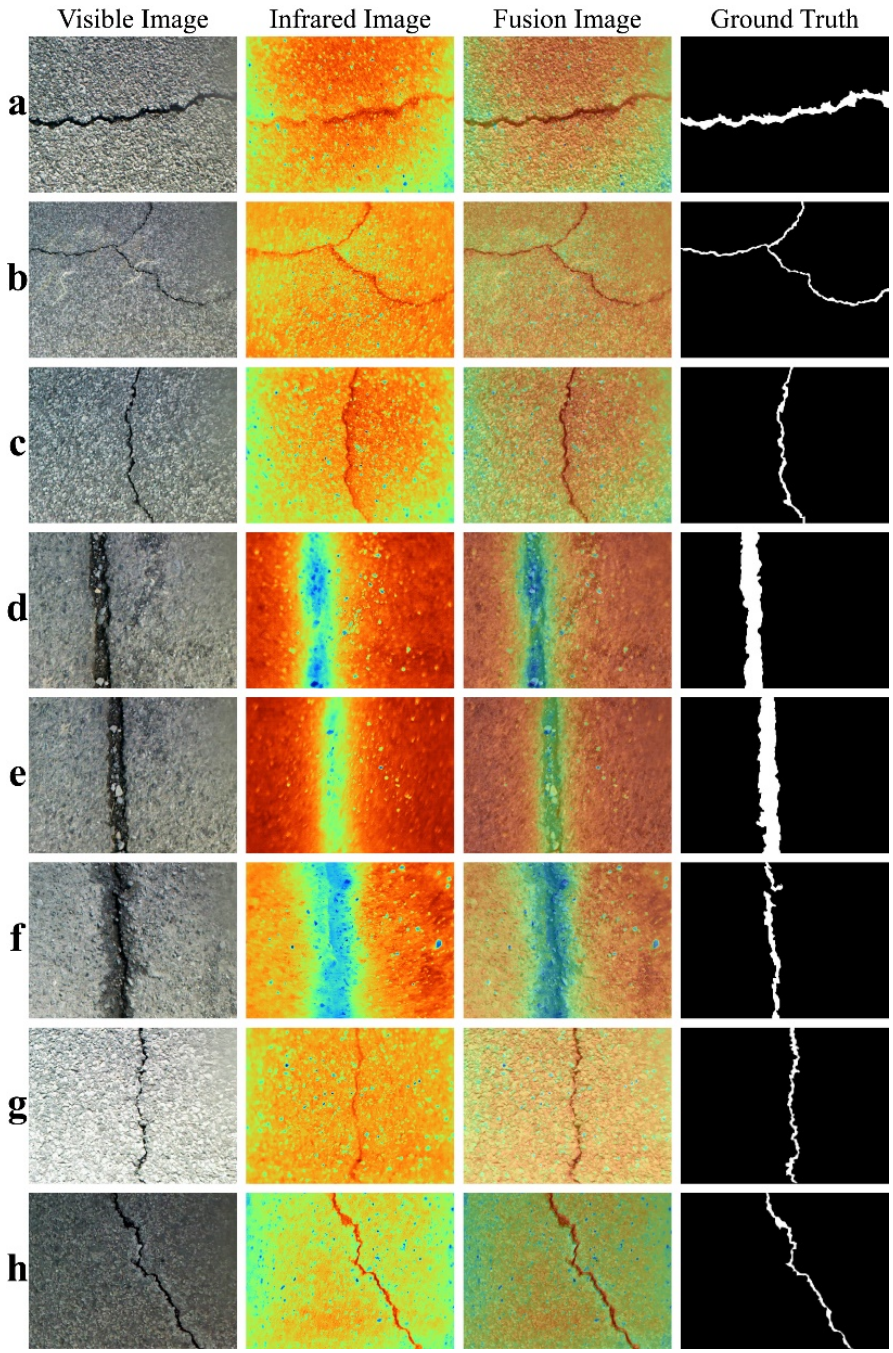

Figure 3-4 Examples of the visible image, infrared image, fusion image, and ground truth in the dataset. (a) Single crack, (b) multi cracks, (c) thin crack, (d) thick crack, (e) clean background, (f) rough background, (g) light background, and (h) dark background.

Table 3-1 Summary of the dataset

Dataset	Number of Images	Crack Pixel (%)	Non-crack Pixel (%)
Morning	186	3.85	96.15
Noon	142	3.97	96.03
Dusk	120	3.20	96.80
Train	382	3.86	96.14
Test	66	2.88	97.12
Total	448	3.71	96.29

3.4 Results and Discussion

3.4.1 Loss

Figure 3-5 shows the loss curves of the visible image, infrared image, and fusion image. These three types of images have different patterns for the loss curves. For the visible image (Figure 3-5a), the loss of almost all segmentation models tends to converge with the increasing epochs but the loss curve of DeepLabv3 experiences huge fluctuations, while UNest-ResNet101 has a slight fluctuation. Different from the visible image, the loss curve of DeepLabv3 for the infrared image (Figure 3-5b) tends to be stable at a low value as the epoch increases and so does the loss curve of FPN. Other segmentation models face the overfitting problem to some degree. Their loss curves firstly decrease and then gradually increase, especially PSPNet and UNest-ResNet101. To eliminate the effect of the overfitting problem, the final results of models were selected as the best results during all epochs rather than the results of the final epoch. The loss curves for the fusion image (Figure 3-5c) have a different pattern. All segmentation models tend to converge when the epoch increases and their loss curves are almost overlapped at a very small value after Epoch 40. From the perspective of the loss curve, the fusion image is a better choice as the loss curves of all segmentation models could converge at a small value.

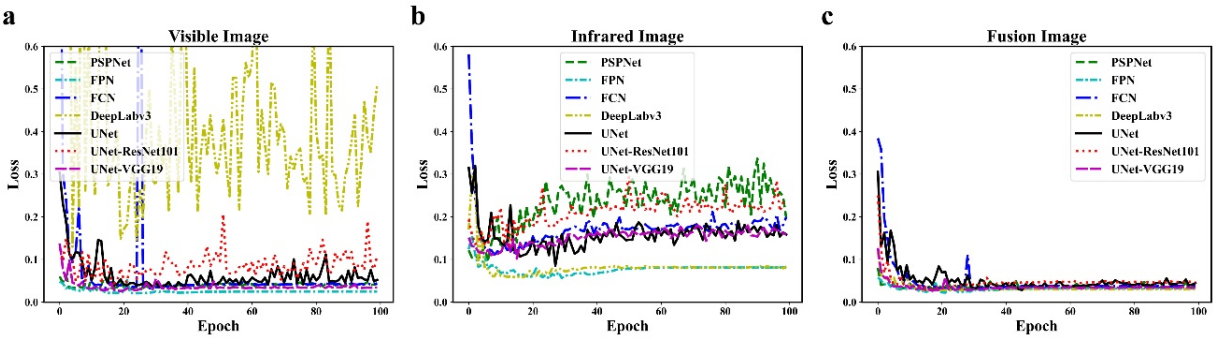

Figure 3-5 Loss curves of the test set. (a) Visible images, (b) Infrared images, (c) Fusion images

3.4.2 Evaluation Metrics

3.4.2.1 Accuracy

The outputs (predicted probability maps) of segmentation models range from 0 to 1. To get the binary image and compute the accuracy, it needs to binarize these outputs to 0 or 1 with thresholds. 0.5 was selected as the threshold as it is often used in the previous literature [36] and is also simple.

Figure 3-6 shows the curves of accuracy metrics of the visible image, including F_1 score, PA, MPA, and MIoU. For the pixel accuracy (Figure 3-6b), these segmentation models tend to be stable and reach a high value after Epoch 40, except for UNet-ResNet101. In comparison, for other accuracy metrics UNet, DeepLabv3, and UNet-ResNet101 have experienced huge fluctuations while UNet-ResNet101 has a good performance in MPA (Figure 3-6c). The other segmentation models tended to stabilize in all accuracy metrics as the epoch increases. Table 3-2 summarizes the final performance of segmentation models on the visible image. FPN obtains the highest value in three accuracy metrics (F_1 score, MPA, and MIoU) and it is 0.001 smaller than PSPNet in PA. In comparison, UNet-ResNet101 has the lowest value in almost all accuracy metrics except for MPA. According to the performance during the training and final results, FPN is the best segmentation model for the visible image among these segmentation models.

The results of the infrared image show a different pattern. Figure 3-7 shows the curves of accuracy metrics of infrared images. The values of FPN and DeepLabv3 are much higher (significant gap) than that of other segmentation models in F_1 score, MPA, and MIoU, while all segmentation models have similar performance in PA (Figure 3-7b) except for UNet-ResNet101. There is a little difference between other segmentation models in F_1 score, MPA, and MIoU. Table 3-3 illustrates the final results of the segmentation models on the infrared image. The value of all accuracy metrics for the infrared image is much lower than that of the visible image. DeepLabv3 achieves the highest value in all accuracy metrics for the infrared image although it has the second-lowest value in all accuracy metrics for the visible image. All accuracy metrics of FPN are slightly lower than that of DeepLabv3. These two models have a far better performance than the other segmentation models.

The results of the fusion image are similar to that of the visible images. Figure 3-8 shows the curves of accuracy metrics for the fusion image. Although UNet reveals some fluctuations in accuracy metrics, the other segmentation models tend to stabilize at a high value. Compared with the visible image, all segmentation models have a more stable performance in the fusion image. Table 3-4 reveals the final results of segmentation models on the fusion image. Similar to the visible image, FPN has the highest value in three accuracy metrics (F_1 score, PA, and MIoU). The high value of MPA has been observed in DeepLabv3 whose values of other accuracy metrics are a little lower than that of FPN. This is different from the results of the visible image.

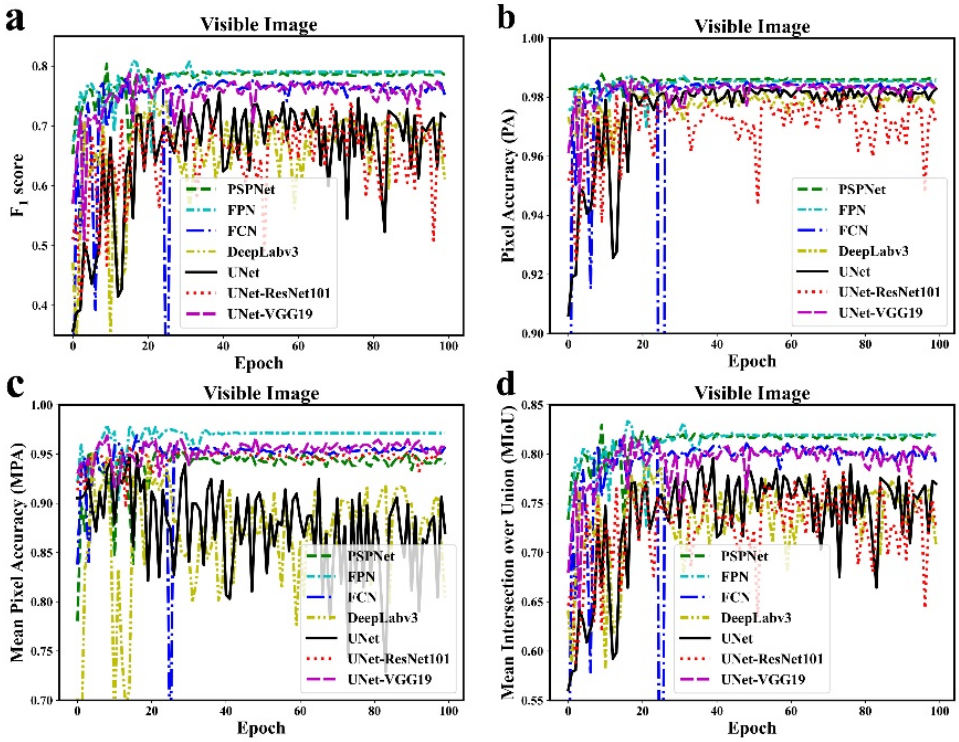

Figure 3-6 Accuracy metrics of the visible image. (a) F_1 score, (b) PA, (c) MPA, and (d) MIoU

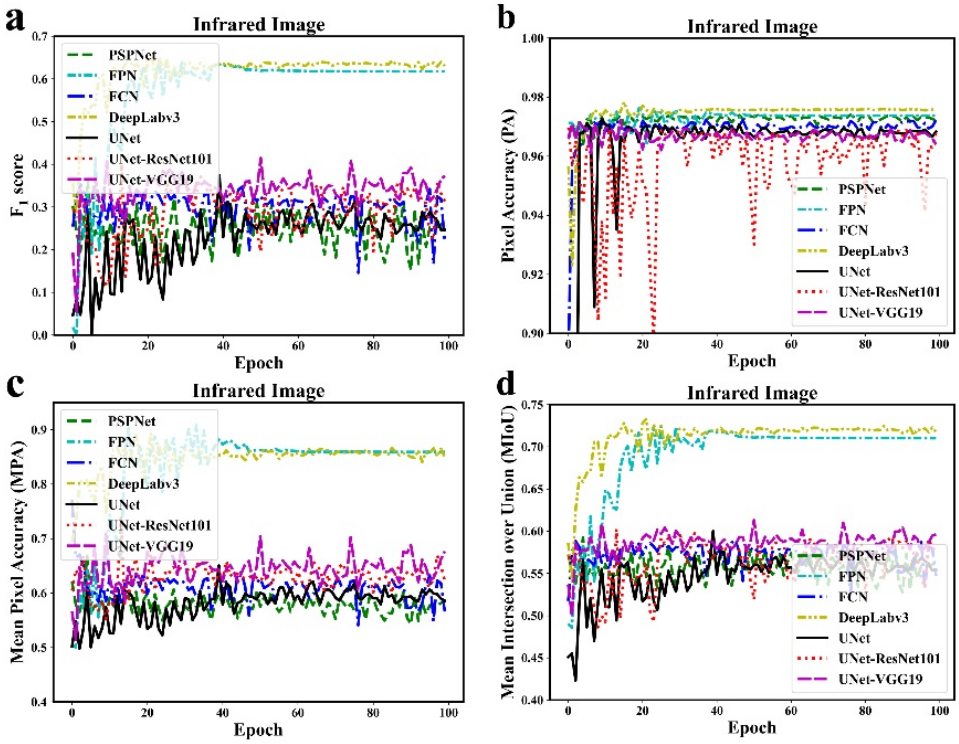

Figure 3-7 Accuracy metrics of the infrared image. (a) F_1 score, (b) PA, (c) MPA, and (d) MIoU

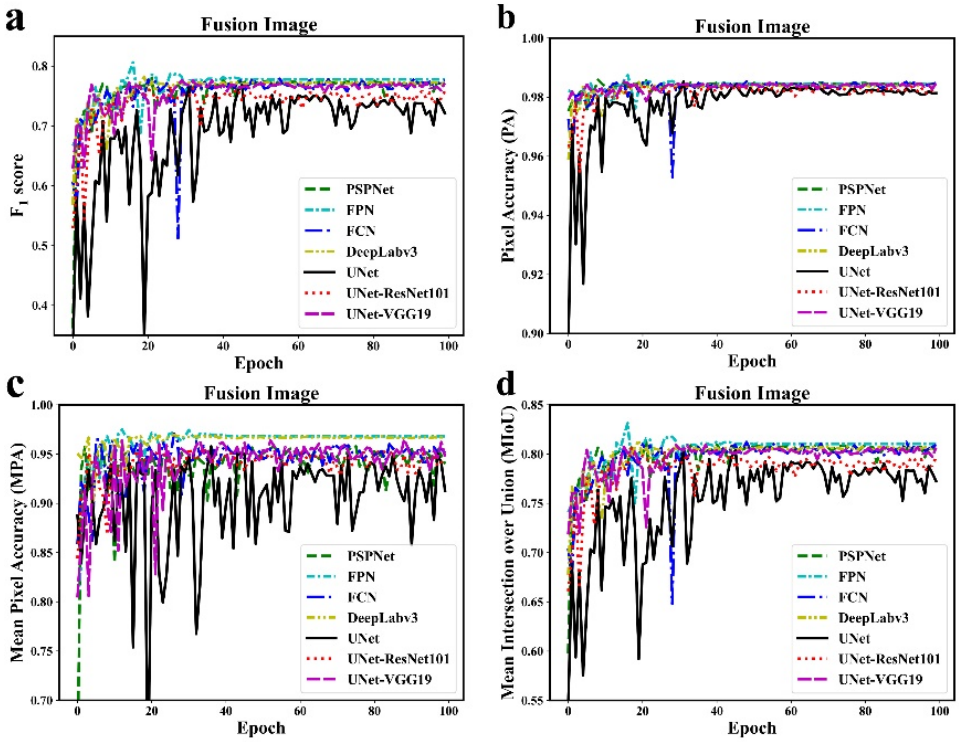

Figure 3-8 Accuracy metrics of the fusion image. (a) F_1 score, (b) PA, (c) MPA, and (d) MIoU

To better present the experimental results, Figure 3-9 summarizes the results of the accuracy metrics of segmentation models for three types of images. In general, the accuracy metrics of the visible image and fusion image are similar and even identical for all segmentation models, which are much better than that of the infrared image. FPN has a satisfying performance in all three types of images. And DeepLabv3 has a good performance in the infrared image and fusion image while compared with other segmentation models, its performance in the visible image is relatively poor but acceptable. The difference in the same accuracy metrics was caused by two main aspects: (1) the type of datasets. For the visible image, it could work as the baseline in this work as it has been always used in all segmentation tasks until now. For the infrared image, when the width of cracks is small, the boundaries between cracks and the background are very clear (Figure 3-4a-c, g, and h). But the boundaries fade out and the buffer zones (color gradient areas) appear as cracks become wider (Figure 3-4d and e). These phenomena increase the difficulty of crack detection and make the same segmentation models behave differently in the visible image and infrared image. For the fusion image, as it integrates 50% of the visible image and 50% of the infrared image, the boundaries keep clear no matter whether cracks are thin or wide. This enables segmentation models to have similar performances in the fusion image with the visible image. In addition, with the help of the gray-level information and temperature information [14, 19], the fusion image makes segmentation models more stable (Figure 3-5 and Figure 3-8). (2) the type of segmentation models. For CNN, shallower feature maps are of lower-level semantics but more accurately localized, while deeper feature maps are semantically stronger but spatially coarser [22, 37]. FPN merges these two types of features to maximize their corresponding advantages, namely, the high resolution from shallower feature maps and the strong

semantics from deeper feature maps [22, 37]. In addition, different from the traditional top-down architecture (e.g. UNet) which makes predictions on the finest level, FPN regards the top-down architecture as a feature pyramid and makes predictions independently at all levels [22]. Therefore, FPN shows a prominent performance in all three types of images. For DeepLab v3, it used the cascaded module to double the atrous rates while it also applies the atrous spatial pyramid pooling module augmented with image-level features for detecting features [38]. These enable DeepLab v3 to maintain a good performance when the type of the input image changes.

In addition to the types of segmentation models and images, these four accuracy metrics also influence evaluating the performance of segmentation models on crack detection. The PA of all images is very high (>0.95) and much larger than the other accuracy metrics. The three types of images do not have a huge difference in PA, and the largest gap among all segmentation models is less than 0.04. In comparison, for the other accuracy metrics, there is an obvious and big difference between these three types of images for all segmentation models and the largest difference is up to 0.44 in the F_1 score. This is because the pixel accuracy only computes the correctly classified pixels without considering the effect of the class ('crack' and 'non-crack'), while the other accuracy metrics (F_1 score, MPA, and MIoU) take the class into account to get the final value. The ground truth of crack detection (Figure 3-4) is the class imbalance and the crack pixel only occupies a very little part (less than 4%) of the whole image (Table 3-1). The high value (>0.95) of PA and low difference (<0.04) between the three types of images are mainly contributed to the correct classification of the background. After considering the effect of the class, the value of MPA decreases, and the difference between the three types of images increases. The F_1 score and MIoU have already considered the effect of the class in their formula. The effect of the class would distinguish the difference between the three types of images. However, crack detection mainly focuses on the crack area and this needs more emphasis on the class of 'crack' rather than all classes. In this case, the visible image and fusion image are better choices for crack detection than the infrared image.

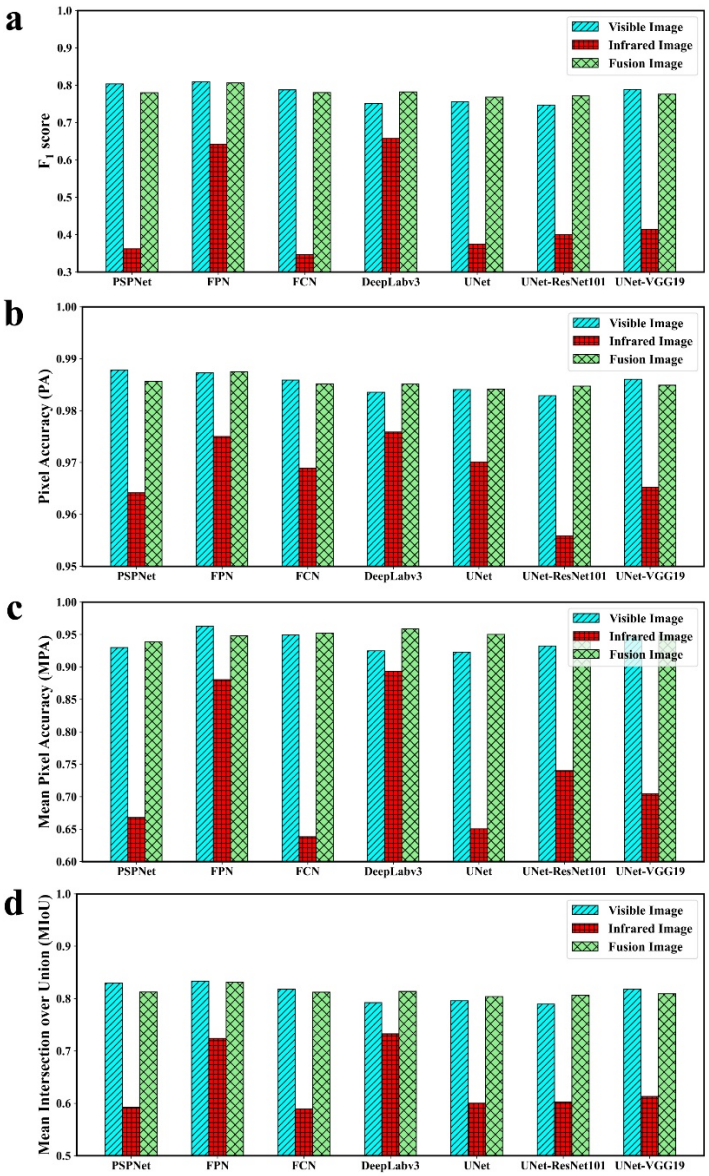

Figure 3-9 Accuracy metrics of segmentation models for three types of images. (a) F_1 score, (b) PA, (c) MPA, and (d) MIoU.

3.4.2.2 Computational and Model Complexity

Table 3-2-Table 3-4 show the computational and model complexity. As they are the internal characteristics of models, they are not affected by the types of images and they are the same for the three types of images. As mentioned above, the computational complexity is evaluated by FLOPs and the model complexity is determined by the number of parameters. DeepLabv3 has the highest value in the number of parameters and FLOPs. This means DeepLabv3 spends more execution time and also needs more training data to reduce the risk of overfitting. UNet has the lowest number of parameters, which is one-third of the value of DeepLabv3. The lowest FLOPs are observed in UNet-ResNet101, which is almost one-quarter of the value

of DeepLabv3, but the number of parameters for UNet-ResNet101 is approximately three times that of UNet. The FLOPs of UNet are about 2.5 times bigger than that of UNet-ResNet101. FPN has the third-lowest value in these two metrics. It could be found that there is no significant pattern between the number of parameters and FLOPs.

Table 3-2 Performance of different models for the visible image

Model	Accuracy Metrics				Parameters	FLOPs
	F_1	PA	MPA	MIoU	(M)	($1x10^9$)
PSPNet [21]	0.804	0.988	0.930	0.830	20.82	111
FPN [22]	0.809	0.987	0.963	0.833	22.11	131
FCN [23]	0.788	0.986	0.949	0.818	23.93	199
DeepLabv3 [24]	0.751	0.984	0.925	0.792	58.63	283
UNet [25]	0.756	0.984	0.923	0.796	17.27	188
UNet-ResNet101 [28]	0.747	0.983	0.933	0.790	51.51	73
UNet-VGG19 [28]	0.789	0.986	0.947	0.818	29.06	142

Table 3-3 Performance of different models for the infrared image

Model	Accuracy Metrics				Parameters	FLOPs
	F_1	PA	MPA	MIoU	(M)	($1x10^9$)
PSPNet [21]	0.362	0.964	0.668	0.592	20.82	111
FPN [22]	0.642	0.975	0.880	0.724	22.11	131
FCN [23]	0.347	0.969	0.638	0.589	23.93	199
DeepLabv3 [24]	0.658	0.976	0.893	0.733	58.63	283
UNet [25]	0.375	0.970	0.650	0.600	17.27	188
UNet-ResNet101 [28]	0.400	0.956	0.740	0.603	51.51	73
UNet-VGG19 [28]	0.415	0.965	0.704	0.613	29.06	142

Table 3-4 Performance of different models for the fusion image

Model	Accuracy Metrics				Parameters	FLOPs
	F_1	PA	MPA	MIoU	(M)	($1x10^9$)
PSPNet [21]	0.780	0.986	0.938	0.812	20.82	111
FPN [22]	0.806	0.987	0.948	0.831	22.11	131
FCN [23]	0.780	0.985	0.952	0.812	23.93	199
DeepLabv3 [24]	0.782	0.985	0.959	0.814	58.63	283
UNet [25]	0.768	0.984	0.950	0.804	17.27	188
UNet-ResNet101 [28]	0.771	0.985	0.942	0.806	51.51	73
UNet-VGG19 [28]	0.776	0.985	0.949	0.809	29.06	142

3.4.2.3 Accuracy vs. Computational and Model Complexity

To comprehensively evaluate the performance of segmentation models and the types of images, accuracy metrics, computational complexity, and model complexity are combined to make an analysis, and the results are shown in Figure 3-10. Considering the effect of the class imbalance, MIoU and MPA are selected to represent the accuracy. The best model in the three types of images is different while FPN could be ranked the top two in almost all accuracy metrics with the low FLOPs and the number of parameters. With the highest FLOPs (x coordinates) and the number of parameters (the size of the ball), DeepLabv3 could be the top two in the infrared image and fusion image but it is almost the worst model for the visible image. UNet-VGG19 could also be a good alternative for detecting cracks in the three types of images.

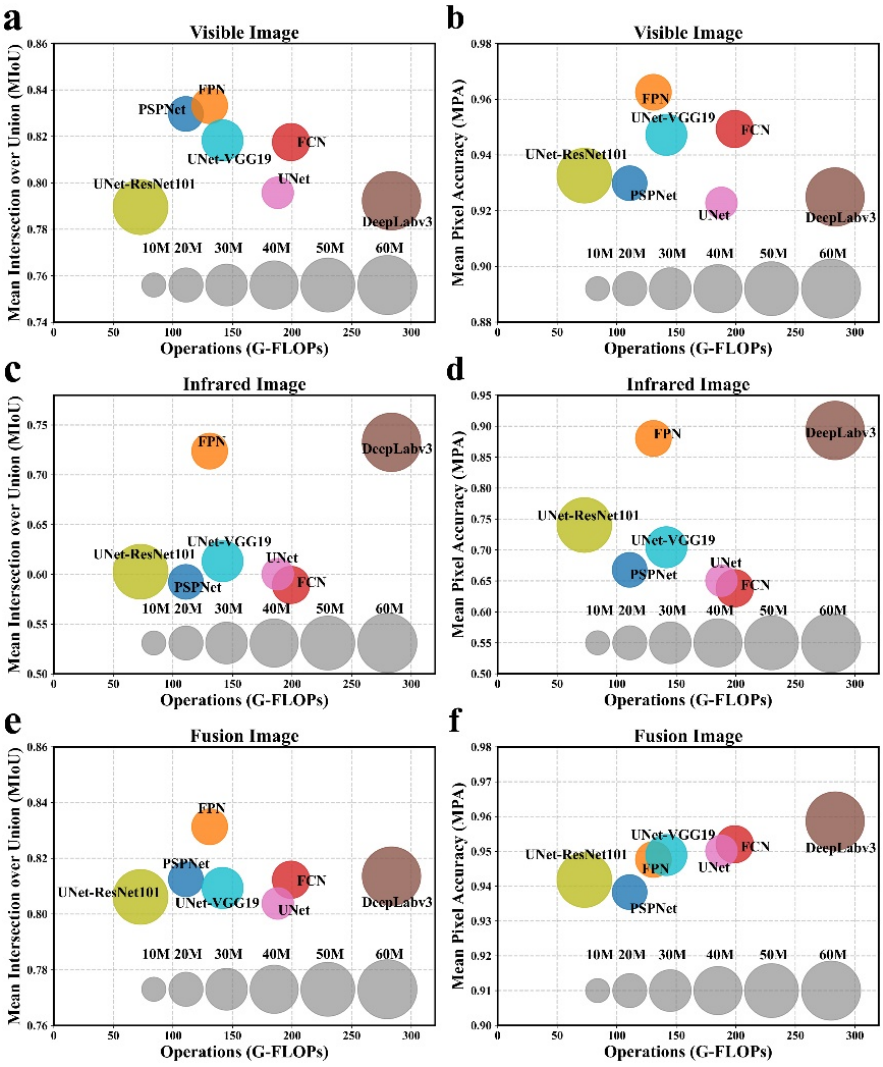

Figure 3-10 Ball chart reporting the accuracy metrics (MIoU and MPA) vs. computational complexity

(floating-point operations (FLOPs), G-FLOPs=1x10^9 FLOPs) with same thresholds. The size of each ball corresponds to the number of parameters for each model. (a) MIoU and (b) MPA for the visible image, (c) MIoU and (d) MPA for the infrared image, (e) MIoU and (f) MPA for the fusion image.

3.4.3 Prediction

In addition to evaluation metrics, prediction is also important to evaluate the performance of segmentation models as it could provide an intuitive assessment. As mentioned above, various conditions have been considered to build the dataset. The prediction under different conditions is used to further evaluate the performance of segmentation models. Based on evaluation metrics, three segmentation models (FPN, DeepLabv3, and UNet-VGG19) are selected to show their predictions on different conditions.

Figure 3-11 shows the predictions of three segmentation models for different cracks based on three types of images. In short, the predictions of the visible image and fusion image are almost identical for all segmentation models, which are much better than the infrared image, especially for multi and thick cracks. As for the performance of the three segmentation models, there is no significant difference in terms of the visible image and fusion image and their predictions are almost identical to the ground truth. DeepLabv3 has a much better prediction of the infrared image than the other two segmentation models.

The dataset also includes various backgrounds. Figure 3-12 shows the predictions of three segmentation models for different backgrounds based on three types of images. The predictions of the visible image and fusion image are still better than that of the infrared image for all segmentation models, while the visible image has a similar prediction with that of the fusion image. However, it is worth noting that when the background is rough, such as with water, the fusion image might be a better choice. The color of the pavement wetted by water is very close to that of cracks (Figure 3-4f and Figure 3-12b), thereby increasing the difficulty of detecting cracks based on the visible image. This is why the predictions of the three segmentation models are unsatisfactory. In comparison, water decreases the temperature and turns the wet areas blue in the infrared image and the fusion image (Figure 3-4f and Figure 3-12b). The wet cracks tend to have a lower temperature and their areas become bluer, thus helping to detect cracks for the infrared image and fusion image.

In addition to various cracks and backgrounds, the dataset takes the periods into account. Different periods in a day have different temperatures and crack detection heavily depends on the temperature difference between cracks and backgrounds for the infrared image. Therefore, it needs to investigate the effect of the temperature on crack detection for the infrared image. Figure 3-13 shows the predictions of three segmentation models for different periods based on three types of images. Although visible images in the three periods are almost identical, their corresponding infrared images and fusions are different (Figure 3-3 and Figure 3-13). The infrared image and fusion image in the morning and at dusk are similar and they are different at noon. It is observed that different periods in a day have no significant influence on the predictions of segmentation models for the infrared image and the fusion image.

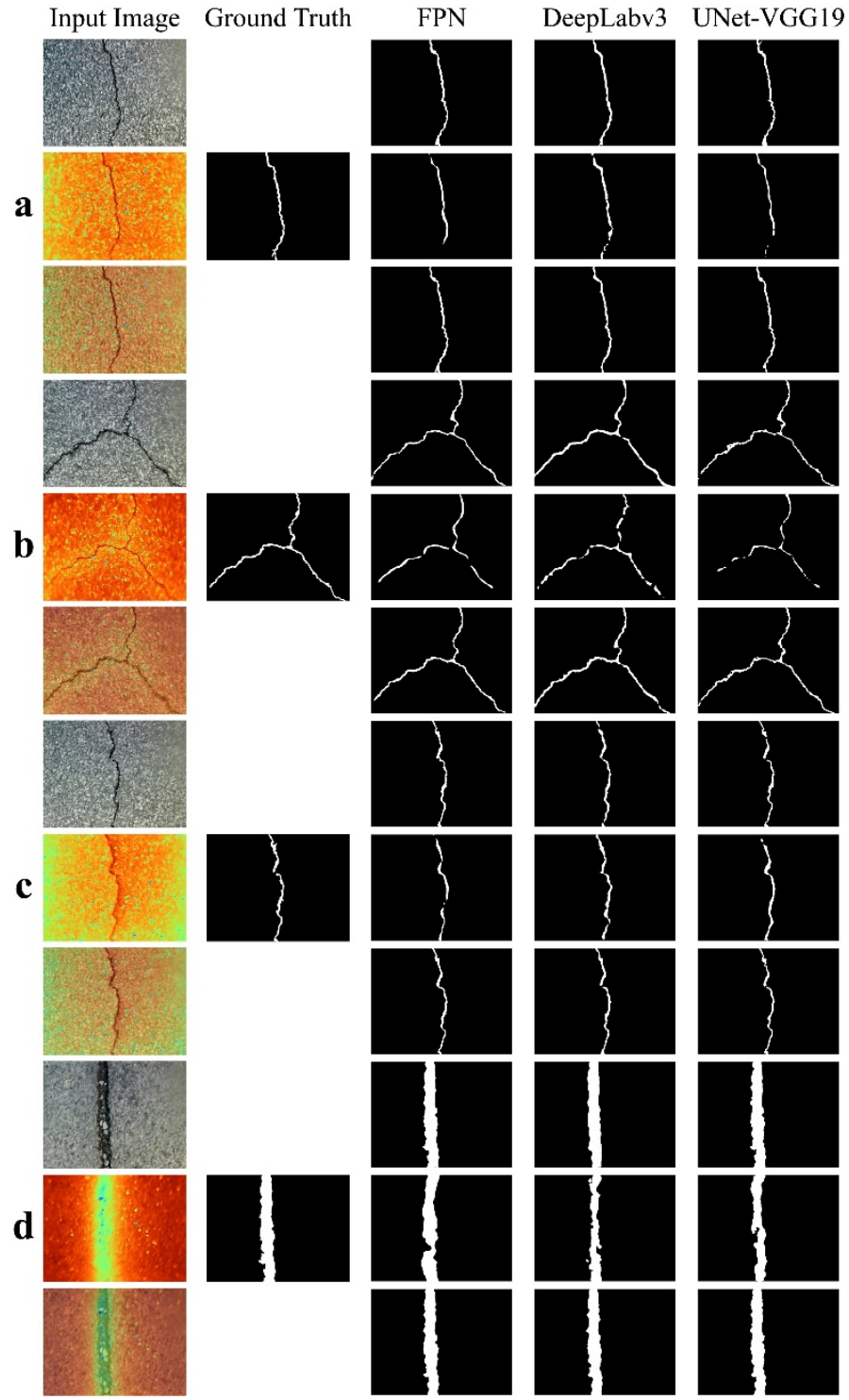

Figure 3-11 Predictions of three segmentation models for different cracks based on three types of images. (a) Single crack, (b) multi cracks, (c) thin cracks, and (d) thick cracks.

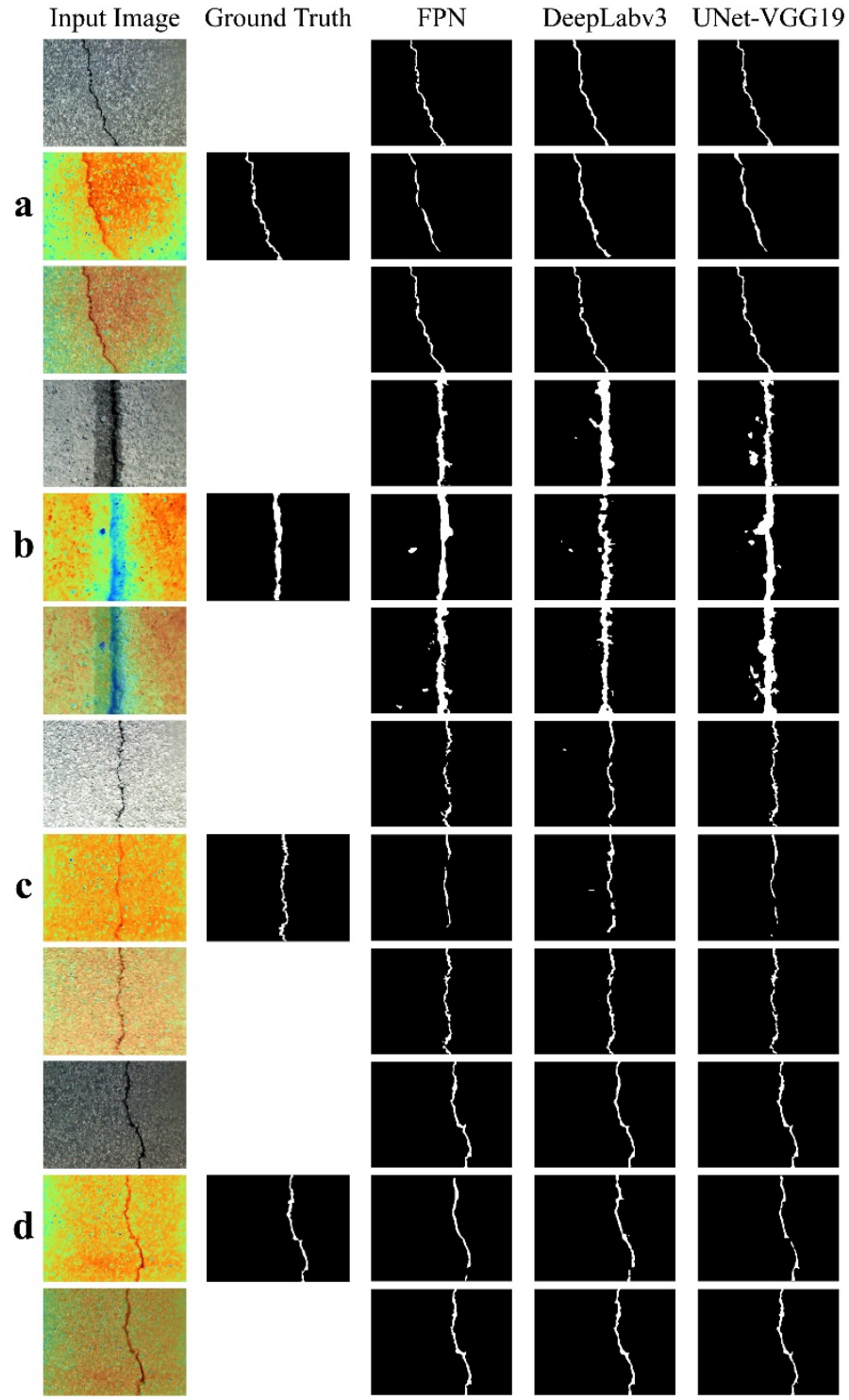

Figure 3-12 Predictions of three segmentation models for different backgrounds based on three types of images. (a) Clean background, (b) rough background, (c) light background, and (d) dark background.

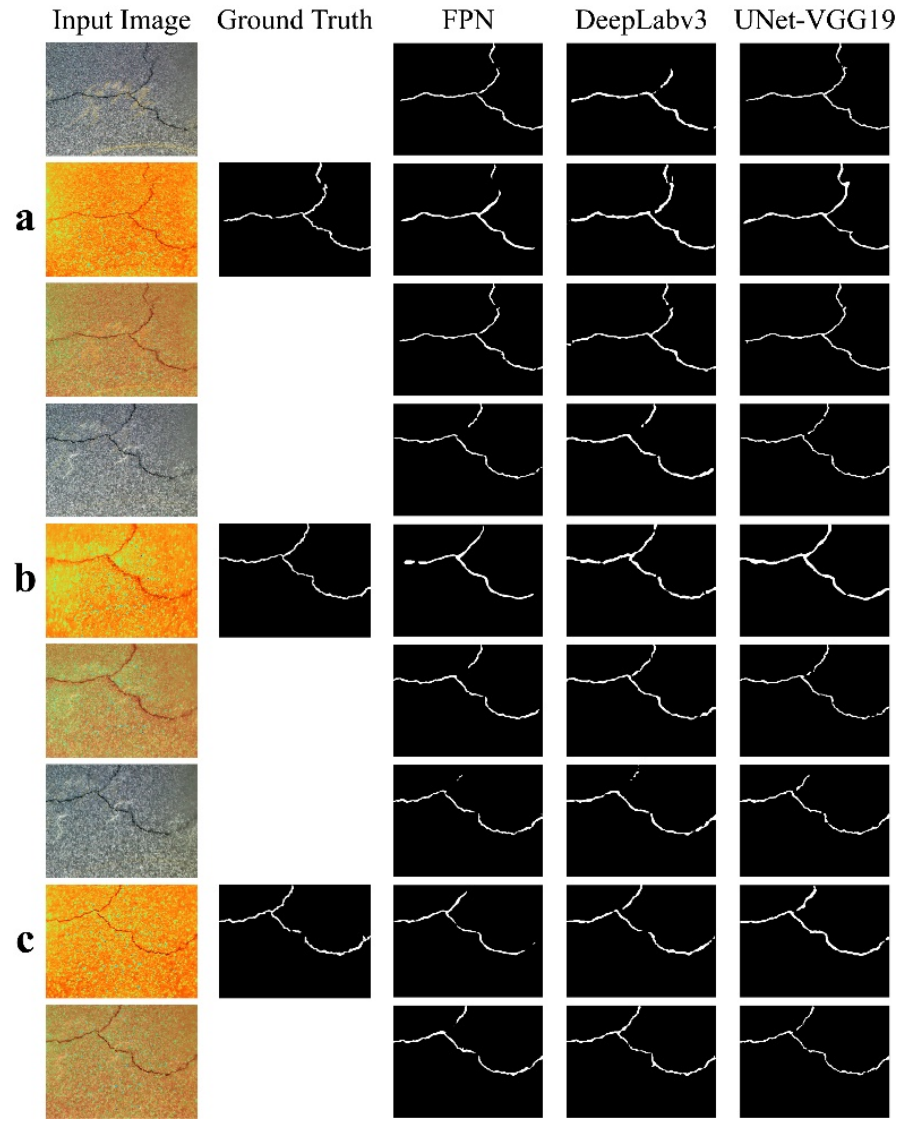

Figure 3-13 Predictions of three segmentation models for different periods based on three types of images. (a) Morning, (b) noon, and (c) dusk.

Another thing worth noting is the indistinct edges of cracks and asphalt pavement surface. When cracks are similar to the background (e.g. pavement texture), it is difficult for segmentation models to detect cracks based on visible images (Figure 3-12c and Figure 3-13). The predictions of segmentation models possibly miss some parts of cracks in terms of the visible image. In comparison, the temperature difference would make cracks more distinguished in the infrared image and fusion image than the visible image (Figure 3-12c and Figure 3-13), thereby improving the quality of predictions. As shown in Figure 3-12c and Figure 3-13, the predictions of all segmentation models for the fusion image are better than the visible image. Furthermore, the predictions of the infrared image are similar to that of the visible image (Figure 3-12c and Figure 3-13), although they are worse than the visible image under other conditions. Therefore, when it is difficult to distinguish the edge of cracks and asphalt pavement surface, the fusion image is a better choice

for crack detection.

To show the training process, the prediction of three segmentation models during different training epochs for three types of images is shown in Figure 3-14. Predictions of the visible image and fusion image tend to become satisfactory and stable after 15 epochs for all segmentation models, while predictions of the infrared image need almost 50 epochs to stabilize and are still worse than predictions of the visible image and fusion image. Segmentation models have different performances on three types of images. All these models predict accurately for the visible image and fusion image, while DeepLabv3 has a better prediction of the infrared image than the other two segmentation models. For FPN, the fusion image allows the model to obtain a satisfying prediction faster (Figure 3-14a in Epoch-5). But the predictions of Deeplabv3 are just the opposite and it is faster to predict accurately for the visible image. UNet-VGG19 gets a good prediction for the visible image and fusion image approximately at the same time.

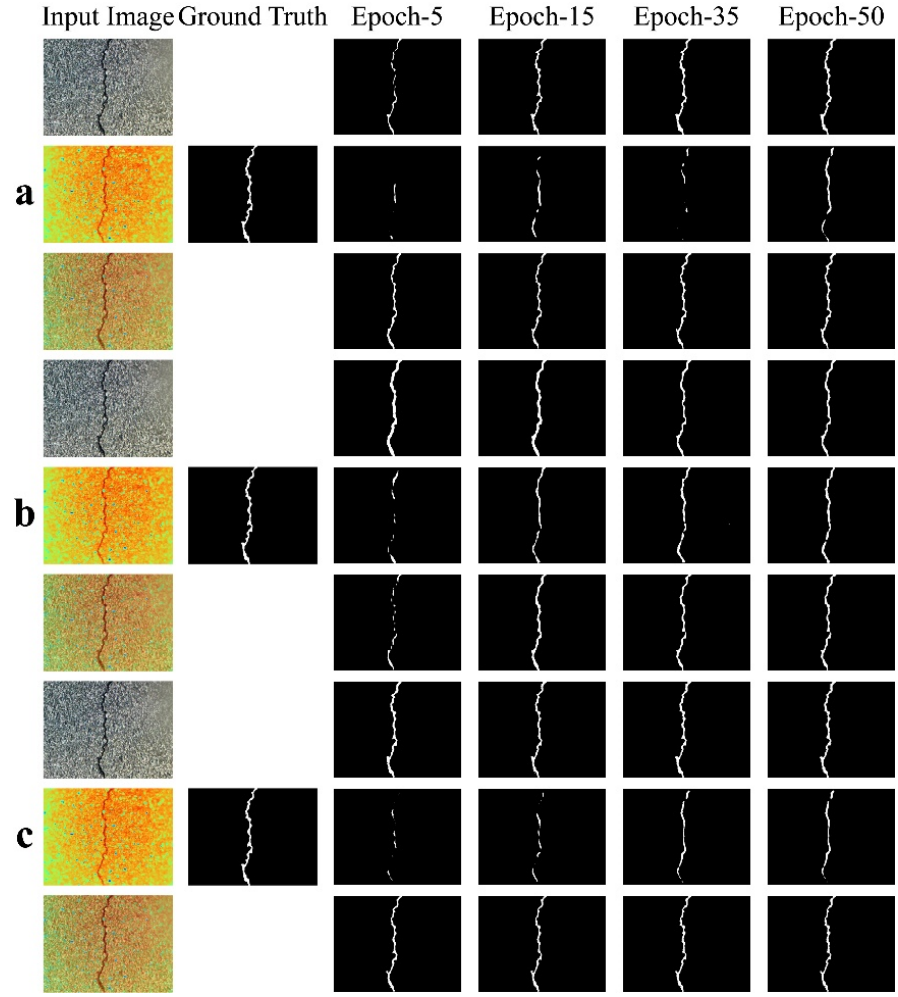

Figure 3-14 Predictions of three segmentation models for different epochs based on three types of images. (a) FPN, (b) DeepLabv3, and (c) UNet-VGG19.

3.5 Conclusion

CNN has become a powerful and efficient tool for detecting pavement cracks. One issue has been raised to find an accurate and efficient CNN model to keep a balance between accuracy and complexity in terms of crack detection. Another issue is the indistinct edges of cracks and asphalt pavement surface. This work aims to propose a robust crack detection method based on CNN and IRT. The main contribution of this work includes: (1) As generating gray-level information and temperature information, infrared thermography is used to better distinguish cracks, eliminating the effect of the indistinct edges between cracks and asphalt pavement surfaces. (2) Build an open benchmark dataset based on three types of images, including visible images, infrared images, and the fusion of visible and infrared images. This dataset also considers different conditions and periods, including single and multi cracks, thin and thick cracks, clean and rough backgrounds, light and dark backgrounds, and three periods in a day. (3) Seven CNN segmentation models are trained and evaluated on the aforementioned dataset to make a benchmark. (4) Evaluation metrics (accuracy, computational complexity, and model complexity) are used to have an overall evaluation of CNN segmentation models rather than only the accuracy metrics, thus finding an accurate and efficient CNN segmentation model.

Five classical CNN segmentation models and two UNet-based models are trained and evaluated on this dataset. The results show that the accuracy metrics and predictions of the visible image and fusion image are similar and even identical for all segmentation models, which are much better than that of the infrared image. When the background is rough (e.g. wet) or cracks are similar to the background (e.g. pavement texture), the fusion image is a better choice for crack detection as IRT distinguishes cracks by temperature differences. The different periods in a day have no significant influence on the predictions of segmentation models for the infrared image and the fusion image. Compared with the visible image, all segmentation models have a more stable performance for the fusion image. Among seven segmentation models, FPN could be the best model because of its high accuracy and low complexity.

Moreover, most of the images in this dataset include several cracks rather than a high percentage of cracks, which could be found by the ratio of crack pixels. In the future, more images would be collected from pavements with highly texted surfaces or with more complicated cracking patterns to enrich the dataset. And the size of the dataset would also increase by collecting and adding more images.

3.6 Reprint and Permission

©[2022] IEEE. Reprinted, with permission, from [Fangyu Liu, Jian Liu, and Linbing Wang. "Asphalt Pavement Crack Detection Based on Convolutional Neural Network and Infrared Thermography." IEEE Transactions on Intelligent Transportation Systems (2022).]. In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of Virginia Polytechnic Institute and State University's products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for

creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

References

[1] Y. Hou, Q. Li, C. Zhang, G. Lu, Z. Ye, Y. Chen, L. Wang, D. Cao, The state-of-the-art review on applications of intrusive sensing, image processing techniques, and machine learning methods in pavement monitoring and analysis, Engineering (2020).

[2] Y. Hou, Q. Li, Q. Han, B. Peng, L. Wang, X. Gu, D. Wang, MobileCrack: object classification in asphalt pavements using an adaptive lightweight deep learning, Journal of Transportation Engineering, Part B: Pavements 147(1) (2021) 04020092.

[3] D. Kang, S.S. Benipal, D.L. Gopal, Y.-J. Cha, Hybrid pixel-level concrete crack segmentation and quantification across complex backgrounds using deep learning, Automation in Construction 118 (2020) 103291.

[4] D. ASTM, Standard practice for roads and parking lots pavement condition index surveys, (2009).

[5] A. Cubero-Fernandez, F.J. Rodriguez-Lozano, R. Villatoro, J. Olivares, J.M. Palomares, Efficient pavement crack detection and classification, EURASIP Journal on Image and Video Processing 2017(1) (2017) 1-11.

[6] M. Salman, S. Mathavan, K. Kamal, M. Rahman, Pavement crack detection using the Gabor filter, 16th international IEEE conference on intelligent transportation systems (ITSC 2013), IEEE, 2013, pp. 2039-2044.

[7] Z. Liu, Y. Cao, Y. Wang, W. Wang, Computer vision-based concrete crack detection using U-net fully convolutional networks, Automation in Construction 104 (2019) 129-139.

[8] B. Li, K.C. Wang, A. Zhang, E. Yang, G. Wang, Automatic classification of pavement crack using deep convolutional neural network, International Journal of Pavement Engineering 21(4) (2020) 457-463.

[9] Z. Fan, Y. Wu, J. Lu, W. Li, Automatic pavement crack detection based on structured prediction with the convolutional neural network, arXiv preprint arXiv:1802.02208 (2018).

[10] H. Nhat-Duc, Q.-L. Nguyen, V.-D. Tran, Automatic recognition of asphalt pavement cracks using metaheuristic optimized edge detection algorithms and convolution neural network, Automation in Construction 94 (2018) 203-213.

[11] J. Huyan, W. Li, S. Tighe, Z. Xu, J. Zhai, CrackU‐net: A novel deep convolutional neural network for pixelwise pavement crack detection, Structural Control and Health Monitoring 27(8) (2020) e2551.

[12] K. He, J. Sun, Convolutional neural networks at constrained time cost, Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 5353-5360.

[13] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, P. Martinez-Gonzalez, J. Garcia-Rodriguez, A survey on deep learning techniques for image and video semantic segmentation, Applied Soft Computing 70 (2018) 41-65.

[14] Y. Du, X. Zhang, F. Li, L. Sun, Detection of Crack Growth in asphalt pavement through Use of infrared imaging, Transportation Research Record 2645(1) (2017) 24-31.

[15] Q. Zou, Y. Cao, Q. Li, Q. Mao, S. Wang, CrackTree: Automatic crack detection from pavement images,

Pattern Recognition Letters 33(3) (2012) 227-238.
[16] C. Plati, P. Georgiou, A. Loizos, Use of infrared thermography for assessing HMA paving and compaction, Transportation Research Part C: Emerging Technologies 46 (2014) 192-208.
[17] V. Vyas, V.J. Patil, A.P. Singh, A. Srivastava, Application of infrared thermography for debonding detection in asphalt pavements, Journal of Civil Structural Health Monitoring 9(3) (2019) 325-337.
[18] H.-T. Bang, S. Park, H. Jeon, Defect identification in composite materials via thermography and deep learning techniques, Composite Structures 246 (2020) 112405.
[19] T. Yu, A. Zhu, Y. Chen, Efficient crack detection method for tunnel lining surface cracks based on infrared images, Journal of Computing in Civil Engineering 31(3) (2017) 04016067.
[20] A.A. Oloufa, H.S. Mahgoub, H. Ali, Infrared thermography for asphalt crack imaging and automated detection, Transportation research record 1889(1) (2004) 126-133.
[21] H. Zhao, J. Shi, X. Qi, X. Wang, J. Jia, Pyramid scene parsing network, Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2881-2890.
[22] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2117-2125.
[23] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3431-3440.
[24] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A.L. Yuille, Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, IEEE transactions on pattern analysis and machine intelligence 40(4) (2017) 834-848.
[25] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, International Conference on Medical image computing and computer-assisted intervention, Springer, 2015, pp. 234-241.
[26] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 (2014).
[27] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770-778.
[28] P. Yakubovskiy, Segmentation Models Pytorch, (2020).
[29] P.C. Huang, The easiest implementation of fully convolutional networks, (2020).
[30] M. Alexandre, UNet: semantic segmentation with PyTorch, (2020).
[31] N.P. Jouppi, C. Young, N. Patil, D. Patterson, G. Agrawal, R. Bajwa, S. Bates, S. Bhatia, N. Boden, A. Borchers, In-datacenter performance analysis of a tensor processing unit, Proceedings of the 44th annual international symposium on computer architecture, 2017, pp. 1-12.
[32] V. Sovrasov, Flops counter for convolutional networks in PyTorch framework, (2020).
[33] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, Pytorch: An imperative style, high-performance deep learning library, arXiv preprint arXiv:1912.01703 (2019).
[34] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).

[35] S. Li, X. Zhao, G. Zhou, Automatic pixel‐level multiple damage detection of concrete structure using fully convolutional network, Computer‐Aided Civil and Infrastructure Engineering 34(7) (2019) 616-634.

[36] Y. Liu, J. Yao, X. Lu, R. Xie, L. Li, DeepCrack: A deep hierarchical feature learning architecture for crack segmentation, Neurocomputing 338 (2019) 139-153.

[37] W. Liu, S. Liao, W. Ren, W. Hu, Y. Yu, High-level semantic feature detection: A new perspective for pedestrian detection, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 5187-5196.

[38] L.-C. Chen, G. Papandreou, F. Schroff, H. Adam, Rethinking atrous convolution for semantic image segmentation, arXiv preprint arXiv:1706.05587 (2017).

Chapter 4 AN ARTIFICIAL NEURAL NETWORK MODEL ON TENSILE BEHAVIOR OF HYBRID STEEL-PVA FIBER REINFORCED CONCRETE CONTAINING FLY ASH AND SLAG POWDER[1]

4.1 Introduction

Fiber reinforced concrete (FRC) has become a prevalent type of concrete and is widely used in many civil infrastructures [1-3], because of its improvement in mechanical properties [4-6] and durability [7, 8]. Many types of fibers, including steel fibers, polyvinyl alcohol (PVA) fibers, polypropylene (PP) fibers, and basalt fibers, have been used to produce FRC [3, 7, 9, 10]. For a given fiber type, the geometry and the volume content of fibers could significantly influence the mechanical properties of concrete [11-15]. Moreover, the hybridization of different fibers could enhance reinforcement [13]. Therefore, hybrid fiber reinforced concrete (HFRC) was developed [16] and two different types of fibers are usually adopted. For example, steel fiber, a type of macrofibers, and PVA fiber, a type of microfibers, are often used to prepare the HFRC, which has more improvement in the mechanical behavior than the steel or PVA fiber reinforced concrete alone.

Constitutive modeling of HFRC is important for the design of the HFRC structure. Many constitutive models of HFRC have been proposed to present the stress-strain relationship of HFRC [10, 17, 18] by considering many fiber characteristics. But it is hard to develop one model from a purely mathematical perspective because of the complexity of HFRC, such as the strain-softening/hardening, and volumetric dilatancy [17]. Recently, the deep learning method has been successfully applied in modeling concrete to overcome these limitations [19, 20]. As one of the basic deep learning methods, artificial neural network (ANN) has become a popular method to model the mechanical properties of FRC [20], to design the mix compositions of FRC [19], to describe the stress-strain model of FRC [21], to predict the fracture energy of polymer–nanoparticle composites (PNCs) [22, 23], to compute the flexoelectricity effect in truncated pyramid structures [24], to build a deep collocation method for thin plate bending problems [25], to solve boundary values problem [26], and to act as function approximation machines to approximate the solution of partial differential equations [27]. Compared with the conventional constitutive model, the ANN model could consider more factors and provide a better prediction of the mechanical properties of concrete. However, few ANN models were proposed to predict the whole stress-strain curve of HFRC and to investigate the effect of HFRC's contents on its mechanical behavior.

Although the ANN model could implicitly detect complex non-linear relationships between dependent and independent variables, it is usually recognized as 'black boxes' that provide little information about relationships among different variables, and therefore, are difficult to be understood [28]. Sensitivity

[1] Part of this chapter is published as journal papers:

Fangyu Liu, Wenqi Ding, Yafei Qiao, and Linbing Wang. "An artificial neural network model on tensile behavior of hybrid steel-PVA fiber reinforced concrete containing fly ash and slag powder." Frontiers of Structural and Civil Engineering 14.6 (2020): 1299-1315

analysis has been proposed to explain how the uncertainty in the output of the ANN model can be divided and allocated to different sources of uncertainty in its inputs [29, 30]. There are several methods to investigate the sensitivity of the ANN model: the 'PaD' method (Partial Derivatives), the 'Weights' method, the 'Profile' method, and the 'Stepwise' method [30]. Considering the values of the inputs and weights, the 'PaD' method could overcome some disadvantages of other methods and be an easy method to calculate the contribution of each input feature [28].

Therefore, this study aims to develop a new approach to describe the tensile stress-strain curve of HFRC by using the ANN model; and to highlight the ability of the ANN model by investigating the effect of different contents of HFRC on the tensile behavior of HFRC. To make a better prediction and consider experimental conditions, many factors are introduced into the ANN model, including the fiber characteristics (fiber volume, weight, length, diameter, the aspect ratio, and the reinforcement index), mechanical properties of plain concrete (elastic modulus, compressive strength, and strain corresponding to tensile strength), and the composition of concrete mixture (cement, fly ash, slag powder, water, coarse aggregate, fine aggregate, and water binder ratio). Moreover, three methods used to process output features are also assessed and compared.

In the following sections, the framework of the ANN model is firstly introduced. Then experimental results of many direct tensile tests on HFRC carried out by our group [9] and other data from previous literature are used to build the database. After that, the ANN model is trained and evaluated based on the database, meanwhile the equation-based model is also built. The comparison between these two models is made via the description of the experiment data. The results of three processed methods are discussed and the most suitable method for the ANN model is recommended. The sensitivity analysis is finally done to show how input features make contributions to output features of the ANN model.

4.2 Framework of the ANN model

4.2.1 ANN Approach

Typically, ANN has three main layers: the input layer, hidden layer, and output layer. The architecture of the ANN model plays an essential role in building such a model. Different hidden neurons and different hidden layers are investigated, and 5-fold cross-validation [31] is used to evaluate and get a better ANN model. Figure 4-1 shows the procedure 5-fold cross-validation which includes finding better parameters, such as neurons and hidden layers, on the training dataset and making the final evaluation on the test dataset. During the first procedure, the training dataset is split into the inner training dataset (blue block in Figure 4-1) and the validation dataset (green block in Figure 4-1). The maximal epoch is 80 during the cross-validation. The cross-validation results reveal that two hidden layers are most suitable for the ANN model in this work. Figure 4-2 shows the architecture of the used ANN model, including one input layer, two hidden layers, and one output layer.

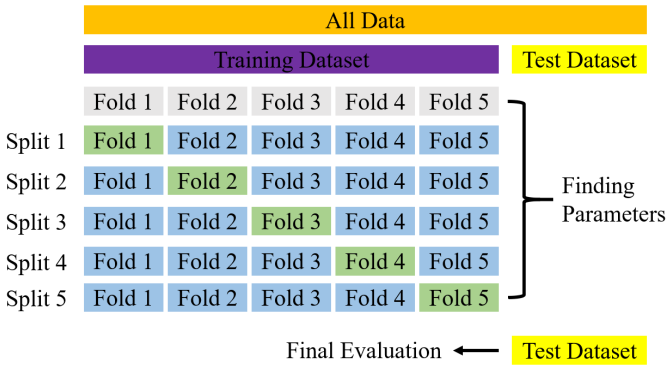

Figure 4-1 5-fold cross-validation

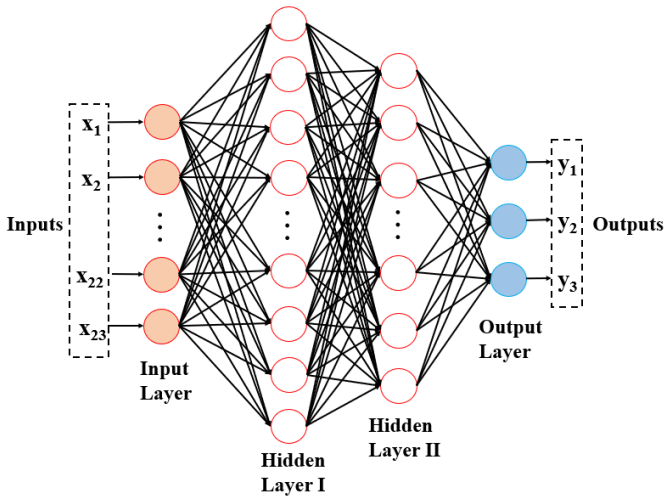

Figure 4-2 Scheme of the ANN approach in this study

The calculation process of the ANN approach is that considering the input vector of one neuron in one layer is $\boldsymbol{x} = [x_1, x_2, \ldots \ldots x_m]$, the neuron in the next layer is computed: (1) $\boldsymbol{x}$ will be multiplied by the weight; (2) the value of the neuron in the next layer z is obtained by the activation function, as shown in Eq.(4-1).

$$z = f\left(\boldsymbol{\omega}^{\boldsymbol{T}}\boldsymbol{x} + b\right) = f\left(\sum_{i=1}^{m} \omega_i x_i + b\right) \tag{4-1}$$

$$\boldsymbol{x} \in R^{1\times m},\ \boldsymbol{\omega} \in R^{1\times m},\ z \in R$$

As increasing the nonlinearity of the neural network, the activation function plays an important role in the neural network. This study uses the popular activation function, the rectified linear unit (ReLU) function [32], which is

$$f(x) = max(0, x) \tag{4-2}$$

4.2.2 Procedures of the ANN model

There are four main procedures of the ANN model to describe the tensile behavior of HFRC, as illustrated in Figure 4-3, including the dataset preprocess, dataset splitting, the ANN model training and evaluating, and prediction.

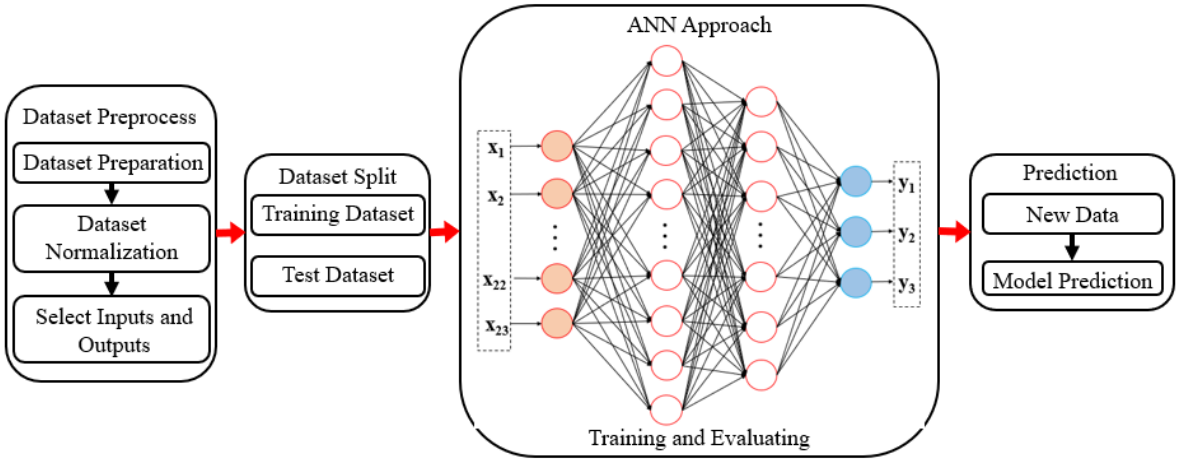

Figure 4-3 Framework of the ANN model in this study

(1) Dataset preprocess

The quality of the dataset has a strong influence on the performance of the ANN model. Therefore, several tensile test results of HFRC which this research group did before are combined with the relevant data collected in the literature [9, 10, 33-35] to improve the quantity and diversity of the dataset. Since the ANN model is used to predict the tensile stress-strain curve, the dataset would include many tensile stress-strain curves of HFRC rather than only the tensile strength and strain corresponding to tensile strength (SCTS). According to the factors influencing the tensile behavior of HFRC, the input data has a total of 23 features in this study, which are summarized in Table 4-1. Correspondingly, the output data is the stress, the tensile strength of HFRC, and the strain corresponding to tensile strength of HFRC, which could be used to obtain the tensile stress-strain curve and evaluate the tensile behavior of HFRC.

Table 4-1 Features of the input data

Parts	Feature
Strain	Strain
Steel fiber	Fiber volume of steel fiber
	Weight of steel fiber
	Length of steel fiber
	Diameter of steel fiber
	Aspect ratio of steel fiber
	Reinforcement index of steel fiber

PVA fiber	Fiber volume of PVA fiber
	Weight of PVA fiber
	Length of PVA fiber
	Diameter of PVA fiber
	Aspect ratio of PVA fiber
	Reinforcement index of PVA fiber
Mechanical properties of plain concrete	Elastic modulus of plain concrete
	Tensile strength of plain concrete
	Strain corresponding to tensile strength of plain concrete
Components of HFRC	Weight of the cement
	Weight of fly ash
	Weight of slag powder
	Weight of water
	Weight of coarse aggregate
	Weight of fine aggregate
	Water binder ratio

The raw dataset needs preprocessing because the amplitude and the range of different input features vary a lot and could have a significant influence on the prediction of the ANN model. For example, the strain of HFRC varies from 0 to 0.01 and it has at least 1000 data points for one HFRC specimen, but the weight of the cement only has two options (225 and 449 kg/m^3) which are far higher than the maximal strain. Without normalization, large differences among input features could make it hard to train and evaluate the ANN model. Therefore, before training and evaluating the ANN model, the input data needs normalization and the Min-Max normalization method is used in this work. The Min-Max normalization could normalize the data to a range from 0 to 1 by using the following equation,

$$x_{j,scaled}^{(i)} = \frac{x_j^{(i)} - x_{min}^{(i)}}{x_{max}^{(i)} - x_{min}^{(i)}} \tag{4-3}$$

where $i = 1, 2, \dots, m$ (m is the number of features), $j = 1, 2, \dots, n$ (n is the number of data samples); $x_{max}^{(i)}$ and $x_{min}^{(i)}$ are the maximal and minimal values of the ith feature, respectively; $x_j^{(i)}$ is the original jth data sample of the ith feature, and $x_{j,scaled}^{(i)}$ is the scaled jth data sample of the ith feature. The Min-Max normalization is completed by the preprocessing module of scikit-learn [36].

There is also a large difference among different output features, like the tensile strength of HFRC (greater than 3 MPa) and the strain corresponding to tensile strength of HFRC (less than 0.03), so it would be better to normalize output features. To illustrate this normalization effect, this study uses three methods to process

output features: no-processed, mid-processed, and processed. No-processed means that output features keep their original value and mid-processed means that only the strain corresponding to tensile strength of HFRC is normalized. Processed means that all three output features are normalized. Because of applying three methods to process output features, triplicate ANN models use these three methods separately and are trained and evaluated in parallel.

(2) Dataset split

The dataset needs to be separated into two subsets, the training dataset and the test dataset, to train and evaluate the ANN model. After shuffling the whole dataset, it is randomly split into the training and test dataset via the model selection module of scikit-learn, 'train_test_split' [36]. The training dataset, 70% of the whole dataset, is used to train the ANN model; and the test dataset, the other 30% of the whole dataset, is used to evaluate the ANN model.

(3) ANN model training and evaluating

The ANN model (Figure 4-2) runs on the popular deep learning library of Pytorch [37] and Skorch [38]. As this is the regression problem, the loss function adopts the mean square error (MSE) function. The used optimizer is the adaptive moment estimation (Adam) [39] and the batch size of the dataset is 128. The maximal epoch is 1000 epochs; during each epoch, the ANN model is firstly trained by the training dataset and then evaluated by the test dataset. Meanwhile, the losses of the training dataset and test dataset are printed and compared to ensure that neither underfitting nor overfitting happens.

The learning rate is a crucial hyperparameter that determines how fast the parameters are updated. The initial learning rate is 0.01; when the decreasing rate of loss value is limited (i.e. 480 epochs in this work), the learning rate would be reduced by 0.1 times. The learning rate decay could improve the convergence of the ANN model.

After finishing training the ANN model, the parameters of the ANN model would be saved to predict the new data and do the sensitivity analysis.

(4) Prediction

The new input dataset is fed into the ANN model after completing training the ANN model. As mentioned above, each data has three output features: stress, tensile strength, and strain corresponding to tensile strength. The predicted stress is combined with the input strain to obtain the tensile stress-strain curve. The final predicted tensile strength and strain corresponding to tensile strength of each HFRC sample are determined by the mean value of these two items of each HFRC sample's data.

4.3 Database of the tensile behavior of HFRC

To build the database, experimental data of a series of direct tensile tests on HFRC containing fly ash and slag powder performed by our group [9], as well as the relevant data in the literature, were used. Table 4-2 summarizes the key information of all collected data. As different cementitious materials seem to have no significant influence on the tensile behavior of HFRC [9], all these data are combined to prepare the database.

Table 4-2 The steel-PVA fiber hybridization of HFRC from previous literature

Cementitious material	Fiber volume (%)		Fiber length (mm)		Fiber Diameter (mm)		Ref.
	Steel fiber	PVA fiber	Steel fiber	PVA fiber	Steel fiber	PVA fiber	
Cement+Fly ash+Slag powder (HFRC)	0.5	0.5	38	12	0.677	0.039	[9]
	0.5	1.0					
	0.5	1.5					
	1.0	0.5					
	1.0	1.0					
	1.0	1.5					
	1.5	0.5					
	1.5	1.0					
	1.5	1.5					
Cement (HFRC)	0.8	0.1	35/50	8/12	0.55/0.75	0.04	[10]
	0.8	0.2					
	1.3	0.1					
	1.3	0.2					
Cement (HFRC)	0.05	0.15	38	12	1.10	0.04	[33]
	0.1	0.3					
	0.3	0.6					
	0.03	0.17					
	0.06	0.34					
	0.11	0.69					
	0.02	0.18					
	0.04	0.36					
	0.08	0.72					
Cement (HFRC)	0.25	0.07	12/15	30/60	0.5/0.73	0.015	[34]
	0.25	0.14					
	0.51	0.07					
	0.51	0.14					

Cement +GGBSa+Silica fume (HFRC)	0.25	0.25	20	6	0.4	0.12	[35]
	0.5	0.25					
	0.75	0.25					
	1	0.25					
	1.25	0.25					
	0.25	0.25					

[a]GGBS: Ground-granulated blast-furnace slag

To describe the tensile stress-strain curves, the whole dataset of these curves is necessary. This is the main difference of the dataset compared to other ANN models that only focus on the tensile strength or the strain corresponding to tensile strength. The direct tensile test can be used to obtain these stress-strain curves, while the dog bone-shaped specimen is suggested to be used [9]. Figure 4-4 shows the typical experimental tensile stress-strain curves of HFRC. The black line is the average tensile stress-strain curve of each type of HFRC and the gray area is drawn by all tensile stress-strain curves of each type of HFRC. The tensile strength and strain corresponding to tensile strength are also illustrated in Figure 4-4. Model details about the tests and results can be found in [9].

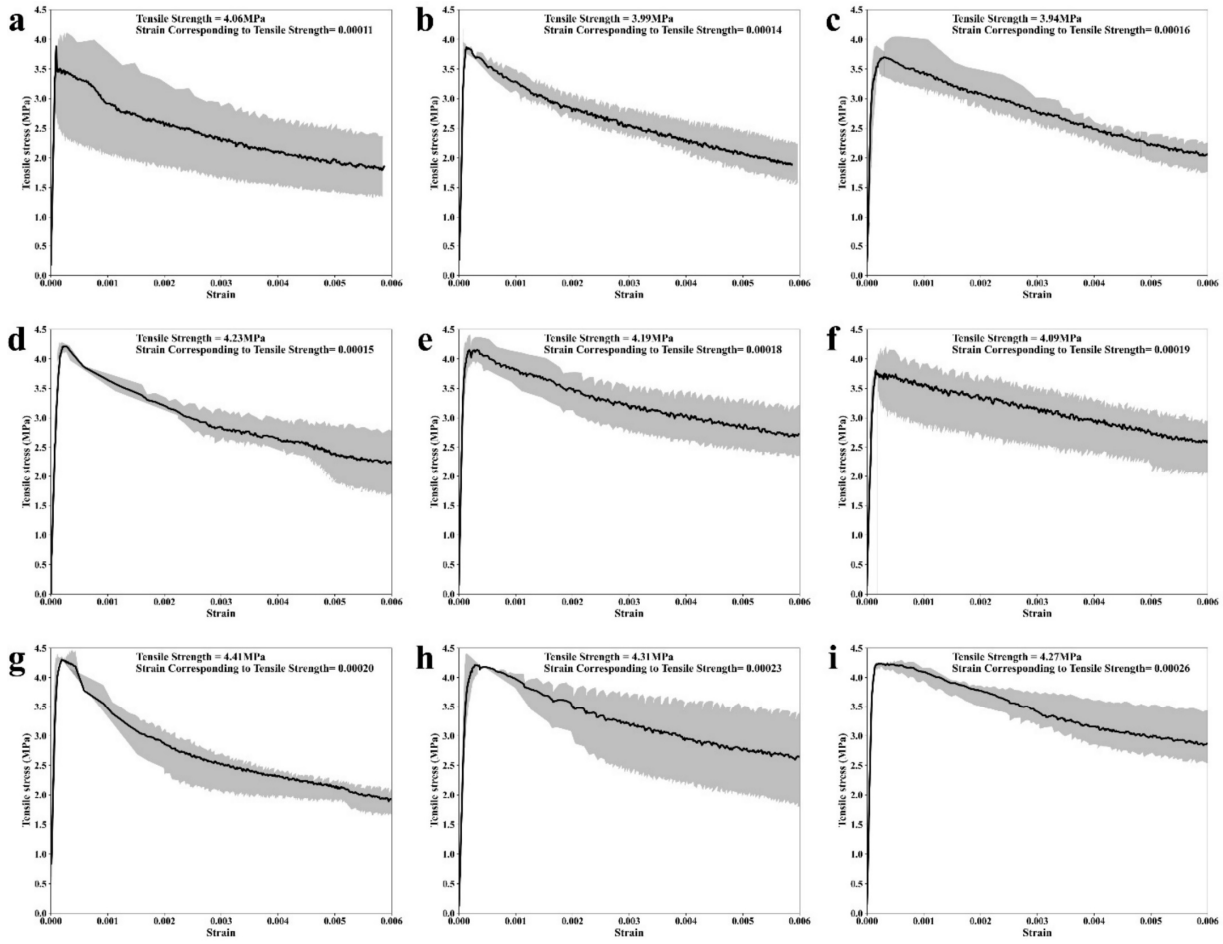

Figure 4-4 Tensile stress-strain curves of (a) $S_{0.5}P_{0.5}$, (b) $S_{0.5}P_{1.0}$, (c) $S_{0.5}P_{1.5}$, (d) $S_{1.0}P_{0.5}$, (e) $S_{1.0}P_{1.0}$, (f) $S_{1.0}P_{1.5}$, (g) $S_{1.5}P_{0.5}$, (h) $S_{1.5}P_{1.0}$, (i) $S_{1.5}P_{1.5}$

With all these data, the main characters of each feature in the output and input can be identified. Table 4-3 summarizes the mean value, the medium value, and the standard deviation of each feature in the database. The size of the whole dataset is 9081. Among all output features, the tensile strength has the biggest mean value and median value, while the tensile stress has the biggest standard deviation. All these values of the strain corresponding to tensile strength are relatively small, which may affect the straining process and will be discussed in Section 4.2.

Table 4-3 Summary of the dataset for the ANN model (Size=9801)

Type	Feature	Mean	Median	Standard deviation
Output	Tensile stress	2.40	2.47	0.97
	Tensile strength	4.10	4.02	0.20
	Strain corresponding to tensile strength	0.000253	0.000168	0.000172
Input	Strain	0.002542	0.001309	0.003017

Fiber volume of steel fiber	1.04	1.00	0.38
Weight of steel fiber	81.84	78.50	29.79
Length of steel fiber	39.53	38.00	4.54
Diameter of steel fiber	0.68	0.68	0.05
Aspect ratio of steel fiber	58.51	56.13	4.13
Reinforcement index of steel fiber	0.61	0.56	0.22
Fiber volume of PVA fiber	0.81	1.00	0.52
Weight of PVA fiber	10.56	13.00	6.81
Length of PVA fiber	11.59	12.00	1.21
Diameter of PVA fiber	0.04	0.04	0.00
Aspect ratio of PVA fiber	295.53	307.69	32.32
Reinforcement index of PVA fiber	2.48	3.08	1.64
Elastic modulus of plain concrete	35915	35369	1135
Tensile strength of plain concrete	3.28	3.41	0.21
Strain corresponding to tensile strength of plain concrete	0.000087	0.000088	0.000019
Weight of the cement	282	225	98
Weight of fly ash	56	75	33
Weight of slag powder	56	75	33
Weight of water	174	165	16
Weight of coarse aggregate	1019	1024	8
Weight of fine aggregate	756	785	50
Water binder ratio	0.443	0.440	0.004

4.4 Performance of the ANN model

4.4.1 Indices for evaluating model

Several indices have been introduced to combine with the loss to evaluate the performance of the ANN model, including the coefficient of determination (R^2) and mean absolute error (MAE) [21]. R^2 is defined in Eq.(4-4); the closer its value is to 1, the better the prediction of the ANN model fits with the true data. MAE is determined in Eq.(4-5) and it is similar to the loss to evaluate the ANN model; the smaller the MAE is, the better the performance of the ANN model is.

$$R^2(y,\hat{y}) = 1 - \frac{\sum_{k=1}^{n}(y_k - \hat{y}_k)^2}{\sum_{k=1}^{n}(y_k - \bar{y})^2}, \quad \bar{y} = \frac{1}{n}\sum_{k=1}^{n} y_k \tag{4-4}$$

$$MAE(y,\hat{y}) = \frac{1}{n}\sum_{k=1}^{n}|y_k - \hat{y}_k| \tag{4-5}$$

where $\hat{y}$ is the predicted value, y is the true value, $\hat{y}_k$ is the predicted value of the kth sample, and y_k is the true value of the kth sample.

4.4.2 ANN model training and evaluating

The ANN model is trained and evaluated using the database presented in Section 3. As the cross-validation is employed to find the better ANN model, Figure 4-5a and b show the R^2 of the training dataset and validation dataset for the 2-hidden layer and 3-hidden-layer ANN models, while the sigmoid function is also acted as the activation function to make a comparison. Meanwhile, some typical machine learning methods, such as support vector machine (SVM) [40], random forests (an ensemble learning method) [41], Gaussian Processes [42], are also used to make a comparison with the ANN model. Figure 4-5c shows the R^2 of the test dataset for these seven models. Although the 3-hidden-layer ANN model with ReLU has the highest R^2 of the validation dataset, its R^2 of the test dataset is a little lower than that of the 2-hidden-layer ANN model with ReLU. Both the 2-hidden-layer and 3-hidden layer ANN models with ReLU are higher than that with Sigmoid in terms of the R^2 of the validation dataset and test dataset, as shown in Figure 4-5. Therefore, ReLU is adopted as the activation function in the following analysis. Compared with the ANN model, these three typical machine learning methods have a lower R^2 of the test dataset.

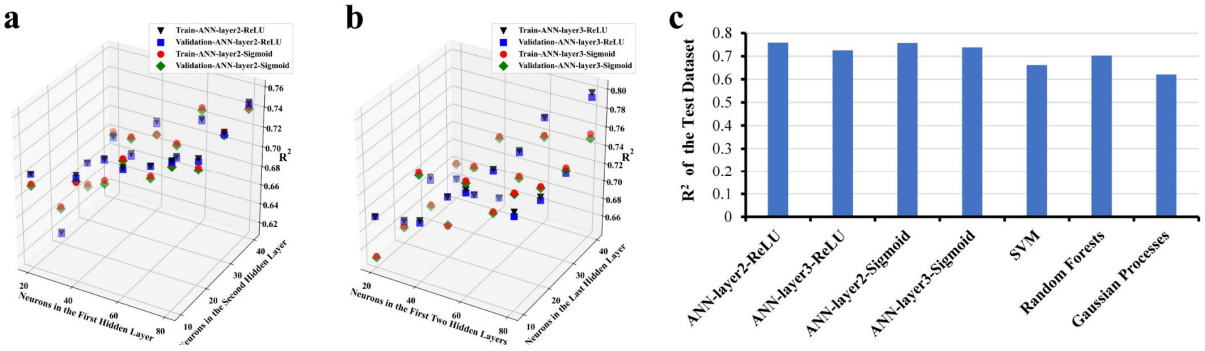

Figure 4-5 R^2 of the training and validation dataset for (a) the 2-hidden-layer ANN model with ReLU and Sigmoid respectively, (b) the 3-hidden-layer ANN model with ReLU and Sigmoid respectively, (c) R^2 of the test dataset for the 2-hidden-layer ANN model with ReLU and Sigmoid respectively, the 3-hidden-layer ANN model with ReLU and Sigmoid respectively, SVM, Random Forests, Gaussian Processes.

According to Figure 4-5, the ANN model with 80 neurons in the hidden layer I and 10 neurons in the hidden layer II is used. Then, training and evaluating this ANN model is followed by the procedure in Section 2.2. Figure 4-6 shows the loss curves, R^2 curves, and MAE curves of the training dataset and test dataset. The loss values of the training dataset and test dataset (Figure 4-6a) drop nonlinearly and rapidly before 480 epochs and then they tend to remain stable and converge. This is because the learning rate changes from

0.01 to 0.001 at the 480 epochs. The lowest flat part could be found in all loss curves after 960 epochs because the learning rate changes from 0.001 to 0.0001. The learning rate determines how fast the model converges. The big learning rate could make the loss decrease quickly but not smoothly, which is usually used in early training. As the learning rate becomes smaller, the loss tends to vary slowly, which could help the model more easily converge in the late period.

Three obvious patterns of loss curves could be found in Figure 4-6a. This is induced by the three different processed methods of output features presented in Section 2. The loss of the no-processed method is the highest among the three processed methods no matter whether the loss belongs to the training dataset or the test dataset. As the degree of processing improves, the loss of the training dataset and test dataset decreases. This is because the Mix-Max normalization could limit the data in a small area (like 0 to 1) and enable the data to distribute evenly. For example, 3.113 MPa and 4.022 MPa are the test data and the difference between these two values is 0.909 MPa. After the Min-Max normalization, the corresponding data are 0.697 and 0.907, and the difference decreases to 0.21. The decrease in the range and the difference could enable the gap between the prediction and test data to become smaller, leading to a decrease in the loss (Figure 4-6a).

The R^2 and MAE of the training dataset and test dataset are shown in Figure 4-6b and c. There are two patterns of MAE curves for three processed methods; the normalized output features have the lowest MAE of the training dataset and test dataset. In contrast, there is no obvious difference between the R^2 of these three processed methods. All R^2 of the training dataset and test dataset increase dramatically to 0.9 before 49 epochs, then gradually rise and finally reach a value higher than 0.99. All these indices reveal the good reliability of the ANN model.

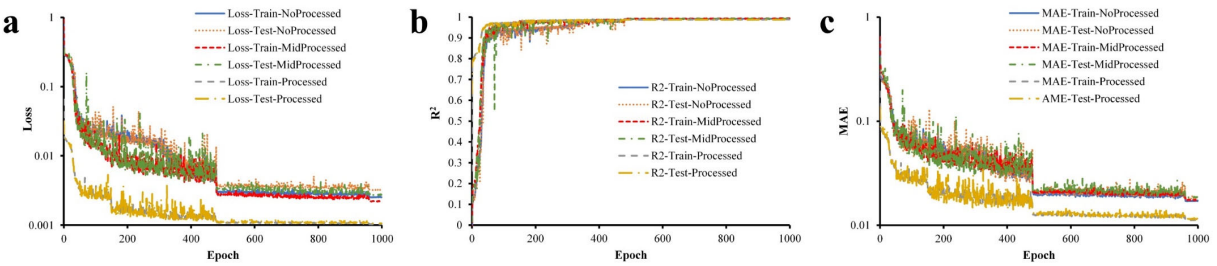

Figure 4-6 Training and evaluating results of the ANN model: (a) Loss, (b) R^2, and (c) MAE.

4.4.3 Equation-based model

An equation-based model is also proposed based on the constitutive model recommended in GB 50010-2010 [43] to compare with the ANN model. The tensile stress-strain curve of HFRC can be described using the following equation,

$$\sigma = (1 - d_t)E_c\varepsilon \tag{4-6}$$

$$d_t = \begin{cases} 1 - \rho_t[1.2 - 0.2x^5] & x \le 1 \\ 1 - \dfrac{\rho_t}{\alpha_t(x-1)^{1.7} + x} & x > 1 \end{cases} \qquad \rho_c = \frac{f_{t,H}}{E_c \varepsilon_{t,H}}, \quad x = \frac{\varepsilon}{\varepsilon_{t,H}} \tag{4-7}$$

where σ is the tensile stress (MPa), ε is the strain; E_c is the elastic modulus (MPa); $f_{t,H}$ is the tensile strength of HFRC (MPa); $\varepsilon_{t,H}$ is the strain corresponding to the tensile strength of HFRC; d_t is the damage evolution parameter and is mainly dependent on the degree of plasticity x that is defined as the ratio of ε to $\varepsilon_{t,H}$; α_t is the descending shape parameter and determines the slope of the descending branch during the post-peak stage. According to the previous literature [10], E_c of HFRC is assumed identical to the corresponding plain concrete in this study.

For the equation-based model of HFRC, $f_{t,H}$ $\varepsilon_{t,H}$, and α_t are the three main parameters and can be determined by back-fitting the experimental data. $f_{t,H}$ and $\varepsilon_{t,H}$ can be easily obtained from the peak point of each tensile stress-strain curve. α_t needs to be determined by fitting each tensile stress-strain curve by using Eq.(4-6)-(4-7). $f_{t,H}$, $\varepsilon_{t,H}$, and α_t of HFRC are related to steel fibers and PVA fibers [9, 10]. Therefore, the back-fitting of these three parameters needs to consider the influences of steel fibers and PVA fibers on the tensile behavior of HFRC. In this study, the polynomial function is proposed, as a first approximation, to back fit the relationship between $f_{t,H}$ $\varepsilon_{t,H}$, α_t of HFRC and the characteristics of steel fibers and PVA fibers. The fitting results are indicated in Eq.(4-8)-(4-10).

$$f_{t,H} = f_{t,p}(1 + 0.886\lambda_S + 0.044\lambda_P - 0.015\lambda_S\lambda_P - 0.589\lambda_S^2 - 0.008\lambda_P^2) \tag{4-8}$$

$$\varepsilon_{t,H} = \varepsilon_{t,p}(1 - 0.296\lambda_S + 0.400\lambda_P + 0.999\lambda_S\lambda_P - 0.060\lambda_S^2 - 0.032\lambda_P^2) \tag{4-9}$$

$$\alpha_t = \frac{1}{9.05 - 19.99\lambda_S + 5.56\lambda_P - 3.82\lambda_S\lambda_P - 33.51\lambda_S^2 - 0.62\lambda_P^2} \qquad \lambda_S = \frac{l_S}{d_S}V_S, \ \lambda_P = \frac{l_P}{d_P}V_P \tag{4-10}$$

Where $f_{t,p}$ is the tensile strength of plain concrete (MPa); $\varepsilon_{t,p}$ is the SCTS of plain concrete; λ_S is the reinforcement index of steel fibers; and λ_P is the reinforcement index of PVA fibers. l_S, d_S, and V_S are the fiber length (mm), fiber diameter (mm), and fiber volume content of steel fibers, respectively; l_P, d_P, and V_P are the fiber length (mm), fiber diameter (mm), and fiber volume content of PVA fibers, respectively.

Steel fibers and PVA fibers have a hybrid effect on the tensile behavior of HFRC [9, 10, 12, 13], and the term $\lambda_S\lambda_P$ is introduced into Eq.(4-8)-(4-10) to consider this hybridization of steel fibers and PVA fibers. Because $\lambda_S\lambda_P$ is the quadratic term, λ_S^2 and λ_P^2 are also listed in Eq.(4-8)-(4-10) to keep the balance. The back-fitting results of $f_{t,H}$, $\varepsilon_{t,H}$, and α_t are presented in Figure 4-7. Eq.(4-8)-(4-10) can reproduce the relationship between $f_{t,H}$ $\varepsilon_{t,H}$, α_t of HFRC and the characteristics of steel fibers and PVA fibers well although the value of R^2 is not very high.

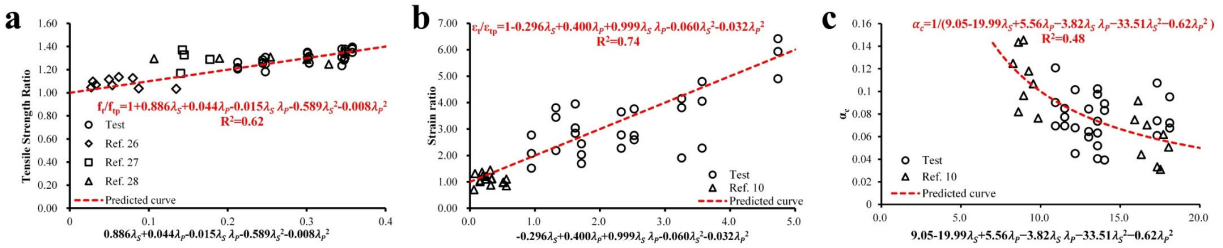

Figure 4-7 Back-fitting results of key parameters in the equation-based model: (a) tensile strength, $f_{t,H}$; (b) SCTS, $\varepsilon_{t,H}$ and (c) descending shape parameter, α_t.

To obtain the equation-based model of HFRC, $f_{t,p}$, $\varepsilon_{t,p}$, and E_c needs to firstly get from the corresponding plain concrete and be acted as the reference parameters in Eq.(4-8)-(4-10). Secondly $f_{t,H}$, $\varepsilon_{t,H}$, and α_t could be obtained by Eq.(4-8)-(4-10) with the fiber characteristic parameter. Then all these three parameters are fed into Eq.(4-6)-(4-7) to plot the tensile stress-strain curve of HFRC.

4.4.4 Model performance analysis

After the training of the ANN model, all input datasets are fed into the ANN model to predict the tensile stress-strain curve, tensile strength, and strain corresponding to tensile strength of HFRC. As mentioned above, triplicate ANN models with different methods to process output features are compared in terms of predicting the tensile behavior of HFRC. The equation-based model is also used to predict the tensile stress-strain curve, tensile strength, and the corresponding strain. Then, the R^2 and MAE of these curves are calculated for the three different ANN models (the same architecture but different data processed methods) and the equation-based model (see Table 4-4), while these two indices of the three methods are also included. Figure 4-8 shows the predicted results of the ANN model and the equation-based model as well as the experimental results of HFRC. Compared with the equation-based model, the ANN model has a better ability to reproduce the experimental results. For the ANN model, the predicted curves of the no-processed method and mid-processed method could fit better and even overlap with the experimental data, while there is a little bit gap between the predicted curves of the processed method and the test data. This is because, among the three processed methods, the no-processed method and the mid-processed method do not process the stress data and keep their original version, while the processed method normalizes the stress data. As mentioned above, the Mix-Max normalization limits the data in the range from 0 to 1 and the difference between the prediction and the test data would be amplified after inversing the Mix-Max normalization as the range expands from 0 to 4.5. For example, assuming that the prediction is 0.821 and the test data is 0.902 in the processed method of the ANN model, the gap between the prediction and the test data is 0.081 and its value is small enough to be acceptable. But after inversing the Mix-Max normalization, the prediction becomes 3.648 and the test data becomes 4 (the peak value of the ANN-Processed curve in Figure 4-8a), and the gap increases to 0.352. This explains the phenomenon that the gap between the test data and the prediction of the processed method is larger than that of the no-processed method and the mid-processed method. The MAE of the processed method is larger and the R^2 of the processed method is smaller than that of the no-processed method and the mid-processed method, which can be seen in Table 4-4. It can also

approve that normalizing the stress could enlarge the gap between the prediction and the truth.

Table 4-4 MAE, R^2 of the ANN model and equation-based model for predicting the tensile stress-strain curve of HFRC

Mix ID	MAE				R^2			
	ANN			Equation	ANN			Equation
	No-Pro.	Mid-Pro.	Pro.		No-Pro.	Mid-Pro.	Pro.	
$S_{0.5}P_{0.5}$	0.047	0.046	0.075	0.119	0.992	0.994	0.986	0.973
$S_{0.5}P_{1.0}$	0.045	0.050	0.094	0.199	0.996	0.995	0.977	0.929
$S_{0.5}P_{1.5}$	0.041	0.045	0.068	0.482	0.996	0.995	0.987	0.561
$S_{1.0}P_{0.5}$	0.077	0.076	0.109	0.234	0.982	0.984	0.968	0.918
$S_{1.0}P_{1.0}$	0.048	0.065	0.092	0.485	0.996	0.992	0.983	0.569
$S_{1.0}P_{1.5}$	0.068	0.059	0.143	0.692	0.989	0.992	0.955	0.350
$S_{1.5}P_{0.5}$	0.124	0.123	0.239	0.688	0.953	0.964	0.882	0.354
$S_{1.5}P_{1.0}$	0.048	0.049	0.132	0.571	0.997	0.997	0.977	0.622
$S_{1.5}P_{1.5}$	0.052	0.032	0.080	1.139	0.997	0.998	0.990	0.139

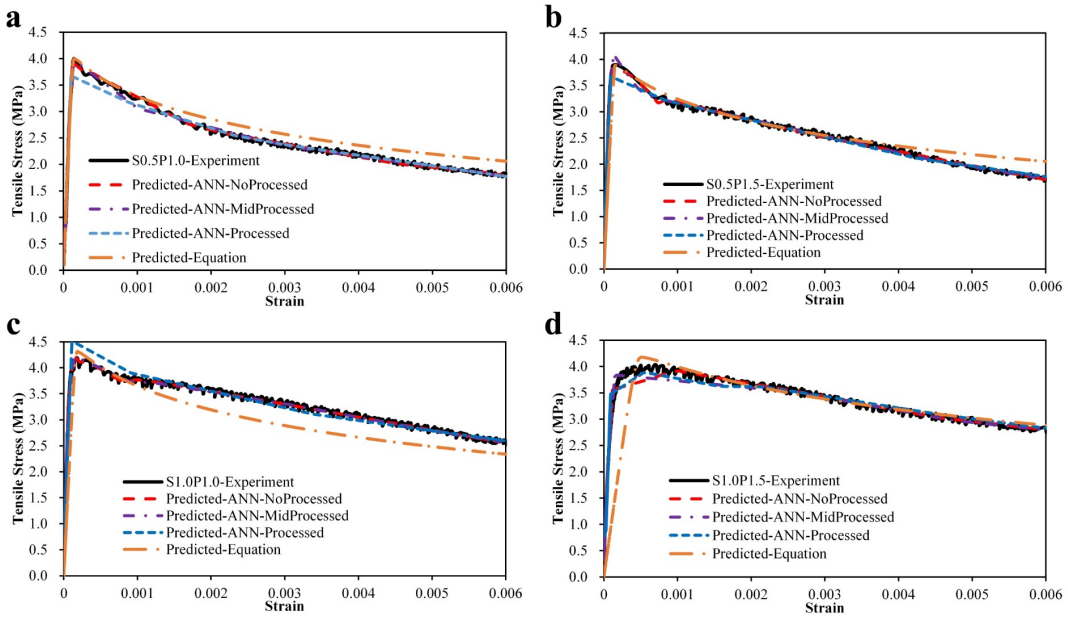

Figure 4-8 Comparison of prediction results and experimental results of the tensile stress-strain curves: (a) $S_{0.5}P_{1.0}$, (b) $S_{0.5}P_{1.5}$, (c) $S_{1.0}P_{1.0}$, and (d) $S_{1.0}P_{1.5}$.

To further evaluate the performance of the ANN model, the other two output features are also evaluated by comparing the equation-based model and the ANN model. Instead of listing the specific predictions, this study introduces the relative error [10] to evaluate the prediction of the tensile strength and SCTS, which can be seen in Figure 4-9. Compared with the equation-based model, the ANN model shows a better

prediction ability. The relative error of the ANN model is lower than that of the equation-based model in terms of the tensile strength and SCTS. Besides, Figure 4-9 also presents the relative error of three processed methods. For the prediction of the tensile strength, all three processed methods have a very small relative error, but there is an obvious gap between these three processed methods in terms of the prediction of SCTS. The maximal relative error of the no-processed could be up to 23.8%, while the other two processed methods have a maximal value of less than 2%. This is because, in the no-processed method, the SCTS is not normalized in the data preprocessing procedure, and both the other two processed methods normalize the SCTS. The SCTS varies in a small range (from 0.00007 to 0.00068), but the minimal loss of the no-processed method is 0.00251 (Figure 4-6a) and the minimal MAE of the no-processed method is 0.01698 (Figure 4-6c). Both these two indices are much higher than that of the SCTS. Without normalization, the small range of the SCTS is difficult for the ANN model to predict when combining the large range of the tensile strength. The Mix-Max normalization would amplify the SCTS to a range from 0 to 1 and it could enable the ANN model to more easily predict the SCTS.

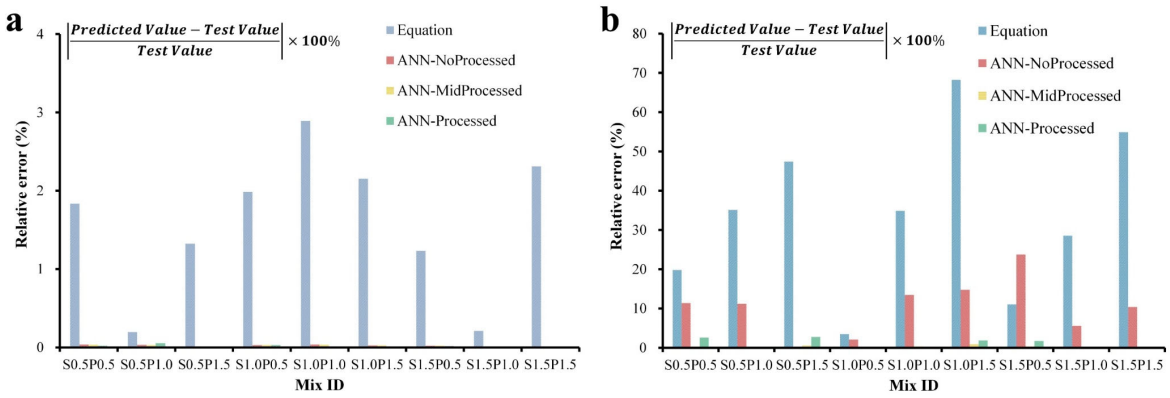

Figure 4-9 The relative error of the ANN model and equation-based model in terms of (a) the tensile strength and (b) SCTS.

4.5 Sensitivity Analysis and Discussion

4.5.1 Sensitivity Analysis

The 'PaD' method focuses on calculating the derivative of the output concerning the inputs of the ANN model [28], which can be used to highlight the contribution of the input features. These partial derivatives are called sensitivity and are defined as Eq.(4-11). As the ANN model in this study has one input layer, two hidden layers, and one output layer, it needs the chain rule to get the final version of the sensitivity, which can be seen in Eq.(4-13). Once the sensitivity has been obtained for each input feature and data sample, two main indices could be calculated to analyze the results: mean sensitivity ($S_{mi,avg}$), and sum squared sensitivity (SSS_{mi}). SSS_{mi}, defined in Eq.(4-14), is the sum squared sensitivity of the output of the m^{th} neuron in the output layer concerning the i^{th} input feature. Based on the sum squared sensitivity, the contribution of each input feature to each output feature could be defined as Eq.(4-15).

$$s_{mi}|_{X_h} = \frac{\partial o_m}{\partial x_i}(X_h) \tag{4-11}$$

$$o_m = \sum_k \omega_{mk} o_k,\ \ o_k = f_2\left(\sum_j \omega_{kj} o_j\right),\ \ o_j = f_1\left(\sum_i \omega_{ji} x_i\right) \tag{4-12}$$

$$s_{mi}|_{X_h} = \frac{\partial o_m}{\partial o_k}\frac{\partial o_k}{\partial o_j}\frac{\partial o_j}{\partial x_i}(X_h) \tag{4-13}$$

$$SSS_{mi} = \sum_{h=1}^{N} (s_{mi}|_{X_h})^2 \tag{4-14}$$

$$Contribution\ of\ i^{th}\ input\ variable = \frac{SSS_{mi}}{\sum_i SSS_{mi}} \tag{4-15}$$

where N is the number of samples in the dataset; X_h is the h[th] sample of the dataset; $s_{mi}|_{X_h}$ is the sensitivity of the output of the mth neuron in the output layer concerning the input of the ith neuron in the input layer evaluated in X_h; x_i is the input of the ith neuron in the input layer; o_m is the output of the neuron m in the output layer; o_k is the output of the neuron k in the hidden layer II; o_j is the output of the neuron j in the hidden layer I; f_2 and f_1 are the activation functions in the hidden layer II and I; ω_{mk} , ω_{kj} , and ω_{ji} are the weight between the neuron m and the neuron k, the weight between the neuron k and the neuron j, the weight between the neuron j and the neuron i.

Figure 4-10 shows the contribution of each input feature to the tensile strength and strain corresponding to tensile strength for each HFRC sample. The color represents the contribution and the darker means the higher contribution. Besides, four main parts are proposed to combine similar input features to investigate the tensile behavior: steel fiber, PVA fiber, mechanical properties of plain concrete, and components of HFRC, which can be seen in Table 4-1. Figure 4-11 shows the contributions of these four parts to the tensile strength and strain corresponding to tensile strength for each HFRC sample and the contributions of these four parts are calculated by summarizing their input features (Table 4-1). According to Figure 4-10 and Figure 4-11, the mechanical properties of plain concrete make the main contribution to the tensile strength and strain corresponding to tensile strength among almost all HFRC samples. The contribution of steel fibers is higher than that of PVA fibers in terms of the tensile strength and strain corresponding to tensile strength. When the fiber volume of steel fibers is below 1.0%, the contributions of steel fibers to the tensile strength and strain corresponding to tensile strength remain stable and then increase as the fiber volume of steel fibers increases to 1.5%. In comparison, the contributions of PVA fibers to the tensile strength and strain corresponding to tensile strength increase gradually when adding more PVA fibers. Besides, the contributions of components of HFRC to the tensile strength and strain corresponding to tensile strength of HFRC remain stable.

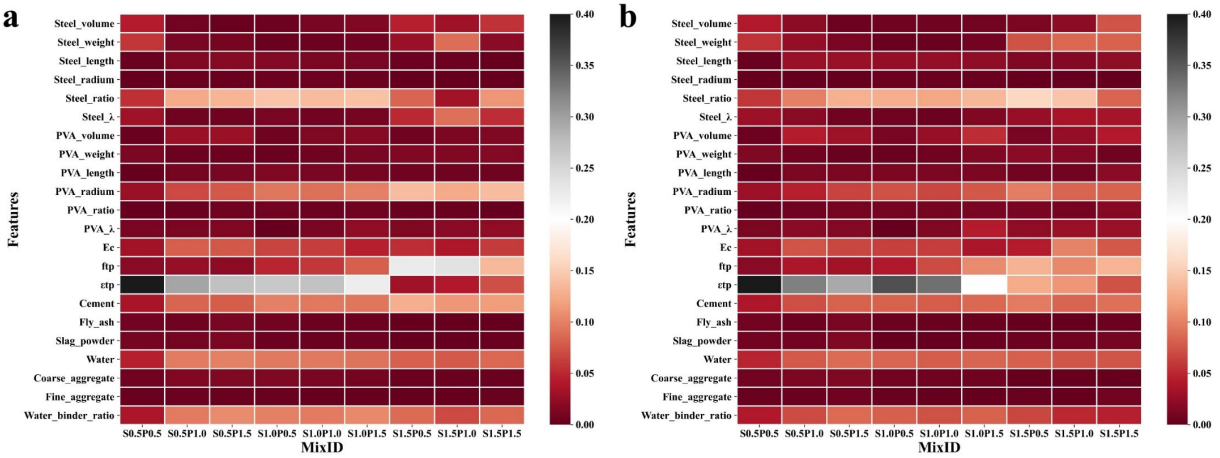

Figure 4-10 Contributions of each input feature to the tensile strength and strain corresponding to tensile strength of HFRC: (a) tensile strength; (b) SCTS

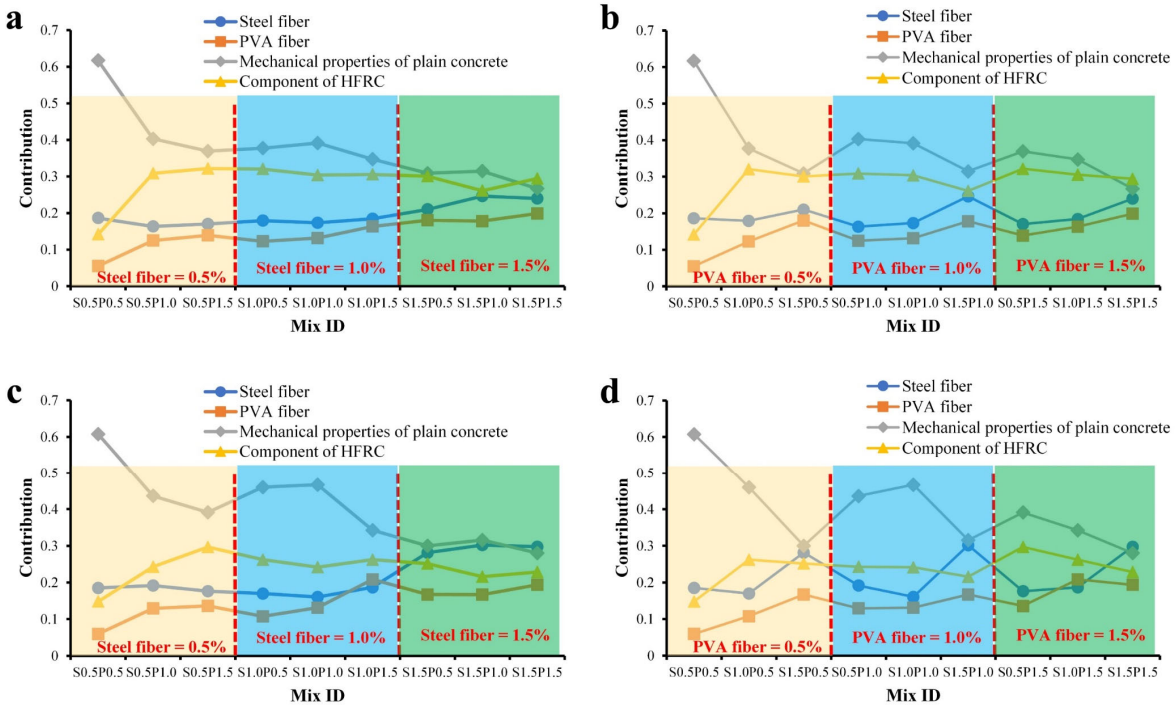

Figure 4-11 Contribution of steel fibers, PVA fibers, mechanical properties of plain concrete, and components of HFRC to the tensile strength and strain corresponding to tensile strength of HFRC: (a) tensile strength as the fiber volume of steel fibers is constant; (b) tensile strength as the fiber volume of PVA fibers is constant;(c) SCTS as the fiber volume of steel fibers is constant; (d) SCTS as the fiber volume of PVA fibers is constant

4.5.2 Discussion

As shown in Figure 4-8, Figure 4-9, and Table 4-4, the ANN model shows a better ability to predict the tensile stress-strain curve, tensile strength, and SCTS of HFRC than the equation-based model. This is

because, compared with the equation-based model, the ANN model could consider more factors (such as components of HFRC), adaptively extract more crucial input features, and present more powerfully nonlinear-fitting capacities. As the performance of the equation-based model are strongly dependent on three key parameters, $f_{t,H}$, $\varepsilon_{t,H}$, and α_t, the fitting results depend largely on the manual effect, namely, the selection of functions. In comparison, the ANN model could avoid this manual effect to a certain extent, which enables the ANN model to better describe the tensile behavior of HFRC. However, the ANN model needs a large and diverse database to improve its performance. Especially, many tensile stress-strain curves of HFRC are necessary because the ANN model is used to predict the tensile stress-strain curve of the HFRC.

Three processed methods are introduced in this study to process the output features: no-processed, mid-processed, and processed. As mentioned above, the processed method has a worse prediction of the tensile stress than the other two methods and the no-processed method shows the worst prediction among the three methods in terms of the SCTS, while all three methods present a similar prediction of the tensile strength. Because the Mix-Max normalization method limits the data in the range from 0 to 1, it seems to have various influences on the different ranges of output features. The Mix-Max normalization is suitable for the output feature which varies in a very small range of less than 1, such as SCTS. In comparison, the output feature changing in a large range of much greater than 1, such as the stress, tends to be worsen predicted by using the Mix-Max normalization.

According to the ANN model, steel fibers tend to make a higher contribution to the tensile strength and strain corresponding to tensile strength than PVA fibers, which is similar to the experimental results that steel fibers play a more important role in the tensile behavior of HFRC [9]. Although their values are lower than that of steel fiber, the contributions of PVA fibers to the tensile strength and strain corresponding to tensile strength keep increasing whatever the fiber volume of steel fiber is.

4.6 Conclusions

This study builds an ANN model to describe the tensile behavior of HFRC containing fly ash and slag powder. The results of many direct tensile tests are combined with the literature data to build the database. Meanwhile, an equation-based model is also established and compared with the ANN model. Three methods to process output features are discussed and the sensitivity analysis is made to better understand the ANN model. The following conclusions can be drawn:

- The ANN model has a better capacity for reproducing the tensile behavior of hybrid steel-PVA fiber concrete containing fly ash and slag powder than the equation-based model. The ANN model shows more powerfully nonlinear-fitting abilities and could consider more factors (a total of 23 factors), including fibers' characteristics, mechanical properties of plain concrete, and concrete composition.

- The normalization has different influences on different types of indices of the HFRC's tensile behavior.

The strain corresponding to tensile strength should be normalized because it is usually very small and varies in a small range (less than one). On the contrary, the tensile stress should not be normalized as its range is often much greater than 1.

- The ANN model can interpret the influence of different features on the tensile behavior of HFRC. Among all input features, the mechanical properties of plain concrete make the main contribution to the tensile strength and strain corresponding to tensile strength; steel fiber tends to make a higher contribution to the tensile strength and strain corresponding to tensile strength than PVA fiber. These outcomes are similar to the experimental results.

4.7 Reprint and Permision

Reprinted by permission from [Springer Nature Customer Service Centre GmbH]: [Springer Nature] [Frontiers of Structural and Civil Engineering] ["An artificial neural network model on tensile behavior of hybrid steel-PVA fiber reinforced concrete containing fly ash and slag power.", Fangyu Liu, Wenqi Ding, Yafei Qiao, and Linbing Wang, [COPYRIGHT] (2020)

References

[1] C. Gong, W. Ding, K.M. Mosalam, S. Günay, K. Soga, Comparison of the structural behavior of reinforced concrete and steel fiber reinforced concrete tunnel segmental joints, Tunnelling and Underground Space Technology 68 (2017) 38-57.

[2] F. Stoll, J.E. Saliba, L.E. Casper, Experimental study of CFRP-prestressed high-strength concrete bridge beams, Composite structures 49(2) (2000) 191-200.

[3] J. Sim, C. Park, Characteristics of basalt fiber as a strengthening material for concrete structures, Composites Part B: Engineering 36(6-7) (2005) 504-512.

[4] L. Xu, H. Xu, Y. Chi, Y. Zhang, Experimental study on tensile strength of steel-polypropylene hybrid fiber reinforced concrete, Advanced Science Letters 4(3) (2011) 911-916.

[5] D.L. Nguyen, D.J. Kim, G.S. Ryu, K.T. Koh, Size effect on flexural behavior of ultra-high-performance hybrid fiber-reinforced concrete, Composites Part B: Engineering 45(1) (2013) 1104-1116.

[6] Z. Wu, C. Shi, W. He, D. Wang, Static and dynamic compressive properties of ultra-high performance concrete (UHPC) with hybrid steel fiber reinforcements, Cement and concrete composites 79 (2017) 148-157.

[7] F. Liu, W. Ding, Y. Qiao, An experimental investigation on the integral waterproofing capacity of polypropylene fiber concrete with fly ash and slag powder, Construction and Building Materials 212 (2019) 675-686.

[8] R. Mu, C. Miao, X. Luo, W. Sun, Interaction between loading, freeze–thaw cycles, and chloride salt attack of concrete with and without steel fiber reinforcement, Cement and Concrete Research 32(7) (2002) 1061-1066.

[9] F. Liu, W. Ding, Y. Qiao, Experimental investigation on the tensile behavior of hybrid steel-PVA fiber

reinforced concrete containing fly ash and slag powder, Construction and Building Materials 241 (2020) 118000.

[10] Y. Zhou, Y. Xiao, A. Gu, G. Zhong, S. Feng, Orthogonal experimental investigation of steel-PVA fiber-reinforced concrete and its uniaxial constitutive model, Construction and Building Materials 197 (2019) 615-625.

[11] F. Liu, W. Ding, Y. Qiao, Experimental investigation on the flexural behavior of hybrid steel-PVA fiber reinforced concrete containing fly ash and slag powder, Construction and Building Materials 228 (2019) 116706.

[12] J. Lawler, T. Wilhelm, D. Zampini, S.P. Shah, Fracture processes of hybrid fiber-reinforced mortar, Materials and Structures 36(3) (2003) 197-208.

[13] J.S. Lawler, D. Zampini, S.P. Shah, Microfiber and macrofiber hybrid fiber-reinforced concrete, Journal of Materials in Civil Engineering 17(5) (2005) 595-604.

[14] D.L. Nguyen, G.S. Ryu, K.T. Koh, D.J. Kim, Size and geometry dependent tensile behavior of ultra-high-performance fiber-reinforced concrete, Composites Part B: Engineering 58 (2014) 279-292.

[15] P. Pujadas, A. Blanco, S. Cavalaro, A. Aguado, Plastic fibres as the only reinforcement for flat suspended slabs: Experimental investigation and numerical simulation, Construction and Building Materials 57 (2014) 92-104.

[16] M. Hsie, C. Tu, P. Song, Mechanical properties of polypropylene hybrid fiber-reinforced concrete, Materials Science and Engineering: A 494(1-2) (2008) 153-157.

[17] Y. Chi, L. Xu, H.-s. Yu, Constitutive modeling of steel-polypropylene hybrid fiber reinforced concrete using a non-associated plasticity and its numerical implementation, Composite Structures 111 (2014) 497-509.

[18] Y. Chi, L. Xu, Y. Zhang, Experimental study on hybrid fiber–reinforced concrete subjected to uniaxial compression, Journal of Materials in Civil Engineering 26(2) (2014) 211-218.

[19] M. Açikgenç, M. Ulaş, K.E. Alyamaç, Using an artificial neural network to predict mix compositions of steel fiber-reinforced concrete, Arabian Journal for Science and Engineering 40(2) (2015) 407-419.

[20] H. Mashhadban, S.S. Kutanaei, M.A. Sayarinejad, Prediction and modeling of mechanical properties in fiber reinforced self-compacting concrete using particle swarm optimization algorithm and artificial neural network, Construction and Building Materials 119 (2016) 277-287.

[21] K. Jiang, Q. Han, Y. Bai, X. Du, Data-driven ultimate conditions prediction and stress-strain model for FRP-confined concrete, Composite Structures (2020) 112094.

[22] K.M. Hamdia, X. Zhuang, T. Rabczuk, An efficient optimization approach for designing machine learning models based on genetic algorithm, Neural Computing and Applications (2020) 1-11.

[23] K.M. Hamdia, T. Lahmer, T. Nguyen-Thoi, T. Rabczuk, Predicting the fracture toughness of PNCs: A stochastic approach based on ANN and ANFIS, Computational Materials Science 102 (2015) 304-313.

[24] K.M. Hamdia, H. Ghasemi, X. Zhuang, N. Alajlan, T. Rabczuk, Computational machine learning representation for the flexoelectricity effect in truncated pyramid structures, Computers, Materials & Continua 59 (2019), Nr. 1 (2019).

[25] H. Guo, X. Zhuang, T. Rabczuk, A deep collocation method for the bending analysis of Kirchhoff plate, Comput Mater Continua 59(2) (2019) 433-456.

[26] C. Anitescu, E. Atroshchenko, N. Alajlan, T. Rabczuk, Artificial neural network methods for the solution of second order boundary value problems, Computers, Materials and Continua 59(1) (2019) 345-359.
[27] E. Samaniego, C. Anitescu, S. Goswami, V.M. Nguyen-Thanh, H. Guo, K. Hamdia, X. Zhuang, T. Rabczuk, An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementation and applications, Computer Methods in Applied Mechanics and Engineering 362 (2020) 112790.
[28] J. Pizarroso, J. Portela, A. Muñoz, NeuralSens: Sensitivity Analysis of Neural Networks, arXiv preprint arXiv:2002.11423 (2020).
[29] A. Saltelli, Sensitivity analysis for importance assessment, Risk analysis 22(3) (2002) 579-590.
[30] M.H. Shojaeefard, M. Akbari, M. Tahani, F. Farhani, Sensitivity analysis of the artificial neural network outputs in friction stir lap joining of aluminum to brass, Advances in Materials Science and Engineering 2013 (2013).
[31] M. Stone, Cross‐validatory choice and assessment of statistical predictions, Journal of the Royal Statistical Society: Series B (Methodological) 36(2) (1974) 111-133.
[32] V. Nair, G.E. Hinton, Rectified linear units improve restricted boltzmann machines, Proceedings of the 27th international conference on machine learning (ICML-10), 2010, pp. 807-814.
[33] B. Liu, Study on Durability and Mechanical Properties of Hybrid Fiber Reinforced Concrete Liaoning University of Technology, Master Thesis (2019).
[34] K.-H. Yang, Tests on Concrete Reinforced with Hybrid or Monolithic Steel and Polyvinyl Alcohol Fibers, ACI Materials Journal 108(6) (2011).
[35] R. Hai, J. Liu, M. Zhang, L. Zhang, Performance of hybrid steel-polyvinyl alcohol fiber reinforced ultra high performance concrete, Concrete (2016).
[36] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, Scikit-learn: Machine learning in Python, the Journal of machine Learning research 12 (2011) 2825-2830.
[37] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, PyTorch: An imperative style, high-performance deep learning library, Advances in Neural Information Processing Systems, 2019, pp. 8024-8035.
[38] T.J.F. Marian Tietz, Daniel Nouri, Benjamin Bossan, {skorch Developers}, skorch: A scikit-learn compatible neural network library that wraps PyTorch, (2017).
[39] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).
[40] C. Cortes, V. Vapnik, Support-vector networks, Machine learning 20(3) (1995) 273-297.
[41] L. Breiman, Random forests, Machine learning 45(1) (2001) 5-32.
[42] C.E. Rasmussen, Gaussian processes in machine learning, Summer School on Machine Learning, Springer, 2003, pp. 63-71.
[43] C. Standard, Code for Design of Concrete Structures (GB 50010-2010), China Building Industry Press, Beijing, China (2010).

Chapter 5 DEEP TRANSFER LEARNING-BASED VEHICLE CLASSIFICATION BY ASPHALT PAVEMENT VIBRATION[1]

5.1 Introduction

How to effectively, rapidly, and accurately classify vehicles? Vehicle classification plays an important role in intelligent traffic management and pavement performance evaluation. Currently, various technologies are used for vehicle classification and they are divided into two main groups, non-intrusive and intrusive technologies. Non-intrusive technologies contain video imaging (e.g. cameras) and radar [1, 2], while intrusive technologies include piezoelectric sensors, inductive loops, fiber optic sensors, and integrated electronic piezoelectric accelerometers [3-6]. Non-intrusive technologies are sensitive to traffic and weather conditions, while the monitoring information and accuracy are limited [7, 8]. Intrusive technologies strongly relied on the sensor, resulting in high installation and maintenance costs [7, 9]. The weigh-in-motion (WIM) systems are the typical and popular intrusive technologies, while they consist of various sensors and can collect numerous traffic information, such as vehicle types, vehicle speed, vehicle axle counts, and vehicle weight. Due to the high requirements for pavement stiffness and roughness, the pavement needs to be improved to install the WIM systems, increasing the construction and maintenance costs. Apart from high costs, the traditional WIM systems are often susceptible to traffic or weather disruptions [9]. In addition, those sensors used for intrusive technologies sometimes have to be equipped with additional adapters and data acquisition devices due to the lack of data processing, storage, and communication techniques [8, 9]. These limit the widespread use of those sensors and make real-time monitoring more difficult.

To overcome the aforementioned limitations, an acceleration sensor based on the Micro-Electro-Mechanical System (MEMS) was developed and applied in the asphalt pavement vibration monitoring. MEMS vibration sensors show strong advantages in terms of low power consumption, miniaturization, and scalability, meanwhile they achieve wireless communication functions with the help of the Internet of Things (IoT) technology. Numerous researchers have successfully applied MEMS vibration sensors to collect traffic information. Bajwa et al. [7] developed a wireless sensor network for vehicle classification based on axle count and spacing and this network included vibration sensors (measure pavement acceleration), vehicle detection sensors (monitor vehicle's time of arrival and departure), and an access point (record the sensors' data). Ma et al. [10] proposed a prototype axle count and spacing automatic vehicle classification system based on wireless accelerometer sensors (detect vehicle axles) and magnetometer sensors (estimate vehicle speed). Huang et al. [11] developed an IoT-based wireless sensor system (based entirely on wireless accelerometer sensors) for traffic volume and vehicle classification, while they conducted a series of laboratory tests, field tests, and numerical simulations to validate the

[1] Part of this chapter is published as journal papers:

Fangyu Liu, Zhoujing Ye, and Linbing Wang. "Deep transfer learning-based vehicle classification by asphalt pavement vibration." Construction and Building Materials 342 (2022): 127997

monitoring system. Ye et al. [9] collected comprehensive traffic information by a vibration-based in-field pavement monitoring system (MEMS acceleration sensor) and applied the artificial neural network and the k-means++ cluster analysis for vehicle classification and the abnormal vehicle weight, separately. Duarte and Hu [12] investigated the wireless sensor network for vehicle classification, while they built a dataset extracted from this wireless sensor network and applied machine learning models for data conditioning and classification. Although all these works used wireless sensors, some of them installed the sensors on the roadside, which might lead to the missing the real pavement vibration caused by vehicle load. This is because pavement vibration fades rapidly during the transmission in the pavement and pavement vibration measured by the sensors on the roadside tends to be smaller than those caused by vehicle loads [13]. Therefore, the pavement vibration should be collected in the center of the road, where it is usually directly subjected to vehicle loads.

Another issue is how to accurately and efficiently classify vehicles by pavement vibration. The vehicle type and pavement vibration are more likely to have a nonlinear relationship and traffic-induced vibration is often influenced by numerous factors, such as temperature, vehicle speed, location of vehicle loads, pavement structure, pavement roughness, and so on [9, 14]. It is quite difficult to propose an empirical and simple function to establish the relationship between vehicle type and pavement vibration. Deep learning is becoming an ideal solution. In recent decades, deep learning has been widely applied in civil engineering [15-23]. As showing the powerful non-linearity and automatic feature extraction capacities, deep learning, especially convolutional neural network (CNN), has become a popular and efficient tool for vibration signal processing. Yu et al. [15] proposed a CNN-based method to identify and localize the damage of building structures by detecting vibration signals from structures. Dang et al. [16] developed a practical end-to-end framework using a hybrid deep learning model that incorporated CNN and long-short term memory (LSTM), which they applied to process vibration signals and detect the damage. Pal et al. [17] developed a CNN-based method to process the vibration signals of a steel frame structure with bolted connections and then detect the damage condition. Alazzawi and Wang [18] proposed a new deep learning-based structural health monitoring method that combined the raw time-domain structural response signals and the deep residual network. Although these works achieved satisfactory accuracy, the size of their datasets tended to be relatively small. This limits the performance of deep learning models, especially CNN which typically requires a large dataset to maximize its performance. In this case, transfer learning (TL) attracts more attention as it relaxes the requirement for the large dataset. How can transfer learning achieve it? Transfer learning is defined as: given a source domain D_S and learning task T_S, and a target domain D_T and learning task T_T, transfer learning aims to improve the performance of predictive function for learning task T_T by transferring the knowledge from D_S and T_S, where $D_S \neq D_T$ and $T_S \neq T_T$, and the size of D_S is much larger than that of D_T in most cases [24, 25]. When transfer learning is applied in deep neural networks, it is often called deep transfer learning which examines how to leverage knowledge from other domains through deep neural networks [25]. Deep transfer learning is divided into four categories, instance-based deep transfer learning, mapping-based deep transfer learning, network-based deep transfer learning, and adversarial-based deep transfer learning [25]. Deep transfer learning needs a small amount of labeled data in the target

domain to fine-tune the pre-trained deep learning models that have already been trained by a large dataset in the source domain. Deep transfer learning has started to be applied in civil engineering. Gao and Mosalam [26] applied deep transfer learning and CNN to structural damage recognition with two strategies, namely feature extractor and fine-tuning. Gopalakrishnan et al. [27] employed deep transfer learning and CNN to automatically detect cracks in pavement images, which include a large number of defects and non-crack anomalies. Kim et al. [28] developed a CNN-based construction object-detection method to accurately identify construction equipment with the help of deep transfer learning. Li et al. [29] proposed a deep transfer learning-based framework for estimating missing sensor data for high arch dams, which was based on the deep-stacked bidirectional LSTM model. These demonstrate that deep transfer learning has great potential for a wide range of applications in civil engineering.

Therefore, this work aims to propose a deep transfer learning-based vehicle classification by asphalt pavement vibration. The key contributions of this work are summarized as follows:

- The wavelet transform was applied to denoise the pavement vibration signals which were collected by the pavement vibration IoT monitoring system.

- The pavement vibration signals were represented in two different ways, including the time-domain graph and the time-frequency graph.

- Two deep transfer learning-based methods were proposed for vehicle classification according to the two different representations of the pavement vibration signals.

5.2 Methods

Figure 5-1 shows the framework of this work. It aims to apply deep transfer learning for vehicle classification by asphalt pavement vibration and includes three main steps. Firstly, a pavement vibration IoT monitoring system was used to monitor pavement vibration, and raw vibration signals were collected through this system. Secondly, raw vibration signals were denoised by the wavelet transform to obtain denoised vibration signals. Finally, deep transfer learning was applied for vehicle classification. During this step, two deep transfer learning-based methods, namely Method I (Time-domain & TL) and Method II (Time-frequency & TL), were proposed for different types of inputs (time-domain graph and time-frequency graph). In Method I, the CNN model processed time-domain graphs to classify vehicles. In Method II, the CNN model employed time-frequency graphs, which were the results of time-frequency analysis, for vehicle classification.

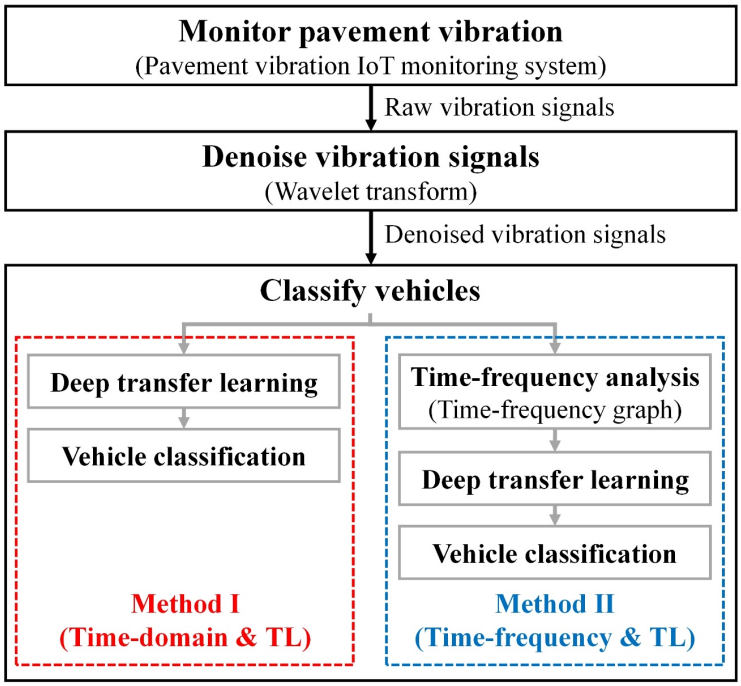

Figure 5-1 Framework of deep transfer learning-based vehicle classification by asphalt pavement vibration

5.2.1 Monitoring pavement vibration

A pavement vibration IoT monitoring system was used to efficiently collect pavement vibration in real-time [8]. This system consisted of acceleration sensing nodes, a gateway, and a cloud platform. This system has been applied in the field to monitor asphalt pavement vibration [8]. Figure 5-2 shows the field layout of the pavement vibration IoT monitoring system. For each traffic direction, two rows of acceleration sensing nodes were deployed, with 7 acceleration sensing nodes in each row. The distance between the two acceleration sensing nodes was 50 cm and the distance between the acceleration sensing node and the roadside was 25 cm and 50 cm in two rows, respectively. The acceleration sensing nodes applied the MEMS technology to support the integration of MEMS sensors, processors, and other electronic components and the detailed information on acceleration sensing nodes could be found in the literature [8]. The acceleration sensing nodes collected pavement vibration signals at a sampling frequency of 500 Hz. Based on low power consumption and high scalability, this system enabled the monitoring, transmission, and analysis of pavement vibration signals [8]. More details could be found in the literature [8].

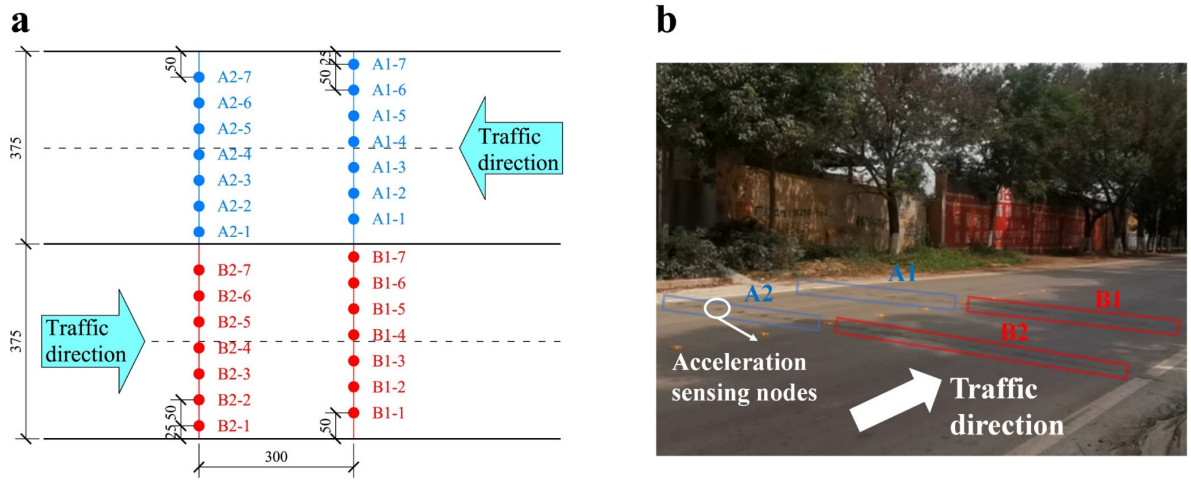

Figure 5-2 The field layout of the pavement vibration IoT monitoring system. (a) Scheme and (b) image of the on-site monitoring.

5.2.2 Denoising vibration signals

5.2.2.1 Wavelet transform

The raw vibration signals collected by the pavement vibration IoT monitoring system contained noises that might affect the accuracy of vehicle classification, highlighting the importance of denoising. Wavelet transform could decompose a signal into several scales representing different frequency bands, and at each scale, the location of the instantaneous structure of the signal can be approximated [30]. Based on this property, wavelet transform has become a popular method for denoising vibration signals [30] in civil engineering [31, 32]. Compared with continuous wavelet transform (CWT), discrete wavelet transform (DWT) is more suitable for denoising signals and more computationally efficient [32]. The threshold function and the wavelet function (i.e. the mother wavelet) play a crucial role in the signal denoising by DWT. The threshold function includes the hard thresholding, the soft thresholding, the compromising method of the hard and soft thresholding, and others. The compromising method of the hard and soft thresholding was proposed in the literature [33] and in the compromising method, the threshold values of the detail coefficients are defined as:

$$\widehat{\omega}_{j,k} = \begin{cases} sign(\omega_{j,k})(|\omega_{j,k}| - \alpha\lambda), & |\omega_{j,k}| \geq \lambda \\ 0, & |\omega_{j,k}| < \lambda \end{cases}, (0 \leq \alpha \leq 1) \quad (5\text{-}1)$$

where, $\widehat{\omega}_{j,k}$ is the estimated wavelet coefficients, $\omega_{j,k}$ is the wavelet coefficients, λ is the selected threshold, *sign*() is the sign function, and α is usually 0.5.

To determine the threshold, Johnstone and Silverman [34] defined a level-dependent threshold applied at each detail level for processing various noises and the threshold is defined as follows:

$$\lambda_m = \sigma_m \times \sqrt{2\,log(N)} = MAD_m/0.6745 \times \sqrt{2\,log(N)} \quad (5\text{-}2)$$

where MAD_m is the median absolute value of the detail coefficients at level m and N is the length of the signal.

The wavelet function is critical in practice as it affects the performance of DWT [32]. Several wavelet functions were investigated in this paper, including Biorthogonal (bior), Symlets (sym), Coiflets (coif), Reverse biorthogonal (rbio), and Daubechies (db), while some wavelet functions are presented in Figure 5-3. The wavelet functions used in this work are based on a Python library named 'PyWavelets' [35].

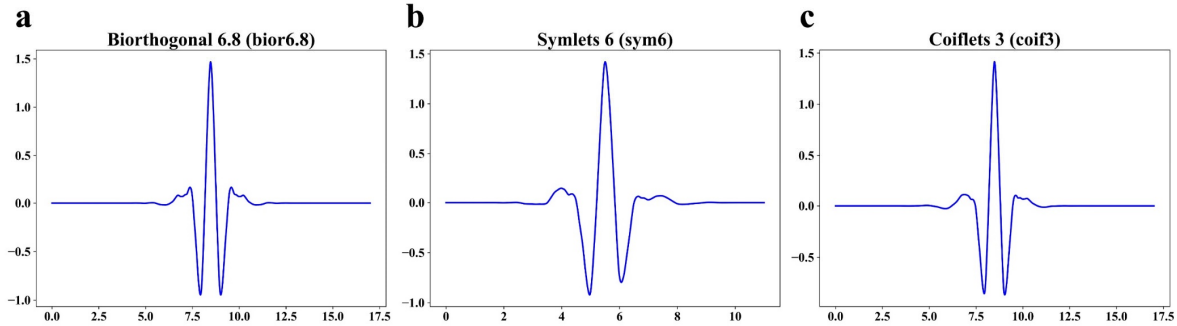

Figure 5-3 Wavelet functions used in this work. (a) Biorthogonal 6.8 (bior6.8), (b) Symlets 6 (sym6), and (c) Coiflets 3 (coif3).

5.2.2.2 Evaluation metrics

Various quantitative evaluation metrics have been used to evaluate the performance of signal denoising in terms of the reconstructed signal quality. The common evaluation metrics include the signal-to-noise ratio (SNR) and the root mean square error (RMSE). The SNR and RMSE were defined as follows [32, 36].

- SNR:

$$SNR = 10log_{10}\left(\frac{P_{signal}}{P_{noise}}\right) \quad (5\text{-}3)$$

- RMSE:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - y_i')^2} \quad (5\text{-}4)$$

Where y_i is the original signal and y_i' is the denoised signal.

5.2.3 Classifying vehicles

During the process of classifying vehicles, two methods were proposed based on deep transfer learning

(Figure 5-1). The difference between these two methods lies in the input image. Method I (Time-domain & TL) was proposed based on the time-domain graph and Method II (Time-frequency & TL) was developed based on the time-frequency graph which was the result of time-frequency analysis.

5.2.3.1 Time-domain graph and Time-frequency graph

The vibration signal has both time and frequency features, and accordingly, the vibration signal is represented as a function of time and frequency. Based on time-domain and frequency-domain representations, the signal graphs are divided into the time-domain graph and the frequency-domain graph. In short, the time-domain graph shows how the signal varies over time (Figure 5-4a), while the frequency-domain graph shows how many signals are in each given frequency band within a certain frequency range. However, either the time-domain graph or the frequency-domain graph ignores one feature of the vibration signal. To address this issue, time-frequency analysis studies the signal in both the time and frequency domains simultaneously and produces time-frequency representations of the vibration signals. Therefore, two different representations were used to represent the pavement vibration signals, including the time-domain graph and the time-frequency graph. The former is the most commonly used method for representing vibration signals, while the latter can study the vibration signals in both the time and frequency domains simultaneously. Based on these two different representations, two different deep transfer learning-based methods would be proposed and compared for vehicle classification, which will be discussed in detail below.

The continuous wavelet transform is an excellent tool for time-frequency analysis, which has already attracted countless attention in the past decades. The continuous wavelet transform generates the time-frequency representation of the signal and plots it in the time-frequency graph (Figure 5-4b). In the time-frequency graph, each data point corresponds to a certain time (X-axis) and frequency (Y-axis), while its color represents the absolute value of the wavelet coefficients. In this work, the continuous wavelet transform was performed by the complex Gaussian wavelet 8 (cgau8), while all these were based on a Python library named 'PyWavelets' [35].

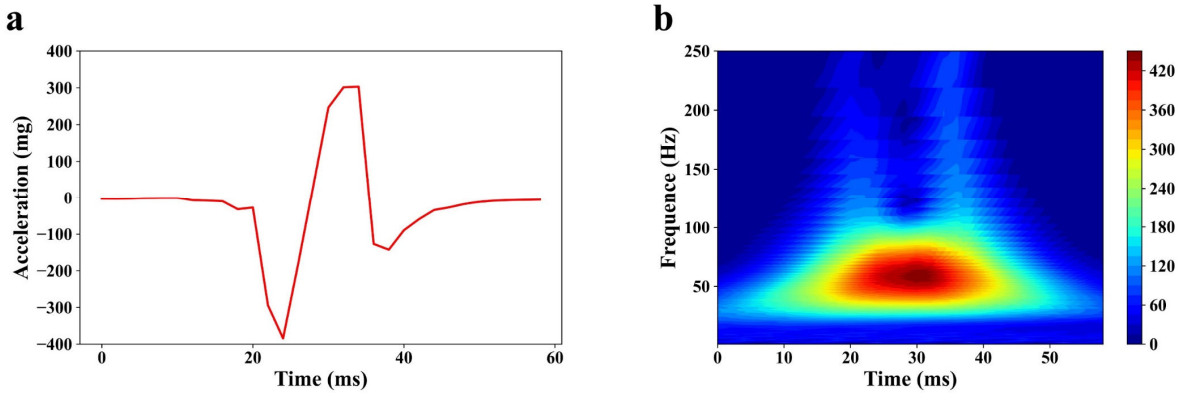

Figure 5-4 Overview of (a) the time-domain graph (a) and (b) the time-frequency graph.

5.2.3.2 Deep transfer learning

For the image classification task (e.g. vehicle classification in this work), deep transfer learning applied to CNN has two main steps (Figure 5-5): (1) For the source domain, the CNN model is trained with a very larger dataset, usually, ImageNet [37] which has 1.2 million images with 1000 categories. The trained CNN model is often referred to as the pre-trained CNN model. (2) For the target domain, the pre-trained CNN model is used to accomplish the target task (vehicle classification in this work). In this process, the convolutional layers of the pre-trained CNN model (i.e., pre-trained convolutional layers) are retained, but the fully connected layers (i.e., FC layers) need to be changed to meet the requirements of the target task. As there are 1000 labels in the source domain and 10 labels in the target domain in this work, the output of the fully connected layers needs to be changed from 1000 to 10. After that, the adjusted pre-trained CNN model is fine-tuned with the dataset of the target domain (i.e. the dataset of asphalt pavement vibration signals).

EfficientNet [38] was used as the CNN model for deep transfer learning in this work. It balances network depth, width, and resolution to have a better performance than traditional CNN models [38]. With the relatively low computation complexity and model complexity, EfficientNet has a satisfactory performance in computer vision tasks. EfficientNet achieves a top-1 accuracy 84.4% of and a top-5 accuracy of 97.1% in ImageNet image classification [38]. Until now, the top-ranked CNN models in the ImageNet image classification task are still based on EfficientNet. Therefore, pre-trained EfficientNets (-B0, -B1, -B2, -B3, -B4, -B5, -B6, and -B7) were used for vehicle classification in this work, which are available on the online sources [39, 40].

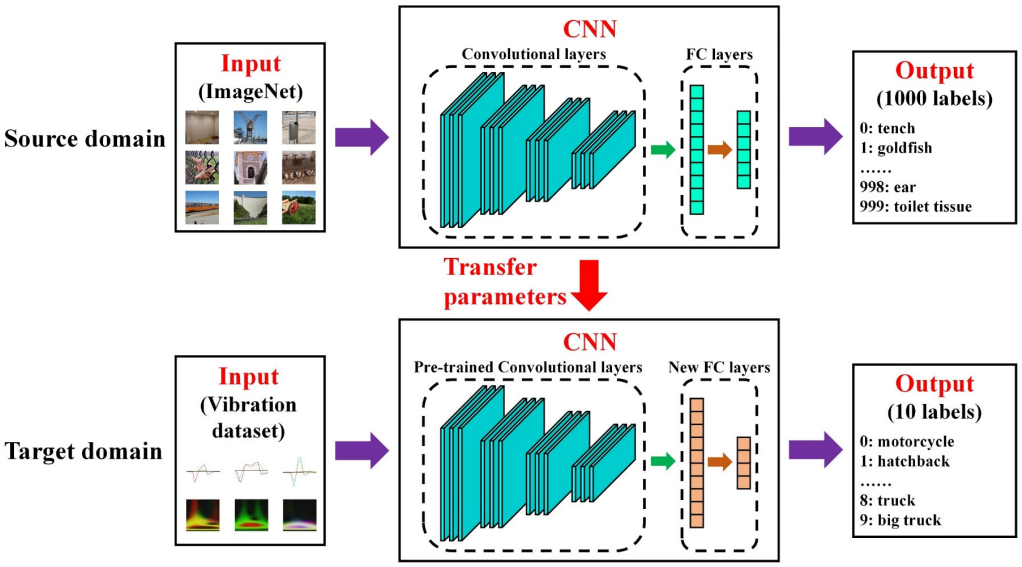

Figure 5-5 Scheme of deep transfer learning applied to CNN

5.2.3.3 Evaluations metrics

Several evaluation metrics are commonly used to evaluate the performance of the CNN model for the image classification task. They include accuracy (Eq.(2-1)), precision (Eq.(2-2)), recall (Eq.(2-3)), and F_1 score (Eq.(2-4)). To consider the effect of multiple classes which are common in the image classification task, the three accuracy metrics have three subsets, 'macro', 'weighted', and 'micro'. 'Macro' means that it needs to calculate accuracy metrics for each class and then obtain unweighted averages, regardless of label imbalances [41]. Similar to'macro', but 'weighted' takes into account label imbalances and calculates weighted averages [41]. In comparison, 'micro' calculates accuracy metrics by counting the total true positives, false negatives, and false positives, and thus the value of micro precision, micro recall, micro F_1 score, and accuracy are identical.

- Accuracy

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5-5}$$

- Precision

$$Precision = \frac{TP}{TP + FP} \tag{5-6}$$

- Recall

$$Recall = \frac{TP}{TP + FN} \tag{5-7}$$

- F_1 score

$$F_1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{5-8}$$

- Macro Precision

$$Precision_{macro} = \frac{\sum_{l \in L} Precision(y_l, \hat{y}_l)}{|L|} \tag{5-9}$$

- Macro Recall

$$Recall_{macro} = \frac{\sum_{l \in L} Recall(y_l, \hat{y}_l)}{|L|} \tag{5-10}$$

- Macro F_1 score

$$F_{1,\ macro} = \frac{\sum_{l \in L} F_1(y_l, \hat{y}_l)}{|L|} \tag{5-11}$$

- Weighted Precision

$$Precision_{weighted} = \frac{\sum_{l \in L} Precision(y_l, \hat{y}_l) \times |y_l|}{\sum_{l \in L} |y_l|} \tag{5-12}$$

- Weighted Recall

$$Recall_{weighted} = \frac{\sum_{l \in L} Recall(y_l, \hat{y}_l) \times |y_l|}{\sum_{l \in L} |y_l|} \quad (5\text{-}13)$$

- Weighted F_1 score

$$F_{1,\ weighted} = \frac{\sum_{l \in L} F_1(y_l, \hat{y}_l) \times |y_l|}{\sum_{l \in L} |y_l|} \quad (5\text{-}14)$$

where L is the set of labels. y is the set of *true* (*sample, label*) pairs and y_l is the subset of y with label l. $\hat{y}$ is the set of *predicted* (*sample, label*) pairs and $\hat{y}_l$ is the subset of $\hat{y}$ with label l. *TP* is true positive, *TN* is true negative, *FP* is false positive, and *FN* is false negative.

5.2.3.4 Image preprocessing and hyperparameters

Image preprocessing is the step of formatting images before images are used for model training. The aim of image preprocessing is to improve the image and make it more suitable for deep learning models. In this work, resizing and normalization, two methods of image preprocessing, are applied to all images. In addition, as a form of image preprocessing, image data augmentation creates different versions of similar content so that the deep learning model is exposed to more training images to avoid the overfitting issue. Image data augmentation used in this work includes random cropping (RandomCrop), random horizontal flipping (RandomHorizontalFlip), and random rotation (RandomRotation). They are only applied in the training set.

The CNN model was based on the online sources [39, 40] and run on PyTorch. The optimization algorithm is Adam [42] and the loss function is the cross-entropy loss. The input image size is 224×224 which meets the requirements of the pre-trained CNN model. Other hyperparameters include the learning rate (0.001), training epoch (100), and batch size (32).

5.3 Results

5.3.1 Denoised vibration signals

As mentioned in Section 2.2, various threshold functions and wavelet functions were used in the wavelet transform for signal denoising. Table 5-1 summarizes the evaluation metrics of the wavelet transform for signal denoising. In three thresholding functions, the compromising method of hard and soft thresholding had the highest SNR and lowest RMSE. Its SNR was more than twice that of the hard thresholding and over three times that of the soft thresholding, while its RMSE is about one-third that of the hard and soft thresholding for Direction A. Although all wavelet functions had similar SNR and RMSE, 'bior6.8' had the highest SNR and lowest RMSE. Therefore, the compromising method of hard and soft thresholding and 'bior6.8' were finally selected as the threshold function and wavelet function.

To visualize the performance of signal denoising, Figure 5-6 shows the images of the raw signals and

denoised signals. As shown in Figure 5-6a, the raw signals had numerous noises and after signal denoising, there were very few noises in the denoised signals which retained the meaningful signal. In the raw signal of acceleration sensing nodes, noises were mainly presented at low signal values (fluctuations at the beginning or end in Figure 5-6b) and the signals with high absolute values (peaks or valleys in Figure 5-6b) had few noises, which are more important for vehicle classification. In the denoised signals of acceleration sensing nodes, the most important signals (peaks or valleys in Figure 5-6b) were almost retained in the original value while noises were almost removed (smoothed curves at the beginning or end in Figure 5-6b). According to Table 5-1 and Figure 5-6, the wavelet transform had a very satisfactory performance in signal denoising, which is important for signal preprocessing.

Table 5-1 Evaluation metrics of the wavelet transform for signal denoising

Threshold function	**Wavelet function**	**Direction A**		**Direction B**	
		SNR	**RMSE**	**SNR**	**RMSE**
Hard thresholding	bior6.8	4.0702	1.7223	3.1715	0.9486
Soft thresholding	bior6.8	2.6665	2.1132	2.0028	1.1278
Compromising method of hard and soft thresholding	bior6.8	**9.3397**	**0.6010**	**4.3913**	**0.5754**
	sym10	9.2976	0.6032	4.3494	0.5770
	sym8	9.3029	0.6029	4.3567	0.5769
	sym6	9.3104	0.6023	4.3540	0.5768
	coif5	9.2897	0.6035	4.3548	0.5768
	coif3	9.3093	0.6025	4.3567	0.5767
	rbio6.8	9.2801	0.6043	4.3421	0.5779
	db10	9.2304	0.6056	4.2475	0.5798
	db8	9.2401	0.6053	4.2641	0.5792
	db6	9.2736	0.6038	4.2914	0.5785
	db4	9.2849	0.6034	4.3116	0.5782

Note: ‘bior’ is Biorthogonal, ‘sym’ is Symlets, ‘coif’ is Coiflets, ‘rbio’ is Reverse biorthogonal, and ‘db’ is Daubechies.

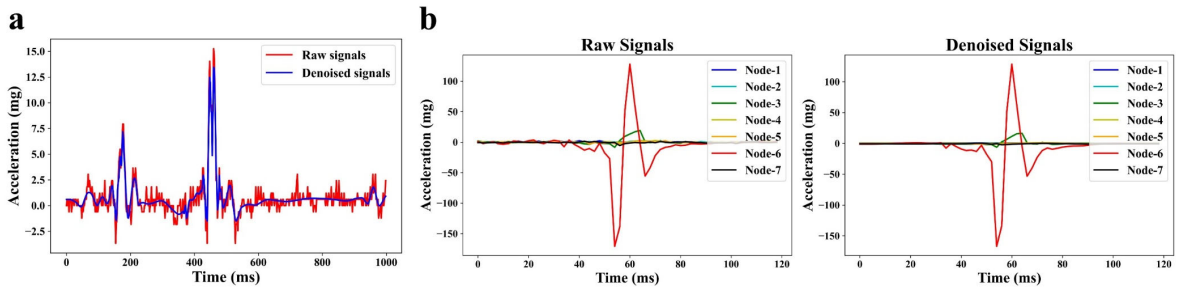

Figure 5-6 Comparison of raw signals and denoised signals. (a) Raw signals and denoised signals of one acceleration sensing node and (b) raw signals and denoised signals of seven acceleration sensing nodes.

5.3.2 Datasets for deep transfer learning

After denoising signals, it needs to build the dataset for deep transfer learning to classify vehicles. As mentioned in Section 2.3, two methods were proposed for vehicle classification based on deep transfer learning. These two methods differed in the type of the input image, the time-domain graph and the time-frequency graph. Figure 5-7a shows the time-domain graph of seven acceleration sensing nodes for one passing vehicle. The time-domain graph allows the vibration signals of all acceleration sensing nodes to be plotted in one graph, which could be found by the legend from 'Node-1' to 'Node-7'. In comparison, the time-frequency graph requires a single graph to plot the vibration signals of only one acceleration sensing node. It means that one passing vehicle needs seven time-frequency graphs to present the vibration signals of all acceleration sensing nodes. Figure 5-7b shows three of the seven time-frequency graphs for the one passing vehicle in the same manner as Figure 5-7a. In Figure 5-7a, Node-6 had the highest vibration signals, followed by Node-3 and the other acceleration sensing nodes had very few vibration signals. Accordingly, Node-6 had the highest and most variable wavelet coefficients, and in Figure 5-7b, it had the most colorful time-frequency graph. The wavelet coefficients of Node-3 were much lower than those of Node-6 and varied in a smaller ranger, so the time-frequency graph of Node-3 was less colorful with a few colors. In comparison, the time-frequency graph of Node-7 was plotted in blue as its vibration signals were almost zero.

When building the dataset, two methods need two different datasets. Method I (Time-domain & TL) is much easier to build its dataset because it only needs one time-domain graph to plot the vibration signals of all acceleration sensing nodes for one passing vehicle. However, Method II (Time-frequency & TL) needs seven time-frequency graphs to present the same corresponding information. How to combine these seven graphs to classify one passing vehicle? As shown in Figure 5-7a, not all acceleration sensing nodes observed vibration signals. Therefore, the number of acceleration sensing nodes successfully obtaining vibration signals for one passing vehicle was summarized in Figure 5-8. A maximum of three acceleration sensing nodes was employed to measure vibration signals for all vehicles. Only trucks and big trucks needed three acceleration sensing nodes and two acceleration sensing nodes were required for the other vehicles except for motorcycles which used one. It means three time-frequency graphs were needed instead of seven graphs to illustrate the vibration signals of all acceleration sensing nodes for one passing vehicle. However, the three time-frequency graphs were numbered differently for different vehicles because vehicles often passed through different locations of one lane. How to select these three time-frequency graphs? When selecting the three time-frequency graphs, it was necessary to choose the time-frequency graphs of acceleration sensing nodes that observed vibration signals. Therefore, it needed to first sort the acceleration sensing nodes by the absolute value of the observed vibration signals and then select the top three acceleration sensing nodes to plot their time-frequency graphs. These three graphs would be used for vehicle classification.

As there are three time-frequency graphs for one passing vehicle, it is possible to consider one graph as one channel of an RGB image. In particular, the time-frequency graph was changed from the RGB image (3

channels) to the gray image (1 channel) and then three time-frequency graphs were combined together into an RGB image. For example, three time-frequency graphs in Figure 5-7b were combined to plot an RGB image in Figure 5-9b. This image would be used to build the dataset for Method II (Time-frequency & TL). In order to eliminate irrelevant information, the axis ticks, axis labels, legends, titles, color bars, etc were removed during the building of the datasets. Figure 5-9 shows the exemplary images of the datasets for the two methods.

Table 5-2 summarizes the information of the dataset built in this work. This dataset contained 10 categories of vehicles, including the motorcycle, hatchback, sedan, SUV, mini VAN, VAN, pickup truck, mini truck, truck, and big truck. The number of each category varied a lot and there was a total of 2098 vehicles collected in this dataset. The number of sedans was the largest, followed by the truck and big truck, which was much larger than the other vehicle types. To train and evaluate the CNN model, the whole dataset was separated into two subsets, training (80%) and test (20%) sets. The number of vehicles in each category in the training and test sets could also be found in Table 5-2.

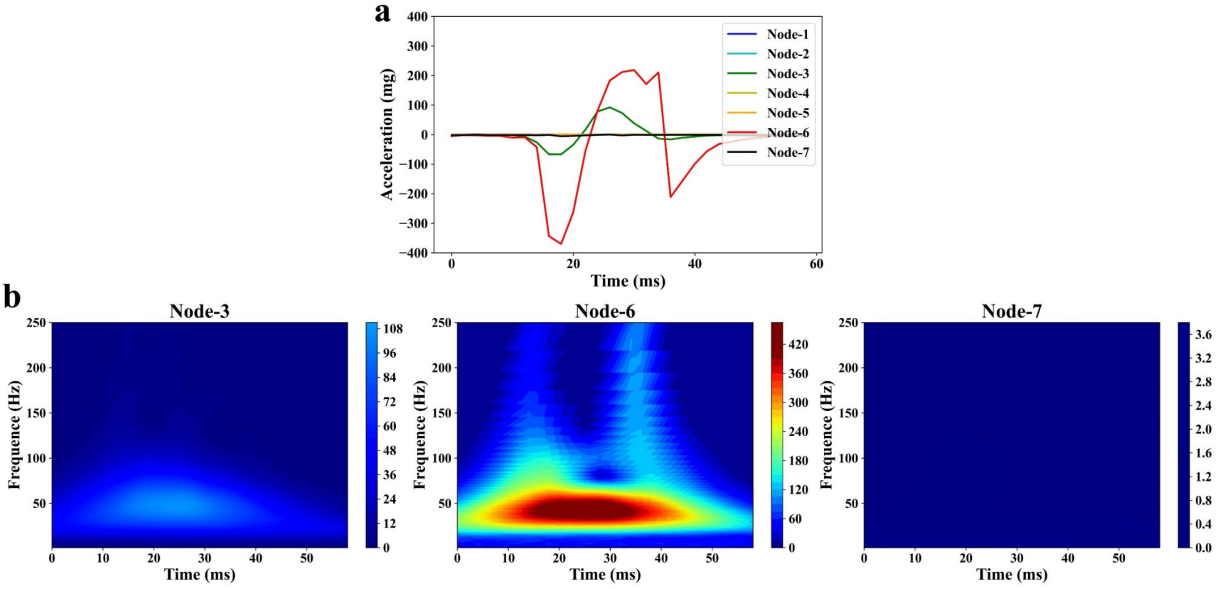

Figure 5-7 Denoised signals of seven acceleration sensing nodes and their corresponding time-frequency graphs. (a) Denoised signals of seven acceleration sensing nodes and (b) time-frequency graphs of Node-3, Node-6, and Node-7.

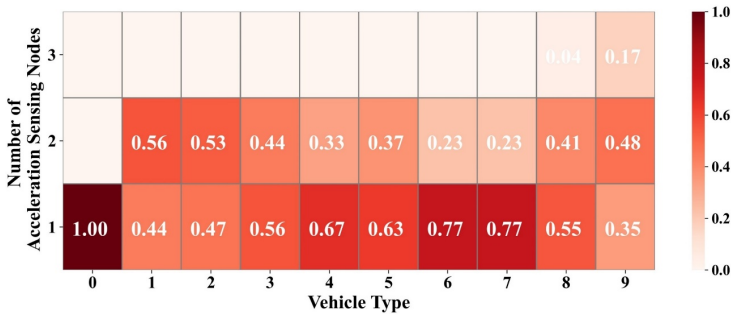

Figure 5-8 The number of acceleration sensing nodes that received effective vibration signals (defined in [8, 13]). 0: motorcycle, 1: hatchback, 2: sedan, 3: SUV, 4: mini VAN, 5: VAN, 6: pickup truck, 7: mini truck, 8: truck, and 9: big truck.

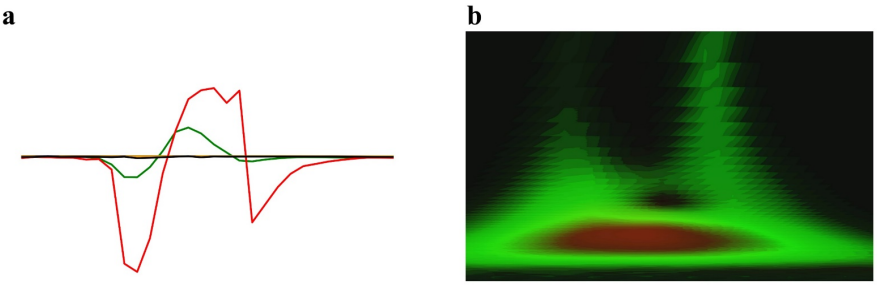

Figure 5-9 Image of the dataset for two methods. (a) Method I (Time-domain & TL) and (b) Method II (Time-frequency & TL).

Table 5-2 Summary of the dataset

	0	1	2	3	4	5	6	7	8	9	Total
Train	32	40	443	206	169	50	18	38	390	301	1687
Test	4	14	133	49	30	7	4	6	56	108	411
Total	36	54	576	255	199	57	22	44	446	409	2098

Note: 0: motorcycle, 1: hatchback, 2: sedan, 3: SUV, 4: mini VAN, 5: VAN, 6: pickup truck, 7: mini truck, 8: truck, and 9: big truck.

5.3.3 Accuracy metrics of vehicle classification

5.3.3.1 Accuracy metrics of image preprocessing methods

As mentioned above, image data augmentation exposes the CNN model to more complex training images to avoid the overfitting issue. Three image data augmentation methods, including RandomCrop, RandomHorizontalFlip, and RandomRotation, were used in this work. Table 5-3 summarizes the accuracy metrics of the test set in terms of image preprocessing methods. Without image data augmentation, the CNN model in both methods faced the overfitting issue. After applying image data augmentation, the accuracy of the CNN model improved a lot no matter which method was chosen. So it is necessary to apply image

data augmentation to the training set. In addition, the same image data augmentation method differently affected the accuracy of the two methods. RandomCrop had the highest accuracy in Method I and RandomHorizontalFlip plus RandomRotation achieved the highest accuracy in Method II. Therefore, Method I and Method II employed different image data augmentation methods to maximize their performance. After applying the selected image data augmentation methods to the training set, the pre-trained CNN models for Method I and Method II were trained and evaluated on the dataset for vehicle classification.

Table 5-3 Accuracy metrics of image preprocessing methods

Image preprocessing	Method I (Time-domain & TL)			Method II (Time-frequency & TL)		
	Accuracy	**Macro F_1**	**Weighted F_1**	**Accuracy**	**Macro F_1**	**Weighted F_1**
Resize	0.8200	0.6431	0.8103	0.8856	0.7731	0.8818
Resize RandomCrop	**0.9489**	**0.9299**	**0.9475**	0.9416	0.8089	0.9372
Resize RandomHorizontalFlip RandomRotation	0.9270	0.8836	0.9259	**0.9611**	**0.8738**	**0.9578**
Resize RandomCrop RandomHorizontalFlip	0.9319	0.8516	0.9286	0.9246	0.7176	0.9148
Resize RandomCrop RandomHorizontalFlip RandomRotation	0.9173	0.8392	0.9146	0.9221	0.7390	0.9155

5.3.3.2 Accuracy metrics of deep transfer learning

Two methods, Method I (Time-domain & TL) and Method II (Time-frequency & TL), were proposed for vehicle classification. Figure 5-10 shows the loss of these two methods. Three EfficientNets (-B0, -B3, and -B7) were selected to present their results because these three CNN models have different accuracy on ImageNet with different computational and model complexity. For Method I (Figure 5-10a), the loss of three CNN models had a similar pattern that the loss decreased with the increasing epoch and then converged despite few fluctuations. The models in Method II performed similarly to those in Method I. However, these three CNN models in Method II converged at approximately Epoch-40, much faster than those in Method I (about Epoch-65), while the final loss values of the three CNN models were similar in these two methods.

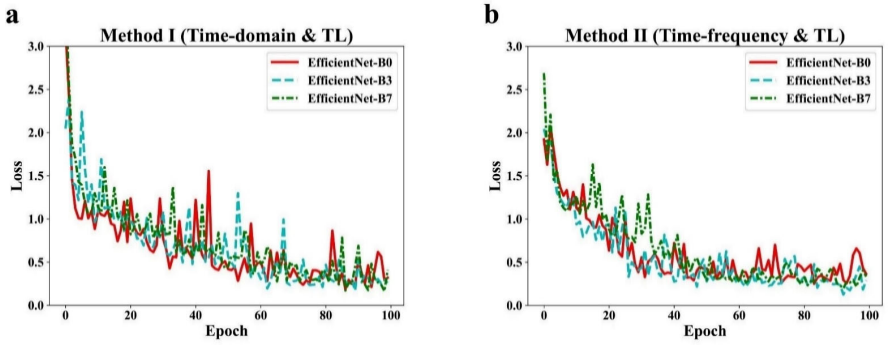

Figure 5-10 Loss of two methods. (a) Method I (Time-domain & TL) and (b) Method II (Time-frequency & TL).

Figure 5-11 shows the accuracy metrics of two methods, Method I (Time-domain & TL) and Method II (Time-frequency & TL). In summary, similar to the loss (Figure 5-10), the CNN models in Method II converged much faster than those in Method I. Although three CNN models eventually reached similar values for all accuracy metrics in both methods, the accuracy metrics of EfficientNet-B0 and -B3 increased more rapidly than EfficientNet-B7, especially in Method II. In both two methods, all three CNN models had similar patterns between the accuracy and weighted F_1 score, which was higher than the macro F_1 score.

Table 5-4 summarizes the accuracy metrics of Method I (Time-domain & TL). EfficientNet-B3 achieved the highest values for all accuracy metrics, except for the macro precision and macro F_1 score, which were the highest in EfficientNet-B0. It means that considering the label imbalance, EfficientNet-B3 had the best performance, but without eliminating the effect of the label imbalance, EfficientNet-B0 performed better than EfficientNet-B3. Table 5-5 summarizes the accuracy metrics of Method II (Time-frequency & TL). Similar to the result of Method I, EfficientNet-B3 had the highest values for all accuracy metrics, except for the macro precision, macro recall, and macro F_1 score. EfficientNet-B2 achieved the highest macro precision and EfficientNet-B7 did it in the macro recall and macro F_1 score. In both two methods, EfficientNet-B3 had the highest accuracy metrics when considering the label imbalance, but other EfficientNets performed better in the accuracy metrics without taking into account the label imbalance. Meanwhile, EfficientNet had a satisfactory performance in both methods with the accuracy of Method I exceeding 0.94 and Method II exceeding 0.95, indicating that the CNN model with deep transfer learning could be an effective, accurate, and reliable tool for vehicle classification based on asphalt pavement vibration. From the perspective of accuracy metrics, EfficientNet-B3 is the best model in both methods.

In addition, there is no obvious relationship between the accuracy and the computation and model complexity. For example, EfficientNet-B2 and EfficientNet-B7 had identical accuracy in Method I while the computation and model complexity of EfficientNet-B7 were much more complex than EfficientNet-B2. In Method II, the accuracy of EfficientNet-B2 was much higher than that of EfficientNet-B7.

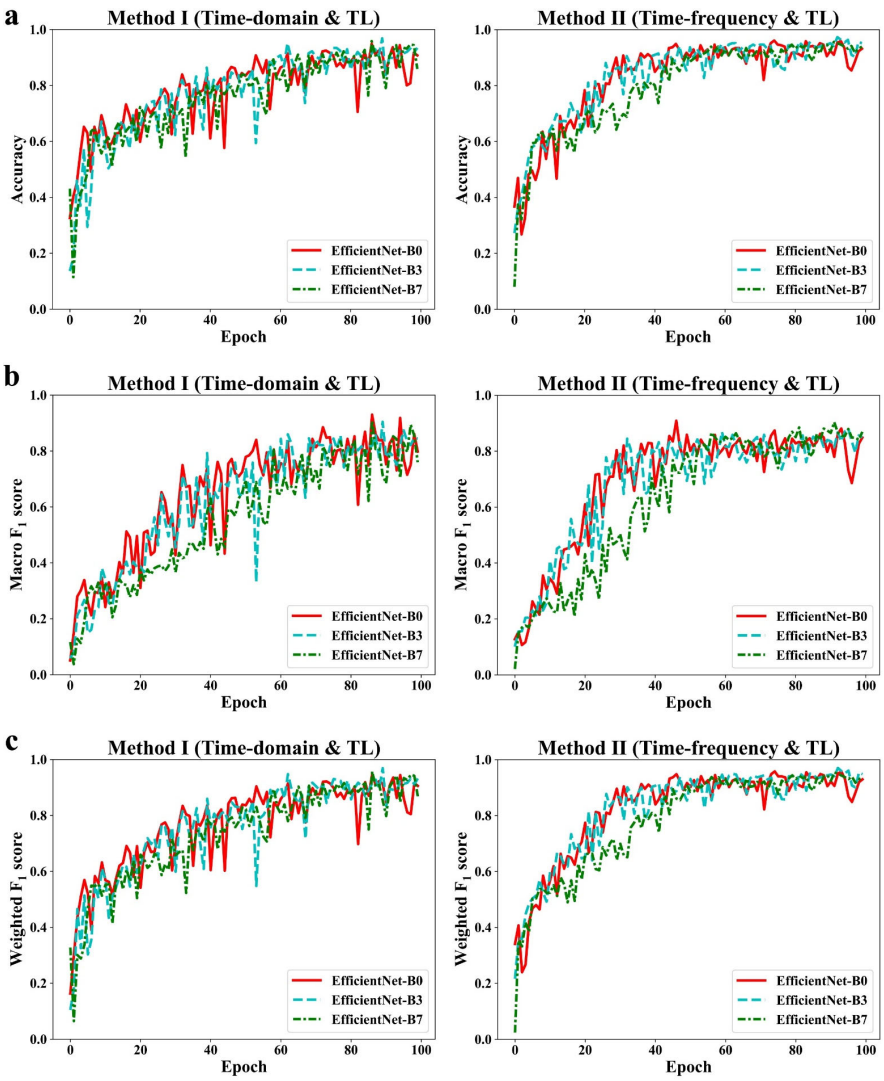

Figure 5-11 Accuracy metrics of two methods, Method I (Time-domain & TL) and Method II (Time-frequency & TL). (a) Accuracy, (b) macro F_1 score, and (c) weighted F_1 score.

Table 5-4 Accuracy metrics of Method I (Time-domain & TL)

Model	Accuracy	Precision		Recall		F_1 score	
		Macro	Weighted	Macro	Weighted	Macro	Weighted
EfficientNet-B0	0.9489	**0.9714**	0.9503	0.9022	0.9489	**0.9299**	0.9475
EfficientNet-B1	0.9513	0.9410	0.9545	0.8982	0.9513	0.9082	0.9509
EfficientNet-B2	0.9586	0.9533	0.9608	0.8840	0.9586	0.9046	0.9571
EfficientNet-B3	**0.9684**	0.9247	**0.9728**	**0.9099**	**0.9684**	0.9042	**0.9688**
EfficientNet-B4	0.9635	0.9518	0.9653	0.8868	0.9635	0.8941	0.9615
EfficientNet-B5	0.9635	0.9437	0.9637	0.8359	0.9635	0.8761	0.9618
EfficientNet-B6	0.9562	0.9534	0.9601	0.8819	0.9562	0.8934	0.9546

EfficientNet-B7	0.9586	0.9631	0.9606	0.8819	0.9586	0.9097	0.9569

Table 5-5 Accuracy metrics of Method II (Time-frequency & TL)

Model	Accuracy	Precision		Recall		F_1 score	
		Macro	Weighted	Macro	Weighted	Macro	Weighted
EfficientNet-B0	0.9611	0.9765	0.9626	0.8356	0.9611	0.8738	0.9578
EfficientNet-B1	0.9611	0.9740	0.9619	0.8501	0.9611	0.8918	0.9590
EfficientNet-B2	0.9708	**0.9844**	0.9715	0.8529	0.9708	0.8909	0.9680
EfficientNet-B3	**0.9732**	0.9702	**0.9744**	0.8340	**0.9732**	0.8673	**0.9700**
EfficientNet-B4	0.9684	0.9735	0.9692	0.8212	0.9684	0.8514	0.9641
EfficientNet-B5	0.9635	0.9752	0.9646	0.8293	0.9635	0.8679	0.9604
EfficientNet-B6	0.9659	0.9628	0.9664	0.8691	0.9659	0.8916	0.9637
EfficientNet-B7	0.9562	0.9698	0.9571	**0.8841**	0.9562	**0.9035**	0.9541

5.4 Discussion

Two deep transfer learning-based methods, Method I (Time-domain & TL) and Method II (Time-frequency & TL), were proposed for vehicle classification in this work and the results of both methods show that the CNN model with deep transfer learning could be an effective, accurate, and reliable technique for vehicle classification based on asphalt pavement vibration. These two methods employed different input types and performed differently on the accuracy metrics. So what is the difference between these two methods? What are the advantages and disadvantages of these two methods? These two methods differ in three aspects:

5.4.1 Difference in input types

These two methods utilized different inputs, the time-domain graph and the time-frequency graph. Their inputs are different in three ways. (1) The features of vibration signals. Method I only used time features of vibration signals and Method II processed both time and frequency features of vibration signals simultaneously. (2) The dimension of inputs. Although both methods employed graphs, the dimension of inputs is different. The time-domain graph is two-dimensional, while the X-axis is time and the Y-axis is the vibration signal (Figure 5-4a). The time-frequency graph could be viewed as three-dimensional, while the X-axis is time, the Y-axis is frequency, and the Z-axis (color) is the wavelet coefficients (Figure 5-4b). (3) The data fusion method. As seven acceleration sensing nodes were used to measure pavement vibration for one passing vehicle, it needs to combine the vibration signals of all acceleration sensing nodes as the input to the CNN model. In Method I, the vibration signals of all acceleration sensing nodes are easily plotted in an image that is considered as the input (Figure 5-9a). In comparison, in Method II, each acceleration sensing node has its own time-frequency graph. After statistical analysis (Figure 5-8), the top three acceleration sensing nodes are selected to plot their time-frequency graphs as gray (1 channel) images.

Then these three gray images are combined as one RGB (3 channels) image that is considered as the input (Figure 5-9b).

5.4.2 Difference in accuracy metrics

To better compare the two methods, Figure 5-12 summarizes the accuracy metrics of the CNN models for two methods and Figure 5-13 presents the normalized confusion matrix of EfficientNet-B0 and -B3 for two methods. In terms of accuracy metrics, these two methods differ in three ways.

(1) The accuracy considers the label imbalance. The label imbalance means that the distribution of images across classes is not identical. In the label imbalance, there are two groups, majority class and minority class, while the former has many images and the latter has few images. The mass of images from the majority class would drown out the minority class, resulting in the deep learning model focusing only on learning the feature of the majority class with abundant images and ignoring the minority class which is in fact equally meaningful [43, 44]. This is because the deep learning model assumes that each class has an equal distribution [44]. The dataset in this work had ten classes with different numbers of images (Table 5-2), resulting in the label imbalance. The accuracy, weighted precision, weighted recall, and weighted F_1 score take into account the label imbalance. As shown in Figure 5-12a and c, the accuracy and weighted F_1 score of almost all CNN models in Method II were higher than those in Method I. When back to the confusion matrix shown in Figure 5-13, compared to Method I, the CNN models in Method II have better performance in the majority class. This is because the important informative region in the time-frequency graph is larger than that of the time-domain graph (Figure 5-14a). Also, the discrepancy between similar vibration signals is more apparent in the time-frequency graph compared to the time-domain graph (Figure 5-14b and c). These characteristics help make the CNN model easier to learn from the time-frequency graph, thus improving accuracy.

(2) The accuracy without taking into account the label imbalance. In comparison, the macro precision, macro recall, and macro F_1 score do not account for the label imbalance and assume that each label is equally important. As shown in Figure 5-12b, the macro F_1 score of all CNN models in Method I was higher than those in Method II. This is due to the fact that the CNN models in Method I perform better in the minority class (e.g. motorcycle and pickup truck) than Method II (Figure 5-13) and the unweighted mean between the majority class and minority class improves the final value of accuracy metrics. The minority class has few examples, making the CNN models in Method II more difficult to be trained because the time-frequency graph is more complex than the time-domain graph. In addition, different time-domain graphs might have similar time-frequency graphs (Figure 5-14d), especially for the motorcycle that collects vibration signals with one acceleration sensing node. This led to the low accuracy of the CNN models in Method II for the motorcycle (Figure 5-13).

(3) The convergence speed and stability of the CNN model. As shown in Figure 5-11, the CNN models in Method II converged faster and performed more stably than those in Method I by using the same

hyperparameters. This is important for training the CNN model. The rapid convergence means that the CNN model needs fewer epochs or training time and the high stability means that the CNN model is more robust.

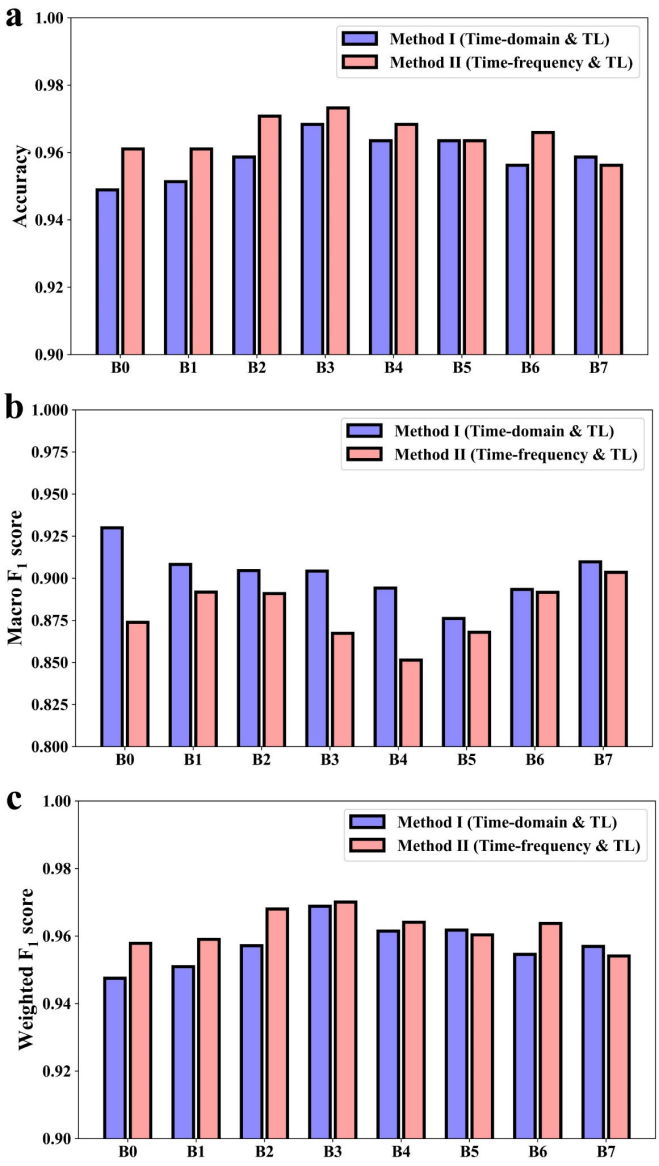

Figure 5-12 Accuracy metrics of the CNN models for two methods, Method I (Time-domain & TL) and Method II (Time-frequency & TL). (a) Accuracy, (b) macro F_1 score, and (c) weighted F_1 score.

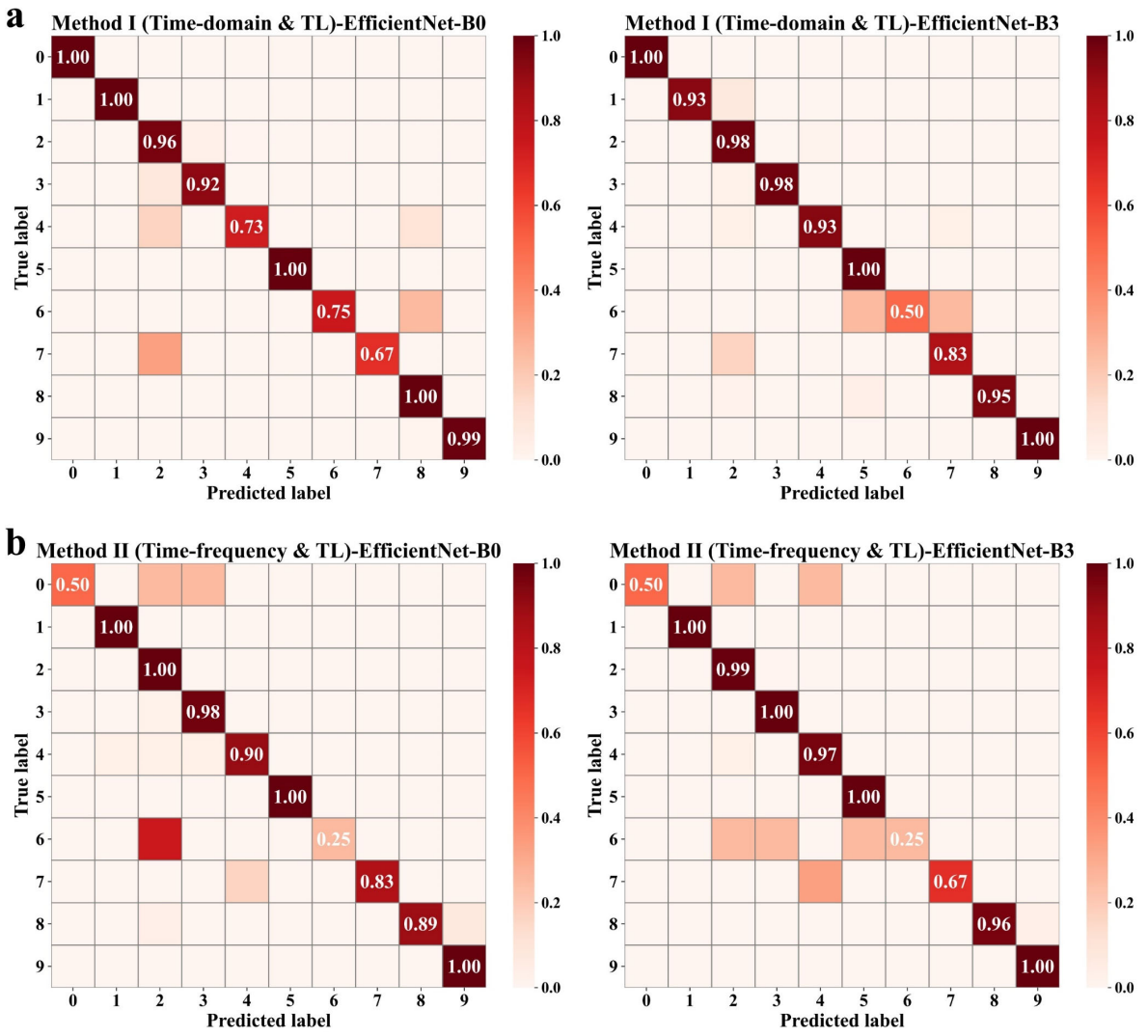

Figure 5-13 Normalized confusion matrix of EfficientNet-B0 and -B3 for two methods. (a) Method I (Time-domain & TL) and (b) Method II (Time-frequency & TL). Note: 0: motorcycle, 1: hatchback, 2: sedan, 3: SUV, 4: mini VAN, 5: VAN, 6: pickup truck, 7: mini truck, 8: truck, and 9: big truck.

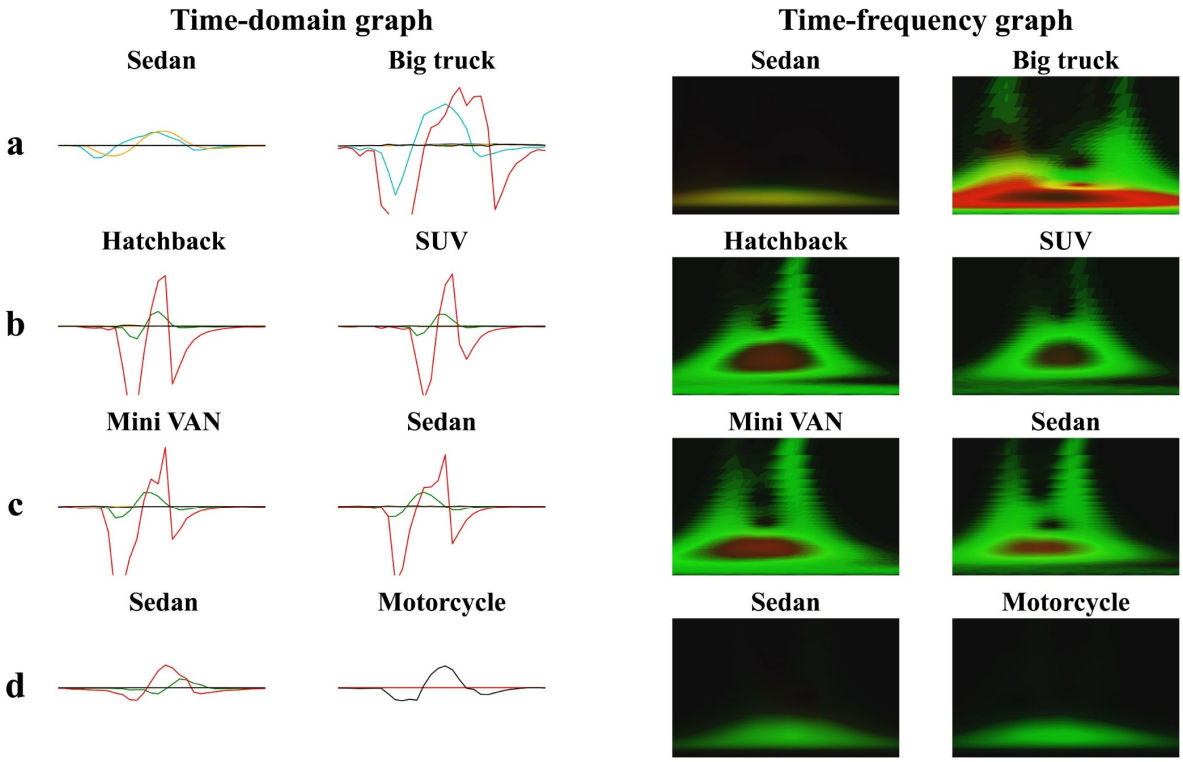

Figure 5-14 Comparison of time-domain graphs and time-frequency graphs

5.4.3 Difference in application prospects

These two methods also have different application prospects. (1) Method I runs at a higher running speed than Method II and requires less time for vehicle classification. Compared to Method I, Method II introduces the time-frequency analysis to obtain the time-frequency graph and although the time-frequency analysis is very fast, it still takes some time. The high usage time for vehicle classification might limit the application of Method II in contrast to Method I. (2) Method II needs more data (image) than Method I. According to Figure 5-12 and Figure 5-13, compared with Method I, the CNN model in Method II performed better in the majority class, but worse in the minority class. The time-frequency graph is more complex than the time-domain graph (Figure 5-14), so the CNN model in Method II requires more data to maximize its performance. In the future, more data will be collected on each vehicle type to increase the dataset size and eliminate the label imbalance in the dataset. (3) Method II employs a different data fusion method than Method I. Unlike Method I, which plots the vibration signals of all acceleration sensing nodes in one graph, Method II requires several graphs to present the same information about one passing vehicle as in Method I. In this work, after statistics analysis, three time-frequency graphs were combined as the input. But if the acceleration sensing nodes are arranged too close together or the vehicle is too wide, more than three acceleration sensing nodes might collect the vibration signal. How to fuse data? There are two alternatives. The first method is to increase the channel of the input to combine all possible time-frequency graphs. If the input has more than three channels, the first convolutional layer of the pre-trained CNN model needs to be adjusted, which may affect its performance. Another method is to use the results of the time-

frequency analysis as the matrix rather than the graph and then apply the LSTM to process the matrix. This method was also investigated by previous literature [45, 46]. The problem with this method is that it is difficult to find the pre-trained LSTM with large datasets, which is different from the pre-trained CNN model. In this case, the LSTM might not achieve a similar performance to the pre-trained CNN model due to the lack of data.

5.5 Conclusion

Deep transfer learning needs a small number of labeled data to achieve satisfactory performance in the target domain, and thus has great potential for a wide range of applications in civil engineering. This work aims to propose a deep transfer learning-based method for vehicle classification by asphalt pavement vibration. The main contributions of this work include: (1) the wavelet transform was used to denoise the pavement vibration signals which were collected by the pavement vibration IoT monitoring system; (2) the pavement vibration signals were represented in two different ways, including the time-domain graph and the time-frequency graph; (3) two deep transfer learning-based methods were proposed for vehicle classification based on the two different representations of the pavement vibration signals. In addition, the input types, accuracy metrics, and application prospects were applied to investigate the feasibility and validity of two deep transfer learning-based methods for vehicle classification. The results show that the CNN model had a satisfactory performance in both methods with the accuracy of Method I exceeding 0.94 and Method II exceeding 0.95. The CNN model in Method II performed better in the accuracy metrics with considering label imbalance, but worse in the accuracy metrics without considering label imbalance than that in Method I. In both two methods, EfficientNet-B3 had the highest accuracy metrics when considering the label imbalance, but other EfficientNets performed better in the accuracy metrics without taking into account the label imbalance. The CNN model with deep transfer learning could be an effective, accurate, and reliable method for vehicle classification based on asphalt pavement vibration.

5.6 Reprint and Permision

©[2022] Elsevier, from [Fangyu Liu, Zhoujing Ye, and Linbing Wang. "Deep transfer learning-based vehicle classification by asphalt pavement vibration." Construction and Building Materials 342 (2022): 127997.] Please note that, as the author of this Elsevier article, you retain the right to include it in a thesis or dissertation, provided it is not published commercially. Permission is not required.

References

[1] N.K. Kanhere, S.T. Birchfield, A taxonomy and analysis of camera calibration methods for traffic monitoring applications, IEEE Transactions on Intelligent Transportation Systems 11(2) (2010) 441-452.

[2] J. Sánchez-Oro, D. Fernández-López, R. Cabido, A.S. Montemayor, J.J. Pantrigo, Radar-based road-traffic monitoring in urban environments, Digital Signal Processing 23(1) (2013) 364-374.

[3] J. Zhang, Y. Lu, Z. Lu, C. Liu, G. Sun, Z. Li, A new smart traffic monitoring method using embedded cement-based piezoelectric sensors, Smart Materials and Structures 24(2) (2015) 025023.

[4] B. Coifman, S. Kim, Speed estimation and length based vehicle classification from freeway single-loop detectors, Transportation research part C: emerging technologies 17(4) (2009) 349-364.
[5] H. Zhao, D. Wu, M. Zeng, S. Zhong, A vibration-based vehicle classification system using distributed optical sensing technology, Transportation Research Record 2672(43) (2018) 12-23.
[6] D. Kleyko, R. Hostettler, N. Lyamin, W. Birk, U. Wiklund, E. Osipov, Vehicle classification using road side sensors and feature-free data smashing approach, 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), IEEE, 2016, pp. 1988-1993.
[7] R. Bajwa, R. Rajagopal, P. Varaiya, R. Kavaler, In-pavement wireless sensor network for vehicle classification, Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks, IEEE, 2011, pp. 85-96.
[8] Z. Ye, G. Yan, Y. Wei, B. Zhou, N. Li, S. Shen, L. Wang, Real-Time and Efficient Traffic Information Acquisition via Pavement Vibration IoT Monitoring System, Sensors 21(8) (2021) 2679.
[9] Z. Ye, H. Xiong, L. Wang, Collecting comprehensive traffic information using pavement vibration monitoring data, Computer‐Aided Civil and Infrastructure Engineering 35(2) (2020) 134-149.
[10] W. Ma, D. Xing, A. McKee, R. Bajwa, C. Flores, B. Fuller, P. Varaiya, A wireless accelerometer-based automatic vehicle classification prototype system, IEEE Transactions on Intelligent Transportation Systems 15(1) (2013) 104-111.
[11] Y. Huang, L. Wang, Y. Hou, W. Zhang, Y. Zhang, A prototype IOT based wireless sensor network for traffic information monitoring, International journal of pavement research and technology 11(2) (2018) 146-152.
[12] M.F. Duarte, Y.H. Hu, Vehicle classification in distributed sensor networks, Journal of Parallel and Distributed Computing 64(7) (2004) 826-838.
[13] Z. Ye, L. Wang, W. Xu, Z. Gao, G. Yan, Monitoring traffic information with a developed acceleration sensing node, Sensors 17(12) (2017) 2817.
[14] Z. Ye, Y. Lu, L. Wang, Investigating the pavement vibration response for roadway service condition evaluation, Advances in Civil Engineering 2018 (2018).
[15] Y. Yu, C. Wang, X. Gu, J. Li, A novel deep learning-based method for damage identification of smart building structures, Structural Health Monitoring 18(1) (2019) 143-163.
[16] H.V. Dang, H. Tran-Ngoc, T.V. Nguyen, T. Bui-Tien, G. De Roeck, H.X. Nguyen, Data-driven structural health monitoring using feature fusion and hybrid deep learning, IEEE Transactions on Automation Science and Engineering 18(4) (2020) 2087-2103.
[17] J. Pal, S. Sikdar, S. Banerjee, A deep‐learning approach for health monitoring of a steel frame structure with bolted connections, Structural Control and Health Monitoring 29(2) (2022) e2873.
[18] O. Alazzawi, D. Wang, A novel structural damage identification method based on the acceleration responses under ambient vibration and an optimized deep residual algorithm, Structural Health Monitoring (2022) 14759217211065009.
[19] F. Liu, W. Ding, Y. Qiao, L. Wang, An artificial neural network model on tensile behavior of hybrid steel-PVA fiber reinforced concrete containing fly ash and slag power, Frontiers of Structural and Civil Engineering 14(6) (2020) 1299-1315.
[20] F.-y. Liu, W.-q. Ding, Y.-f. Qiao, L.-b. Wang, Q.-y. Chen, Compressive behavior of hybrid steel-polyvinyl

alcohol fiber-reinforced concrete containing fly ash and slag powder: experiments and an artificial neural network model, Journal of Zhejiang University-SCIENCE A 22(9) (2021) 721-735.
[21] F. Liu, L. Wang, UNet-based model for crack detection integrating visual explanations, Construction and Building Materials 322 (2022) 126265.
[22] F. Liu, J. Liu, L. Wang, Asphalt Pavement Crack Detection Based on Convolutional Neural Network and Infrared Thermography, IEEE Transactions on Intelligent Transportation Systems (2022).
[23] F. Liu, J. Liu, L. Wang, Deep learning and infrared thermography for asphalt pavement crack severity classification, Automation in Construction 140 (2022) 104383.
[24] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Transactions on knowledge and data engineering 22(10) (2009) 1345-1359.
[25] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, C. Liu, A survey on deep transfer learning, International conference on artificial neural networks, Springer, 2018, pp. 270-279.
[26] Y. Gao, K.M. Mosalam, Deep transfer learning for image‐based structural damage recognition, Computer‐Aided Civil and Infrastructure Engineering 33(9) (2018) 748-768.
[27] K. Gopalakrishnan, S.K. Khaitan, A. Choudhary, A. Agrawal, Deep convolutional neural networks with transfer learning for computer vision-based data-driven pavement distress detection, Construction and building materials 157 (2017) 322-330.
[28] H. Kim, H. Kim, Y.W. Hong, H. Byun, Detecting construction equipment using a region-based fully convolutional network and transfer learning, Journal of computing in Civil Engineering 32(2) (2018) 04017082.
[29] Y. Li, T. Bao, H. Chen, K. Zhang, X. Shu, Z. Chen, Y. Hu, A large-scale sensor missing data imputation framework for dams using deep learning and transfer learning strategy, Measurement 178 (2021) 109377.
[30] Q. Pan, L. Zhang, G. Dai, H. Zhang, Two denoising methods by wavelet transform, IEEE transactions on signal processing 47(12) (1999) 3401-3406.
[31] S.W. Katicha, A. Loulizi, J.E. Khoury, G.W. Flintsch, Adaptive false discovery rate for wavelet denoising of pavement continuous deflection measurements, Journal of computing in civil engineering 31(2) (2017) 04016049.
[32] J. Baili, S. Lahouar, M. Hergli, I.L. Al-Qadi, K. Besbes, GPR signal de-noising by discrete wavelet transform, Ndt & E International 42(8) (2009) 696-703.
[33] S. Guoxiang, Z. Ruizhen, Three novel models of threshold estimator for wavelet coefficients, International Conference on Wavelet Analysis and Its Applications, Springer, 2001, pp. 145-150.
[34] I.M. Johnstone, B.W. Silverman, Wavelet threshold estimators for data with correlated noise, Journal of the royal statistical society: series B (statistical methodology) 59(2) (1997) 319-351.
[35] G. Lee, R. Gommers, F. Waselewski, K. Wohlfahrt, A. O'Leary, PyWavelets: A Python package for wavelet analysis, Journal of Open Source Software 4(36) (2019) 1237.
[36] X. An, J. Yang, Denoising of hydropower unit vibration signal based on variational mode decomposition and approximate entropy, Transactions of the Institute of Measurement and Control 38(3) (2016) 282-292.
[37] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, 2009 IEEE conference on computer vision and pattern recognition, Ieee, 2009, pp. 248-255.

[38] M. Tan, Q. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, International Conference on Machine Learning, PMLR, 2019, pp. 6105-6114.

[39] L. Melas-Kyriazi, EfficientNet PyTorch, 2021. https://github.com/lukemelas/EfficientNet-PyTorch.

[40] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, Pytorch: An imperative style, high-performance deep learning library, Advances in neural information processing systems 32 (2019) 8026-8037.

[41] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, Scikit-learn: Machine learning in Python, the Journal of machine Learning research 12 (2011) 2825-2830.

[42] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).

[43] A. Fernández, S. García, M. Galar, R.C. Prati, B. Krawczyk, F. Herrera, Learning from imbalanced data sets, Springer2018.

[44] J. Brownlee, Imbalanced classification with Python: better metrics, balance skewed classes, cost-sensitive learning, Machine Learning Mastery2020.

[45] M. Liu, S. Liao, Y. Yang, Y. Men, J. He, Y. Huang, Tunnel boring machine vibration-based deep learning for the ground identification of working faces, Journal of Rock Mechanics and Geotechnical Engineering 13(6) (2021) 1340-1357.

[46] M. Rautela, S. Gopalakrishnan, Ultrasonic guided wave based structural damage detection and localization using model assisted convolutional and recurrent neural networks, Expert Systems with Applications 167 (2021) 114189.

Chapter 6 PI-LSTM: PHYSICS-INFORMED LONG SHORT-TERM MEMORY NETWORK FOR DATA-DRIVEN STRUCTURAL RESPONSE MODELING[1]

6.1 Introduction

Infrastructures, including buildings, pavements, bridges, and tunnels, play an important role in public life. They are closely related to people's living standards and life safety. Natural disasters are a huge challenge to infrastructure. Numerous nature disasters, such as earthquakes, could significantly reduce the service life of infrastructure, and sometimes destroy it. This highlights the need for the safe operation of infrastructure. With this goal in mind, structural health monitoring has been developed to sense, detect, and evaluate the safety and health of infrastructure [1, 2]. Modeling the structural response plays a fundamental role in structural health monitoring. The current structural response modeling methods mainly include the analytical method and physics-based method. For the analytical method, some time series modeling methods have been applied to modeling structural response, including the autoregressive model [3], autoregressive moving average model [4], and autoregressive integrated moving average model [5]. In addition, the time history analysis method has also been utilized in modeling structure response based on a structural mechanics model [6, 7]. Lu et al. [6] built an open-source workflow for urban building seismic damage simulation using time history analysis and multiple-degree-of-freedom models. However, some issues still exist in these analytical methods in terms of accuracy [8]. For the physics-based method, the finite element method (FEM) is one of the most common and popular methods and has been widely applied in modeling structural response. Zheng et al. [9] utilized FEM to simulate the small deformation of a three-story reinforced concrete frame, while they proposed a hybrid framework that combines FEM and physics engine to simulate the whole process of building collapse. Wu et al. [10] applied FEM to investigate the seismic performance of a typical two-story, three-span Shanghai Metro station in soft soil and validated their finite element models with the experimental results. Meng and Lui [11] employed FEM to analyze and study the seimic response of skew reinforced concrete box girder bridge and investigated the effects of external forces on the superstructure, boundary conditions on the substructure, inclination and stiffness eccentricity of the structure. Although the finite element model is of high fidelity and accuracy, FEM usually requires high computational cost and its accuracy is susceptible to the settings of strucutal materials, structural boundary conditions, and interaction between substructures. Therefore, how to effectiently and reliably model structural response, especially under extreme loads, remains a challenge.

Recently, with the explosive development of technologies, the data-drive method has started to be applied to structural response modeling. Deep learning has proven to be a powerful and efficient tool for structural response modeling [1, 12-14]. Zhang et al. [12] proposed two schemes of the long-short term memory

[1] Part of this chapter is under review as journal papers:

Fangyu Liu, Junlin Li, and Linbing Wang. PI-LSTM: Physics-informed Long Short-Term Memory Network for Data-driven Structural Response Modeling

(LSTM) network for nonlinear structural response modeling and prediction in conjunction with experimental results of a six-story instrumented building. Huang and Chen [13] adopted the one-dimensional convolutional neural network (1D-CNN) and LSTM network to model the seismic response of a two-story, three-span metro station and also compared the results of 1D-CNN and LSTM with the baseline multi-layer perceptron (MLP) model. Li et al. [14] developed an artificial neural network (ANN) framework for modeling dynamic responses of the vehicle-bridge interaction system which consists of a 3D train vehicle model, a bridge finite element model, and a wheel-rail contact model. Li et al. [1] proposed a new modeling paradigm with 1D-CNN and the gated recurrent unit (GRU) for dam structural response modeling while validating their paradigm on a high arch dam with three monitoring items (dam displacement, crack opening displacement, and seepage). Although these works achieved very high accuracy, they strongly relied on the hypothesis that the data is sufficient to train the deep learning model. However, it is very time-consuming, expensive, and difficult to collect a large amount of data, while a purely data-driven model might not meet physics constraints. One potential solution to these problems is to integrate scientific knowledge into deep learning [8, 15, 16]. It will take full advantage of the data- and knowledge-driven approaches. Several studies has utilized different strategies to incorporate knowledge into the deep learning model. For instance, the algebraic equation or logic rule has been added to the loss function as the constraint for training the deep leanring model [17, 18]. The knowledge graph has been introduced to the deep learning model for providing information about relationships between instances [19], especially in the image classification task [20]. The physical simulation has been used to enrich the training data [21].

Therefore, this work proposes a physics-informed long short-term memory (PI-LSTM) network for data-driven structural response modeling. The key contributions of this work are listed as follows:

- The physics-informed long short-term memory network was proposed for the structural response modeling by incorporating scientific knowledge into deep learning.

- Four cases of the single-degree-of-freedom system under the ground motion were used to investigate and make a comparison between the proposed PI-LSTM network and the regular LSTM. The numerical analysis of the six-story building under different earthquake ground motions was also applied to investigate the performance of the proposed PI-LSTM network.

- The proposed PI-LSTM network was compared and validated with the existing physics-informed deep learning model (PhyCNN) in terms of the numerical simulation results for the single-degree-of-freedom system and the experimental results for the six-story building.

6.2 Physics-informed long short-term memory (PI-LSTM)

Since the structural response is usually the time series data, the structural response modeling could be regarded as the sequence prediction task. Numerous studies have proven that LSTM is an effective and

reliable tool for the sequence prediction task [22, 23]. This is because LSTM could learn long-distance information by storing the important information from the past and forgetting the unimportant or irrelevant information [24, 25]. The traditional training strategy for the deep learning model is purely based on data. Integrating scientific knowledge as the physics constraint into the training process could improve the robustness and reliability of the deep learning model to learn from data [8]. Therefore, a physics-informed long short-term memory network is proposed in this work for structural response modeling.

Let's look back to the fundamental equation (Eq. (6-1)) for structural dynamics based on the single-degree-of-freedom (SDOF) system.

$$m\ddot{x} + c\dot{x} + kx = F_E \tag{6-1}$$

where m, c, and k are the mass, damping coefficient, stiffness; x, $\dot{x}$, and $\ddot{x}$ are the displacement, velocity, and acceleration; F_E is the external force.

When modeling the structural response, the ground motion is one of the most prevalent cases. When infrastructure is subjected to the ground motion, the equation of motion for the SDOF system is written as Eq. (6-2) or (6-3).

$$m\ddot{x} + c\dot{x} + kx = -m\ddot{x}_g \tag{6-2}$$

$$\ddot{x} + 2\xi\omega\dot{x} + \omega^2 x = -\ddot{x}_g, \ \omega = \sqrt{\frac{k}{m}} \text{ and } \xi = \frac{c}{2m\omega} \tag{6-3}$$

where x, $\dot{x}$, and $\ddot{x}$ are the relative displacement, velocity, and acceleration with respect to ground/support; $\ddot{x}_g$ is the ground acceleration; ω is the structural frequency; ξ is the damping ratio.

The physics-informed long short-term memory network is proposed for the structural response modeling under the ground motion. Figure 6-1 shows the scheme of the physics-informed long short-term memory network. When the deep learning model is applied for modeling the structural response under the ground motion, the input is usually the ground acceleration [8, 13, 26]. However, the current structural response is not only related to the current load, but also to the previous structural response. It is necessary to take into account the previous structural response as one of the inputs. Therefore, both the ground acceleration ($\ddot{x}_g$) and previous structural response (z) are considered as the input, while the previous structural response (z) will vary depending on the different design or measurement indicators. For example, the displacement and acceleration of buildings on the ground are easily measured separately by the displacement sensor and accelerometer during the shake table test or field monitoring [27, 28]. However, the displacement is sometimes more difficult to measure directly, so researchers or engineers obtain the displacement by integrating the measured acceleration for monitoring the seismic response of the underground structure [10, 29]. Therefore, the previous structural response (z) might be both the displacement and acceleration or just one of these two indicators. The output is the structural response, which has the same indicators as the previous structural response. This will be discussed in detail in Section 3.2. After getting the input and

output, LSTM is built to establish the relationship between the input and output.

The equation of motion for the structural dynamics (i.e. scientific knowledge) is introduced as the physics constraint. This physics constraint contains the output structural response and the input ground acceleration. However, as shown in Eq. (6-2) or (6-3), three indicators (displacement, velocity, and acceleration) are needed at the same time, which is quite difficult for the experiment or field monitoring. Therefore, the differential method (e.g., the central finite difference method) is used to obtain the unknown index of the structural response. For example, when the output structural response includes the displacement and acceleration, the velocity is obtained from the derivative of the predicted displacement. Then the output structural response (e.g., displacement and acceleration) and the structural response (e.g., velocity) obtained by the differential method as well as the input ground acceleration are introduced into the physics constraint.

The total loss contains two components, including the data loss and physics loss. The data loss is measured by the difference between the prediction and ground truth, which is the same for LSTM applied to the time series prediction task. The physics loss is measured from the physics constraint (e.g., the equation of motion for the structural dynamics). Since the loss function is only utilized when training the deep learning model, the physics loss could regularize the deep learning model, which helps to eliminate the overfitting issue, reduce the requirement for large amounts of training data, and improve the robustness of the deep learning model. Both the data loss and physics loss adopt the mean square loss (i.e., MSELoss) which is common in the time series forecasting task.

Based on the aforementioned concepts, the physics-informed long short-term memory network is built for the structural response modeling under the ground motion. However, this PI-LSTM network could also be applied to other conditions for structural response modeling. It just needs to change the physics constraint and sometimes the target structural response.

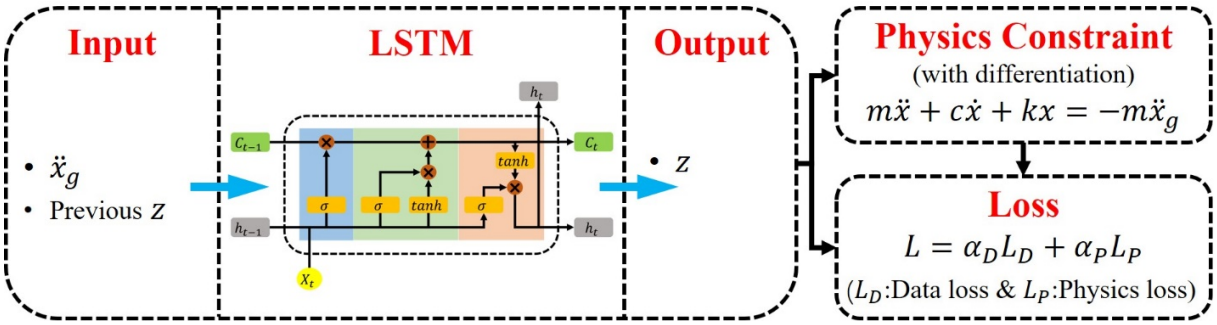

Figure 6-1 Scheme of physics-informed long short-term memory (PI-LSTM) network. The input includes the ground acceleration ($\ddot{x}_g$) and the previous structural response (z). The output is the structural response (z). The total loss consists of the data loss from the difference between the prediction and ground truth and the physics loss from the physics constraint. Both α_D and α_P are set to 1 in this work.

6.3 SDOF numerical validation

The single-degree-of-freedom system is the typical and fundamental object for understanding and studying structural dynamics. Therefore, the SDOF system under the ground motion is firstly applied to investigate the validity and usefulness of the proposed PI-LSTM network. The data of the SDOF system under the ground motion is obtained from the previous literature [8] and numerical analysis by the finite element method.

6.3.1 Comparison with the existing model

Zhang et al. [8] proposed a physics-guided convolutional neural network (PhyCNN) for the structural response modeling under the ground motion and found their proposed PhyCNN outperformed the non-physics-guided neural network. To verify the validity of the PI-LSTM network proposed in this work, the SDOF system results from their work were utilized to make a comparison with the PI-LSTM network. In their work, 100 independent seismic sequences selected from the PEER strong-motion database [30] were fed into the SDOF system to obtain the numerical results. Each simulation result was up to 50s with a frequency of 20Hz, namely 1001 data points per simulation result. Ten samples were randomly selected from all 100 samples as the training data and the other 90 samples were regarded as the test data to evaluate the performance of the model. Since they published their data, codes, and results, it is easier to make a fair comparsion by simply replacing their PhyCNN model with the proposed PI-LSTM network.

Table 6-1 summarizes the information for the comparison between PhyCNN and PI-LSTM. In their work, the input was the ground acceleration and the output was displacement, velocity, and restoring force. For the physical loss, they considered the equation of motion as well as the difference between the predicted velocity and the derivative of the predicted displacement. Figure 6-2 shows the performance of PhyCNN and PI-LSTM for predicting displacement, acceleration, and restoring force. In general, the accuracy of PI-LSTM was much higher than that of PhyCNN for all three indicators. The accuracy of PhyCNN varied over a larger area and the mean accuracy of the test data (90 samples) was lower than the median accuracy for all three indicators. In comparison, the accuracy of PI-LSTM was concentrated on a smaller range and its mean and median accuracy were much closer than that of PhyCNN for all three indicators. From the results of the test data, the proposed PI-LSTM network in this work had much higher accuracy than the PhyCNN network. The distribution of results from the test data showed that the proposed PI-LSTM network in this work was more robust and stable than the PhyCNN model.

Table 6-1 Summary of the comparison between PhyCNN and PI-LSTM

	PhyCNN	**PI-LSTM**
Input	$\ddot{x}_g$	$\ddot{x}_g$ & Previous $\{x, \dot{x}, F_{rs}\}$
Output	$x, \dot{x}, F_{rs}$	$x, \dot{x}, F_{rs}$
Data Loss (L_D)	$\frac{1}{N}\left(\|x^p - x^m\|_2^2 + \|\dot{x}^p - \dot{x}^m\|_2^2 + \left\|F_{rs}^p - F_{rs}^m\right\|_2^2\right)$	

Physics Loss (L_P)	$\frac{1}{N}\left(\left\|\dot{x}^p - \dot{x}_t^p\right\|_2^2 + \left\|\ddot{x}_t^p + F_{rs}^p + \ddot{x}_g\right\|_2^2\right)$
Total Loss (L)	$L = L_D + L_P$

Note: (1) F_{rs} is the mass-normalized restoring force, (2) the superscript p and m denote the prediction and ground truth, respectively, (3) the subscript t denotes the derivative of the prediction. E.g. $\dot{x}_t^p = \frac{dx^p}{dt}$, and (4) $\|u\|_2^2 = \left(\sqrt{\sum_i x_i^2}\right)^2 = \sum_i x_i^2$.

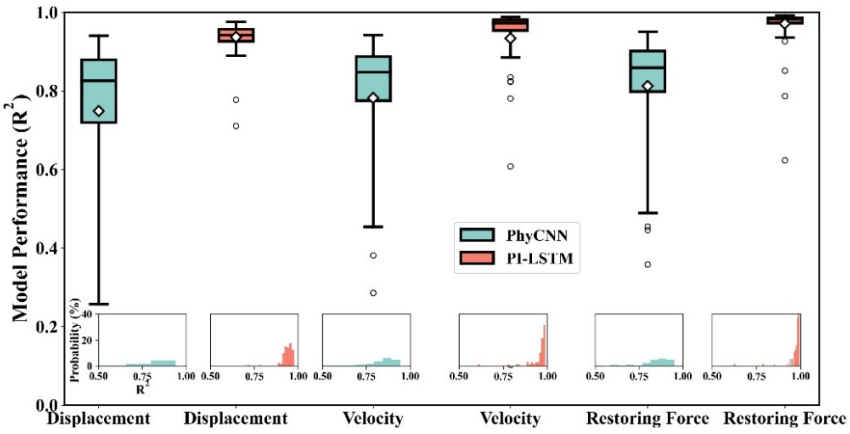

Figure 6-2 Performance of PhyCNN and PI-LSTM for predicting displacement, velocity, and restoring force.

6.3.2 Case study

One problem with the work of Zhang et al. [8] is that their SDOF system was that their SDOF system was more from a mathematical perspective rather than from an engineering application perspective. This is because, in their SDOF system, the mass is 1 kg, the damping coefficient is 1 Ns/m, and the linear stiffness is 20 N/m. This might raise some concerns about the validity of the numerical results. Therefore, the SDOF system with the parameters of a one-story concrete building was utilized to build the dataset. The same earthquake records in the work of Zhang et al. [8] were adopted. In the SDOF system, the mass is 100 tons, the period is 0.5s (i.e., the stiffness is 15791367 N/m), and the damping ratio is 0.05 (i.e., the damping coefficient is 125664 Ns/m). The numerical analysis of the SDOF system under the ground motion was based on OpenSees [31]. The one sample of the ground acceleration, structural displacement, structural velocity, and structural acceleration is shown in Figure 6-3. As mentioned above, there were 100 samples of earthquake records. Ten samples were randomly selected from all 100 samples as the training data and the remaining 90 samples were used to test the model performance. Different cases of the output will be utilized to investigate the validity of the proposed PI-LSTM network. In all cases, the training and test earthquake records were the same. Since the proposed PI-LSTM network has been compared with the existing model, the regular LSTM without the physics loss (i.e., LSTM) is employed to compare with the proposed PI-LSTM network in this section.

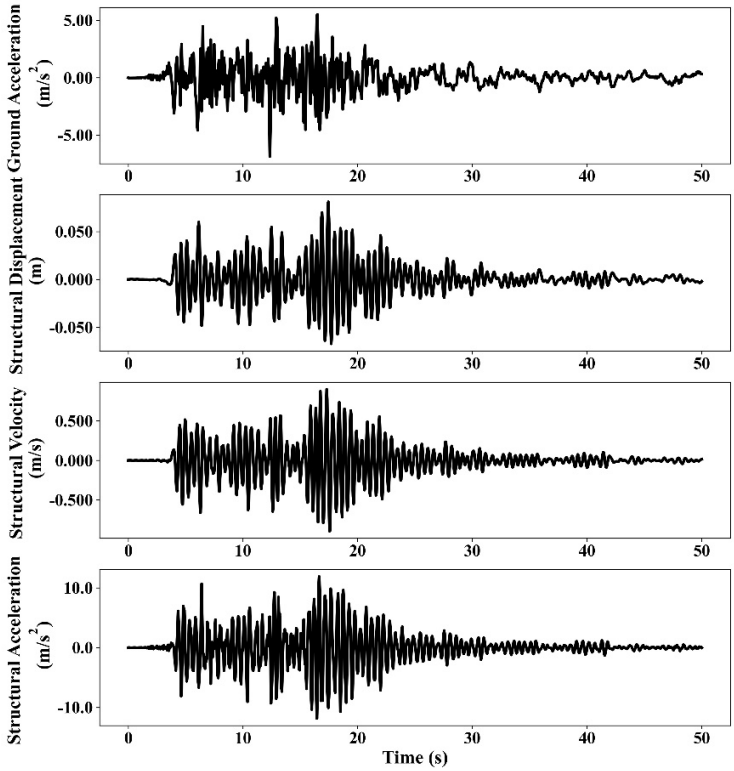

Figure 6-3 Data for the ground acceleration, structural displacement, structural velocity, and structural acceleration

6.3.2.1 Case 1: Predicting x, $\ddot{x}$

Displacement and acceleration are the common indicators to monitor when conducting the shake table test or field monitoring [27, 28]. Therefore, the displacement and acceleration will be regarded as the output structural response. Table 6-2 summarizes the information for the comparison between LSTM and PI-LSTM. The input of PI-LSTM was the ground acceleration and previous structural response (i.e., displacement and acceleration) and the output of PI-LSTM was displacement and acceleration. The total loss of PI-LSTM was the sum of the data loss and physics loss, while only the data loss was the total loss of LSTM. Figure 6-4 summarizes the results of LSTM and PI-LSTM for predicting displacement and acceleration. Overall, PI-LSTM had higher accuracy than LSTM. For the accuracy distribution of the test data, PI-LSTM was more robust and stable than LSTM because the accuracy of PI-LSTM was concentrated on a smaller range.

Table 6-2 Summary of the comparison between LSTM and PI-LSTM for predicting displacement and acceleration

	LSTM	**PI-LSTM**
Input	$\ddot{x}_g$ & Previous $\{x, \ddot{x}\}$	
Output	$x, \ddot{x}$	
Data Loss (L_D)	$\frac{1}{N}(\|\|x^p - x^m\|\|_2^2 + \|\|\ddot{x}^p - \ddot{x}^m\|\|_2^2)$	
Physics Loss (L_P)	-	$\frac{1}{N}\left(\left\|\ddot{x}^p + 2\xi\omega\dot{x}_t^p + \omega^2 x^p + \ddot{x}_g\right\|_2^2\right)$
Total Loss (L)	$L = L_D$	$L = L_D + L_P$

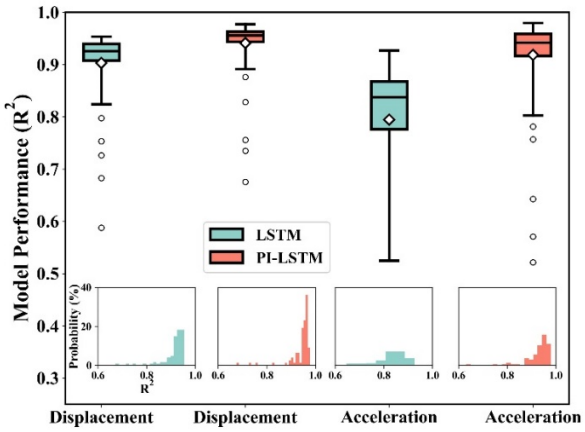

Figure 6-4 Performance of LSTM and PI-LSTM for predicting displacement and acceleration

6.3.2.2 Case 2: Predicting x

The displacement is a good indicator of structural condition and descriptor of the structural formation behavior. Since the accelerometer usually has a lot of noise, it is important to measure the displacement when monitoring the structural condition of a building on the ground. Therefore, the case of predicting only the displacement is applied to investigate the performance of the proposed PI-LSTM network. Table 6-3 summarizes the comparison between LSTM and PI-LSTM for predicting displacement. The input of PI-LSTM was the ground acceleration and previous structural response (i.e., displacement) and the output was the displacement. For the physics loss, both velocity and acceleration were obtained by deriving the displacement. Figure 6-5 shows the results of LSTM and PI-LSTM for predicting displacement. The predicted acceleration was obtained from the second-order derivative of the predicted displacement. This predicted acceleration was then used to calculate the accuracy of the deep learning model for predicting the acceleration by comparing it with the ground truth (acceleration obtained from the numerical results). PI-LSTM had higher accuracy in predicting displacement and performed more robustly and stably than LSTM. After calculating the acceleration from the predicted displacement, the predicted acceleration of PI-LSTM was still closer to the ground truth than that of LSTM.

Table 6-3 Summary of the comparison between LSTM and PI-LSTM for predicting displacement

	LSTM	PI-LSTM
Input	$\ddot{x}_g$ & Previous $\{x\}$	
Output	x	
Data Loss (L_D)	$\frac{1}{N}(\|x^p - x^m\|_2^2)$	
Physics Loss (L_P)	-	$\frac{1}{N}\left(\left\|\ddot{x}_t^p + 2\xi\omega\dot{x}_t^p + \omega^2 x^p + \ddot{x}_g\right\|_2^2\right)$
Total Loss (L)	$L = L_D$	$L = L_D + L_P$

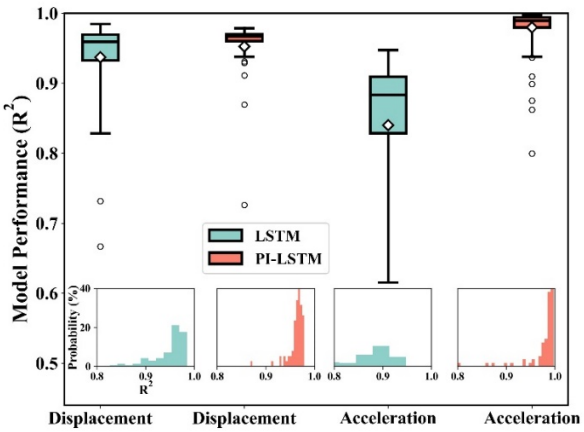

Figure 6-5 Performance of LSTM and PI-LSTM for predicting displacement

6.3.2.3 Case 3: Predicting $\ddot{x}$

Acceleration is the most commonly used in structural health monitoring because it is relatively easy to measure. Sometimes, the monitoring conditions only allow researchers or engineers to measure the acceleration when conducting the shake table test or field monitoring [10, 29]. Therefore, it needs to investigate the performance of the deep learning model that predicts only acceleration. When only the accelerometer is installed on infrastructure, it needs double integration of the acceleration to obtain the displacement. However, there are some problems with the double integration due to the difficulties in the definition of the initial or boundary condition as well as the noise [8, 32]. This raises concerns about predicting the displacement based on the predicted acceleration. Therefore, the displacement is considered as the output of the deep learning model and then the predicted acceleration is obtained from the second-order derivative of the predicted displacement. This predicted acceleration is used to compare with the ground truth (i.e., the acceleration obtained from the numerical results) to train the deep learning model. Table 6-4 summarizes the comparison between LSTM and PI-LSTM for predicting acceleration. As mentioned above, the input of PI-LSTM was the ground acceleration and previous structural response (i.e., displacement) and the output of PI-LSTM was the displacement. The data loss was calculated from the difference between the predicted acceleration (i.e., the second-order derivative of the model output) and the

ground truth (i.e., the acceleration obtained from the numerical results). In the physics loss, both the acceleration and velocity were obtained by deriving the model output. Figure 6-6 summarizes the performance of LSTM and PI-LSTM for predicting acceleration. The accuracy of PI-LSTM was higher and more stable than that of LSTM. The accuracy distribution of the test data shows that PI-LSTM was better and more robust than LSTM.

Table 6-4 Summary of the comparison between LSTM and PI-LSTM for predicting acceleration

	LSTM	**PI-LSTM**
Input	$\ddot{x}_g$ & Previous $\{x\}$	
Output	x	
Data Loss (L_D)	$\frac{1}{N}\left(\left\|\ddot{x}_t^p - \ddot{x}^m\right\|_2^2\right)$	
Physics Loss (L_P)	-	$\frac{1}{N}\left(\left\|\ddot{x}_t^p + 2\xi\omega\dot{x}_t^p + \omega^2 x^p + \ddot{x}_g\right\|_2^2\right)$
Total Loss (L)	$L = L_D$	$L = L_D + L_P$

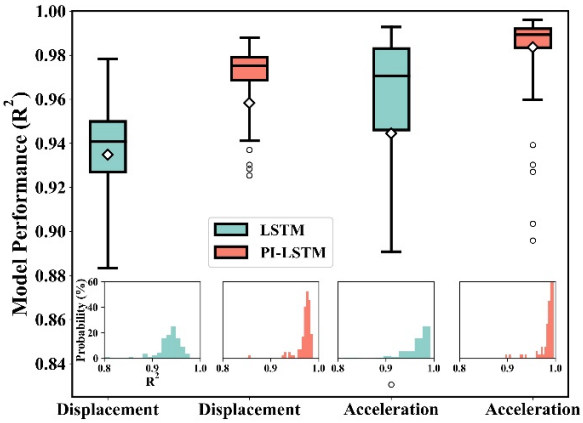

Figure 6-6 Performance of LSTM and PI-LSTM for predicting acceleration

6.3.2.4 Case 4: Predicting $\ddot{x}$ (non-linear analysis)

Since the stiffness of the structure changes when its deformation changes, it is difficult to apply the constant parameters (e.g., stiffness) to describe the motion of structures for the non-linear analysis. This means it is difficult to use the equation of motion (Eq. (6-2) or (6-3)) as the physics constraint for the proposed PI-LSTM. One possible solution is to combine the stiffness force and damping force as the restoring force. The equation of motion (Eq. (6-2) or (6-3)) is re-written as follows:

$$\ddot{x} + F_{rs} = -\ddot{x}_g \tag{6-4}$$

where F_{rs} is the mass-normalized restoring force

This equation (Eq.(6-4)) will be used as the physics constraint for the proposed PI-LSTM. Figure 6-7 shows the modified PI-LSTM network for non-linear analysis. To investigate the performance of this modified PI-LSTM network, one case is selected to make a comparison between this modified PI-LSTM network and LSTM. As mentioned above, acceleration is the most common indicator to monitor the structural condition. Hence, PI-LSTM and LSTM were compared to predict acceleration. The dataset was the same as the other cases, but the mass-normalized restoring force was obtained from the ground acceleration and structural acceleration, namely $F_{rs} = -\ddot{x}_g - \ddot{x}$, to build the dataset.

Table 6-5 summarizes the comparison between LSTM and PI-LSTM for predicting the acceleration in terms of the non-linear analysis. The input of LSTM and PI-LSTM was the ground acceleration and previous structural response (displacement and mass-normalized restoring force), while the output of LSTM and PI-LSTM was the displacement and mass-normalized restoring force. As mentioned in Section 6.3.2.3 (Case 3), the displacement was regarded as the model output to predict acceleration, reducing the errors. The data loss was calculated from the difference between the prediction (acceleration and mass-normalized restoring force) and the ground truth (numerical results) and the physics loss was computed from the physics constraint. Figure 6-8 shows the results of LSTM and PI-LSTM for predicting acceleration (non-linear analysis). PI-LSTM had higher accuracy than LSTM. The accuracy distribution of the test data for PI-LSTM was concentrated on a smaller range compared to that of LSTM in terms of displacement and acceleration.

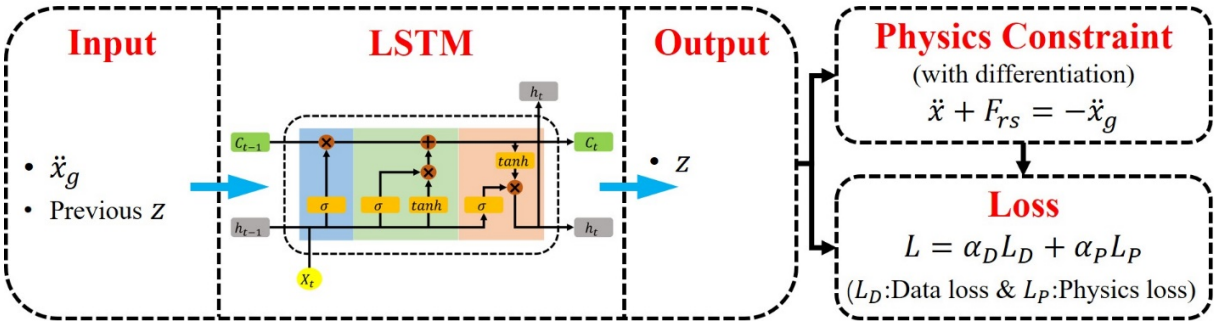

Figure 6-7 Scheme of the modified PI-LSTM network for non-linear analysis.

Table 6-5 Summary of the comparison between LSTM and PI-LSTM for predicting acceleration (non-linear analysis)

	LSTM	**PI-LSTM**
Input	$\ddot{x}_g$ & Previous $\{x, F_{rs}\}$	
Output	x, F_{rs}	
Data Loss (L_D)	$\frac{1}{N}\left(\left\|\ddot{x}_t^p - \ddot{x}^m\right\|_2^2 + \left\|F_{rs}^p - F_{rs}^m\right\|_2^2\right)$	
Physics Loss (L_P)	-	$\frac{1}{N}\left(\left\|\ddot{x}_t^p + F_{rs}^p + \ddot{x}_g\right\|_2^2\right)$

Total Loss (L)	$L = L_D$	$L = L_D + L_P$

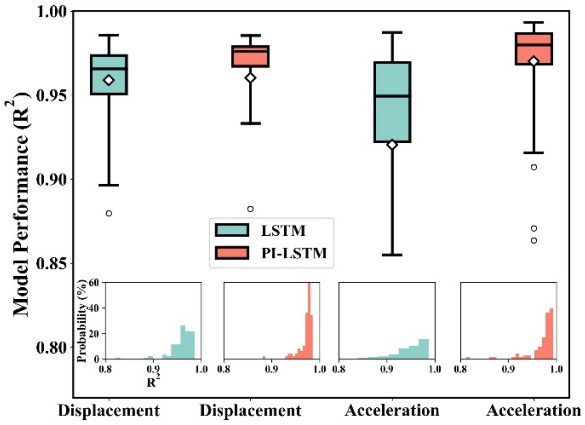

Figure 6-8 Performance of LSTM and PI-LSTM for predicting acceleration (non-linear analysis)

6.4 Experimental and numeral validation

After verifying the performance of the proposed PI-LSTM network on the SDOF system, it is also necessary to investigate its performance on experimental results. Since few benchmark experimental results are publicly available, numerical analysis (FEM) is also a good alternative. It needs to first build the finite element model based on experimental conditions, such as the parameters of buildings and boundary conditions. Then it calculates the structural response of the aforementioned finite element model under experimental loading. If the simulation results are similar to the experimental results, it will be confident to investigate this finite element model under other different conditions, such as different ground motions. Therefore, the experimental and numerical results were used to investigate the performance of the proposed PI-LSTM network for a multi-story building under the ground motion.

6.4.1 Experimental validation

Zhang et al. [8] utilized the experimental results of a 6-story hotel building in San Bernardino, California (CA), to investigate the performance of their proposed PhyCNN model. This building was designed in 1970 and nine accelerometers were installed on the first and third floors and the roof of the building. Twenty-one available records were selected from the Center for Engineering Strong Motion Data (CESMD) [33] to build the dataset. Figure 6-9 shows one sample of the ground acceleration, structural displacement, structural velocity, and structural acceleration. Eleven samples were regarded as the training data, four samples were selected as the evaluation data, and the remaining six samples were chosen as the test data.

Since only accelerometers were installed on the building, the target output of the deep learning model was acceleration. As mentioned above, the double integration might cause some unavoidable errors in obtaining the displacement from the acceleration. Therefore, similar to Section 6.3.2.4, the output of the deep learning model was considered as the displacement and then the predicted acceleration was obtained from the

second-order derivative of the predicted displacement. However, unlike cases illustrated in Section 6.3.2.4, Zhang et al. [8] did not introduce the equation of motion as the physics constraint for the deep learning model. In their proposed PhyCNN model, the output was the displacement and then the predicted acceleration was obtained from the second-order derivative of the predicted displacement. The loss was only obtained from the difference between the predicted acceleration and ground truth. In this work, the proposed PI-LSTM network could be applied for the non-linear analysis of the structural response (shown in Section 3.2.4). Since these experimental results could be categorized as the non-linear analysis of the structural response, the proposed PI-LSTM network shown in Figure 6-7 could be used for structural response modeling in terms of these experimental results. Table 6-6 summarizes the comparison between PhyCNN and PI-LSTM for predicting acceleration, aiming at modeling the experiment results. The input of PI-LSTM was the ground acceleration and previous structural response (displacement and mass-normalized restoring force) and the input of PhyCNN was the ground acceleration. The output of PI-LSTM was the displacement and mass-normalized restoring force and the output of PhyCNN was only the displacement. The loss of PI-LSTM consisted of two components, including the data loss obtained from the prediction and ground truth and the physics loss obtained from the physics constraint, while the loss of PhyCNN was only calculated from the difference between the predicted acceleration and the experimental results.

Figure 6-10 shows the performance of PhyCNN and PI-LSTM for predicting the acceleration in terms of the third floor and roof, while Figure 6-11 shows their corresponding predicted displacement and acceleration for the third floor and roof. For the third floor, although PI-LSTM and PhyCNN had similar accuracy for predicting displacement, the acceleration accuracy of PI-LSTM was much higher than that of PhyCNN. For the roof, the accuracy of PI-LSTM was higher than that of PhyCNN for both displacement and acceleration, especially PI-LSTM was much more accurate than PhyCNN for acceleration. Since the acceleration was the monitoring indicator of the building for the experiment, PI-LSTM could be a better choice for structural response modeling than PhyCNN for this experiment.

According to Figure 6-11, the structural response (including displacement and acceleration) of the roof was higher than that of the third floor. PhyCNN and PI-LSTM had similar performance on the displacement of the third floor (Figure 6-10a). When the displacement increased as the monitoring location changed from the third floor to the roof, PI-LSTM had a better performance than PhyCNN on the displacement of the roof (Figure 6-10b). The superiority of PI-LSTM became more obvious when the indicator became the acceleration. For the acceleration of the third floor, PI-LSTM was much more accurate than PhyCNN (Figure 6-10a), although the accuracy of PhyCNN was satisfactory (R^2=0.86). When the monitoring location was changed to the roof, the acceleration accuracy of PhyCNN dropped to 0.71, but PI-LSTM still maintained a high accuracy, even higher than that of the third floor (Figure 6-10b). The gap between the acceleration accuracy of PI-LSTM and PhyCNN was widened when the amplitude and range of the acceleration increased. In addition, for both the displacement and acceleration, the accuracy distribution of PI-LSTM was concentrated on a smaller range than that of PhyCNN. Therefore, it could be concluded that

PI-LSTM is a more accurate and robust approach for structural response modeling than PhyCNN. In particular, when the amplitude and range of the monitoring indicator increase, the advantage of PI-LSTM will become more prominent.

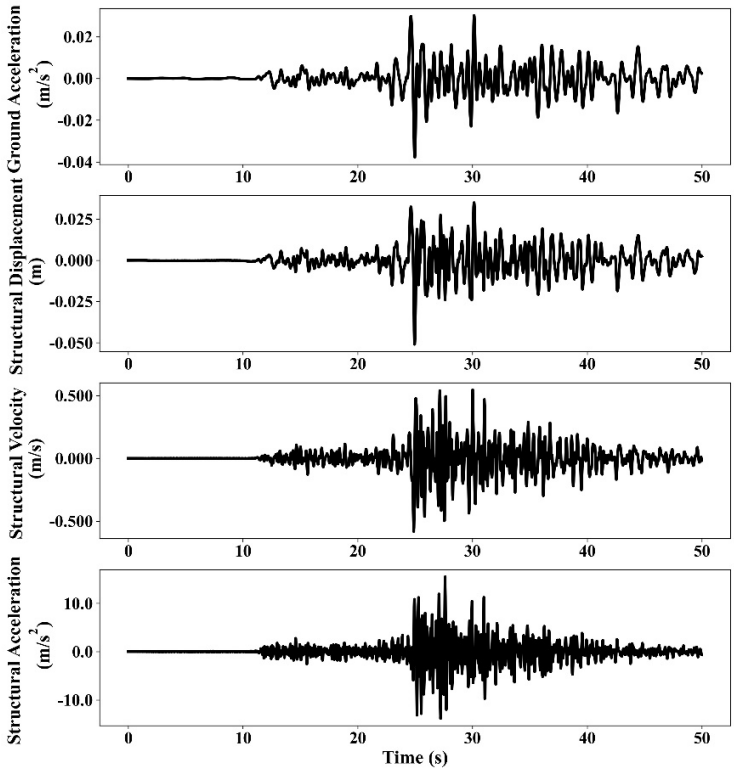

Figure 6-9 Data for the ground acceleration, structural displacement, structural velocity, and structural acceleration

Table 6-6 Summary of the comparison between PhyCNN and PI-LSTM for predicting acceleration

	PhyCNN	**PI-LSTM**
Input	$\ddot{x}_g$	$\ddot{x}_g$ & Previous $\{x, F_{rs}\}$
Output	x	x, F_{rs}
Data Loss (L_D)	$\frac{1}{N}\left(\left\|\ddot{x}_t^p - \ddot{x}^m\right\|_2^2\right)$	$\frac{1}{N}\left(\left\|\ddot{x}_t^p - \ddot{x}^m\right\|_2^2 + \left\|F_{rs}^p - F_{rs}^m\right\|_2^2\right)$
Physics Loss (L_P)	-	$\frac{1}{N}\left(\left\|\ddot{x}_t^p + F_{rs}^p + \ddot{x}_g\right\|_2^2\right)$
Total Loss (L)	$L = L_D$	$L = L_D + L_P$

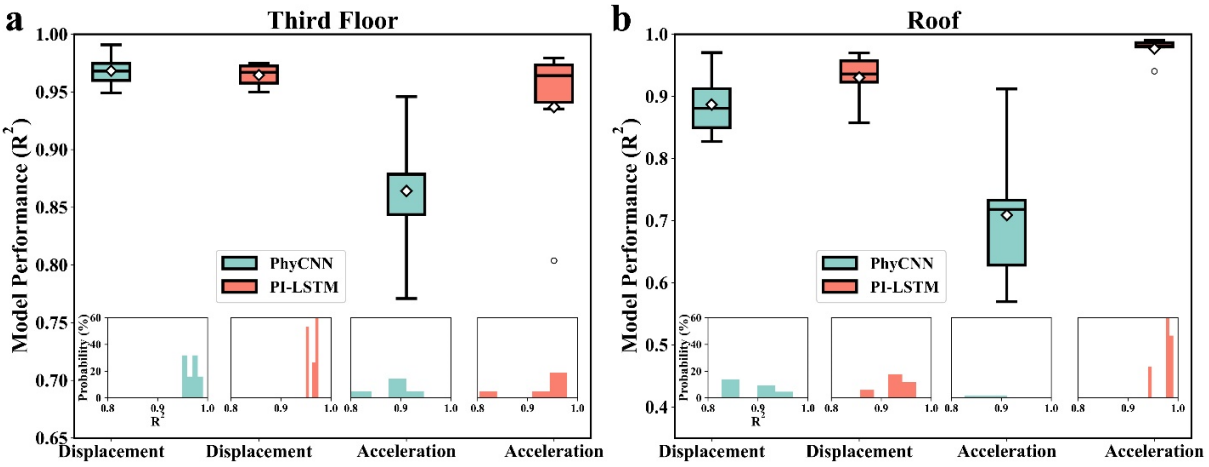

Figure 6-10 Performance of PhyCNN and PI-LSTM for predicting the structural response in terms of the (a) third floor and (b) roof.

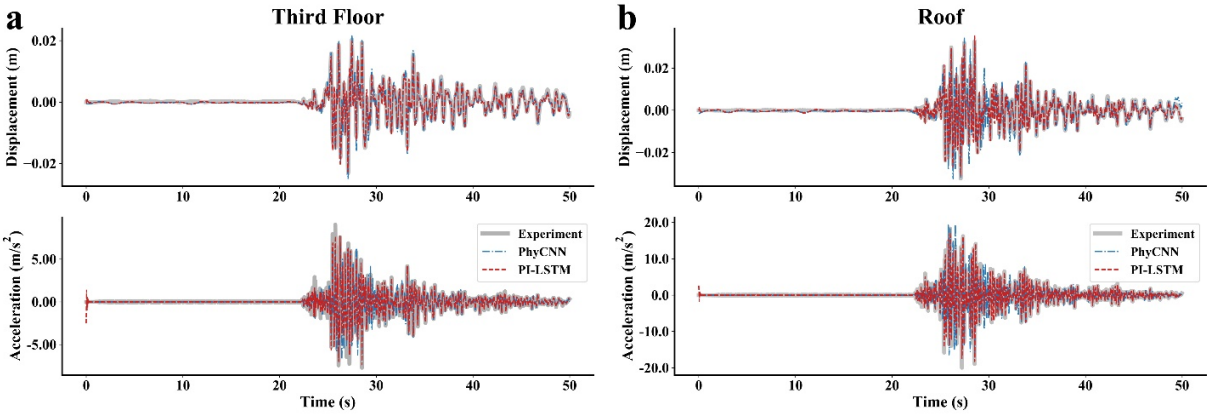

Figure 6-11 Prediction performance of PhyCNN and PI-LSTM in terms of the (a) third floor and (b) roof.

6.4.2 Numerical validation

In addition to the experimental results, the numeral simulation of a six-story building was also utilized to investigate the performance of the proposed PI-LSTM model. According to the previous literature [34], this six-story building was designed in 1976 based on the 1973 UBC requirements [35] and the information about this building was selected by the California Strong Motion Instrumentation Program (CSMIP). This building was 36.60 m by 36.60 m in plan and had six bays in one direction (6 × 6.10 m). The elevation of this building is shown in Figure 6-12, which also includes information on the height of each floor and the cross-section of each column and beam. More details about this building could be found in the previous literature [34]. A total of 13 strong motion sensors were installed on the ground, second, third, and roof levels. This building has recorded response data for three earthquakes, including the 1987 Whittier Narrows earthquake, the 1991 Sierra Madre earthquake, and the 1994 Northridge earthquake.

Based on the previous literature [34], the finite element model of this building was built on OpenSees [31].

The tributary seismic response masses were 235.6 tons per level and the damping ratio was 0.03 for this building. The finite element model of the building had a period of 1.43s, which was very close to the target period in the literature [34].

The popular earthquake ground motions for Los Angeles [36] were selected to simulate the structural response under the ground motion. Since these earthquake ground motions had different probabilities of exceedance and time intervals, a total of 40 earthquake ground motions with the same time interval (0.2s) and various probabilities of exceedance were selected. These earthquake ground motions include LA01-LA02, LA07-LA14, LA17-LA20 (10% probability of exceedance); LA21-LA22, LA27-LA30, LA37-LA40 (2% probability of exceedance); LA45-LA60 (50% probability of exceedance). These earthquake ground motions were input into the aforementioned finite element model to obtain the numerical simulation results for the 6-story building. One sample of the ground acceleration, structural displacement, structural velocity, and structural acceleration is shown in Figure 6-13. Among these 40 samples, 20 samples were randomly selected from each group according to the corresponding proportion as the training data, 10 samples as the validation data, and 10 samples as the test data. For example, 7 samples were randomly selected from the group containing the earthquake ground motion with the 10% probability of exceedance, 5 samples from the group with the 2% probability of exceedance, and 8 samples from the group with the 50% probability of exceedance.

The target output for this numerical validation is acceleration because acceleration was the most common indicator for structural health monitoring. The target floors included the third floor and roof (Figure 6-12). Since this is a 6-story building and it is difficult to directly calculate the parameters for the equation of motion, the proposed PI-LSTM model shown in Figure 6-7 could be used to model the structural response of different floors for the multi-story building. Two models, LSTM and PI-LSTM, were used to make a comparison for modeling the structural response. The information for LSTM and PI-LSTM is summarized in Table 6-5. The input of LSTM and PI-LSTM was the ground acceleration and previous structural response (displacement and mass-normalized restoring force) and the output was the displacement and mass-normalized restoring force. The predicted acceleration was obtained from the second-order derivative of the predicted displacement. The data loss was calculated by the difference between the predicted structural response (acceleration and mass-normalized restoring force) and numerical simulation results, while the physics loss for PI-LSTM was obtained from the physics constraint. Figure 6-14 shows the performance of LSTM and PI-LSTM for predicting the structural response in terms of the third floor and roof. PI-LSTM was significantly more accurate than LSTM for the displacement, while the accuracy of PI-LSTM was slightly higher than that of LSTM in terms of acceleration. Compared to LSTM, the accuracy of PI-LSTM was concentrated on a smaller range for both displacement and acceleration. Figure 6-15 shows the prediction of LSTM and PI-LSTM for predicting the structural response for the third floor and roof. The prediction performance of PI-LSTM was much better in terms of acceleration than displacement. PI-LSTM could predict the trend of the displacement very well, although the amplitude of the predicted displacement of PI-LSTM was slightly lower than the numerical simulation results. Therefore, PI-LSTM

could be an accurate and robust approach for modeling the structural response under the ground motion.

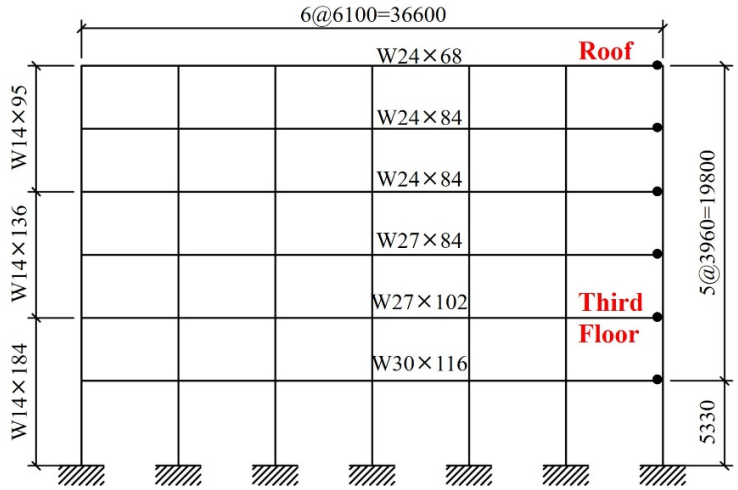

Figure 6-12 Elevation of the six-story building (from the literature [34])

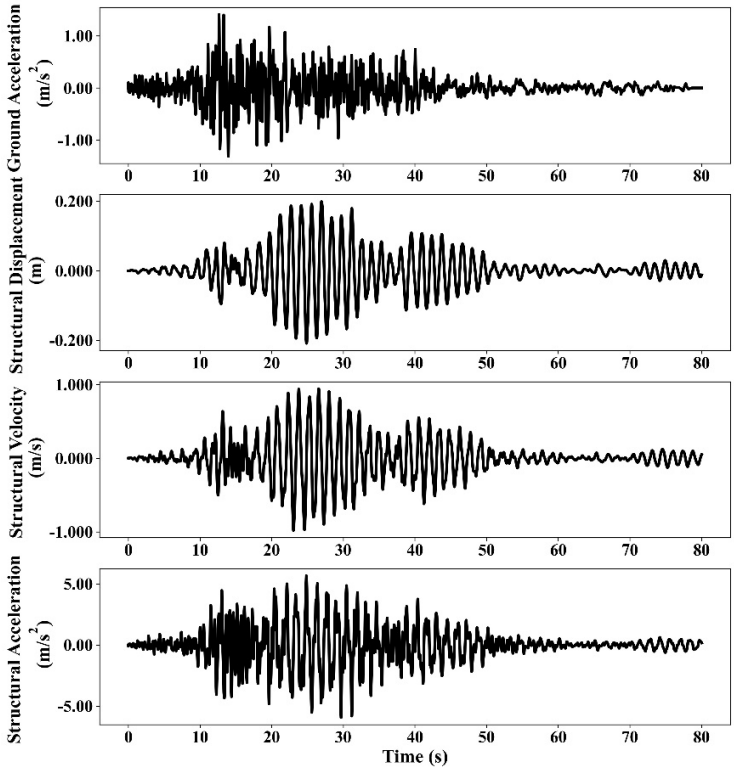

Figure 6-13 Data for the ground acceleration, structural displacement, structural velocity, and structural acceleration

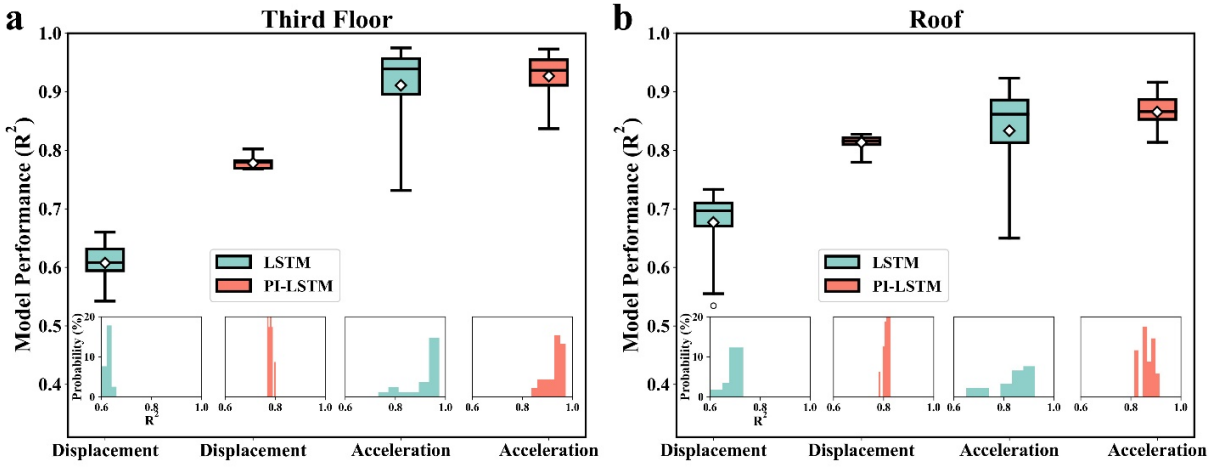

Figure 6-14 Performance of PhyCNN and PI-LSTM for predicting the structural response in terms of the (a) third floor and (b) roof.

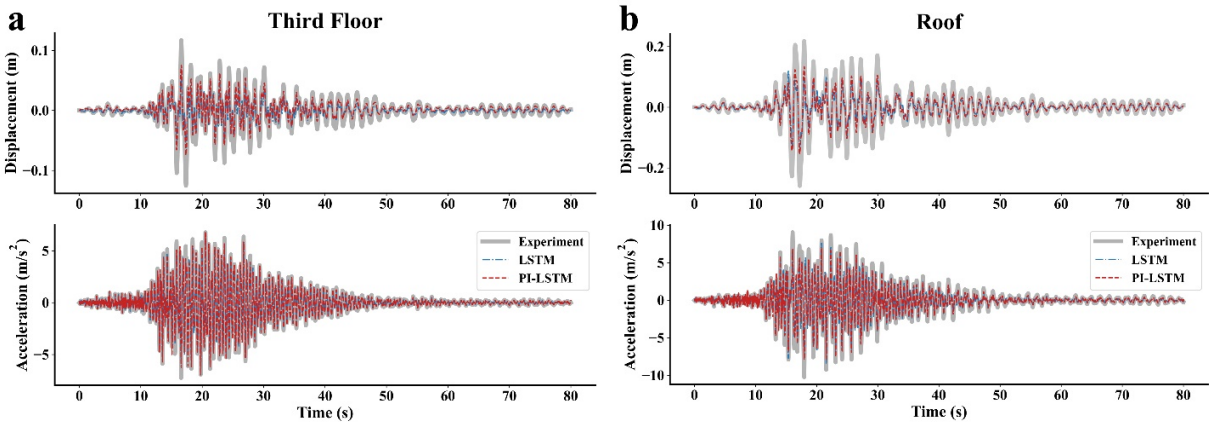

Figure 6-15 Prediction performance of LSTM and PI-LSTM in terms of the (a) third floor and (b) roof.

6.5 Conclusion

Deep learning is emerging as a powerful and efficient tool for structural response modeling. Although the deep learning model achieved very high accuracy, it requires sufficient training data. However, collecting large amounts of data is very time-consuming and difficult and the data-driven model sometimes fails to meet the physics constraint. One possible solution to these problems is to incorporate scientific knowledge into deep learning, namely physics-informed deep learning. Therefore, this work proposed a physics-informed long short-term memory network for data-driven structural response modeling by incorporating scientific knowledge into deep learning. The single-degree-of-freedom system was firstly used to verify the performance of the proposed model. The proposed PI-LSTM network was compared with the existing physics-informed model (PhyCNN) regarding its published results for the SDOF system, while the proposed PI-LSTM network was also compared with the regular LSTM for the four cases of the SDOF system. After that, the performance of the proposed PI-LSTM network was further investigated and

validated in terms of the experimental results of the six-story building using the PhyCNN model and the numerical simulation results of the six-story building using the regular LSTM. The results showed that the accuracy of the proposed PI-LSTM network was higher and concentrated on a smaller range than the PhyCNN model for the numerical simulation results of the SDOF system and experimental results of the six-story building. The proposed PI-LSTM network was also more accurate and robust than the regular LSTM for the numerical simulation results of the SDOF system and the six-story building. The proposed PI-LSTM network could be a reliable and robust approach for structural response modeling. In the future, this model will be applied to model the structural response of different infrastructures, such as tunnels and bridges, under different loads or hazard events.

References

[1] Y. Li, T. Bao, Z. Gao, X. Shu, K. Zhang, L. Xie, Z. Zhang, A new dam structural response estimation paradigm powered by deep learning and transfer learning techniques, Structural Health Monitoring (2021) 14759217211009780.

[2] R.-T. Wu, M.R. Jahanshahi, Data fusion approaches for structural health monitoring and system identification: past, present, and future, Structural Health Monitoring 19(2) (2020) 552-586.

[3] R. Guidorzi, R. Diversi, L. Vincenzi, C. Mazzotti, V. Simioli, Structural monitoring of a tower by means of MEMS-based sensing and enhanced autoregressive models, European Journal of Control 20(1) (2014) 4-13.

[4] Y. Dong, Y. Li, M. Lai, Structural damage detection using empirical-mode decomposition and vector autoregressive moving average model, Soil Dynamics and Earthquake Engineering 30(3) (2010) 133-145.

[5] Y. Gao, K.M. Mosalam, Y. Chen, W. Wang, Y. Chen, Auto-Regressive Integrated Moving-Average Machine Learning for Damage Identification of Steel Frames, Applied Sciences 11(13) (2021) 6084.

[6] X. Lu, F. McKenna, Q. Cheng, Z. Xu, X. Zeng, S.A. Mahin, An open-source framework for regional earthquake loss estimation using the city-scale nonlinear time history analysis, Earthquake Spectra 36(2) (2020) 806-831.

[7] X. Lu, Q. Cheng, Z. Xu, Y. Xu, C. Sun, Real-time city-scale time-history analysis and its application in resilience-oriented earthquake emergency responses, Applied Sciences 9(17) (2019) 3497.

[8] R. Zhang, Y. Liu, H. Sun, Physics-guided convolutional neural network (PhyCNN) for data-driven seismic response modeling, Engineering Structures 215 (2020) 110704.

[9] Z. Zheng, Y. Tian, Z. Yang, X. Lu, Hybrid framework for simulating building collapse and ruin scenarios using finite element method and physics engine, Applied Sciences 10(12) (2020) 4408.

[10] W. Wu, S. Ge, Y. Yuan, W. Ding, I. Anastasopoulos, Seismic response of subway station in soft soil: Shaking table testing versus numerical analysis, Tunnelling and Underground Space Technology 100 (2020) 103389.

[11] J.Y. Meng, E.M. Lui, Seismic analysis and assessment of a skew highway bridge, Engineering Structures 22(11) (2000) 1433-1452.

[12] R. Zhang, Z. Chen, S. Chen, J. Zheng, O. Büyüköztürk, H. Sun, Deep long short-term memory networks for nonlinear structural seismic response prediction, Computers & Structures 220 (2019) 55-68.

[13] P. Huang, Z. Chen, Deep learning for nonlinear seismic responses prediction of subway station, Engineering

Structures 244 (2021) 112735.
[14] H. Li, T. Wang, G. Wu, Dynamic response prediction of vehicle-bridge interaction system using feedforward neural network and deep long short-term memory network, Structures, Elsevier, 2021, pp. 2415-2431.
[15] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, Journal of Computational physics 378 (2019) 686-707.
[16] L. von Rueden, S. Mayer, K. Beckh, B. Georgiev, S. Giesselbach, R. Heese, B. Kirsch, J. Pfrommer, A. Pick, R. Ramamurthy, Informed Machine Learning--A Taxonomy and Survey of Integrating Knowledge into Learning Systems, arXiv preprint arXiv:1903.12394 (2019).
[17] M. Diligenti, S. Roychowdhury, M. Gori, Integrating prior knowledge into deep learning, 2017 16th IEEE international conference on machine learning and applications (ICMLA), IEEE, 2017, pp. 920-923.
[18] A. Daw, A. Karpatne, W. Watkins, J. Read, V. Kumar, Physics-guided neural networks (pgnn): An application in lake temperature modeling, arXiv preprint arXiv:1710.11431 (2017).
[19] P. Battaglia, R. Pascanu, M. Lai, D. Jimenez Rezende, Interaction networks for learning about objects, relations and physics, Advances in neural information processing systems 29 (2016).
[20] K. Marino, R. Salakhutdinov, A. Gupta, The more you know: Using knowledge graphs for image classification, arXiv preprint arXiv:1612.04844 (2016).
[21] K.-H. Lee, G. Ros, J. Li, A. Gaidon, Spigan: Privileged adversarial learning from simulation, arXiv preprint arXiv:1810.03756 (2018).
[22] I. Sutskever, O. Vinyals, Q.V. Le, Sequence to sequence learning with neural networks, Advances in neural information processing systems 27 (2014).
[23] O. Vinyals, A. Toshev, S. Bengio, D. Erhan, Show and tell: A neural image caption generator, Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3156-3164.
[24] G. Dai, C. Ma, X. Xu, Short-term traffic flow prediction method for urban road sections based on space–time analysis and GRU, IEEE Access 7 (2019) 143025-143035.
[25] Y. Chen, G. Peng, Z. Zhu, S. Li, A novel deep learning method based on attention mechanism for bearing remaining useful life prediction, Applied Soft Computing 86 (2020) 105919.
[26] R. Zhang, Y. Liu, H. Sun, Physics-informed multi-LSTM networks for metamodeling of nonlinear structures, Computer Methods in Applied Mechanics and Engineering 369 (2020) 113226.
[27] A. Filiatrault, I.P. Christovasilis, A. Wanitkorkul, J.W. van de Lindt, Experimental seismic response of a full-scale light-frame wood building, Journal of structural engineering 136(3) (2010) 246-254.
[28] T. Kijewski-Correa, D.K. Kwon, A. Kareem, A. Bentz, Y. Guo, S. Bobby, A. Abdelrazaq, SmartSync: An integrated real-time structural health monitoring and structural identification system for tall buildings, Journal of Structural Engineering 139(10) (2013) 1675-1687.
[29] S. Chen, B. Tang, K. Zhao, X. Li, H. Zhuang, Seismic response of irregular underground structures under adverse soil conditions using shaking table tests, Tunnelling and Underground Space Technology 95 (2020) 103145.
[30] B. Chiou, R. Darragh, N. Gregor, W. Silva, NGA project strong-motion database, Earthquake Spectra 24(1) (2008) 23-44.

[31] F. McKenna, OpenSees: a framework for earthquake engineering simulation, Computing in Science & Engineering 13(4) (2011) 58-66.

[32] J. Li, H. Hao, K. Fan, J. Brownjohn, Development and application of a relative displacement sensor for structural health monitoring of composite bridges, Structural Control and Health Monitoring 22(4) (2015) 726-742.

[33] H. Haddadi, A. Shakal, C. Stephens, W. Savage, M. Huang, W. Leith, J. Parrish, R. Borcherdt, Center for engineering strong-motion data (CESMD), 14th World Conference on Earthquake Engineering, Beijing, China, 2008.

[34] S.K. Kunnath, Q. Nghiem, S. El-Tawil, Modeling and response prediction in performance-based seismic evaluation: case studies of instrumented steel moment-frame buildings, Earthquake Spectra 20(3) (2004) 883-915.

[35] U. Uniform Building Code, International conference of building officials, Uniform Building Code, Whittier, California (1997).

[36] P.G. Somerville, Development of ground motion time histories for phase 2 of the FEMA/SAC steel project, SAC Joint Venture1997.

Chapter 7 CONCLUSION

7.1 Overview

This work aims to apply deep learning to the condition assessment of civil infrastructure and materials. The first chapter summarizes the applications of deep learning in civil engineering based on static and dynamic settings. It mainly covers the applications of deep learning in structural damage detection and structural vibration analysis for infrastructure (e.g., buildings, pavements, bridges, and tunnels). The second chapter presents the applications of deep learning and infrared thermography in asphalt pavement crack severity classification. This chapter focuses on the longitudinal or transverse crack. The third chapter shows the applications of convolutional neural networks and infrared thermography for asphalt pavement crack segmentation. This chapter presents the performance of the proposed method under different conditions. The fourth chapter introduces the application of artificial neural networks in modeling the tensile behavior of hybrid steel-PVA fiber reinforced concrete containing fly ash and slag powder. This chapter presents the sensitivity analysis to analyze the contribution of the components of HFRC to the tensile behavior of HFRC. The fifth chapter presents the application of deep transfer learning to vehicle classification through asphalt pavement vibration. This chapter introduces two deep transfer learning-based methods based on different inputs (time-domain graph and time-frequency graph). The sixth chapter illustrates the application of the physics-informed deep learning model (LSTM) to data-driven structural response modeling. This chapter describes the integration of the deep learning model and physical theory (structural dynamics) in structural response modeling.

7.2 Major findings

The major findings are summarized as follows:

- For asphalt pavement crack severity classification, the CNN models had the highest accuracy on the fusion image for deep learning from scratch, but the highest accuracy on the visible image for transfer learning. The CNN models had a very satisfying performance on both the no crack and low crack for all three types of images. But for the medium crack and high crack, three types of images had different influences on the performance of the CNN models, while misclassification occurs mainly on the medium crack and high crack for all three types of images. EfficientNet-B3 had the highest accuracy on all three types of images for both deep learning from scratch and transfer learning.

- For asphalt pavement crack segmentation, the accuracy metrics and predictions of the visible image and fusion image are similar and even identical for all segmentation models, which are much better than that of the infrared image. When the background is rough (e.g. wet) or cracks are similar to the background (e.g. pavement texture), the fusion image is a better choice for crack detection as IRT distinguishes cracks by temperature differences. The different periods in a day have no significant influence on the predictions of segmentation models for the infrared image and the fusion image. Compared with the visible image, all segmentation models have a more stable performance for the

fusion image. Among seven segmentation models, FPN could be the best model because of its high accuracy and low complexity.

- For modeling the tensile behavior of HFRC, the ANN model shows more powerfully nonlinear-fitting abilities and could consider more factors (a total of 23 factors), including fibers' characteristics, mechanical properties of plain concrete, and concrete composition. Among all input features, the mechanical properties of plain concrete make the main contribution to the tensile strength and strain corresponding to tensile strength; steel fiber tends to make a higher contribution to the tensile strength and strain corresponding to tensile strength than PVA fiber. These outcomes are similar to the experimental results.

- For vehicle classification by asphalt pavement vibration, the CNN model had a satisfactory performance in both methods with the accuracy of Method I exceeding 0.94 and Method II exceeding 0.95. The CNN model in Method II performed better in the accuracy metrics with considering label imbalance, but worse in the accuracy metrics without considering label imbalance than that in Method I. In both two methods, EfficientNet-B3 had the highest accuracy metrics when considering the label imbalance, but other EfficientNets performed better in the accuracy metrics without taking into account the label imbalance. The CNN model with deep transfer learning could be an effective, accurate, and reliable method for vehicle classification based on asphalt pavement vibration.

- For structural response modeling, the accuracy of the proposed PI-LSTM network was higher and concentrated on a smaller range than the PhyCNN model for the numerical simulation results of the SDOF system and experimental results of the six-story building. The proposed PI-LSTM network was also more accurate and robust than the regular LSTM for the numerical simulation results of the SDOF system and the six-story building. The proposed PI-LSTM network could be a reliable and robust approach for structural response modeling.

7.3 Recommendation for future work

This work presents some deep learning studies for the condition assessment of civil infrastructure and materials. Some future work should be conducted to further improve and expand the current work:

- Build a crack (or damage) detection system based on deep learning and infrared thermography. This work has finished the crack severity classification and crack segmentation. In the future, we should future investigate the severity classification of different crack types (e.g. fatigue cracking and edge cracking). We should also increase the dataset for crack segmentation and take into account more complex patterns or more complicated environments. In addition to crack severity classification and crack segmentation, we should carry out new work on crack detection based on deep learning and infrared thermography. Based on these works, we can build a real-time crack detection system and then a new pavement condition assessment system.

- Build a deep learning-based HFRC mix design method based on the prediction of the mechanical properties. This work has finished the ANN-based tensile behavior prediction method. In the future, we should future apply deep learning methods to predict the mechanical properties (e.g. compressive and flexural behaviors). Then based on these mechanical property predictions, we can build a deep learning-based HFRC mix design method.

- Build a physics-informed deep learning method for structural condition assessment. This work has finished the work on deep learning-based object identification by vibration and deep learning-based structural response modeling. In the future, we will apply the physics-informed deep learning model to predict the structural response and then build a structural condition assessment method through the predicted and actual structural response.